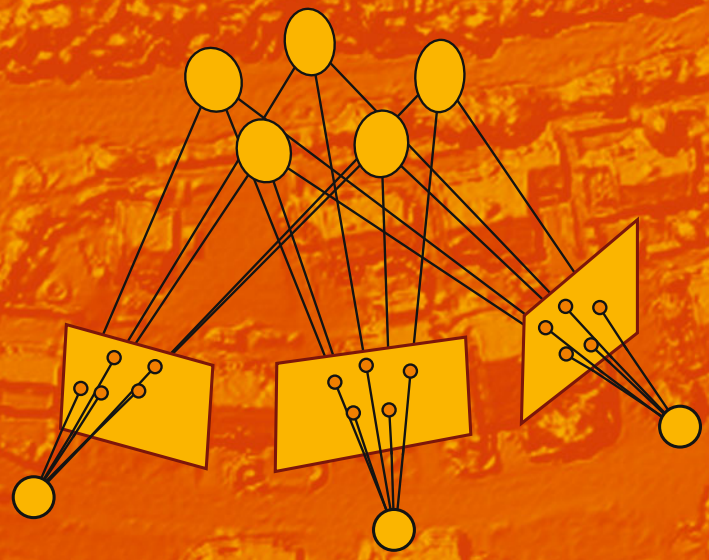
Geometry and Computing 11



Wolfgang Förstner
Bernhard P. Wrobel

Photogrammetric Computer Vision

Statistics, Geometry, Orientation and Reconstruction

Springer

Geometry and Computing

Volume 11

Series editors

Herbert Edelsbrunner, Department Computer Science, Durham, NC, USA
Leif Kobbelt, RWTH Aachen University, Aachen, Germany
Konrad Polthier, AG Mathematical Geometry Processing, Freie Universität Berlin, Berlin, Germany

Geometric shapes belong to our every-day life, and modeling and optimization of such forms determine biological and industrial success. Similar to the digital revolution in image processing, which turned digital cameras and online video downloads into consumer products, nowadays we encounter a strong industrial need and scientific research on geometry processing technologies for 3D shapes.
Several disciplines are involved, many with their origins in mathematics, revived with computational emphasis within computer science, and motivated by applications in the sciences and engineering. Just to mention one example, the renewed interest in discrete differential geometry is motivated by the need for a theoretical foundation for geometry processing algorithms, which cannot be found in classical differential geometry.

Scope: This book series is devoted to new developments in geometry and computation and its applications. It provides a scientific resource library for education, research, and industry. The series constitutes a platform for publication of the latest research in mathematics and computer science on topics in this field.

- Discrete geometry
- Computational geometry
- Differential geometry
- Discrete differential geometry
- Computer graphics
- Geometry processing
- CAD/CAM
- Computer-aided geometric design
- Geometric topology
- Computational topology
- Statistical shape analysis
- Structural molecular biology
- Shape optimization
- Geometric data structures
- Geometric probability
- Geometric constraint solving
- Algebraic geometry
- Graph theory
- Physics-based modeling
- Kinematics
- Symbolic computation
- Approximation theory
- Scientific computing
- Computer vision

More information about this series at http://www.springer.com/series/7580

Wolfgang Förstner · Bernhard P. Wrobel

Photogrammetric Computer Vision

Statistics, Geometry, Orientation and Reconstruction

Springer

Wolfgang Förstner
Institut für Geodäsie und Geoinformation
Rheinische Friedrich-Wilhelms-Universität Bonn
Bonn
Germany

Bernhard P. Wrobel
Institut für Geodäsie
Technische Universität Darmstadt
Darmstadt
Germany

ISSN 1866-6795 ISSN 1866-6809 (electronic)
Geometry and Computing
ISBN 978-3-319-11549-8 ISBN 978-3-319-11550-4 (eBook)
DOI 10.1007/978-3-319-11550-4

Library of Congress Control Number: 2016954546

© Springer International Publishing Switzerland 2016
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This textbook on *Photogrammetric Computer Vision – Statistics, Geometry, Orientation and Reconstruction* provides a statistical treatment of the geometry of multiple view analysis useful for camera calibration, orientation, and geometric scene reconstruction.

The book is the first to offer a joint view of photogrammetry and computer vision, two fields that have converged in recent decades. It is motivated by the need for a conceptually consistent theory aiming at generic solutions for orientation and reconstruction problems.

Large parts of the book result from teaching bachelor's and master's courses for students of geodesy within their education in photogrammetry. Most of these courses were simultaneously offered as subjects in the computer science faculty.

The book provides algorithms for various problems in geometric computation and in vision metrology, together with mathematical justification and statistical analysis allowing thorough evaluation.

The book aims at enabling researchers, software developers, and practitioners in the photogrammetric and GIS industry to design, write, and test their own algorithms and application software using statistically founded concepts to obtain optimal solutions and to realize self-diagnostics within algorithms. This is essential when applying vision techniques in practice. The material of the book can serve as a source for different levels of undergraduate and graduate courses in photogrammetry, computer vision, and computer graphics, and for research and development in statistically based geometric computer vision methods.

The sixteen chapters of the book are self-contained, are illustrated with numerous figures, have exercises, and are supported by an appendix and an index. Many of the examples and exercises can be verified or solved using the MATLAB routines available on the home page of the book, which also contains solutions to some of the exercises.

Acknowledgements: The book gained a lot through the significant support of numerous colleagues. We thank Camillo Ressl and Jochen Meidow for their careful reading of the manuscript and Carl Gerstenecker and Boris Kargoll for their critical review of Part I on statistics. The language proofreading by Silja Weber, Indiana University, is highly appreciated. Thanks for fruitful comments, discussions and support of the accompanying MATLAB Software to Martin Drauschke, Susanne Wenzel, Falko Schindler, Thomas Läbe, Richard Steffen, Johannes Schneider, and Lutz Plümer. We thank the American Society for Photogrammetry and Remote Sensing for granting us permission to use material of the sixth edition of the 'Manual of Photogrammetry'.

Wolfgang Förstner
Bernhard P. Wrobel

Bonn, 2016

Contents

1 Introduction 1
- 1.1 Tasks for Photogrammetric Computer Vision 2
- 1.2 Modelling in Photogrammetric Computer Vision 6
- 1.3 The Book 11
- 1.4 On Notation 16

Part I Statistics and Estimation

2 Probability Theory and Random Variables 21
- 2.1 Notions of Probability 21
- 2.2 Axiomatic Definition of Probability 22
- 2.3 Random Variables 24
- 2.4 Distributions 28
- 2.5 Moments 36
- 2.6 Quantiles of a Distribution 40
- 2.7 Functions of Random Variables 40
- 2.8 Stochastic Processes 48
- 2.9 Generating Random Numbers 55
- 2.10 Exercises 56

3 Testing 61
- 3.1 Principles of Hypothesis Testing 61
- 3.2 Testability of an Alternative Hypothesis 65
- 3.3 Common Tests 69
- 3.4 Exercises 72

4 Estimation 75
- 4.1 Estimation Theory 75
- 4.2 The Linear Gauss–Markov Model 81
- 4.3 Gauss–Markov Model with Constraints 99
- 4.4 The Nonlinear Gauss–Markov Model 102
- 4.5 Datum or Gauge Definitions and Transformations 108
- 4.6 Evaluation 115
- 4.7 Robust Estimation and Outlier Detection 141
- 4.8 Estimation with Implicit Functional Models 160
- 4.9 Methods for Closed Form Estimations 176
- 4.10 Estimation in Autoregressive Models 183
- 4.11 Exercises 185

Part II Geometry

5 Homogeneous Representations of Points, Lines and Planes 195
5.1 Homogeneous Vectors and Matrices 195
5.2 Homogeneous Representations of Points and Lines in 2D 205
5.3 Homogeneous Representations in $\mathbb{P}^n$ 209
5.4 Homogeneous Representations of 3D Lines 216
5.5 On Plücker Coordinates for Points, Lines and Planes 221
5.6 The Principle of Duality 229
5.7 Conics and Quadrics 236
5.8 Normalizations of Homogeneous Vectors 241
5.9 Canonical Elements of Coordinate Systems 242
5.10 Exercises 245

6 Transformations 247
6.1 Structure of Projective Collineations 248
6.2 Basic Transformations 250
6.3 Concatenation and Inversion of Transformations 261
6.4 Invariants of Projective Mappings 266
6.5 Perspective Collineations 277
6.6 Projective Correlations 282
6.7 Hierarchy of Projective Transformations and Their Characteristics 284
6.8 Normalizations of Transformations 285
6.9 Conditioning 286
6.10 Exercises 287

7 Geometric Operations 291
7.1 Geometric Operations in 2D Space 292
7.2 Geometric Operations in 3D Space 299
7.3 Vector and Matrix Representations for Geometric Entities 311
7.4 Minimal Solutions for Conics and Transformations 316
7.5 Exercises 322

8 Rotations 325
8.1 Rotations in 3D 325
8.2 Concatenation of Rotations 337
8.3 Relations Between the Representations for Rotations 338
8.4 Rotations from Corresponding Vector Pairs 339
8.5 Exercises 340

9 Oriented Projective Geometry 343
9.1 Oriented Entities and Constructions 344
9.2 Transformation of Oriented Entities 355
9.3 Exercises 358

10 Reasoning with Uncertain Geometric Entities 359
10.1 Motivation 360
10.2 Representing Uncertain Geometric Elements 364
10.3 Propagation of the Uncertainty of Homogeneous Entities 386
10.4 Evaluating Statistically Uncertain Relations 393
10.5 Closed Form Solutions for Estimating Geometric Entities 395
10.6 Iterative Solutions for Maximum Likelihood Estimation 414
10.7 Exercises 432

Part III Orientation and Reconstruction

11 Overview ... 441
11.1 Scene, Camera, and Image Models ... 441
11.2 The Setup of Orientation, Calibration, and Reconstruction ... 449
11.3 Exercises ... 453

12 Geometry and Orientation of the Single Image ... 455
12.1 Geometry of the Single Image ... 456
12.2 Orientation of the Single Image ... 489
12.3 Inverse Perspective and 3D Information from a Single Image ... 523
12.4 Exercises ... 537

13 Geometry and Orientation of the Image Pair ... 547
13.1 Motivation ... 547
13.2 The Geometry of the Image Pair ... 549
13.3 Relative Orientation of the Image Pair ... 568
13.4 Triangulation ... 596
13.5 Absolute Orientation and Spatial Similarity Transformation ... 607
13.6 Orientation of the Image Pair and Its Quality ... 608
13.7 Exercises ... 615

14 Geometry and Orientation of the Image Triplet ... 621
14.1 Geometry of the Image Triplet ... 622
14.2 Relative Orientation of the Image Triplet ... 632
14.3 Exercises ... 641

15 Bundle Adjustment ... 643
15.1 Motivation for Bundle Adjustment and Its Tasks ... 644
15.2 Block Adjustment ... 645
15.3 Sparsity of Matrices, Free Adjustment and Theoretical Precision ... 651
15.4 Self-calibrating Bundle Adjustment ... 674
15.5 Camera Calibration ... 696
15.6 Outlier Detection and Approximate Values ... 707
15.7 View Planning ... 715
15.8 Exercises ... 722

16 Surface Reconstruction ... 727
16.1 Introduction ... 727
16.2 Parametric $2^1/_2$D Surfaces ... 733
16.3 Models for Reconstructing One-Dimensional Surface Profiles ... 742
16.4 Reconstruction of $2^1/_2$D Surfaces from 3D Point Clouds ... 757
16.5 Examples for Surface Reconstruction ... 763
16.6 Exercises ... 765

Appendix: Basics and Useful Relations from Linear Algebra ... 767
A.1 Inner Product ... 767
A.2 Determinant ... 767
A.3 Inverse, Adjugate, and Cofactor Matrix ... 769
A.4 Skew Symmetric Matrices ... 770
A.5 Eigenvalues ... 772
A.6 Idempotent Matrices ... 774
A.7 Kronecker Product, vec(·) Operator, vech(·) Operator ... 775

A.8 Hadamard Product 776
A.9 Cholesky and QR Decomposition 776
A.10 Singular Value Decomposition 777
A.11 The Null Space and the Column Space of a Matrix 777
A.12 The Pseudo-inverse 779
A.13 Matrix Exponential 781
A.14 Tensor Notation 782
A.15 Variance Propagation of Spectrally Normalized Matrix 783

References 785

Index 799

List of Algorithms

1 Estimation in the linear Gauss–Markov model 91
2 Estimation in the Gauss–Markov model with constraints 108
3 Random sample consensus 156
4 Robust estimation in the Gauss–Helmert model with constraints 168
5 Reweighting constraints 169
6 Estimation in the model with constraints between the observations only 171

7 Algebraic solution for estimating 2D homography from point pairs 389
8 Direct LS estimation of 2D line from points with isotropic accuracy 401
9 Direct LS estimation of a 2D point from lines with positional uncertainty . . . 403
10 Direct LS estimation of the mean of directions with isotropic uncertainty. . . . 404
11 Direct LS estimation of the mean of axes with isotropic uncertainty 405
12 Direct LS estimation of a rotation from direction pairs 408
13 Direct LS estimation of similarity 411
14 Direct LS estimation of 3D line from points 414
15 Estimation in the Gauss–Helmert model with reduced coordinates 416

16 Algebraic estimation of uncertain projection from six or more points 496
17 Optimal estimation of a projection matrix from observed image points 499
18 Decomposition of uncertain projection matrix 500
19 3D circle with given radius determined from its image 536

20 Base direction and rotation from essential matrix 583
21 Optimal triangulation from two images and spherical camera model 600

22 Sequential spatial resections 709
23 Sequential similarity transformations 710

List of Symbols

Table 0.1 List of symbols: A – M

symbol	meaning
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$	names of planes, sets
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$	homogeneous vectors of planes
A_0, $\boldsymbol{A}_h$	Euclidean, homogeneous part of the homogeneous coordinate vector $\mathbf{A}$ of plane $\mathcal{A}$
$\mathbf{A}_X, \mathbf{A}_Y, \mathbf{A}_Z$	homogeneous vectors of coordinate planes, perpendicular to the axes X, Y, and Z
$\mathcal{B}^d$	d-dimensional unit ball in $\mathbb{R}^d$
$\mathbb{C}\text{ov}(\bullet, \bullet)$	covariance operator
$\text{CR}(\bullet, \bullet, \bullet, \bullet)$	cross ratio
D	6×6 matrix dualizing a line
$\delta(x)$	Dirac's delta function
$\text{Diag}(\bullet)$	diagonal matrix of vector or list of matrices
$\text{diag}(\bullet)$	vector of diagonal elements of a matrix
$\det(\bullet) = \|\bullet\|$	determinant
$\boldsymbol{e}_i^{[d]}$	ith basic unit vector in d-space, e.g., $\boldsymbol{e}_2^{[3]} = [0, 1, 0]^\mathsf{T}$
$\mathbb{D}(\bullet)$	dispersion operator
$\mathbb{E}(\bullet)$	expectation operator
$\mathsf{\Gamma}(\mathbf{L})$	Plücker matrix of a 3D line
$\overline{\mathsf{\Gamma}}(\mathbf{L})$	dual Plücker matrix of a 3D line
$\mathsf{\Gamma}^{(s)}(\mathbf{L})$	2×4 matrix of selected independent rows
I_n	$n \times n$ unit matrix
$\mathcal{J} = \{1, ..., j, ..., J\}$	set of indices
J_{xy}, $\mathsf{J}_{x,y}$	Jacobian $\partial \boldsymbol{x} / \partial \boldsymbol{y}$
J_r	Jacobian $\partial \mathbf{x} / \partial \boldsymbol{x}_r$, with reduced vector $\boldsymbol{x}_r$ of $\mathbf{x}$
J_s	Jacobian $\partial \mathbf{x}^s / \partial \mathbf{x}$ of spherical normalization
$\mathsf{H}f$ or $\mathsf{H}(f)$	Hessian matrix $[\partial^2 f(\boldsymbol{x}) / (\partial x_i \partial x_j)]$ of function $f(\boldsymbol{x})$
$\mathcal{H}$	name of homography
H	general homography, 2×2, 3×3, or 4×4 matrix
$\boldsymbol{l}$	vector of observations in an estimation procedure
ℓ, m, n	names of 2D lines
$\mathbf{l}$, $\mathbf{m}$, $\mathbf{n}$	homogeneous vectors of 2D lines
$\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$	names of 3D lines
$\mathbf{L}$, $\mathbf{M}$, $\mathbf{N}$	homogeneous vectors of 3D lines
$\boldsymbol{l}_0$, $\boldsymbol{l}_h$	Euclidean, homogeneous part of homogeneous coordinate vector $\mathbf{l}$ of 2D line ℓ
$\boldsymbol{L}_0$, $\boldsymbol{L}_h$	Euclidean, homogeneous part of homogeneous coordinate vector $\mathbf{L}$ of 3D line $\mathcal{L}$
$\mathbf{l}_x, \mathbf{l}_y, \mathbf{L}_X, \mathbf{L}_Y, \mathbf{L}_Z$	line parameters of coordinate axes
$\overline{\mathbf{L}}$	coordinates of 3D line $\overline{\mathcal{L}}$ dual to 3D line $\mathcal{L}$
$\mathcal{M}$	motion, special homography in 2D or 3D
$\mathbb{N}$	set of natural numbers
$\mathcal{M}(\boldsymbol{\mu}, \mathsf{\Sigma})$	distribution characterized only by mean $\boldsymbol{\mu}$ and covariance matrix $\mathsf{\Sigma}$

Table 0.2 List of symbols: N – Z

symbol	meaning
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ
N	normal equation matrix
$\mathsf{N}(\bullet)$	operator for achieving Frobenius norm 1, for vectors: spherical normalization
$\mathsf{N}^{\mathsf{e}}(\bullet)$	operator for Euclidean normalization of homogeneous vectors
$\mathsf{N}^{\sigma}(\bullet)$	operator for spectral normalization of matrices
$\mathsf{null}(\bullet)$, $\mathsf{null}^{\mathsf{T}}(\bullet)$	orthonormal matrix: basis vectors of null space as columns, transpose
o	origin of coordinate system
$O(\boldsymbol{Z}), o(\boldsymbol{z})$	coordinates of the centre of perspectivity
$\mathbb{P}^n$	n-dimensional projective space
$\mathbb{P}^{*n}$	dual n-dimensional projective space
$\mathsf{\Pi}(\mathbf{X})$, $\mathsf{\Pi}(\mathbf{A})$	Pi-matrix of a 3D point or a plane
$\overline{\mathsf{\Pi}}(\mathbf{X})$, $\overline{\mathsf{\Pi}}(\mathbf{A})$	dual Pi-matrix of a 3D point or a plane
$\mathsf{\Pi}^{(s)}(\mathbf{X})$, $\mathsf{\Pi}^{(s)}(\mathbf{A})$	3×4 matrix of selected independent rows
$r(x\|a, b)$	rectangle function in the range $[a, b]$
r_{xy}	correlation coefficient of $\underline{x}$ and $\underline{y}$
R	rotation matrix, correlation matrix
$\mathbb{R}^n$	n-dimensional Euclidean space over $\mathbb{R}$
$\mathbb{R}^n \setminus \mathbf{0}$	n-dimensional Euclidean space without origin
$s(x)$	step function
$SL(n)$	special group of linear transformations with determinant 1
$SO(n)$	special group of orthogonal transformations (rotations)
$so(n)$	Lie group of skew matrices
$\mathsf{S}_a, \mathsf{S}(\boldsymbol{a}), \mathsf{S}_{\mathrm{a}}, \mathsf{S}(\mathbf{a})$	inhomogeneous, homogeneous skew symmetric matrix depending on a 3-vector
S_i	3×3 skew symmetric matrix of 3×1 vector $\boldsymbol{e}_i^{[3]}$
$\mathsf{S}^{(s)}(\mathbf{x})$	2×3 matrix with two selected independent rows
S^d	unit sphere of dimension d in $\mathbb{R}^{d+1}$, set of points $\boldsymbol{x} \in \mathbb{R}^{d+1}$ with $\|\boldsymbol{x}\| = 1$
σ_x	standard deviation of $\underline{x}$
σ_{xy}	covariance of $\underline{x}$ and $\underline{y}$
Σ_{xy}	covariance matrix of $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$
$\mathbb{T}^n$	oriented projective space
$\mathbb{T}^{*n}$	dual oriented projective space
W_{xx}	weight matrix of parameters $\underline{\boldsymbol{x}}$
$\boldsymbol{x}$	unknown parameters in an estimation procedure
$\mathcal{x}$, $\mathcal{y}$, $\mathcal{z}$	names of 2D points
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$	homogeneous vectors of points in 2D
$\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$	names of 3D points
$\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$	homogeneous vectors of points in 3D
$\boldsymbol{x}_0, x_h$	Euclidean, homogeneous part of the homogeneous coordinate vector $\mathbf{x}$ of point $\mathcal{x}$
$\boldsymbol{X}_0, X_h$	Euclidean, homogeneous part of the homogeneous coordinate vector $\mathbf{X}$ of point $\mathcal{X}$

Table 0.3 List of symbols: fonts, operators

symbol	meaning
$\%_0$	permille
$\underset{n\times 1}{\boldsymbol{x}}$, $\boldsymbol{\mu}$	inhomogeneous vectors, with indicated size
$\mathbf{x}$, $\boldsymbol{\mu}$	homogeneous vectors
$\underset{m\times n}{\mathsf{A}}$, R	inhomogeneous matrices, with indicated size, generally $n \leq m$
K, P	homogeneous matrices
$\lambda_{\max}(\bullet)$	largest eigenvalue
$(\bullet)_\infty$	entity at infinity, transformation referring to entities at infinity
$i \in \mathcal{I} = \{1, ..., I\}$	index and index set
$(\bullet)^\mathsf{T}$	transpose
$(\bullet)^{-\mathsf{T}}$	transpose of inverse matrix
$(\bullet)^+$	pseudo-inverse matrix
$(\bullet)^a$	approximated vector or matrix within iterative estimation procedure
$(\bullet)^*$	adjugate matrix
$(\bullet)^\circlearrowleft$	cofactor matrix
$(\bullet)_r$	reduced, minimal vector
$(\bullet)^{(s)}$	reduced matrix with selected independent rows
$\|\bullet\|$	absolute value of scalar, Euclidean norm of a vector, determinant of matrix
$\|\|\bullet\|\|$	Frobenius norm
$\langle \bullet, \bullet \rangle_A$	inner product, e.g., $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_A = \boldsymbol{x}^\mathsf{T} \mathsf{A} \boldsymbol{y}$
$\langle \bullet, \bullet, \bullet \rangle$	triple product of three 3-vectors, identical to the determinant of their 3×3 matrix $[\bullet, \bullet, \bullet]$
$\langle \bullet, \bullet, \bullet, \bullet \rangle$	cross ratio of four numbers
$\circ$	operation, defined locally
$\overline{(\bullet)}$	dualizing or Hodge operator
$\boldsymbol{x}^\perp$	vector perpendicular to $\boldsymbol{x}$
$\nabla \boldsymbol{x}(\boldsymbol{p})$	nabla operator, gradient, Jacobian $\partial x / \partial p$
$\underline{(\bullet)}$	stochastic variable
$\underline{\mathrm{vec}\mathsf{A}}$	vec operator
$\underline{x} \sim \mathcal{H}(\boldsymbol{q})$	stochastic variable $\underline{x}$ follows distribution $\mathcal{H}(\boldsymbol{q})$
$\widehat{(\bullet)}$	estimated value
$\widetilde{(\bullet)}$	true value
$\exists$	there exists
$\mathsf{A} \odot \mathsf{B}$	Hadamard product
$\mathsf{A} \otimes \mathsf{B}$	Kronecker product
$\cap$	intersection operator ('cap')
$\wedge$	join operator ('wedge')
$\cong$	proportional to (vectors, matrices)
$\propto$	proportional to (functions)
$\neg$	not, antipode of an entity having negative homogeneous coordinates
$\Leftrightarrow$	if and only if
$\doteq$	defining equation
$:=$	assignment
$a \overset{!}{=} b$	constraint: a should be equal to b, or $\mathbb{E}(\underline{a}) = b$
$\boldsymbol{a} \overset{+}{=} \boldsymbol{b}$, $\mathbf{a} \overset{+}{=} \mathbf{b}$	two elements are equivalent in oriented projective geometry
$[\bullet, \bullet]$	closed interval
$(\bullet, \bullet]$	semi-open interval
$\lfloor x \rfloor$	floor function, largest integer smaller than x
$\lceil x \rceil$	ceiling function, smallest integer larger than x

Table 0.4 List of Symbols in Part III (1)

abbreviation	meaning
α	parallactic angle between two rays
A	infinite homography, mapping from plane at infinity to image plane, also called H_∞
$A = [C, D]$	design matrix, Jacobian w.r.t. parameters, partitioned for scene coordinates and orientation parameters
$(\mathbf{A}, \mathbf{B}, \mathbf{C})$	principal planes of camera coordinate system, rows of projection matrix P
$\mathcal{A}_{l'}(\mathbf{A}_{l'})$	projection plane to image line l'
B	Jacobian of constraints w.r.t. observations
$\mathbf{b}$, $\boldsymbol{B}$	base vector
c	principal distance
${}^c(\bullet)$	coordinate in camera coordinate system
$\boldsymbol{c}(\mathbf{x})$	function to derive inhomogeneous from homogeneous coordinates
D_E	number of parameters of observed image feature
C	3×3 matrix for conics
D_T	number of parameters for transformation or projection
D_I	number of parameters for scene feature
$(it) \in \mathcal{E}$	index set $\mathcal{E} \subset \mathcal{I} \times \mathcal{T}$ for observed image features f_{it}
$e'(\mathbf{e}')$, $e''(\mathbf{e}'')$	epipoles of image pair
$\mathcal{E}$	epipolar plane
E, $\mathsf{E}_{tt'}$	essential matrix, of images t and t'
E_{it}	matrix for selecting scene points observed in images
F, $\mathsf{F}_{tt'}$	fundamental matrix, of images t and t'
$\mathcal{F}_i(\boldsymbol{k}_i)$	scene feature $\mathcal{F}_i$ with coordinates $\boldsymbol{k}_i$, indices $i \in \mathcal{I}$
$\mathcal{F}_{i0}(\boldsymbol{k}_{i0})$	control scene feature $\mathcal{F}_{i0}$ with coordinates $\boldsymbol{k}_{i0}$, indices $i \in \mathcal{I}_0$
$f_{it}(\boldsymbol{l}_{it})$	image feature f_{it} with observed coordinates $\boldsymbol{l}_{it}$, indices $(it) \in \mathcal{E}$
$\boldsymbol{f}_{it}$	projection function for scene feature i and image t
$\boldsymbol{g}_{it}$	projection relation for scene feature i and image t
$\mathsf{G}_3, \mathsf{G}_4$	$d \times d$ selection matrix $\mathrm{Diag}([\mathbf{1}_d^\mathsf{T}, 0])$
G_6	6×6 selection matrix $\mathrm{Diag}(\{I_3, 0_{3\times 3}\})$
$\mathcal{H}(\mathsf{H})$	homography, perspective mapping
H_∞	infinite homography, mapping from plane at infinity to image plane, also called A
H_A	homography, mapping plane $\mathcal{A}$ in object space to image plane
$\mathcal{H}(\boldsymbol{x}_H)$	principal point
H	matrix of constraints for fixing gauge
H_g	flight height over ground
${}^i(\bullet)$	coordinate in image coordinate system
$\{1, ..., i, ...I\} = \mathcal{I}$	index set for scene features
(i, j)	discrete image coordinates, unit pixels
$l'(\mathcal{X})$, $l'(\mathcal{X}'')$	epipolar lines of image pair, depending on scene or on image point
κ	rotation angle around Z-axis of camera system, gear angle
κ_1, κ_2	principal curvatures of surface
$\boldsymbol{k}$	vector of unknown coordinates
$\mathcal{K}_1$, $\mathcal{K}_2$	principal points of optics
K	calibration matrix
$\ell(l'', l''')$	projection operator to obtain line l'
$\mathcal{L}_{x'}(\mathbf{L}_{x'})$	projection ray to image point $\mathcal{X}'$
${}^m(\bullet)$	coordinate in model coordinate system of two or more images
m	scale difference of x'- and y'-image coordinates
$\mathcal{M}(\mathsf{M})$	motion or similarity

Table 0.5 List of Symbols in Part III (2)

abbreviation	meaning
${}^{n}(\bullet)$	coordinate in normal camera coordinate system (parallel to scene coordinate system)
N	normal equation matrix
$\overline{\mathsf{N}}_{pp}$, $\overline{\mathsf{N}}_{kk}$	normal equation matrices reduced to orientation parameters and coordinates
ω	rotation angle around X-axis of camera, roll angle
$O(\boldsymbol{Z})$	coordinates of projection centre
$\wp_2(\mathcal{x}', \mathcal{l}'')$, $\wp_3(\mathcal{x}', \mathcal{l}'')$	prediction of point from point and line in two other images
ϕ	rotation around Y-axis of camera, tilt angle
$\mathcal{P}$	projection with projection matrix for points
$\mathcal{P}_t(\boldsymbol{p}_t)$	tth image with parameters of projection
$\mathcal{P}_{0t}(\boldsymbol{p}_{0t})$	image with observed parameters $\boldsymbol{p}_{0t}$ of projection $\mathcal{P}_{0t}$, indices $t \in \mathcal{T}_0$
$\boldsymbol{p}$	vector of unknown orientation parameters
P	projection matrix for points
P_d	$(d-1) \times d$ unit projection matrix $[\boldsymbol{I}_{d-1}\|\boldsymbol{0}]$
$\boldsymbol{q}$	vector of parameters modelling non-linear image distortions
Q	3×6-projection matrix for lines, 4×4 matrix for quadrics
Q_6	3×6 unit projection matrix $[\boldsymbol{I}_3\|\boldsymbol{0}_{3\times 3}]$
$\mathcal{R}(R)$	rotation matrix
s, S	image scale s and image scale number $1/S$
s	shear of image coordinate system
$\boldsymbol{s}$	vector of additional parameters for modelling systematic errors
$\{1, ..., t, ...T\} = \mathcal{T}$	index set for images (time)
$\mathsf{T} = [[T_{i,jk}]]$	trifocal tensor
$v'(\mathbf{v}')$	vanishing point
$\mathcal{x}'$, $\overline{\mathcal{x}}'$	observable image point, ideal image point (without distortion)
$\boldsymbol{Z}(\boldsymbol{a})$	2×2 matrix operator $\boldsymbol{Z}: \underset{2\times 1}{\boldsymbol{a}} \rightarrow \begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix}$

Table 0.6 Abbreviations

abbreviation	meaning
AO	absolute orientation
AR	autoregressive
BLUE	best linear unbiased estimator
DLT	direct linear transformation
EO	exterior orientation
GIS	geoinformation system(s), geoinformation science
GHM	Gauss–Helmert model
GPS	global positioning system
GSD	ground sampling distance
IMU	inertial measuring unit
IO	interior orientation
LS	least squares
MAD	median absolute difference
MAP	maximum a posteriori
ML	maximum likelihood
MSE	mean square error
PCA	principal component analysis
RANSAC	random sample consensus
RMSE	root mean square error
SLERP	spherical linear interpolation
SVD	singular value decomposition

Chapter 1
Introduction

Images have always served as an inspiration to perceive our environment. Naturalism in art, supported in the sixteenth century by the knowledge of the perspective projection, was replaced in the nineteenth century by the technique of registering perspective images as photographs. Photographs not only initiated the transition to modernity in art but soon were used for solving engineering tasks, such as the 3D mensuration of buildings for preserving cultural heritage (Albertz, 2001). Today, digital images are omnipresent and used as a matter of course for documentation, communication, reconnaissance, and surveillance. Professional application domains are medicine, industrial inspection, quality control, and mapping and remote sensing. Computers not only serve as image storage and allow image processing but also enable image analysis and interpretation. Early examples are barcode readers; late examples, 3D models of complete cities as in Google Maps or Microsoft's Bing Map.

This book is about concepts and methods for developing computer vision systems for automatically analysing images, with a focus on the main application areas of photogrammetry, specifically mapping and image-based metrology.

Photogrammetry is the science and technology of obtaining information about the physical environment from images, with a focus on applications in mapping, surveying and high-precision metrology. The aim of photogrammetry is to provide automated or semi-automated procedures for these engineering tasks, with emphasis on a specified accuracy, reliability and completeness of the extracted information.

Computer vision, a science and technology of obtaining information about the physical environment from images, does not focus on specific applications. On the contrary, its roots are in the area of what is called artificial intelligence, which aims at mimicking intelligent human behaviour. It has deep links to cognitive science via the analysis of the visual system of animals and humans. As such it is a special part of artificial intelligence, addressing among other things the development of systems, e.g., robots, which mimic the cognitive capabilities of natural systems having vision sensors. Such sensors may be cameras or video cameras, laser scanners or tomographic sensors, as long as they yield a dense description of the object.

Photogrammetric computer vision comprises photogrammetric theories, methods and techniques related to computer vision problems and relevant for automatically solving tasks in mapping and metrology using software systems and the necessary tools for design and evaluation of the results. As such it is intimately linked with methods from mathematics, statistics, physics, and computer science. It is closely coupled especially to methods in computational geometry, image processing and computer graphics (cf. Fig. 1.1). As photogrammetry can be seen as a part of remote sensing, the mentioned aspects are also valid for analysing satellite images or images with many bands of the physical spectrum.

This book is the first of two volumes on photogrammetric computer vision. This first volume addresses all aspects of statistically founded geometric image analysis, the second volume will focus on methods of image processing, analysis, interpretation. Both volumes address the mathematical concepts for developing vision software systems. They do not

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_1

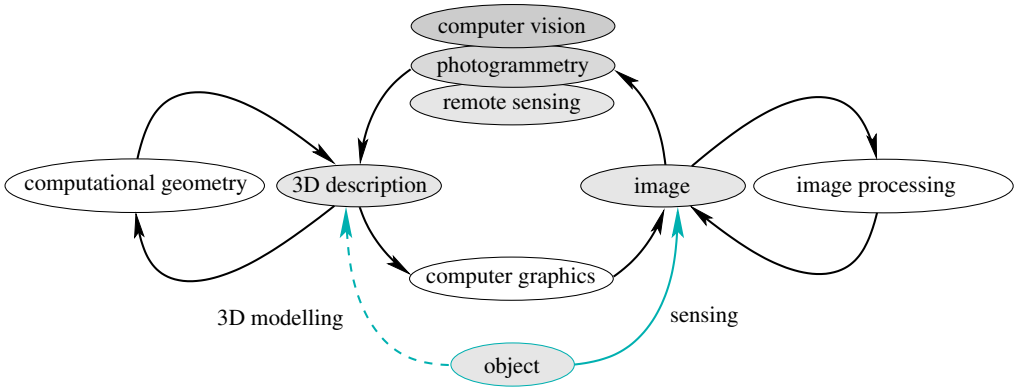


Fig. 1.1 Computer internal processing of 3D descriptions of objects and their images touches several disciplines: *image processing* (right) transforming images to images, with goals such as noise suppression, coding, warping or generating computer tomographic images; *computational geometry* (left) transforming 3D objects, with tasks such as constructing 3D objects in a CAD system, ray tracing or path planning; *computer graphics* (bottom) generating realistic images from 3D scenes consisting of objects – mimicking the process of physical *sensing* – with tasks such as photo-realistic rendering, computer animation, or visualization of physical phenomena. The inverse process deriving scene information from images is addressed by *computer vision, photogrammetry, or remote sensing* (top) with tasks such as understanding videos, generating maps from aerial images or inferring environmental parameters from satellite images – all supporting 3D modelling

deal with the hardware technology as such, e.g., the man–machine interfaces for supporting practical mapping tasks. However, to perform image analysis, they use knowledge about the physics of the image formation process, in the context of this volume specifically about the *physical and geometrical structure of cameras and laser range scanners*.

In the following we illustrate classical tasks of photogrammetric computer vision, and elaborate on our view of modelling for automatic image analysis, which motivates the structure of the book and suggests how to use it.

1.1 Tasks for Photogrammetric Computer Vision

1.1.1 Data Capture for GIS

The generation of topographic maps since the 1930s is based on aerial images, taken with specialized cameras on aeroplanes from altitudes up to ten kilometres. The maps first were manually drawn on paper; today the still mainly manually acquired information is stored in geoinformation systems, at whose core is a spatial database. The information is used, among other things, for navigation systems or urban planning (cf. Fig. 1.2, top row). Since the 1970s satellite scanners provide digital data from space with resolutions on the ground between a kilometre for meteorological purposes and below a meter for cartographic purposes. For a few decades mobile mapping systems, which acquire images and also laser scan data from moving vehicles, are used to regionally obtain high-resolution 3D models, especially of cities (Fig. 1.2, second row). Only recently have unmanned aerial vehicles reached the civil market, supporting local mapping on demand.

The most elementary product derived by photogrammetry is the elevation model, in analogue maps presented by contour lines, since the 1980s realized as the digital elevation model (DEM). It refers to what is called the topographic surface, i.e., the bare ground without buildings and vegetation (Fig. 1.2, third row). Automatic methods based on aerial images in a first instance derive the visible surface represented as the digital surface model

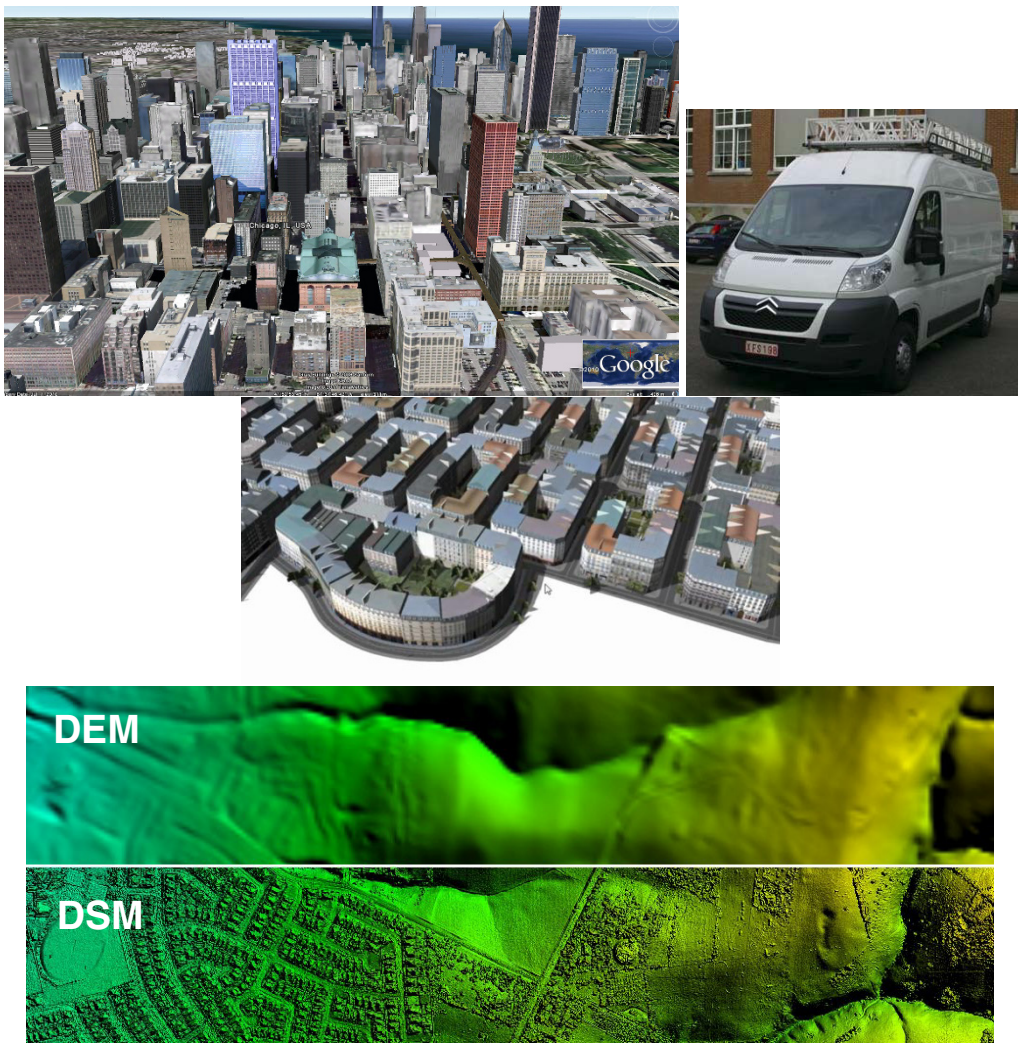


Fig. 1.2 Photogrammetric computer vision for GIS data capture. **First row:** 3D-view of Chicago in Google Earth. **Second row:** Rendered 3D city model aquired with mobile mapping system of VarCity. (Sources: top: Google Maps, left: http://www.ngi.be/Common/articles/eurosdr_sem/5_Maarten_Vergauwen.pdf right: http://www.varcity.eu/pix/cityengine_procedural_640.jpg. **Third row:** Topographic surface, surface of the bare earth (digital elevation model (DEM)). **Fourth row:** Visible surface (digital surface model (DSM)), pictures courtesy of Aerometrex.)

(DSM) (Fig. 1.2, fourth row). The DEM can be derived from the DSM by appropriate filter techniques.

The processing pipeline is pretty standardized, cf. Fig. 1.3. In most cases image capture follows a prespecified flight plan in order to guarantee full coverage, reliability of object identification and accuracy of pose and form of the acquired objects. Determining the camera's poses or orientation, i.e., position and attitude, usually is supported by the global positioning system (GPS) and often by inertial measurement units (IMUs). Cameras are manufactured such that the perspective model is valid up to small distortions, which are determined by precalibration. All orientation parameters as well as a large number of well-identifiable scene points are optimally estimated in statistical terms using what is called the bundle adjustment. It exploits the constraint that corresponding rays of the bundles of projection rays intersect. In essence, the old idea of geodetic triangulation (cf. Gauss, 1903) is applied in 3D. Given the images and their poses, non-semantic information such as digital elevation models or 3D surface models can be derived automatically from two or more images using automated stereo techniques. Deriving semantic scene information,

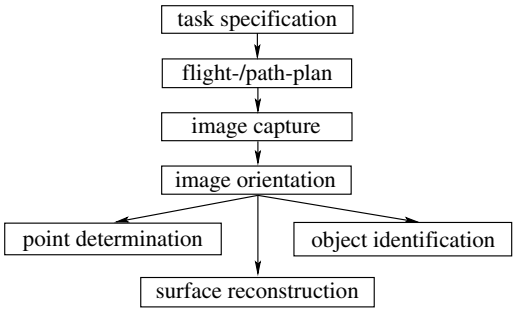


Fig. 1.3 Photogrammetric pipeline

i.e., objects of prespecified classes, up to now is predominantly derived by exploiting human interpretation capabilities. Automatic pattern recognition mainly is used to evaluate satellite images, e.g., for land cover classification.

Besides the management of the huge amount of data, three technical problems have to be solved when deriving maps from images: (1) finding the pose or orientation of the cameras at the times of image exposure with an accuracy and reliability which allows us to proceed to step (2), reconstructing the 3D form of the earth's or the visible surface from multiple images using stereo techniques, and (3) interpreting the images according to the legend of the envisaged map, which is specified by the user.

1.1.2 Vision Metrology

Having its roots in geodesy and surveying, photogrammetry also is used for high-precision metrology. This refers to two important tasks, (1) 3D scene point determination and (2) surface reconstruction.

1.1.2.1 3D Scene Point Determination

If the user is interested in specific points, say border points of land parcels or points on a machine part, they have to be either targeted or manually identified by an operator.

Automation of target detection is simplified with coded targets, which are regularly used in industrial inspection (Fig. 1.4, top row), where a mixture of coded and non-coded targets is utilized. The object is captured with high-resolution digital cameras from a large number of directions, which guarantees a reliable and accurate determination of all targeted points. Their coordinates are compared to their nominal values in order to check the production of the object. Relying on proper calibration of the cameras and on proper image resolution, relative accuracies of 1:150 000 can be realized, thereby referring accuracy to the object size.

If the object cannot be targeted or surface features are to be inspected, identification of surface points or lines can be realized by an operator using specific probes, as shown in Fig. 1.4, bottom row, observed and positioned in realtime by one or several video cameras which measure the 3D position of LED lights mounted at the probe. Depending on the form of the probe, this mensuration mode allows us to check parts of the surface which are not directly visible, e.g., the interior of cylinders. With such systems typical relative accuracies of the measured 3D coordinates of 1:50 000 can be achieved.

Especially for large objects, the bringing of the mensuration device to the object is the main advantage of visual inspection systems in contrast to systems where the object has to be brought into a special inspection lab with mechanical mensuration tools.

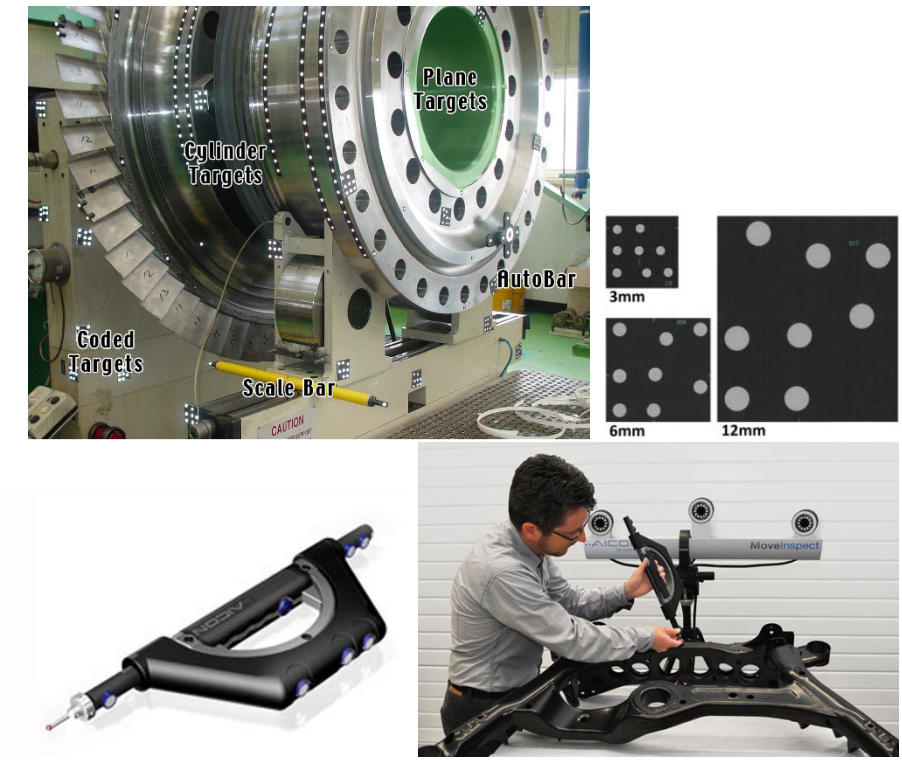


Fig. 1.4 Photogrammetric computer vision for metrology. **Top row:** High precision inspection of a turbofan (left) with retro reflective targets using the System V-Star from Geodetic Systems (Photo courtesy of Geodetic Systems, Melbourne, Florida, USA. and GANCELL Pty. Ltd., Moonee Ponds, Victoria, Australia). The targets allow the system to automatically identify and locate points of interest. Such targets are coded by the spatial arrangement of circles, cf. the example p. 272. Three of them with different patterns and circle sizes are shown in the upper right. **Bottom row:** Aicon's System MoveInspect. The probe (left) is positioned at 3D points by an operator. Its LED lights are used to determine the 3D pose, i.e., the six degrees of freedom, translation and rotation, in real-time with high precision using the three cameras (right). The system allows static or dynamic measurements

1.1.2.2 Surface Reconstruction

Surface reconstruction from images – relevant for deriving digital elevation models for geoinformation systems – in vision metrology relies on natural or, more often, on artificial texture.

In order to guarantee a reliable reconstruction of surfaces, with low or no texture, an appropriate dense texture may be projected onto the object's surface and observed in one or more images (Fig. 1.5). If the projected texture is unknown, it needs to be observed by at least two cameras, as realized early in the system Indusurf by Zeiss (Schewe, 1988). Otherwise one camera is sufficient to derive the surface structure, as realized in the depth camera Kinect by Microsoft, where the known calibrated pattern is projected using infrared light so as not to interfere with the image of an additional colour camera.

User requirements mostly are specified by tolerances, by agreeing on a certain significance level, i.e., accepting a certain but low percentage of small deviations. Furthermore, tolerances refer to the standard metre. Photogrammetric systems therefore need to (1) allow a transfer of a standard metre to the coordinates derived from image measurements, which has to be guaranteed by external control points in a coordinate system defined by the user, and (2) guarantee the required reliability and accuracy. This is why statistical techniques for design, estimation and testing are provided when developing vision metrology systems.

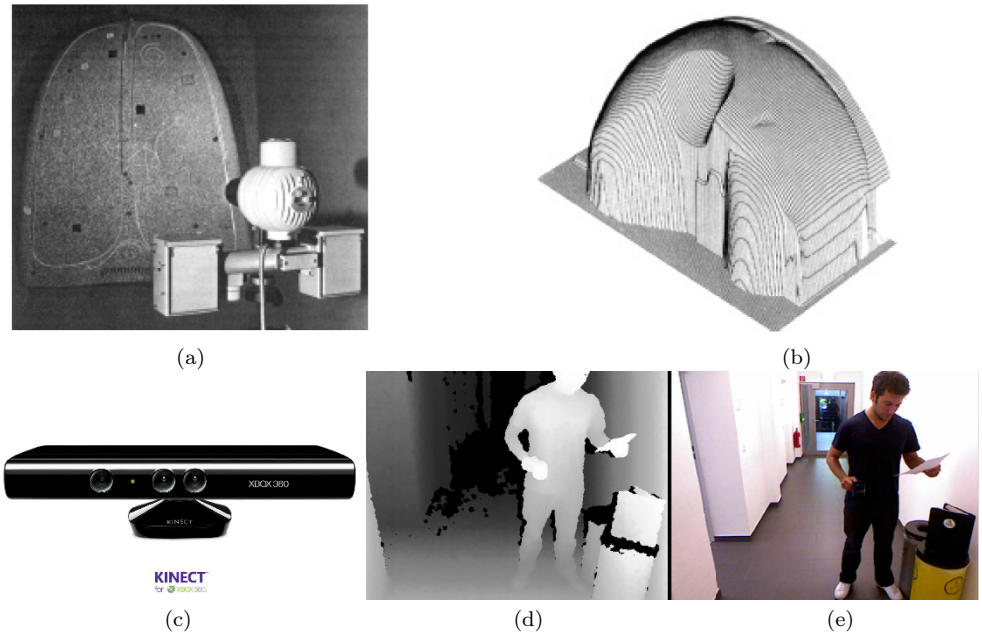


Fig. 1.5 Surface reconstruction systems using a texture projector and cameras. **Top row:** (a) The system Indusurf by Zeiss (Schewe, 1988) was designed to capture the form of industrial objects, here parts of a car body for reverse engineering: (b) the inner part of a wheel house. The analogue images were digitized and evaluated automatically. **Bottom row:** Microsoft's Kinect (c) as man–machine interface. It captures a depth map (d) and an intensity map (e) in real time (Steinke, 2012, Fig. 5)

1.2 Modelling in Photogrammetric Computer Vision

Automated systems for photogrammetric computer vision require adequate models. Depending on the task to be solved such models refer to the complete processing chain, from capturing data to their final interpretation. As this volume addresses automatic methods for orientation and reconstruction we will, besides referring to the general view, always refer to specific tasks dealt within this book. Tasks requiring methods from image processing or interpretation which are presented in the second volume, however, are kept in mind when discussing modelling aspects in the following.

1.2.1 A Meta Model of Photogrammetric Computer Vision

A meta model for photogrammetric computer vision is shown in Fig. 1.6. It enrolls the relation between the generation of images and their analysis, required for establishing proper algorithmic models, and explicitly relates these models to the application within a real scenario. This meta model therefore is the basis for our view on building models for vision algorithms.

Though we restrict ourselves to the geometric analysis of images in this volume the meta model is general enough for more general tasks: It can be used for very simple tasks such as the detection of the horizon in the image of an urban scene, tasks of medium complexity such as the determination of the pose of a set of cameras, or quite complex tasks such as the identification of the road network in a satellite image. We will take an excerpt of this model for discussing orientation and reconstruction procedures in Part III, cf. Fig. (11.1), p. 442.

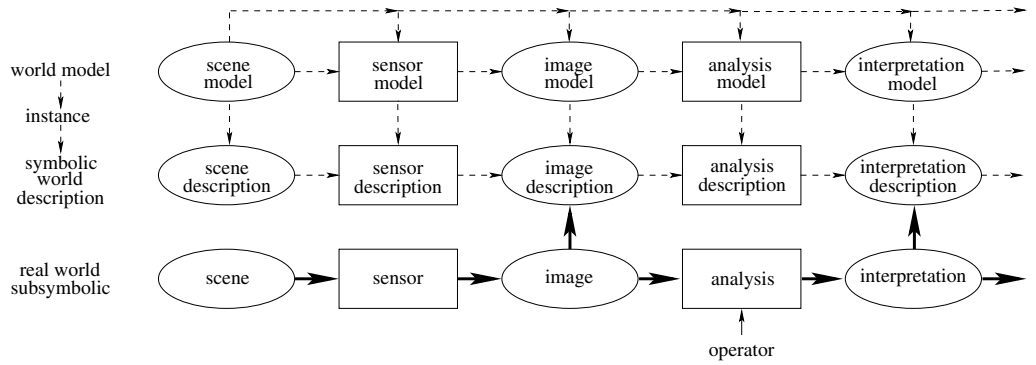


Fig. 1.6 Components of a model for image interpretation consisting of three levels. **Lowest level**: the real world from the scene to the interpretation by a human operator, not necessarily described by human related notions (subsymbolic). **Middle level**: Description of the real world using words, symbols or mathematical structures representing certain aspects of the real world, relevant for the application. **Upper level**: Models for describing the real world using an adequate language, a useful set of symbols or a mathematical model. The elements of the middle level are instances of the concepts of the upper level. Relevant parts of the world, from left to right: the scene, i.e., the object of interest, the used sensor, the generated images, the analysis process or, synonymously, interpretation process and the result of the analysis or interpretation. The results of the interpretation may be fed into a visualization tool not made explicit here. The dashed paths in the upper level indicate the sequence of modelling, where each model possibly is relevant for all following ones. The dashed paths in the middle level indicate information flow. The vertical dashed arrows indicate instantiation. The sequence indicated by thick arrows is the one realized by an image analysis or interpretation system. Adapted from Förstner (1993)

Figure 1.6 depicts three levels of reasoning:

1. The subsymbolic level, i.e., the *real world*. The world is assumed to exist independently of a symbolic or linguistic description of any kind whatsoever. This leaves open all aspects of possible use of the world and the relevance of parts or aspects of the world.
2. The symbolic level of a *world description*. It is chosen such that it is adequate for the task to be solved. It is symbolic in the sense that it may be described by names, numbers, etc., often referring to parts of the world and their relations.
3. The symbolic level of a *world model*, which specifies the language for the symbolic description. The model contains variables for entities and their relations, which, when describing the world, are instantiated, i.e., given specific values.
   We strictly distinguish between the world model, a structure of variables, and the world description containing fixed values. For example, a *digital surface model* representing a part of the earth's surface is a specific symbolic scene description.

There is no describable relation between the world model, and consequently the world description, and the real world, as the complete model of the world is unknown.

Figure 1.6 contains the path from the scene on the subsymbolic level, via the sensors and the images, e.g., symbolically described by a set of image points and lines, to arrive, via the image analysis (e.g., a triangulation procedure) at an interpretation of the scene, e.g., described by a set of 3D points or lines. To design the analysis process we need models for all components involved, namely the scene, the sensors, the images, the analysis process and the interpretation result.

Observe, we take the sensors as being separate from the scene, which appears adequate for our application domain. An integrated view would be necessary when embedding a set of robots having visual sensors and actuators into a scene for learning visual perception in a perception-action cycle, where learning the properties of an object is supported by manipulation of that object, e.g., its passing a cup of milk to another robot, and seeing its effects on the scene. We do not address these aspects of artificial perception.

We will specialize Fig. 1.6, when discussing orientation and reconstruction procedures in Part III.

1.2.2 Models for Imaging and Interpretation

The imaging model in Fig. 1.7a is a specialization of the general scheme from the scene to the images in Fig. 1.6. It makes explicit the components involved, the scene, the cameras and the environment and the physical nature of the imaging process.

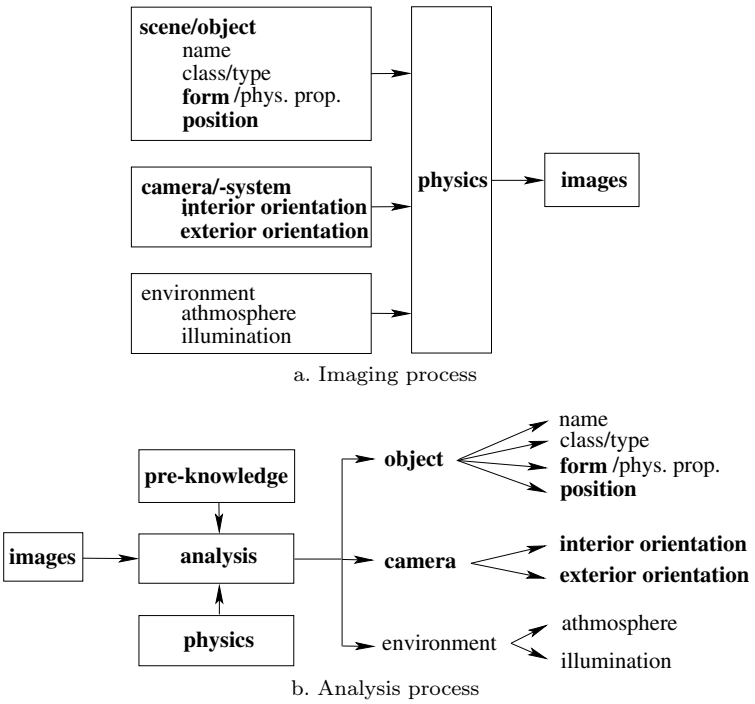


Fig. 1.7 Imaging process and its inversion, the analysis process. **a.)** The scene is supposed to consist of identifiable objects. The camera (or more general the cameras) can be described by its (their) interior and exterior orientation. The environment, especially the lighting conditions, possibly the atmospheric situation, influences the imaging process. Physical laws dictate the image model. This includes all optical laws, especially those of geometric optics. **b.)** Given some images, knowledge about the physical imaging process is needed and pre-knowledge about the scene in order to recover descriptions of the objects, the cameras and the environment. This knowledge may consist of the basic geometry of the internal structure of the cameras. It may consist of the form of the object or of complex models for the illumination and the structure and appearance of the scene and its parts. This book addresses the concepts shown in bold letters, the other concepts are planned to be discussed in the second volume

Image analysis (Fig. 1.7b) can be seen as an inversion of the sensing model leading to an interpretation, starting from *images* in the sense of an imaging description. In general these are intensity, colour, multi-spectral, or depth images given in a grid or an irregular structure that is a field-based representation. In the context of geometric image analysis the result may consist of image features, such as points or lines, possibly their mutual relations, and their identity, derived by some image preprocessing including key point and line extraction and correspondence analysis, possibly leading to a geometric scene description that is an object-based representation. Identifying objects or object classes may be based directly on the image information, on derived image features or on a geometric-radiometric scene description.

Developing an analysis tool requires an adequate *physical model*, used in the imaging model. Since this is an inverse problem we also need *pre-knowledge* about the scene in order to arrive at unique interpretation results. In general this refers to all aspects of the scene model, including geometry, material properties and semantic classes.

In the context of this volume we use known geometric surface properties, e.g., the smoothness of the observed surface. We specifically require a one-to-one mapping between

the scene and the images. Thus we assume the correspondence problem has been solved at least partially, in the sense that putative matches between features in different images or between image and scene elements are available by some automatic, semi-automatic or manual procedure, which will be discussed in detail in the second volume. Finally, we assume the uncertainty of the given image features to be known to an adequate fidelity.

correspondence problem, matching

The model for the analysis contains all methods for inverting the imaging process. In general this may be as simple as a classification of the given image or of certain regions of the image, based on local image properties such as colour or colour variations, to obtain the class labels, e.g., sky, building, road and vegetation. Then we can expect to arrive at methods which are optimal in some statistical sense. Or the mode may be highly complex when extracting cartographic features from aerial images, such as roads or buildings, or complex geographic entities such as biotopes or arid areas. Methods solving such complex analysis tasks often do not have a complete model addressing all aspects, but use well-understood methods for a stepwise solution.

In our context of geometric image analysis the situation is similar. We have statistical and geometrical analysis models for parameter estimation problems, which are well-understood, if the data are not contaminated by outliers and we have sufficiently accurate approximate values for the parameters. If these conditions are not fulfilled, we also arrive at sequences of methods whose mutual links are understood fairly well, and contain well-motivated heuristics.

statistical and geometrical models

The aim of the analysis is an interpretation of the given images. In general it refers to all aspects of the scene, the cameras and the environment, e.g., the light sources, and is modelled the same way as when deriving the image model. The notion 'interpretation' suggests that the result may not be unique. This is true, as it depends on the task specific pre-knowledge made available; thus, results may differ widely due to different aspects induced by the application.

interpretation

In our context the result in general is a – possibly large – set of parameters. In many cases it also depends on the given pre-knowledge, e.g., the assumed accuracy of the image features or the outlier model. Though in most cases there is a canonical way to solve the orientation and reconstruction tasks, results may differ due to different pre-knowledge, e.g., if the geometric configuration is close to singular. This especially holds in the presence of outliers, where we – in a very restricted sense – have an interpretation problem, namely that of classifying observations as in- and outliers.

1.2.3 Probabilistic and Statistical Reasoning

Uncertainty of all states and processes needs to be represented within an automatic analysis procedure. Therefore we have a look at the process of modelling when solving a certain interpretation task.

Let us start with a simple example. Take the drawing of the cover of the book, represented in Fig. 1.8. It shows uncertain 3D points $\mathcal{X}_i$, uncertain centres $\mathcal{O}_t$ of pinhole cameras, and uncertain image points $\mathcal{x}'_{it}$.

Given two of the three – scene points, pinhole centres or image points – we have three geometric analysis tasks, namely deriving the other, provided we have enough 3D points visible in two or more images:

- Given uncertain scene points and uncertain poses of the cameras, which above the pinhole centres include their rotational components, the task is to derive the image points together with their uncertainty. This may be useful when checking the validity of measurements of image points w.r.t. the already given information.
- Given uncertain image points and uncertain camera poses, the task is to derive the scene points together with their uncertainty, e.g., by finding the scene point closest to the projection rays $\mathcal{O}_t\mathcal{X}_i$. This is the central task of scene reconstruction, assuming the scene can be represented as a set of 3D points.

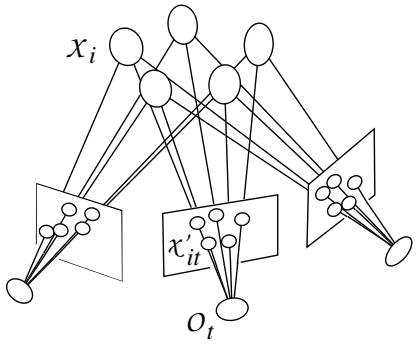


Fig. 1.8 Uncertain geometry in geometric image analysis. All entities involved may be statistically uncertain: 3D points $\mathcal{X}_i$, centres $\mathcal{O}_t$ of pinhole cameras, and the image points x'_{it} in the image planes

- Given uncertain scene points and corresponding uncertain image points the task is to derive the poses of the cameras during image capture together with their uncertainty. When dealing with a stream of images taken by a video camera this is the classical task of (ego) motion determination.

  The situation can be generalized: if only parts of the scene points are given the task is to simultaneously determine the poses of the cameras during image capture and the unknown scene points using bundle adjustment. When performed in real time, the task sometimes is called *simultaneous localization and mapping*, localization referring to the determination of the camera poses and mapping referring to the determination of 3D scene points. Also in this case the resultant poses and 3D positions will be uncertain.

Therefore, we need to be able to adequately represent, estimate and propagate the uncertainty of the geometric entities involved, namely the 2D and 3D points, the poses of the cameras in space and possibly the projection rays $\mathcal{O}_t\mathcal{X}_i$.

Figure 1.9 sketches the reasoning network during the performance of a possibly complex *task*, e.g., the reconstruction of a 3D scene from images with a required accuracy.

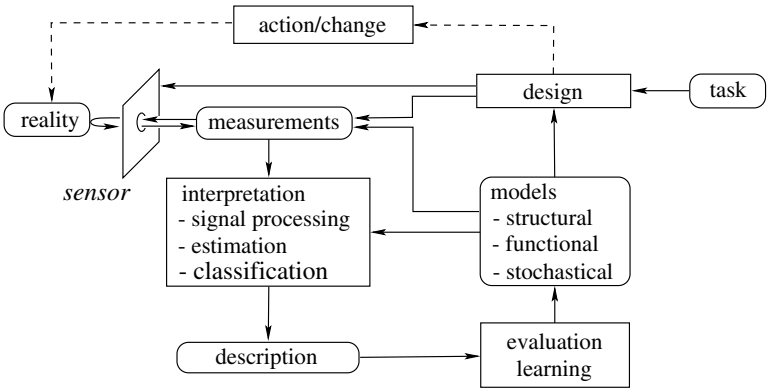


Fig. 1.9 Observation–analysis–modelling–planning loop. Reality can be seen as "unregarded", except when answering "questions" of a sensor. The resulting measurements are interpreted following models about reality and its interaction with the sensors, and which are to be chosen according to the design, such that they are adequate for solving a certain task

Following a certain design, e.g., choice and arrangement of sensors and 3D scene points, *measurements* are treated as answers by the – otherwise unregarded – reality to certain sensor requests, e.g., for the coordinates of an image point or the direction to a scene point. These measurements are used to derive a *description* of the world in terms of the envisaged set of points, or – similarly – surfaces, properties or object classes using methods from signal processing, estimation or decision theory. This interpretation process is guided by the

available *models*, which in general have structural, functional, and statistical components and can be viewed as prior information about the relation between measurements and the description of reality, e.g., the light rays are straight, i.e., the scene point, the pinhole and the image point are collinear. Strictly speaking, already the measurements are interpretations of reality based on the sensor model, following Einstein's statement: *Die Theorie bestimmt, was wir beobachten können*,[1] cited in Watzlawick (1978, p. 70). The resultant desciption is *evaluated* and used to *learn*, i.e., update components of the model, e.g., using the discrepancies between the measurements and the model: either compensating the directions due to lens distortion or correcting the assumptions about the measurement precision. Given a task, the measurement process is *designed* to fulfil the user's requirements. This design may just be the choice of the type and the position of available sensors. But it may also refer to the architecture of the sensors, including some processing, easing the interpretation process. The design may also refer to actions in or to changes of reality, which then may be checked by a specific measurement and interpretation process, a topic not addressed in this book.

Uncertainty refers to all steps in the observation and analysis chain. Measurements cannot be repeated perfectly; deviations may be small, e.g., due to electronic noise, or large, e.g., due to the coarseness of the specified design or outliers. Models of reality, which never are true, simplify reality w.r.t. certain aspects relevant for the task, e.g., reducing spatial objects such as buildings to polyhedra, neglecting not only physical and geometrical details but also the class the building belongs to. Models of sensors, which are models of a specific part of reality, also are sometimes crude approximations, e.g., when using the pinhole camera model for real cameras.

We will use probabilistic and statistical tools for handling uncertainty. This choice has a twofold motivation: (1) Probability theory allows us to map uncertainty to crisp mathematical relations, based on Kolmogorov's axioms. (2) Statistics allows us to relate measurements to models for evaluating, selecting and learning, thus improving models.

1.3 The Book

1.3.1 The Structure of the Book

The book consists of three parts.

I. Part I on *Statistics and Estimation* (Fig. 1.10) covers elements from probability theory, especially continuous random variables, their properties and dependencies, necessary for describing measurements and for performing parameter estimation and its evaluation, based on statistical testing. These methods are used in Part II, especially in Chap. 10 for uncertain geometric reasoning, namely uncertainty propagation, testing geometric relations and estimating geometric entities and transformations. These methods are basic for all orientation and reconstruction procedures discussed in Part III, and allow us to explicitly describe the uncertainty of the results with respect to random or systematic model deviations and to perform mensuration design to guarantee user requirements.

II. Part II on *Geometry* (Fig. 1.11)covers projective geometry in 2D and 3D, especially linear and quadratic elements and their transformations together with the representation and exploitation of their uncertainty. These representations and methods are useful for 2D and 3D reconstruction based on geometric elements, such as 2D straight line segments or 3D point clouds and their transformation, including the modelling and estimation of 3D rotations. The methods are essential for modelling and orientation of cameras. As far as possible geometric elements and transformations are

[1] "It is the theory that decides what we can observe".

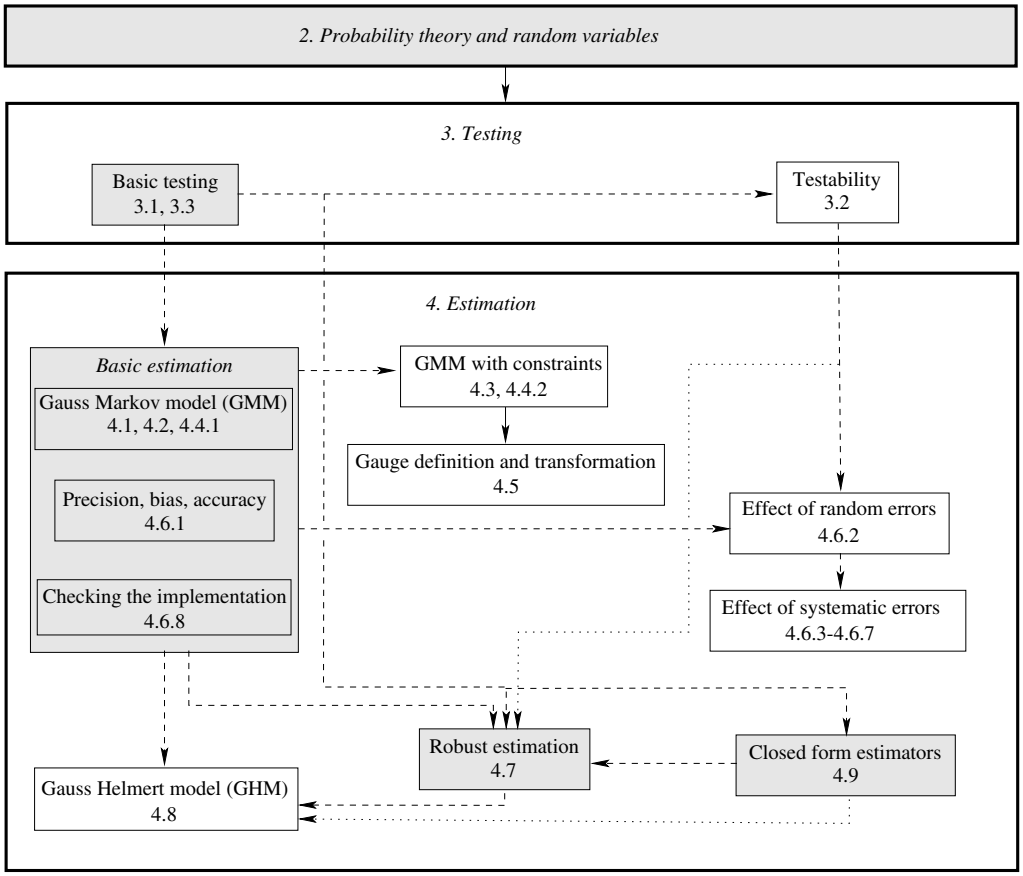


Fig. 1.10 Structure of Part I, GMM = Gauss–Markov model. Topics useful in basic courses are in grey boxes, advanced topics in white boxes.

treated as oriented, useful for transferring knowledge about direction and right/left or inside/outside relations through the reasoning chain.

III. Part III on *Orientation and Reconstruction* (Fig. 1.12) addresses aspects of single and multiple view analysis, especially camera calibration and orientation, and the reconstruction of scene elements and surfaces. We assume the images are represented by image features, say points or line segments, and the correspondence problem is at least partially solved. The discussed methods cover (1) direct solutions not assuming approximate values, which are useful for outlier detection and (2) statistically optimal solutions which exploit all information about the uncertainty of the given measurements and the assumed models. These allow proper planning of measurements and finally an accuracy and reliability evaluation of the results w.r.t. user requirements.

The appendix contains basic material mainly from linear algebra. Navigation through the book is supported by an extensive index.

1.3.2 How to Use This Book

Large parts of the book are the result of lectures offered within geodetic Bachelor's and Master's courses, which is why it covers more than a single course. Figures 1.10–1.12 show the internal dependencies of the chapters. Topics useful in undergraduate courses have a grey background. The numbers indicate the sections, including the corresponding introductions.

1.3.2.1 Standalone Chapters

Clusters of chapters can be used as stand alone text.

- Part I on its own is useful for problems using parameter estimation, evaluation and design, also other problems not discussed in this book.
- Sects. 5.1 to Chap. 9 treat projective geometry of 2D and 3D without using the uncertainty of geometric elements and transformations. It includes a Chap. 9 on oriented projective geometry.
- Parts I and II can be seen as covering uncertain projective geometry without explicitly discussing the uncertainty of the imaging process using cameras.

1.3.2.2 Basic Courses

Basic courses useful for undergraduate students may address the following topics. We refer to the basic section shown in Figs. 1.10 to 1.12 and only provide details down to the section level. If the basics of Chaps. 1, 2 and 3 are known from other courses, they may be omitted, or perhaps summarized.

- Single image orientation.
  - [I] Basics of Chaps. 2, 3, and 4.
  - [II] Basics of Chaps. 5 and 6, Chap. 7, Sects. 10.1–2, 10.4–5.
  - [III] Essence of Chap. 11, Sects. 12.1–2.
- Orientation of the image pair
  - Single image orientation (see above).
  - [I] Estimation in the Gauss–Helmert model Sect. 4.8.
  - [III] Basics of Chap. 13.
- Uncertain projective geometry in the plane (vanishing points, homography)
  - [I] Basics of Chaps. 2, 3 and 4, possibly Sect. 4.8.
  - [II] Basics of Chaps. 5, 6, 7 and Chap. 10.
- Random Sample Census (RANSAC) for identifying inliers
  - [I] Basics of Chaps. 2, 3 and 4, Sect. 4.8.
  - Applications: Fitting a circle (in Sect. 4.9), fitting a straight line (in Sect. 7.4), spatial resection (in Sect. 12.2), estimating the fundamental matrix (in Sect. 13.3), spatial intersection (in Sect. 13.4).
- Elements of bundle adjustment
  - Orientation of the image pair (see above).
  - Basics of Chap. 15.

1.3.2.3 Advanced Courses

The following advanced courses either require knowledge of the more general estimation scheme with the Gauss–Helmert model or address issues which appear more adequate for graduate students.

- Quality of different estimators
  - [I] Basics of Chaps. 2, 3 and 4, possibly Sect. 4.5.
  - [II] Short course Sect. 5.1, Chap. 10.

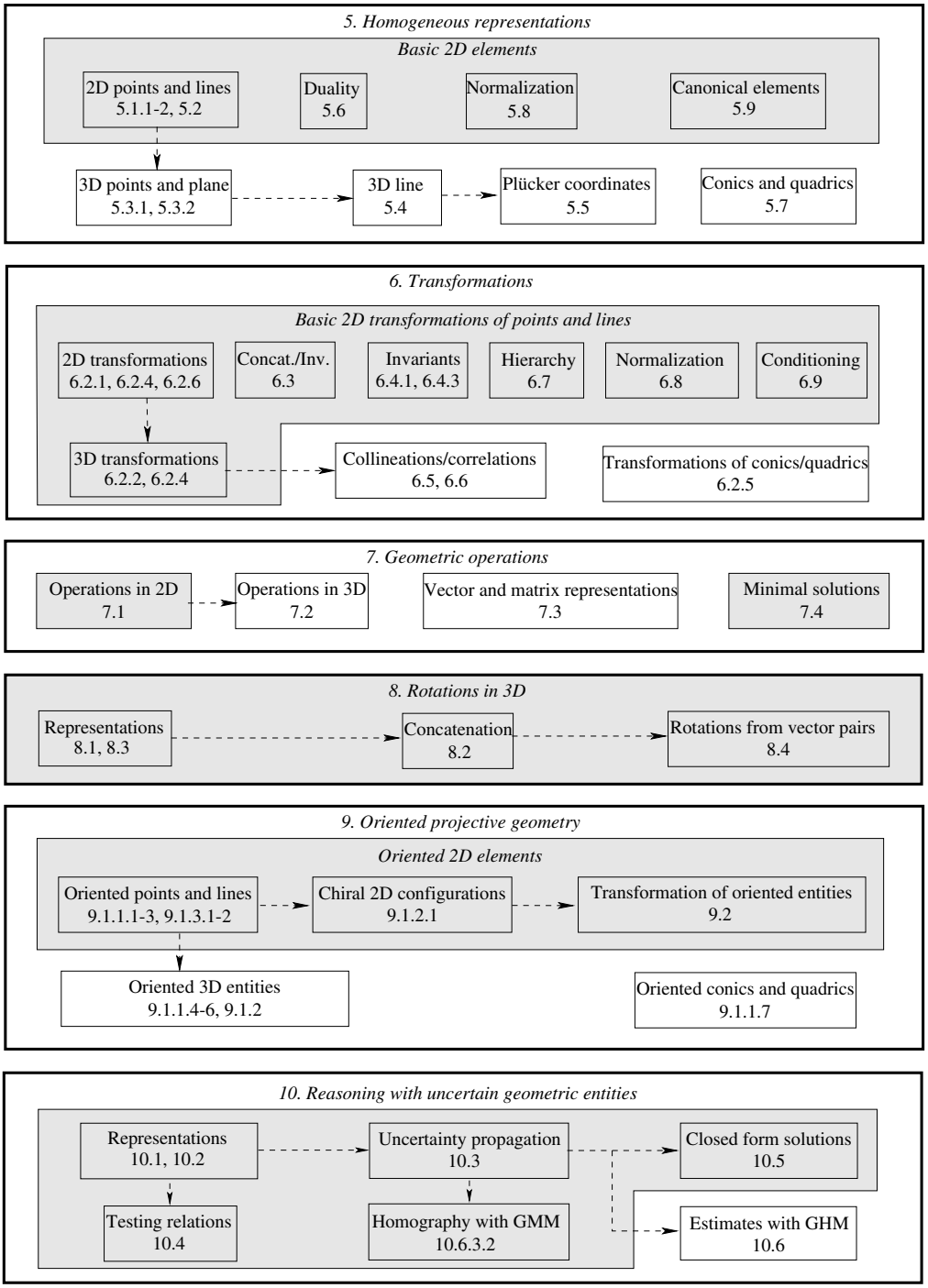


Fig. 1.11 Structure of Part II, GMM = Gauss–Markov model. Topics useful in basic courses are in grey boxes, advanced topics in white boxes.

  - Applications in Sect. 10.5: intersecting straight 2D lines, estimating a 2D homography, additional applications, e.g., circle or ellipse fitting (in Sect. 4.6).

- The image triplet
  - Orientation of the image pair (see above).
  - [I] Estimation in the Gauss–Helmert model Sect. 4.5.
  - [III] Chap. 14.

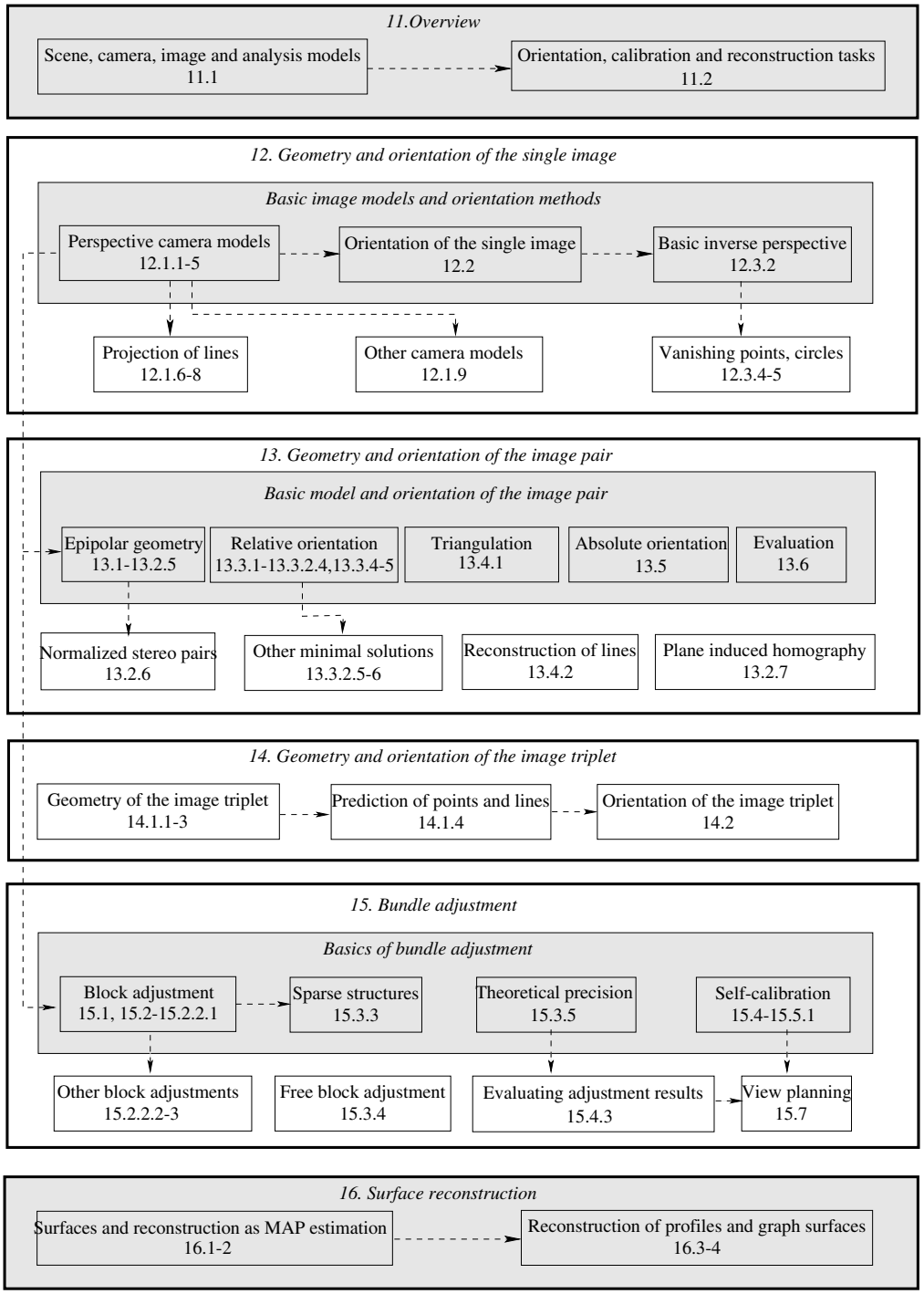


Fig. 1.12 Structure of Part III. Topics useful in basic courses are in grey boxes, advanced topics in white boxes.

- Camera calibration using self-calibrating bundle adjustment
  - Elements of bundle adjustment (see above).
  - [I] Advanced methods in Chaps. 2, 3, and 4.
  - [III] Evaluating block adjustments, view planning.

1.3.2.4 Exercises

Exercises are provided for each chapter. The first part of the exercises is meant to directly help to understand the basic text. Some of the proofs, which use previous results and give insight into the statement to be proven, are shifted into the exercises. Computer experiments help to understand procedures and techniques for evaluating results of algorithms. An approximate time estimate is given for each exercise:

(1) The exercise can be done while reading using a pencil and paper. It is meant to take less than 15 minutes.
(2) The exercise takes a while, usually below one hour.
(3) The exercise is an extended one.

Some exercises require additional data or code, which is given on the home page of the book at the link

http://www.ipb.uni-bonn.de/book-pcv/

References to the exercises are given in teletext font as `HOME/file_name.ending`, where `HOME` stands for `http://www.ipb.uni-bonn.de/book-pcv/exercises`.

1.4 On Notation

The notation used in this book is chosen to make the types of the variables explicit.

As geometric entities may be represented in different ways, e.g., a 2D line $\mathcal{l}$ using the slope and intercept (m, k) or the two parameters (ϕ, d) of the Hessian form, we distinguish between the name $\mathcal{l}$ of the geometric entity, using calligraphic letters, and its representation. Thus $\mathcal{l}(m, k)$ and $\mathcal{l}(\phi, d)$ represent the same line represented differently.

The choice of the variable names refers to two contexts, which are made as explicit as possible: estimation theory and geometry. As an example, the symbol $\boldsymbol{x}$ has two meanings: it refers to (1) the parameter of unknown values within estimation theory and (2) the coordinates of a 2D point in geometry. Switching between the two is made explicit by an assignment. E.g. if $\boldsymbol{h} = [\phi, d]^\mathsf{T}$ is the vector of the Hessian parameters of a 2D line, the assignment $\boldsymbol{x} := \boldsymbol{h}$ indicates that the vector $\boldsymbol{h}$, which represents some geometric entity, is assigned to the unknown vector $\boldsymbol{x}$ within an estimation process.

We used the convention of what is called the Dutch school for indicating random variables by underscoring (see Abadir and Magnus, 2002, Sect. 6). For details see the list of symbols.

The convention is used that if a matrix is square or has more rows than columns, we then denote it by A, and if a matrix is square or has fewer columns than rows, we then denote it by A^T. Vectors are column vectors. Exceptions will be indicated.

Variable and function names longer than one symbol are written with upright letters, such as ‘sin’ or ‘BIC’.

Part I
Statistics and Estimation

Probability theory handles uncertainty representation and reasoning. Linking probabilities with real observations is the domain of *statistics*.

In our context of photogrammetric computer vision, we have three types of tasks, where we need tools from probability theory and statistics:

1. Given the outcome of an experiment and a parametric model for the underlying probability, determine the parameters of that model.

   In the case of continuous parameters, this problem is solved by *parameter estimation*. Depending on the model and the optimization criterion, we may distinguish between several principles for estimation.

   In the case of discrete parameters, the problem is solved by *classification*.

   That a model simultaneously depends on both, continuous and discrete parameters, often occurs when there are several alternative models. Then we have the problem of *model selection*. Thus, estimation and classification may occur simultaneously.

   In this volume we are only concerned with parameter estimation. We discuss basic concepts of model selection. Classification is dealt with in the second volume.
2. Given the outcome of an experiment and a hypothesis for the underlying probability, decide whether there are reasons to reject the hypothesis. This problem is solved by *hypothesis testing*. As such decisions may be uncertain, we need to investigate the effect of making wrong decisions, e.g., caused by weaknesses in the configuration of the measurements. *Diagnostics* provide safeguards against situations which are sensitive to deviations from the underlying probability.
3. Finally, when planning an experiment with the goal to achieve a certain quality of the result, we use probability theory for *predicting the expected precision and accuracy*.

Part I collects the necessary concepts and tools to perform parameter estimation and hypothesis testing. Chap. 2, p. 21 describes the basic concepts of using random variables and their statistical distributions, including the propagation of uncertainty and the representation and visualization of the uncertainty of random variables. Chap. 3, p. 61 deals with hypothesis testing, which is useful for identifying possible deviations from assumptions of a given distribution, and tools for measuring the ability to perform such tests. Chap. 4, p. 75 describes the various forms of parameter estimation useful within photogrammetry and computer vision, including the means necessary to evaluate the resultant estimates and to handle deviations from the underlying models.

We assume that the reader is familiar with the basic notions of probability theory.

Chapter 2
Probability Theory and Random Variables

2.1 Notions of Probability 21
2.2 Axiomatic Definition of Probability 22
2.3 Random Variables 24
2.4 Distributions 28
2.5 Moments 36
2.6 Quantiles of a Distribution 40
2.7 Functions of Random Variables 40
2.8 Stochastic Processes 48
2.9 Generating Random Numbers 55
2.10 Exercises 56

This chapter collects the basic terms from probability theory and statistics. It motivates the axiomatic approach for the concept of probability, introduces the concept of a random variable, describes the key properties of the main distributions of random variables occurring when modelling observational uncertainties and testing hypotheses, and provides an introduction to stochastic processes. We give the key methods for determining the uncertainty of derived entities, especially for explicit and implicit functions of single and multiple variables. The reader who has had a basic course on statistics may take a quick look at the notation used and the lines of thought employed. The concepts can be found in the excellent textbooks by Papoulis (1965) and Papoulis and Pillai (2002) and online at `http://www.math.uah.edu/stat/index.html`.

2.1 Notions of Probability

Probability theory is the most powerful tool for working with uncertainty. The notion of probability has changed over the last two centuries.

- The *classical definition* of probability P according to Laplace is the ratio of the number n_+ of favourable to the number n of possible cases of an event $\mathcal{E}$,

$$P(\mathcal{E}) \doteq \frac{n_+}{n}\,. \tag{2.1}$$

  When modelling the outcome of throwing a die, e.g., this definition leads to the usually assumed probability $1/6$ for each possible event.
  But when modelling the outcome of a modified die, e.g., one that yields more sixes, we encounter difficulties with this definition. We would need to define conditions for the different events under which they occur with the same probability, thus requiring the notion of probability.
  In the case of alternatives which are not countable, e.g., when the event is to be represented by a real number, we have difficulties in defining equally probable events.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_2

Bertrand's paradox

This is impressively demonstrated by *Bertrand's paradox* (Fig. 2.1), which answers the question: What is the probability of an *arbitrarily chosen* secant in a circle longer than the side of an inscribing equilateral triangle? We have three alternatives for specifying the experiment:

1. Choose an arbitrary point in the circle. If it lies within the concentric circle with half the radius, then the secant having this point as centre point is longer than the sides of the inscribing triangle. The probability is then $1/4$.
2. Choose an arbitrary point on the circle. The second point of the secant lies on one of the three segments inducing sectors of $60°$. If the second point lies in the middle sector the secant through these points is longer than the side of the inscribing triangle. The probability is then $1/3$.
3. Choose an arbitrary direction for the secant. If its centre point lies in one of the two centre quarters of the diameter perpendicular to this direction the secant is longer than the side of the inscribing triangle. The probability is then $1/2$.

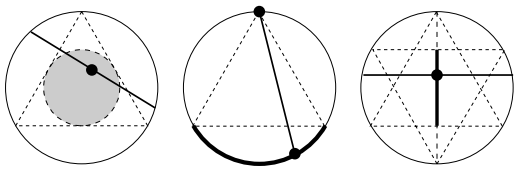


Fig. 2.1 Bertrand's paradox: Three alternatives for choosing an arbitrary secant in a circle. **Left:** choosing an arbitrary point in the small circle with half radius, and interpreting it as the middle of the secant; **Middle:** by first choosing a point on the boundary, then the second point must lie in a certain range of the boundary, namely in between the secants belonging to an equilateral triangle; **Right:** choosing an arbitrary point on a diameter, in the middle range of the secant

Obviously the definition of the notion *arbitrarily chosen*, i.e., an equal probability, is not simple. However, this definition is often used, as it follows the classical logic under certain conditions.

- The definition of probability as *relative frequency* following von Mises. This definition follows the empirical finding that the empirical relative frequency seems to converge to a limiting value
$$P(\mathcal{E}) \doteq \lim_{n\to\infty} \frac{n_+}{n}\,. \tag{2.2}$$
This plausible definition fails in practice, as the number of experiments will not be sufficiently large and the conditions for an experiment cannot be held stable over a long enough time.
- Probability as the degree of *subjective certainty*, e.g., in the sentence: *"There is a large probability this statement, A, is correct."*
Due to its subjectivity, this definition is not suitable as a basis for a theory. However, sometimes we use subjective probabilities, which then requires a rigorous definition of the concept.

All three definitions are plausible and form the basis for the following axiomatic definition.

2.2 Axiomatic Definition of Probability

The following axiomatic definition of probability follows Kolmogorov and solves the issues of the previous definitions (Fig. 2.2).

Kolmogorov's Axiomatic Definition of Probability. Basis is a space S of elementary events $A_i \in S$. Events A are subsets of S. The certain event is S, the impossible event is $\emptyset$. Each combination of events A and B again is an event; thus, the alternative event $A \cup B$, the joint event $A \cap B$ and the negated event $\overline{A} = S - A$ are events.

Each event can be characterized by a corresponding number, $P(A)$, its probability, which fulfils the following three axioms:

axiomatic definition of probability

1. For any event, we have
$$P(A) \geq 0\,. \tag{2.3}$$
2. The certain event has probability 1,
$$P(S) = 1\,. \tag{2.4}$$
3. For two mutually exclusive events, $A \cap B = \emptyset$ (Fig. 2.2, a),
$$P(A \cup B) = P(A) + P(B)\,. \tag{2.5}$$

Conditional Probability. Moreover, we have the conditional probability of an event A given the event B has occurred. The probability
$$P(A \mid B) = \frac{P(A,B)}{P(B)} \tag{2.6}$$
is the ratio of the joint probability $P(A,B) = P(A \cap B)$ of events A and B occurring simultaneously and the probability $P(B)$ of only B occurring (Fig. 2.2, b).

Total Probability. The total probability of an event A in the presence of a second event $B = \bigcup_{i=1}^{I} B_i$ therefore is (Fig. 2.2, c)
$$P(A) = \sum_{i=1}^{I} P(A \mid B_i) P(B_i)\,. \tag{2.7}$$

Independent Events. Two events A and B are called independent (Fig. 2.2, d) if
$$P(A,B) = P(A)P(B)\,. \tag{2.8}$$

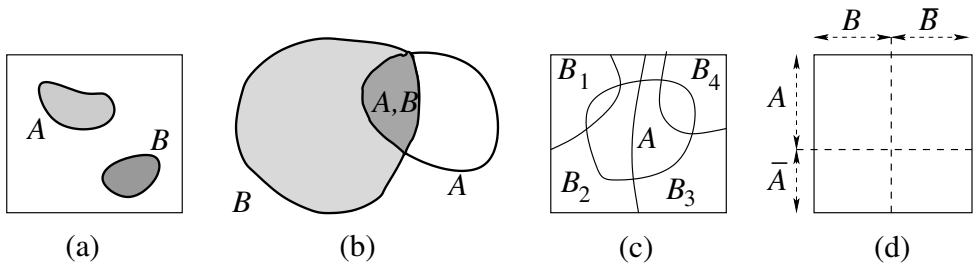


Fig. 2.2 Independence, conditional and total probability. (a) Disjoint events A and B, (b) conditional probability $P(A \mid B) = P(A,B)/P(B)$, (c) total probability $P(A)$, (d) independent events A and B

These axioms coincide with the classical definition of probability if the definition of elementary events is unique and can be considered as equally probable.

Example 2.2.1: ***Throwing a die.*** (1) When throwing a die, we have the set S of elementary events

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6\},$$

and for each s_i

$$P(s_i) = \frac{1}{6}.$$

(2) Throwing two dice i, j, we have

$$S = \{(s_i, s_j)\} \quad \text{and} \quad P((s_i, s_j)) = \frac{1}{36}.$$

(3) The *conditional probability* $P(s_2 \mid \mathsf{even})$ of throwing a 2, i.e., event s_2, under the condition that we know that an even number was thrown, and using (2.6) is

$$P(s_2 \mid \{s_2, s_4, s_6\}) = \frac{P(s_2, \{s_2, s_4, s_6\})}{P(\{s_2, s_4, s_6\})} = \frac{P(s_2)}{P(\{s_2, s_4, s_6\})} = \frac{1/6}{1/2} = \frac{1}{3}.$$

(4) The *total probability* for $A = \mathsf{even} := \{s_2, s_4, s_6\}$ (i.e., throwing an even number) if having thrown a $B_1 = \mathsf{small}$ or $B_2 = \mathsf{large}$ number (with $B = \{\mathsf{small}, \mathsf{large}\} := \{\{s_1, s_2, s_3\}, \{s_4, s_5, s_6\}\}$), is

$$P(\mathsf{even}) = P(\mathsf{even} \mid \{s_1, s_2, s_3\})P(\{s_1, s_2, s_3\}) + P(\mathsf{even} \mid \{s_4, s_5, s_6\})P(\{s_4, s_5, s_6\}) = \frac{1}{3}\frac{1}{2} + \frac{2}{3}\frac{1}{2} = \frac{1}{2}.$$

(5) The events $A = s_1$ to first throw 1 and $B = \{s_2, s_4, s_6\}$ to secondly throw even are *independent*;

$$\begin{aligned} P(1, \mathsf{even}) &= P(\{(s_1, s_2), (s_1, s_4), (s_1, s_6)\}) = \frac{3}{36} \\ &= P(1)P(\mathsf{even}) = \frac{1}{6}\frac{1}{2} = \frac{1}{12}. \end{aligned}$$

◇

2.3 Random Variables

2.3.1 Characterizing Probability Distributions 24
2.3.2 Probability Density Function 26
2.3.3 Continuous and Discrete Random Variables 26
2.3.4 Vectors and Matrices of Random Variables 27
2.3.5 Statistical Independence 28

random variables for unifying numerical and nonnumerical experimental outcomes

For experiments with a nonnumerical outcome, e.g., a colour, it is useful to map the experimental outcome to a real value and describe the probabilistic properties of the experiment using a real-valued *random variable*.

Since such a mapping in a natural way can be defined for experiments with discrete or continuous outcome, random variables in a unifying manner play a central role in stochastic modelling.

2.3.1 Characterizing Probability Distributions

With each outcome $s \in S$ of an experiment, we associate a real number $\underline{x}(s) \in \mathbb{R}$. The function $\underline{x}$

$$\underline{x} : S \to \mathbb{R} \qquad \underline{x} = \underline{x}(s) \tag{2.9}$$

is called a random variable. In order to specify the randomness of the experiment, thus, instead of characterizing the possible outcomes s, we characterize the function $\underline{x}$ (cf. Papoulis and Pillai, 2002). Observe: we distinguish between a sample value $x(s)$ (without underscore) depending on the outcome s of a *specific* experiment and the random variable $\underline{x}(s)$ (with underscore) which describes the experiment *as a whole*, for all $s \in S$. We regularly denote the random variable by $\underline{x}$, omitting the dependency of s.

Specifically, the experiment is characterized by what is called the *distribution* or *probability function*,

$$P_x(x) = P(\underline{x} < x)\,. \tag{2.10}$$

The argument $\underline{x} < x$ is the set of all possible outcomes for which $\underline{x}(s) < x$ holds. This definition assumes that there exists an event for all $x \in \mathbb{R}$.

The *index* x in the probability function $P_x(x)$ refers to the associated random variable, whereas the *argument* in $P_x(x)$ is the variable of the function. For simplicity, we sometimes omit the index.

observation process characterized by random variables

We will regularly characterize the statistical properties of an *observation process* by one or more random variables, catching that aspect of the concrete observation procedure which is relevant for the analysis task.

We can now derive the probability of a random variable to be in an interval,

$$P(\underline{x} \in [a, b]) = P_x(b) - P_x(a)\,. \tag{2.11}$$

Obviously, a probability function must fulfil

- $P_x(-\infty) = 0$,
- $P_x(x)$ is not decreasing, and
- $P_x(\infty) = 1$.

Example 2.3.2: ***Throwing a coin.*** When throwing a coin, we assume that

$$\underline{x}(\mathsf{heads}) = 0 \qquad \underline{x}(\mathsf{tails}) = 1\,. \tag{2.12}$$

In the case of equal probability of each outcome, we obtain the probability function

$$P_c(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1/2 & \text{if } 0 < x \leq 1 \\ 1 & \text{else} \end{cases}\,. \tag{2.13}$$

Observe, the index c in P_c is part of the name P_c of the probability function, here referring to throwing a coin. For the range $x \in (-\infty, 0]$, the corresponding event is the empty set $\emptyset$: it is unlikely that throwing a coin leads to neither heads nor tails. For the range $x \in (0, 1]$, the corresponding event is heads as $P(\underline{x}(\mathsf{heads}) < x) = 1/2$. For the range $x \in (1, \infty)$, the corresponding event is the certain event S. The probability of the event tails is given by $P(\mathsf{tails}) = P(\neg\mathsf{heads}) = 1 - P(\mathsf{heads}) = 1/2$, as the events heads and tails are mutually exclusive. Thus the event tails cannot be represented by some interval. ◇

unit step function

Using the unit-step function $s(x)$ (Fig. 2.3),

$$s(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{else} \end{cases} \tag{2.14}$$

the probability function P_c can be written as

$$P_c(x) = \frac{1}{2}s(x) + \frac{1}{2}s(x-1)\,. \tag{2.15}$$

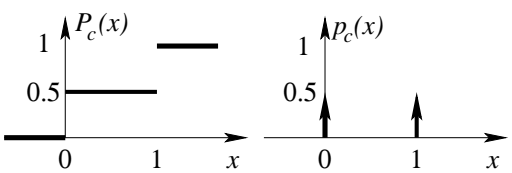


Fig. 2.3 Probability function $P_c(x)$ and density function $p_c(x)$ for throwing a coin

2.3.2 Probability Density Function

For experiments with continuous outcomes, e.g., a length measurement, we usually choose[1] $\underline{x}(x) = x$. We characterize the experiment by the first derivative of the probability function, which is called the *probability density function* or just *density function*

probability density function or density function

$$p_x(x) = \frac{\mathrm{d}P_x(x)}{\mathrm{d}x} \,. \tag{2.16}$$

Since integrating $p_x(x)$ yields $P_x(x)$ (cf. (2.10), p. 25)

$$P_x(x) = \int_{t=-\infty}^{x} p_x(t)\,\mathrm{d}t \,. \tag{2.17}$$

The function $P_x(x)$ is also is called the *cumulative distribution function* or just *cumulative distribution*. It is the same function as in (2.10), p. 25.

Example 2.3.3: ***Rounding errors.*** Rounding errors e lie in the interval $[-\frac{1}{2}, \frac{1}{2}]$. The probability of a rounding error to lie in the subinterval $[a, b] \subset [-\frac{1}{2}, \frac{1}{2}]$ is proportional to the ratio of the length $b - a$ to the length 1 of the complete interval. Therefore the probability density is

$$p_e(x) = r\left(x\middle| -\frac{1}{2}, \frac{1}{2}\right) \doteq \begin{cases} 1 & \text{if } x \in \left[-\frac{1}{2}, \frac{1}{2}\right] \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

This is the density of the uniform distribution in the interval $[-\frac{1}{2}, \frac{1}{2}]$, see Fig. 2.4. ◇

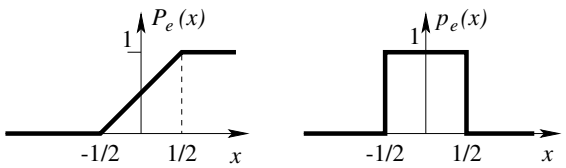


Fig. 2.4 Probability distribution $P_e(x)$ and probability density function $p_e(x)$ of the rounding error $\underline{e}$

2.3.3 Continuous and Discrete Random Variables

Random variables are called *continuous* if their probability distribution is continuous or, equivalently, if their density function is bounded. A random variable is called *discrete* if the probability function contains only steps or, equivalently, if the probability density function is either zero or infinite at a countable number of values x.

Example 2.3.4: ***Discrete probability density function.*** The probability density function of the random variable $\underline{x}$ of tossing a coin is

$$p_x(x) = \frac{1}{2}\delta(x) + \frac{1}{2}\delta(x-1) \,,$$

where $\delta(x)$ is Dirac's delta function. ◇

Dirac's delta function is the first derivative of the unit step function

$$\delta(x) \doteq \frac{\mathrm{d}s(x)}{\mathrm{d}x} \tag{2.19}$$

and is defined by a limiting process, e.g., by:

[1] The random variable depends on the unit in which x is measured, e.g., m or cm.

$$\delta(x) = \lim_{d\to 0} r(x|-d,+d) \tag{2.20}$$

with the rectangle function

$$r(x|a,b) \doteq \begin{cases} \dfrac{1}{b-a} & \text{if } x \in [a,b] \\ 0 & \text{else} \end{cases} . \tag{2.21}$$

The Dirac function has the following properties: The area under the delta function is 1:

$$\int_{t=-\infty}^{\infty} \delta(t)\mathrm{d}t = \lim_{x\to 0} \int_{t=-x}^{x} \delta(t)\mathrm{d}t = \lim_{x\to 0}(s(x)-s(-x)) = 1\,. \tag{2.22}$$

Therefore,

$$\int_{t=-\infty}^{\infty} f(x-t)\delta(t)\mathrm{d}t \overset{t\to \underline{x}-t}{=} \int_{t=-\infty}^{\infty} f(t)\delta(x-t)\mathrm{d}t \tag{2.23}$$

$$\overset{\delta(x)=0 \text{ for } x\neq 0}{=} \lim_{d\to 0} \int_{t=x-d}^{x+d} f(t) r(x|t-d,t+d)\mathrm{d}t \tag{2.24}$$

$$\overset{\xi\in[t-d,t+d]}{=} \lim_{d\to 0} \int_{t=x-d}^{x+d} \frac{1}{2d} f(\xi)\mathrm{d}t = f(x), \tag{2.25}$$

the second last step using the mean value theorem for integration. The delta function can thus be used to select a certain value of a function $f(x)$.

In graphs, the delta function is visualized as an *arrow* with the height indicating the local area under the function. For discrete random variables, therefore, we draw the heights of these *arrows*, i.e., the probabilities that one of the countable number of events occurs. Instead of the density function $p_x(x) = 1/2\,\delta(x) + 1/2\,\delta(x-1)$ for tossing a coin, e.g., we give the two probabilities $P(x=0) = P(x=1) = 1/2$.

The distribution of a random variable is often given a name, e.g., $\mathcal{H}$, and we write $\underline{x} \sim \mathcal{H}$ or, if the distribution depends on parameters $\boldsymbol{p}$,

$$\underline{x} \sim \mathcal{H}(\boldsymbol{p})\,. \tag{2.26}$$

2.3.4 Vectors and Matrices of Random Variables

We often have experiments with multiple outcomes. The corresponding I random variables $\underline{x}_i$ are usually collected in a vector called a *random vector*,

$$\underline{\boldsymbol{x}} = [\underline{x}_1, ..., \underline{x}_i, ..., \underline{x}_I]^\mathsf{T}\,. \tag{2.27}$$

The experiment is then characterized by the multi-dimensional probability function

$$P_{x_1,...,x_i,...,x_I}(\underline{x}_1 \leq x_1, ..., \underline{x}_i \leq x_i, ..., \underline{x}_I) = P(x_1, ..., x_i, ..., x_I) \tag{2.28}$$

or

$$P_x(\underline{\boldsymbol{x}} \leq \boldsymbol{x}) = P(\boldsymbol{x}), \tag{2.29}$$

or by the multi-dimensional probability density function

$$p_x(\boldsymbol{x}) = \frac{\partial^I P(\boldsymbol{x})}{\partial x_1 ... \partial x_i ... \partial x_I}\,. \tag{2.30}$$

random matrices

We will regularly use random matrices, e.g., when dealing with uncertain transformations. Let the $N \times M$ matrix $\underline{\boldsymbol{X}} = [X_{nm}]$ contain NM random variables. Then it is of

advantage to vectorize the matrix,

$$\underset{NM\times 1}{\underline{\boldsymbol{x}}} = \mathrm{vec}\underline{\boldsymbol{X}} = [\underline{X}_{11}, \underline{X}_{21}, ..., \underline{X}_{N1}, \underline{X}_{12}, ..., \underline{X}_{NM}]^{\mathsf{T}} \tag{2.31}$$

and represent the uncertainty by the joint probability of the random NM-vector $\underline{\boldsymbol{x}}$.

2.3.5 Statistical Independence

If two random variables $\underline{x}$ and $\underline{y}$ are statistically independent, their joint probability function and their joint probability density function are separable functions, i.e.,

$$P_{xy}(x, y) = P_x(x)P_y(y) \quad \text{or} \quad p_{xy}(x, y) = p_x(x)p_y(y)\,. \tag{2.32}$$

2.4 Distributions

2.4.1 Binomial Distribution 28
2.4.2 Uniform Distribution 28
2.4.3 Exponential and Laplace Distribution 29
2.4.4 Normal Distribution 29
2.4.5 Chi-Square Distribution 33
2.4.6 Wishart Distribution 34
2.4.7 Fisher Distribution 34
2.4.8 Student's t-Distribution 35

We now list a number of distributions relevant for statistical reasoning.

2.4.1 Binomial Distribution

A discrete random variable $\underline{n}$ follows a binomial distribution,

$$\underline{n} \sim \mathrm{Bin}(N, p), \tag{2.33}$$

if its discrete density function is

$$P(n) = \binom{N}{n} p^n (1-p)^{N-n} \qquad n = 0, 1, ..., N \qquad 0 \leq p \leq 1 \tag{2.34}$$

where $\binom{N}{n}$ are binomial coefficients. It models the probability of n successes if an experiment for which the probability of success p is repeated N times.

For $p = \frac{1}{2}$, we obtain the probability $P(n)$ of observing n heads when tossing a coin N times (Table 2.1).

2.4.2 Uniform Distribution

A continuous random variable follows the general uniform distribution,

$$\underline{x} \sim \mathcal{U}(a, b) \qquad a, b \in \mathbb{R} \quad b > a, \tag{2.35}$$

Table 2.1 Probability $P(n)$ of obtaining n heads when tossing a coin N times

	$n=0$	1	2	3	4	5	6
$N=1$	$\frac{1}{2}$	$\frac{1}{2}$					
2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$				
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			
4	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{4}$	$\frac{1}{16}$		
5	$\frac{1}{32}$	$\frac{5}{32}$	$\frac{5}{16}$	$\frac{5}{16}$	$\frac{5}{32}$	$\frac{1}{32}$	
6	$\frac{1}{64}$	$\frac{3}{32}$	$\frac{15}{64}$	$\frac{5}{16}$	$\frac{15}{64}$	$\frac{3}{32}$	$\frac{1}{64}$

if it has the density $r(x|a,b)$ ((2.21), p. 27). For example, *rounding errors* $\underline{e}$ have uniform distribution $\underline{e} \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$.

Two random variables $\underline{x}$ and $\underline{y}$ jointly follow a uniform distribution,

$$(\underline{x}, \underline{y}) \sim \mathcal{U}(a, b; c, d), \tag{2.36}$$

if they have the density function

$$r_{xy}(x, y \mid a, b; c, d) = r(x \mid a, b)\; r(y \mid c, d)\,, \tag{2.37}$$

where $x \in [a, b]$ and $y \in [c, d]$. Due to (2.37) the random variables $\underline{x}$ and $\underline{y}$ are independent.

2.4.3 Exponential and Laplace Distribution

A random variable $\underline{x}$ follows an exponential distribution with real parameter $\mu > 0$ if its density function is given by

$$p_x(x) = \frac{1}{\mu} \mathrm{e}^{-\dfrac{x}{\mu}}\,, \qquad x \geq 0, \quad \mu > 0\,. \tag{2.38}$$

This is also called the *Rayleigh distribution*. *Rayleigh distribution*

A random variable $\underline{x}$ is Laplacian distributed with real parameter $\sigma > 0$,

$$\underline{x} \sim \mathrm{Lapl}(\sigma), \tag{2.39}$$

if its density function is given by

$$p_x(x) = \frac{1}{\sqrt{2}\,\sigma} \mathrm{e}^{-\sqrt{2}\left|\frac{x}{\sigma}\right|}\,, \qquad \sigma > 0\,. \tag{2.40}$$

2.4.4 Normal Distribution

2.4.4.1 Univariate Normal distribution

A random variable $\underline{x}$ is normally or Gaussian distributed with real parameters μ and $\sigma > 0$,

$$\underline{x} \sim \mathcal{N}(\mu, \sigma^2), \tag{2.41}$$

if its density function is given by

$$p_x(x) = g(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma} \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \qquad \sigma > 0\,. \tag{2.42}$$

The density function is symmetric with respect to μ, there having the value $1/(\sqrt{2\pi}\,\sigma) \approx 0.4/\sigma$; the inflection points are at $\mu - \sigma$ and $\mu + \sigma$, there having the value $1/(\sqrt{2\pi\mathrm{e}}\,\sigma) \approx 0.24/\sigma$, hence 3/5th of the value at the mean. The tangents at the inflection points intersect the x-axis at $\mu \pm 2\sigma$.

Large deviations from the mean value μ are unlikely:

$$P(\underline{x} \in [\mu - \sigma, \mu + \sigma]) = \int_{x=\mu-\sigma}^{x=\mu+\sigma} g(x \mid \mu, \sigma^2)\,\mathrm{d}x \approx 0.6827\,, \tag{2.43}$$

$$P(\underline{x} \in [\mu - 2\sigma, \mu + 2\sigma]) = \int_{x=\mu-2\sigma}^{x=\mu+2\sigma} g(x \mid \mu, \sigma^2)\,\mathrm{d}x \approx 0.9545\,, \tag{2.44}$$

$$P(\underline{x} \in [\mu - 3\sigma, \mu + 3\sigma]) = \int_{x=\mu-3\sigma}^{x=\mu+3\sigma} g(x \mid \mu, \sigma^2)\,\mathrm{d}x \approx 0.9973\,. \tag{2.45}$$

Thus the probability of a value lying outside the interval $[\mu-3\sigma, \mu+3\sigma]$ is very low, 0.3 %.

standard normal distribution, normalized Gaussian distribution

The *standard normal distribution* or *normalized Gaussian distribution* is given by $\mu = 0$ and $\sigma = 1$ (Fig. 2.5)

$$\phi(x) = g(x \mid 0, 1) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}\,. \tag{2.46}$$

Its cumulative distribution is

$$\Phi(x) = \int_{t=-\infty}^{x} \phi(t)\,\mathrm{d}t\,. \tag{2.47}$$

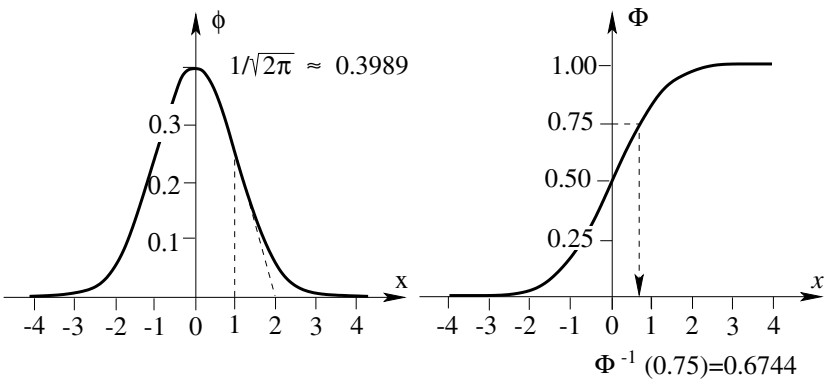


Fig. 2.5 Left: normal or Gaussian density function $\phi(x)$. Inflection points at $x = +1$ and $x = -1$. The ratio of the function values on the symmetry axis and at the inflection point is $\sqrt{\mathrm{e}} = 1.6487... \approx 5/3$; the tangent in the inflection point intersects the x-axis at $x = 2$, such that the x-coordinate of the inflection point is in the middle of this intersection point and the line of symmetry. **Right:** cumulative distribution function $\Phi(x)$. 75th percentile at $x = \Phi^{-1}(0.75) = 0.6745$

central limit theorem

The normal distribution is the most important distribution. This follows from the *central limit theorem*: The sum of a large number of independent, identically distributed random variables with bounded variance is approximately normally distributed (cf. Papoulis, 1965, Sect. 8–6).

2.4.4.2 Multi-dimensional Normal Distribution

If two independent random variables are normally distributed according to

$$\underline{x} \sim \mathcal{N}(\mu_x, \sigma_x^2) \qquad \underline{y} \sim \mathcal{N}(\mu_y, \sigma_y^2), \tag{2.48}$$

their joint density function is

$$p_{xy}(x, y) = g_x(x \mid \mu_x, \sigma_x^2)\; g_y(y \mid \mu_y, \sigma_y^2) \tag{2.49}$$

$$= \frac{1}{2\pi\sigma_x\sigma_y} \mathrm{e}^{-\frac{1}{2}\left(\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right)}. \tag{2.50}$$

With the vectors

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \tag{2.51}$$

and the 2×2 matrix,

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \tag{2.52}$$

this can be written as

$$g_{xy}(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{2\pi\sqrt{|\Sigma|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \tag{2.53}$$

If the 2×2 matrix Σ is a general symmetric positive definite matrix

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}, \tag{2.54}$$

the two random variables are dependent. The correlation coefficient,

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \in [-1, 1], \tag{2.55}$$

measures the degree of linear dependency. If $\rho_{xy} = 0$, the two random variables are *uncorrelated*, and if they are normally distributed, they are *independent*, due to (2.32), p. 28. The 2D normal distribution is an elliptic bell-shaped function and can be visualized by one of its contour lines, cf. Fig. 2.6. The *standard ellipse*, sometimes also called standard error ellipse, is defined by

uncorrelated, independent random variables

standard ellipse

$$(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T}\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) = 1\,. \tag{2.56}$$

The standard ellipse allows the visualization of important properties of the uncertain point:

- The standard ellipse is centred at $\boldsymbol{\mu}_x$.
- The bounding box has size $2\sigma_x \times 2\sigma_y$.
- The semi-axes are the square roots of the eigenvalues λ_i of the covariance matrix, namely $\sigma_{\max} = \sqrt{\lambda_1}$ and $\sigma_{\min} = \sqrt{\lambda_2}$, which are the square roots of the eigenvalues of Σ,

$$\sigma^2_{\max,\min} = \frac{1}{2}(\sigma_x^2 + \sigma_y^2) \pm \frac{1}{2}\sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2}\,. \tag{2.57}$$

- If the two coordinates are correlated, the major axis is not parallel to the coordinate system. The angle α is given by

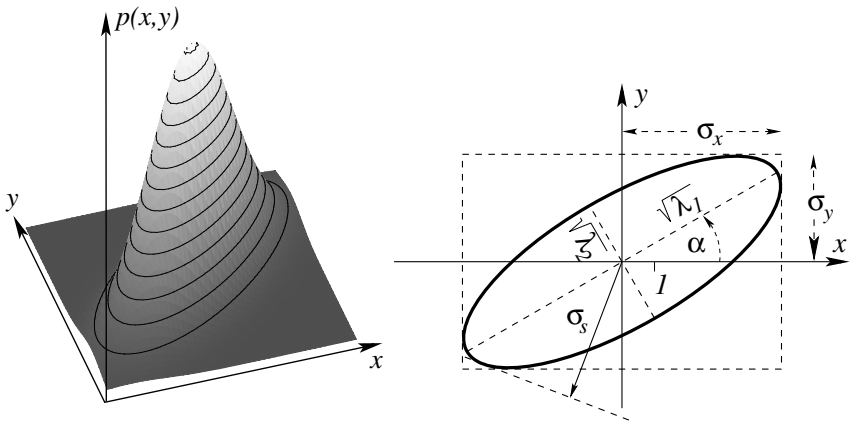


Fig. 2.6 General 2D normal or Gaussian distribution, centred at the origin. **Left:** density function. **Right:** standard ellipse. Actual values: $\mu_x = \mu_y = 0$, $\sigma_x = 4.9$, $\sigma_y = 3.2$, $\rho = 0.7$

$$\alpha = \frac{1}{2}\operatorname{atan2}\left(2\sigma_{xy}, \sigma_x^2 - \sigma_y^2\right) \in (-\pi/2, +\pi/2] \tag{2.58}$$

using a two-argument version of the arctan function.

The sign of the angle follows the sign of the correlation coefficient ρ_{xy} or the covariance σ_{xy}.

- The standard deviation σ_s of a distance s between the point $\boldsymbol{\mu}_x$ and a fixed point in an arbitrary direction, indicated here by an arrow, is given by the distance of $\boldsymbol{\mu}_x$ from the tangent to the standard ellipse perpendicular to that direction. This shows that the minor and the major axes of the standard ellipse give the minimum and the maximum of the directional uncertainty of the point.

In higher dimensions, (2.56) represents an ellipsoid or a hyper-ellipsoid $\mathcal{E}$. The probability $S = P(\underline{x} \in \mathcal{E})$ that a random point lies within the standard ellipsoid depends on the dimension as shown in the first line of Table 2.2, and rapidly diminishes with the dimension.

confidence ellipse

Instead of showing the standard ellipse or standard ellipsoid, we therefore can show the *confidence ellipse* or *confidence ellipsoid*. The confidence ellipsoid is the k-fold standard ellipsoid, such that the probability $P(\underline{x} \in \mathcal{E}(k))$ that a sample lies within the ellipsoid is a certain prespecified value S

$$\mathcal{E}(k): \quad (\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) = k^2\,, \quad P(\underline{x} \in \mathcal{E}(k)) = S\,. \tag{2.59}$$

The standard ellipse is identical to the confidence ellipse for $k = 1$: $\mathcal{E} = \mathcal{E}(1)$. For the dimension $d = 1$ and a probability $P(\underline{x} \in \mathcal{E}(k)) = S = 0.9973$, we would obtain $k = 3$, as shown in (2.45), p. 30. Here the ellipse reduces to the interval $[-k\sigma_x, +k\sigma_x]$.

For $S = 95\%$, $S = 99\%$ and $S = 99.9\%$, the values $k(S)$ determined from the right equation in (2.59) are given in Table 2.2 for different dimensions.

Table 2.2 Confidence regions. **First row:** Probabilities $P(\underline{x} \in \mathcal{E})$ for different dimensions d of a random vector $\underline{x}$. **Other rows:** Factor $k(S)$ for the confidence ellipsoids $\mathcal{E}(k(S))$ for $S = 0.95, 0.99, 0.999$ and for different dimensions d.

d	1	2	3	4	5	10	20	50	100
$P(\underline{x} \in \mathcal{E})$	0.68	0.40	0.20	0.09	$3.7 \cdot 10^{-2}$	$1.7 \cdot 10^{-4}$	$1.7 \cdot 10^{-10}$	$1.2 \cdot 10^{-33}$	$1.8 \cdot 10^{-80}$
$k(0.95)$	1.96	2.45	2.80	3.08	3.33	4.28	5.60	8.22	11.2
$k(0.99)$	2.58	3.03	3.37	3.64	3.88	4.82	6.13	8.73	11.6
$k(0.999)$	3.29	3.72	4.03	4.30	4.53	5.44	6.73	9.31	12.2

Gaussian distributed matrix

Matrices of Gaussian distributed random variables can be represented using their vector representation, (2.31), p. 28. Let the $N \times M$ matrix $\underline{\boldsymbol{X}}$ contain NM random variables which are normally distributed; we represent its uncertain covariance matrix using the

random vector

$$\underline{\boldsymbol{x}} = \mathrm{vec}\underline{X}: \quad \underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})\,. \tag{2.60}$$

Or we may keep the matrix representation for the mean matrix and write

$$\underline{X} \sim \mathcal{N}(\boldsymbol{\mu}_X, \Sigma_{xx})\,. \tag{2.61}$$

Sometimes we will refer to Σ_{xx} as the *covariance matrix of the random matrix* $\underline{X}$.

2.4.4.3 Normal Distribution with Zero or Infinite Variance

When representing fixed values, such as the third component in a homogeneous vector $[\underline{x}, \underline{y}, 1]^\mathsf{T}$, we might track this property through the reasoning chain, which is cumbersome, or just treat the value 1 as a stochastic variable with mean 1 and variance 0. The second alternative has implicitly been chosen by Kanatani (1996) and Criminisi (2001). This method needs some care, as the density function for a Gaussian random variable is not defined for zero variance.

The distribution of a random variable $\underline{y} \sim \mathcal{N}(\mu_y, 0)$ can be defined in a limiting process ((2.22), p. 27), by a δ-function:

$$p_y(y) = \lim_{\sigma_y \to 0} g(y; \mu_y, \sigma_y^2) = \delta(y - \mu_y)\,. \tag{2.62}$$

Now a 2-vector can be constructed with a singular 2×2 covariance matrix. Assume that $\underline{x} \sim N(\mu_x, 1)$ and $\underline{y} \sim N(\mu_y, 0)$ are independent stochastic variables; thus,

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \right)\,. \tag{2.63}$$

As $\underline{x}$ and $\underline{y}$ are stochastically independent, their *joint generalized probability density function* is ((2.32), p. 28)

$$g_{xy} = g_x(x; \mu_x, 1)\ \delta(y - \mu_y)\,. \tag{2.64}$$

Obviously, working with a product of Gaussians and δ-functions will be cumbersome in cases when stochastic variables are not independent.

In most cases, reasoning can be done using the moments (cf. Sects. 2.5); therefore, the complicated distribution is not of primary concern. The propagation of uncertainty with second moments (cf. Sect. 2.7, p. 40) only relies on the covariance matrices, not on their inverses, and can be derived usng what is called the *moment generating function* (Papoulis, 1965), which is also defined for generalized probability density functions. Thus uncertainty propagation can also be performed in mixed cases.

Similar reasoning can be used to allow random variables with zero weights $1/\sigma^2$, or infinite variance, or, more general, singular weight matrices $W = \Sigma^{-1}$ (Dempster, 1969).

2.4.5 Chi-Square Distribution

A random variable $\underline{y}$ is χ^2_n-distributed with n degrees of freedom,

$$\underline{y} \sim \chi^2_n\,, \quad \text{or} \quad \underline{y} \sim \chi^2(n), \tag{2.65}$$

if it has the density function

$$p_y(y, n) = \frac{y^{(n/2)-1} \mathrm{e}^{-y/2}}{2^{n/2}\Gamma\left(\frac{n}{2}\right)}\,, \qquad n \in \mathbb{N}\,, \qquad y > 0 \tag{2.66}$$

with the Gamma function $\Gamma(\cdot)$ (cf. Koch, 1999, Sect. 2.6.1). This distribution is used for testing quadratic forms. In particular, the sum

$$\underline{y} = \sum_{i=1}^{n} \underline{z}_i^2 \tag{2.67}$$

of n independent random variables $\underline{z}_i$, which follow a standard normal distribution ($\underline{z}_i \sim \mathcal{N}(0,1)$), is χ_n^2 distributed. For $n = 2$, we obtain the exponential distribution

$$p_y(y,2) = \frac{1}{2}\mathrm{e}^{-y/2} \qquad y \geq 0\,. \tag{2.68}$$

noncentral χ'^2 distribution

Given the n mutually independent random variables which follow noncentral normal distributions $\underline{z}_i \sim \mathcal{N}(\mu_i, 1)$, then the random variable

$$\underline{y} = \sum_{i=1}^{n} \underline{z}_i^2 \sim \chi_d'^2(\delta^2) \quad \text{with} \quad \underline{z}_i \sim \mathcal{N}(\mu_i, 1) \tag{2.69}$$

has a *noncentral chi-square distribution* $\chi_n'^2(\delta)$ with n degrees of freedom and noncentrality parameter $\delta^2 = \sum_{i=1}^{n} \mu_i^2$.

Sometimes we need the distribution of the square root $\underline{s} = \sqrt{\underline{y}}$ and thus of the length $\underline{s} = |\underline{\boldsymbol{x}}|$ of a random vector $\underline{\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0}, I_n)$. The resulting distribution is the χ distribution, having density

Exercise 2.28

χ distribution

$$p_s(s,n) = \frac{2^{1-n/2} s^{n-1} \mathrm{e}^{-s^2/2}}{\Gamma(n/2)}\,. \tag{2.70}$$

2.4.6 Wishart Distribution

A symmetric positive definite $p \times p$ matrix $\underline{V}$ is Wishart distributed, $\mathcal{W}(n, \Sigma)$, with n degrees of freedom and matrix parameter Σ if its density function is (cf. Koch, 1999, Sect. 2.8.1)

$$p_W(V|n,\Sigma) = k_W \cdot |V|^{(n-p-1)/2} \mathrm{e}^{-\mathrm{tr}\left(\Sigma^{-1}V\right)/2}, n \in \mathbb{N}\,,\ |V| > 0\,,\ |\Sigma| > 0 \tag{2.71}$$

with some normalization constant k_W. This distribution is useful for evaluating empirical covariance matrices. Let N mutually independent random vectors $\underline{\boldsymbol{x}}_n$ of length p be given which follow a multivariate central normal distribution, $\underline{\boldsymbol{x}}_n \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Then the matrix

$$\underline{V} = \sum_{n=1}^{N} \underline{\boldsymbol{x}}_n \underline{\boldsymbol{x}}_n^{\mathsf{T}} \sim \mathcal{W}(n, \Sigma) \tag{2.72}$$

follows a Wishart distribution. For $\Sigma = 1$ the Wishart distribution reduces to the χ^2 distribution.

Exercise 2.29

2.4.7 Fisher Distribution

A random variable $\underline{F}$ is Fisher-distributed or F-distributed,

$$\underline{F} \sim \mathcal{F}(m,n), \tag{2.73}$$

with m and n degrees of freedom if its density is

$$p_F(x|m,n) = k_F \cdot s(x) \cdot x^{\frac{m}{2}-1}(mx+n)^{-\frac{m+n}{2}} \tag{2.74}$$

with the step function $s(x)$ and a normalization constant k_F.

If two independent random variables $\underline{y}_1$ and $\underline{y}_2$ are χ^2 distributed, namely

$$\underline{y}_1 \sim \chi^2_m \qquad \underline{y}_2 \sim \chi^2_n\,, \tag{2.75}$$

then the random variable

$$\underline{F} = \frac{\underline{y}_1/m}{\underline{y}_2/n} \sim \mathcal{F}(m,n) \tag{2.76}$$

is Fisher distributed with (m,n) degrees of freedom. This distribution is used for testing results of estimation processes.

2.4.8 Student's t-Distribution

A random variable is t-distributed,

$$\underline{t} \sim t(n), \tag{2.77}$$

with n degrees of freedom, if its density is given by

$$p_t(x|n) = k_t \cdot \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}\,, \tag{2.78}$$

with some normalization constant k_t. If two independent random variables $\underline{z}$ and $\underline{y}$ are distributed according to

$$\underline{z} \sim \mathcal{N}(0,1) \qquad \underline{y} \sim \chi^2_n\,, \tag{2.79}$$

the random variable

$$\underline{t} = \frac{\underline{z}}{\sqrt{\underline{y}/n}} \sim t(n) \qquad n \in \mathbb{N} \tag{2.80}$$

follows Student's t-distribution with n degrees of freedom. This distribution may be used for testing residuals of observations after parameter estimation.

The relationships among the different distributions is given in Fig. 2.7. The normal distribution $\mathcal{N}$ is a special case of Student's t_n distribution and of the χ^2_m distribution, which themselves are special cases of the Fisher $\mathcal{F}_{m,n}$ distribution, obtained by setting one or both parameters to infinity.

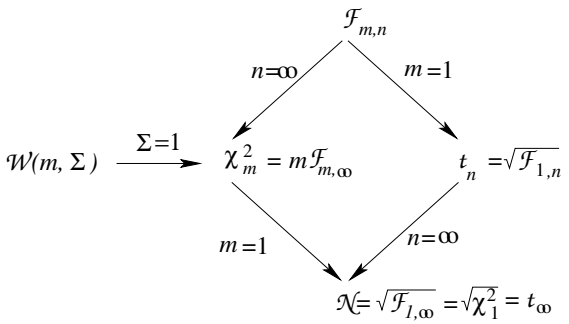


Fig. 2.7 Fisher's $\mathcal{F}_{m,n}$ and Wishart distribution $\mathcal{W}(m,\Sigma)$ and its specializations: χ^2_m, Student's t_n and normal distribution $\mathcal{N}(0,1)$. For example, taking the square root of a random variable, which is $\mathcal{F}_{1,n}$ distributed can be shown to be t_n-distributed

2.5 Moments

2.5.1 General Moments 36
2.5.2 Central Moments 37
2.5.3 Moments of Normally Distributed Variables 39
2.5.4 Moments of the Uniform Distribution 39

Moments are used to characterize probability distributions. They are mathematically equivalent to moments in physics, if the probability density function is interpreted as a mass density function.

2.5.1 General Moments

With the density functions $p_x(x)$ or $p_{xy}(x, y)$, *general moments* are defined as

$$m_r = \int_{x=-\infty}^{+\infty} x^r p_x(x)\,\mathrm{d}x \qquad r \geq 0 \tag{2.81}$$

or

$$m_{r,s} = \int_{x=-\infty}^{+\infty}\int_{y=-\infty}^{+\infty} x^r y^s p_{xy}(x, y)\,\mathrm{d}x\,\mathrm{d}y \qquad r, s \geq 0\,. \tag{2.82}$$

The values m_k and $m_{r,k-r}$, with $r \leq k$, are called *kth-order moments*. For discrete random variables with probabilities $P_x(\underline{x} = x)$ and $P_{xy}(\underline{x} = x, \underline{y} = y)$, general moments are defined as

$$m_r = \sum_{i=1}^{\infty} x_i^r P_x(\underline{x} = x_i) \qquad r \geq 0 \tag{2.83}$$

or

$$m_{r,s} = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} x_i^r y_j^s P_{xy}(\underline{x} = x_i, \underline{y} = y_j)\,\mathrm{d}x\mathrm{d}y \qquad r, s \geq 0\,. \tag{2.84}$$

We will restrict the derivations to continuous variables. The moment of the order zero is always 1. The moments m_1 or $m_{1,0}$ and $m_{0,1}$ are the *mean values* or the expected values $\mathbb{E}(\underline{x})$,

$$\mu_x \doteq m_1 = \int x p_x(x)\,\mathrm{d}x\,, \tag{2.85}$$

or

$$\mu_x \doteq m_{1,0} = \int x p_{xy}(x, y)\,\mathrm{d}x\,\mathrm{d}y\,, \tag{2.86}$$

$$\mu_y \doteq m_{0,1} = \int y p_{xy}(x, y)\,\mathrm{d}x\,\mathrm{d}y\,, \tag{2.87}$$

respectively, omitting the boundaries of the integrals.

The higher-order moments can be interpreted more easily if they refer to the mean values.

2.5.2 Central Moments

The *central moments* are defined as[2]

$$\mu_r = \int (x - \mu_x)^r p_x(x)\, \mathrm{d}x \tag{2.88}$$

and, for random d-vectors,

$$\mu_{r,s} = \int (x - \mu_x)^r (y - \mu_y)^s p_{xy}(x, y)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{2.89}$$

In general, we have

$$\mu_0 = 1 \qquad \mu_1 = 0 \qquad \mu_{0,0} = 1 \qquad \mu_{1,0} = \mu_{0,1} = 0\,. \tag{2.90}$$

The central moments of a random variable yield their *variance*,

$$\sigma_x^2 \doteq \mu_2 = \int (x - \mu_x)^2 p_x(x)\, \mathrm{d}x\,, \tag{2.91}$$

$$\sigma_x^2 \doteq \mu_{2,0} = \int (x - \mu_x)^2 p_{xy}(x, y)\, \mathrm{d}x\, \mathrm{d}y\,, \tag{2.92}$$

and

$$\sigma_y^2 \doteq \mu_{0,2} = \int (y - \mu_y)^2 p_{xy}(x, y)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{2.93}$$

We can easily show that the following relation holds, which in physics is called *Steiner's theorem*: *Steiner's theorem*

$$\mu_2 = m_2 - m_1^2 \qquad \text{or} \qquad \sigma_x^2 = m_2 - \mu_x^2\,. \tag{2.94}$$

Therefore, the central moments can be easily derived from the noncentral moments. The positive square root of the variance is called the *standard deviation*,

$$\sigma_x = +\sqrt{\sigma_x^2}, \tag{2.95}$$

of the random variable $\underline{x}$. The mixed second central moment of two random variables is their *covariance*

$$\sigma_{xy} \doteq \mu_{1,1} = \int (x - \mu_x)(y - \mu_y) p_{xy}(x, y)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{2.96}$$

As it is difficult to interpret, it is usually related to the standard deviations σ_x and σ_y via the correlation coefficient (2.55) by

$$\sigma_{xy} = \rho_{xy} \sigma_x \sigma_y\,. \tag{2.97}$$

The second central moments of a vector $\underline{\boldsymbol{x}}$ of several random variables $\underline{\boldsymbol{x}} = [\underline{x}_i]$ usually are collected in its covariance matrix

$$\boldsymbol{\Sigma}_{xx} = [\sigma_{x_i x_j}]\,. \tag{2.98}$$

Similarly, the covariances $\sigma_{x_i y_j}$ of the random variables collected in two vectors $\underline{\boldsymbol{x}} = [\underline{x}_i]$ and $\underline{\boldsymbol{y}} = [\underline{y}_j]$ are collected in their covariance matrix

$$\boldsymbol{\Sigma}_{xy} = [\sigma_{x_i y_j}]\,. \tag{2.99}$$

Due to the symmetry of covariance matrices we have

[2] Not to be confused with the mean value μ_x.

$$\Sigma_{xy} = \Sigma_{yx}^{\mathsf{T}} \,. \tag{2.100}$$

With the diagonal matrices

$$S_x = \mathrm{Diag}([\sigma_{x_i}) \qquad S_y = \mathrm{Diag}([\sigma_{y_j}]) \tag{2.101}$$

containing the standard deviations, we can also express the covariance matrix as

$$\Sigma_{xy} = S_x R_{xy} S_y \tag{2.102}$$

using the correlation matrix

$$R_{xy} = [\rho_{x_i y_j}] = \left[\frac{\sigma_{x_i y_j}}{\sigma_{x_i} \sigma_{y_j}} \right] . \tag{2.103}$$

In the case of two random variables $\underline{x}$ and $\underline{y}$ we have their covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \begin{bmatrix} 1 & \rho_{xy} \\ \rho_{xy} & 1 \end{bmatrix} \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} . \tag{2.104}$$

We can show that covariance matrices always are positive semidefinite and the correlation coefficients ρ_{ij} always lie in $[-1, +1]$.

expectation $\mathbb{E}(\cdot)$

We use the expectation operator or mean operator $\mathbb{E}(\cdot)$ as an abbreviation. It yields the mean value of a random variable $\underline{x}$ or of a random vector $\underline{\boldsymbol{x}}$,

$$\mathbb{E}(\underline{x}) = \int_{x=-\infty}^{\infty} x p_x(x) \, \mathrm{d}x \tag{2.105}$$

and, for a d-vector $\underline{\boldsymbol{x}}$,

$$\mathbb{E}(\underline{x}) = \int_{x=-\infty}^{\infty} \boldsymbol{x} p_x(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \,. \tag{2.106}$$

The kth moments therefore are the expected or mean values of the kth power of the random variable,

$$m_k = \mathbb{E}(\underline{x}^k) \qquad m_{r,s} = \mathbb{E}(\underline{x}^r \underline{y}^s) \quad \text{with} \quad k = r + s \,. \tag{2.107}$$

The central moments thus are the expected mean values of the kth power of the difference of the random variable and its *expected* or *mean value*,

$$\mu_k = \mathbb{E}([\underline{x} - \mu_x]^k) \qquad \mu_{r,s} = \mathbb{E}([\underline{x} - \mu_x]^r [\underline{y} - \mu_y]^s) \,. \tag{2.108}$$

linearity of $\mathbb{E}(\cdot)$

The expectation operator is linear,

$$\mathbb{E}(a\underline{x} + b) = a\mathbb{E}(\underline{x}) + b \qquad \text{or} \qquad \mathbb{E}(A\underline{\boldsymbol{x}} + \boldsymbol{b}) = A\,\mathbb{E}(\underline{\boldsymbol{x}}) + \boldsymbol{b} \,, \tag{2.109}$$

which results from the linearity of the integration, a property which we often use.

Based on the expectation operator we also can define the *dispersion operator* $\mathbb{D}(\cdot)$ or $\mathbb{V}(\cdot)$ and the *covariance operator* $\mathbb{C}\mathrm{ov}(\cdot, \cdot)$, which operates on one or two vectors of random variables, respectively. The dispersion operator leads to the variance–covariance matrix of a random variable:

variance $\mathbb{V}(\cdot)$
dispersion $\mathbb{D}(\cdot)$

$$\mathbb{D}(\underline{\boldsymbol{x}}) = \mathbb{V}(\underline{\boldsymbol{x}}) = \Sigma_{xx} = \mathbb{E}[\{\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}})\}\{\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}})\}^{\mathsf{T}}] \,. \tag{2.110}$$

covariance $\mathbb{C}ov(\cdot, \cdot)$

The covariance operator leads to the covariance matrix of two random variables:

$$\mathbb{C}\mathrm{ov}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \Sigma_{xy} = \mathbb{E}\left[\{\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}})\}\{\underline{\boldsymbol{y}} - \mathbb{E}(\underline{\boldsymbol{y}})\}^{\mathsf{T}}\right] = \Sigma_{yx}^{\mathsf{T}} = \mathbb{C}\mathrm{ov}(\underline{\boldsymbol{y}}, \underline{\boldsymbol{x}})^{\mathsf{T}} \,, \tag{2.111}$$

thus

$$\mathbb{D}(\underline{\boldsymbol{x}}) = \mathbb{V}(\underline{\boldsymbol{x}}) = \mathbb{C}\mathrm{ov}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{x}})\,. \tag{2.112}$$

Observe the convention for scalar random variables x_i and y_j:

$$\Sigma_{x_i x_i} = \sigma^2_{x_i} \qquad \Sigma_{x_i y_j} = \sigma_{x_i, y_j}\,. \tag{2.113}$$

For single variables, the dispersion operator is often replaced by the variance operator, e.g., $\mathbb{V}(\underline{x}) = \sigma^2_x$.

2.5.3 Moments of Normally Distributed Variables

A variable following a one-dimensional normal distribution $\mathcal{N}(\mu, \sigma^2)$ has the first moments,

$$m_0 = 1\,, \quad m_1 = \mu\,, \quad m_2 = \mu^2 + \sigma^2\,, \quad m_3 = \mu^3 + 3\mu\sigma^2 \tag{2.114}$$

and

$$m_4 = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 \tag{2.115}$$

and the corresponding central moments

$$\mu_0 = 1\,, \quad \mu_1 = 0\,, \quad \mu_2 = \sigma^2\,, \quad \mu_3 = 0\,, \quad \mu_4 = 3\sigma^4\,. \tag{2.116}$$

In general, the odd central moments are zero due to the symmetry of the density function. The even central moments, $\mu_{2k}, k = 0, 1, \ldots$, of the normal distribution with density $g(x \mid \mu, \sigma^2)$ only depend on the variance

$$\mu_{2k} = \int (x-\mu)^{2k} g(x \mid \mu, \sigma^2)\, \mathrm{d}x = 1 \cdot 3 \cdot \ldots \cdot (2k-1)\sigma^{2k}\,. \tag{2.117}$$

The parameters μ and σ^2 of the one-dimensional normal distribution are the mean and the variance. The two parameters $\boldsymbol{\mu}$ and Σ of the multi-dimensional normal distribution are the mean vector and the covariance matrix.

The second (central) moment of a multi-dimensional normal distribution is the covariance matrix Σ. It exists even if the covariance matrix is singular and the density function is not a proper function.

2.5.4 Moments of the Uniform Distribution

The moments of the uniform distribution $\mathcal{U}(a, b)$ are

$$m_k = \frac{1}{k+1} \frac{b^{k+1} - a^{k+1}}{b-a}\,. \tag{2.118}$$

We obtain the even central moments $\mu_0 = 1$ and

$$\mu_2 = \sigma^2 = \frac{1}{12}(b-a)^2 \qquad \mu_4 = \frac{1}{80}(b-a)^4\,. \tag{2.119}$$

Thus, the standard deviation of the rounding error, modelled as $\underline{r} \sim \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)$, is *rounding error*

$$\sigma_r = \sqrt{1/12} \approx 0.28 \tag{2.120}$$

of the last and rounded digit.

2.6 Quantiles of a Distribution

We are often interested in the value x such that the value of the cumulative distribution $P_x(x) = P(\underline{x} < x)$ is a prespecified probability α

$$P_x(x) = \int_{t=-\infty}^{x} p_x(t)\,\mathrm{d}t = \alpha\,. \tag{2.121}$$

This *α-quantile* can be determined using the *inverse cumulative distribution*

$$x = P_x^{-1}(\alpha)\,. \tag{2.122}$$

If the random variable follows a certain distribution, e.g. $\underline{x} \sim \mathcal{F}_{m,n}$, the α-quantile can be written as $x = F_{m,n;\alpha}$.

median

The *median* is the 0.5-quantile or 50th percentile

$$\mathrm{med}(x) = P_x^{-1}(0.5)\,. \tag{2.123}$$

For normally distributed random variables, it coincides with the mean, thus $N_{\mu_x,\sigma_x^2;0.5} = \mathrm{med}(x) = \mu_x$.

median absolute difference

Instead of the standard deviation, it is also possible to use the *median of the absolute differences* (MAD) from the median to characterize the spread of the random variable. It is given by

$$\mathrm{MAD}_x = \mathrm{med}(|x - \mathrm{med}(x)|)\,. \tag{2.124}$$

For normally distributed random variables, it is related to the standard deviation by

$$\mathrm{MAD}_x = \Phi^{-1}(0.75)\,\sigma_x \approx 0.6745\,\sigma_x \tag{2.125}$$

and

$$\sigma_x = \frac{1}{\Phi^{-1}(0.75)}\,\mathrm{MAD}_x \approx 1.4826\,\mathrm{MAD}_x\,, \tag{2.126}$$

(Fig. 2.5, p. 30, right).

2.7 Functions of Random Variables

2.7.1 Transformation of a Random Variable 41
2.7.2 Distribution of the Sum of Two Random Variables 42
2.7.3 Variance Propagation of Linear Functions 42
2.7.4 Variance Propagation of Nonlinear Functions 43
2.7.5 Implicit Variance Propagation 43
2.7.6 Bias Induced by Linearization 44
2.7.7 On the Mean and the Variance of Ratios 46
2.7.8 Unscented Transformation 47

Propagation of uncertainty can be formalized as follows: Given one or several random variables collected in the random vector $\underline{\boldsymbol{x}}$, together with its probability density function $p_x(\boldsymbol{x})$, and a function $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, derive the probability density function of the random vector $\underline{\boldsymbol{y}}$.

There are several methods for solving this problem (cf. Papoulis and Pillai, 2002). We want to present two important cases with one and two random variables having arbitrary distribution and then discuss linear and nonlinear functions of Gaussian variables.

2.7.1 Transformation of a Random Variable

We first discuss the case of a monotonically increasing function $\underline{y} = f(\underline{x})$ of a single variable $\underline{x}$ with its given probability density function $p_x(x)$. The unknown probability density function of the random variable $\underline{y}$ is $p_y(y)$.

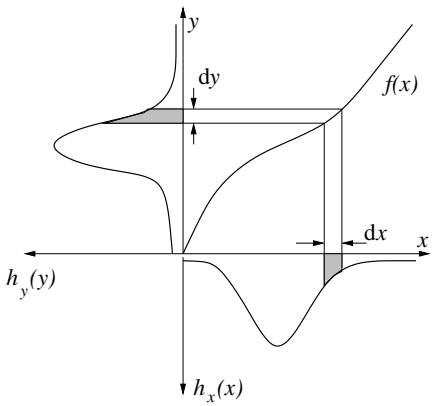


Fig. 2.8 Transformation of a random variable $\underline{x}$ with a monotonic function $y = f(x)$

With Fig. 2.8 we have $p_y(y)\,\mathrm{d}y = p_x(x)\,\mathrm{d}x$ as $P(\underline{y} \in [y, y+\mathrm{d}y]) = P(\underline{x} \in [x, x+\mathrm{d}x])$ for differential $\mathrm{d}x$ and $\mathrm{d}y$. Thus, with monotonic $f(x)$, we obtain

$$p_y(y) = \frac{p_x(x)}{\left|\frac{\mathrm{d}y}{\mathrm{d}x}\right|} = \frac{p_x(x)}{|f'(x)|}\,. \tag{2.127}$$

With the inverse function $x = f^{-1}(y)$, we finally obtain the density $p_y(y)$ of $\underline{y}$ as a function of y,

$$p_y(y) = \frac{p_x\left(f^{-1}(y)\right)}{\left|f'\left(f^{-1}(y)\right)\right|}\,. \tag{2.128}$$

This result generalizes to vector-valued variables (cf. Papoulis and Pillai, 2002, p. 142). Exercise 2.28

Example 2.7.5: ***Linear transformation of a random variable.*** For the linear transformation $y = f(x) = k + mx$, we use the first derivative $f'(x) = m$ and the inverse function

$$f^{-1}(y) = \frac{y-k}{m}$$

to obtain the density

$$p_y(y) = \frac{p_x\left(\dfrac{y-k}{m}\right)}{|m|}\,. \tag{2.129}$$

Obviously, the original density function $p_x(x)$ is translated by k and scaled by m in the y- and p_y-directions in order to obtain the area 1 under $p_y(y)$.

A Gaussian random variable $\underline{x} \sim \mathcal{N}(\mu, \sigma^2)$ thus can be transformed into a normalized Gaussian random variable $\underline{z} = \mathcal{N}(0,1)$ by

$$z = \frac{x-\mu}{\sigma}\,. \tag{2.130}$$

This can be generalized to a normally distributed random d-vector $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The vector *whitening*

$$\underline{\boldsymbol{z}} = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - \boldsymbol{\mu}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d) \tag{2.131}$$

follows a normalized multivariate normal distribution. The inverse square root of the matrix $\boldsymbol{\Sigma}$ with eigenvalue decomposition $\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{\mathsf{T}}$ can be determined by $\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{R}\mathrm{Diag}([1/\sqrt{\lambda_i}])\boldsymbol{R}^{\mathsf{T}}$. As a vector whose elements $\underline{z}_i \sim \mathcal{N}(0,1)$ are mutually independent with zero mean is called *white*, the operation (2.131) is called *whitening*. ◇

2.7.2 Distribution of the Sum of Two Random Variables

The density of the sum $\underline{z} = \underline{x} + \underline{y}$ of two independent random variables with densities $p_x(x)$ and $p_y(y)$ is

$$p_z(z) = \int p_x(z-y)p_y(y)\,\mathrm{d}y \tag{2.132}$$

$$p_z = p_x * p_y \tag{2.133}$$

and is thus identical to the *convolution* $p_x * p_y$ of the two densities p_x and p_y (Castleman, 1996).

In many cases, we have several random variables $\underline{x}_i$ which follow a joint normal distribution and which are possibly mutually correlated, $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$. We are interested in the distribution of new random variables $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}}) = [f_i(\underline{\boldsymbol{x}})]$. Due to the nonlinearity of the functions f_i, the resulting density $p_y(\boldsymbol{y})$ is complicated.

2.7.3 Variance Propagation of Linear Functions

Probability functions often are smooth and thus may be locally approximated by a linear function. Moreover, the relative precision of the quantities involved (the random variables $\underline{\boldsymbol{x}}$) is high; thus, their standard deviations are small compared to the curvature of the functions. Under these conditions, we may approximate the resulting distribution by a normal distribution and characterize it by its first two moments, the mean and the covariance matrix.

We first give the distribution of linear functions, for which the variance propagation follows.

Given random variables $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$ and the linear function $\underline{\boldsymbol{y}} = A\underline{\boldsymbol{x}} + \boldsymbol{b}$, the random vector $\underline{\boldsymbol{y}}$ is normally distributed as

$$\underline{\boldsymbol{y}} \sim \mathcal{N}(A\boldsymbol{\mu}_x + \boldsymbol{b}, A\Sigma_{xx}A^\mathsf{T})\,, \tag{2.134}$$

or

$$\mathbb{E}(\underline{\boldsymbol{y}}) = A\mathbb{E}(\underline{\boldsymbol{x}}) + \boldsymbol{b}\,, \qquad \mathbb{D}(\underline{\boldsymbol{y}}) = A\mathbb{D}(\underline{\boldsymbol{x}})A^\mathsf{T}\,. \tag{2.135}$$

The proof for the preservation of the distribution uses the result of the transformation of random variables.

The proof for the first two moments uses the linearity of the expectation operator, which allows us to exchange the expectation and matrix multiplication $\mathbb{E}(\underline{\boldsymbol{y}}) = \mathbb{E}(A\underline{\boldsymbol{x}} + \boldsymbol{b}) = A\mathbb{E}(\underline{\boldsymbol{x}}) + \boldsymbol{b} = A\boldsymbol{\mu}_x + \boldsymbol{b}$ with a similar proof for the second central moments.

Comments:

- As the variance $\mathbb{V}(\underline{y}_i) = \sigma^2_{y_i}$ of an arbitrary element $\underline{y}_i$ for arbitrary matrices A needs to be nonnegative, the covariance matrix Σ_{xx} needs to be positive semi-definite.
- Though the density function of the normal distribution is not defined for singular covariance matrices, the probability function exists. Variance propagation uses only the moments, so it is allowed for singular covariance matrices as well. If A does not have full rank, then Σ_{yy} is singular.
- The proof only uses the moments. It is thus valid for arbitrary distributions $\mathcal{M}_x(\boldsymbol{\mu}_x, \Sigma_{xx})$ for which we only use the first two moments, $\boldsymbol{\mu}_x$ and Σ_{xx}. Therefore, we have the following law of *variance propagation*:

variance propagation

$$\underline{\boldsymbol{x}} \sim \mathcal{M}_x(\boldsymbol{\mu}_x, \Sigma_{xx}) \quad \text{and} \quad \underline{\boldsymbol{y}} = A\underline{\boldsymbol{x}} + \boldsymbol{b} \quad \rightarrow \quad \underline{\boldsymbol{y}} \sim \mathcal{M}_y(A\boldsymbol{\mu}_x + \boldsymbol{b}, A\Sigma_{xx}A^\mathsf{T})\,. \tag{2.136}$$

- The inverse W_{xx} of a regular covariance matrix Σ_{xx} is sometimes called a *weight matrix* or the *precision matrix* (cf. Bishop, 2006),

  weight matrix, precision matrix

  $$W_{xx} = \Sigma_{xx}^{-1}\,, \tag{2.137}$$

  as random variables with smaller variances have higher weights and higher precision when performing an estimation (Sect. 4.1.4, p. 79).
  If A is invertible, we also have a propagation law for weight matrices,

  $$W_{yy} = A^{-1} W_{xx} A^{-\mathsf{T}}\,. \tag{2.138}$$

- We can transfer the result to linear functions of random matrices. Given the random matrix $\underline{X} \sim \mathcal{M}(\mathbb{E}(\underline{X}), \mathbb{D}(\mathrm{vec}\underline{X}))$ and the linear function $\underline{Y} = A\underline{X}B + C$, the random matrix $\underline{Y}$ is normally distributed since

  $$\underline{Y} \sim \mathcal{M}(A\mathbb{E}(\underline{X})B + C, (B^{\mathsf{T}} \otimes A)\Sigma_{xx}(B^{\mathsf{T}} \otimes A)^{\mathsf{T}})\,. \tag{2.139}$$

  Using the vectors $\underline{\boldsymbol{x}} = \mathrm{vec}\underline{X}$ and $\underline{\boldsymbol{y}} = \mathrm{vec}\underline{Y}$ this result immediately follows from the vectorized function $\underline{\boldsymbol{y}} = (B^{\mathsf{T}} \otimes A)\underline{\boldsymbol{x}} + \mathrm{vec}C$ (cf. (A.95), p. 775).

2.7.4 Variance Propagation of Nonlinear Functions

In the case of nonlinear functions $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}})$, we first perform a Taylor series expansion,

$$\underline{\boldsymbol{y}} = \boldsymbol{y}^{(0)} + \mathrm{d}\underline{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}^{(0)}) + J\mathrm{d}\underline{\boldsymbol{x}} + O(|\mathrm{d}\underline{\boldsymbol{x}}|^2), \tag{2.140}$$

with the Jacobian

$$J = [J_{ij}] = \left[\frac{\partial f_i(\boldsymbol{x})}{\partial x_j}\right]\Bigg|_{x=x^{(0)}}\,, \tag{2.141}$$

where – to simplify notation – the subscript $x = x^{(0)}$ refers to the vector $\boldsymbol{x}$. If we use $\boldsymbol{x}^{(0)} = \boldsymbol{\mu}_x$ with $\boldsymbol{y}^{(0)} = \boldsymbol{f}(\boldsymbol{x}^{(0)})$, we obtain

$$\mathrm{d}\underline{\boldsymbol{y}} = J\,\mathrm{d}\underline{\boldsymbol{x}}, \tag{2.142}$$

and therefore in a first-order approximation

$$\mathbb{E}(\underline{\boldsymbol{y}}) \approx \boldsymbol{\mu}_y^{(1)} = \boldsymbol{f}(\boldsymbol{\mu}_x)\,, \qquad \mathbb{D}(\underline{\boldsymbol{y}}) \approx \Sigma_{yy}^{(1)} = J\Sigma_{xx}J^{\mathsf{T}} \tag{2.143}$$

since, up to a first-order approximation,

$$\Sigma_{yy} = \Sigma_{\mathrm{d}y\,\mathrm{d}y} \tag{2.144}$$

due to $\underline{\boldsymbol{y}} \approx \boldsymbol{y}^{(0)} + \mathrm{d}\underline{\boldsymbol{y}}$.

It can be shown that with relative errors $r_{x_j} = \sigma_{x_j}/\mu_{x_j}$ of the variables $\underline{x}_i$, the error in the standard deviations σ_{y_j} due to linearization is less than $r_{x_j}\sigma_{y_i}$, and is thus negligible in most practical applications; cf. Sect. 2.7.6, p. 44.

2.7.5 Implicit Variance Propagation

If we have an implicit relation

$$\boldsymbol{f}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \boldsymbol{0} \tag{2.145}$$

between two stochastic variables $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$, the variance propagation can be performed with the Jacobians

$$A = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\right|_{x=\mu_x, y=\mu_y} \qquad B = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}}\right|_{x=\mu_x, y=\mu_y} \tag{2.146}$$

if B is invertible. From $\mathrm{d}\boldsymbol{f} = A\,\mathrm{d}\boldsymbol{x} + B\,\mathrm{d}\boldsymbol{y} = \boldsymbol{0}$ we obtain $\mathrm{d}\underline{\boldsymbol{y}} = -B^{-1}A\,\mathrm{d}\underline{\boldsymbol{x}}$ with given Σ_{xx}, again, in a first-order approximation,

$$\Sigma_{yy} = B^{-1}A\Sigma_{xx}A^{\mathsf{T}}B^{-\mathsf{T}}\,. \tag{2.147}$$

This allows the derivation of the covariance matrix of $\underline{\boldsymbol{y}}$ even if the procedure for deriving $\boldsymbol{y}$ from $\boldsymbol{x}$ is very complicated.

2.7.6 Bias Induced by Linearization

bias: deviation from the true value

Moment propagation (2.143) of nonlinear functions using only the first-order Taylor series of the nonlinear function leads to a systematic deviation from the *true value*, also called *bias*. Analysing higher-order terms yields expressions for the bias due to linearization.

For a scalar function $y = f(x)$ of a scalar x, it is based on the Taylor expansion of the stochastic variable at $f(\mu_x)$,

$$\begin{aligned}\underline{y} = f(\underline{x}) = f(\mu_x) + f'(\mu_x)(\underline{x}-\mu_x) + \frac{1}{2}f''(\mu_x)(\underline{x}-\mu_x)^2 \\ + \frac{1}{6}f'''(\mu_x)(\underline{x}-\mu_x)^3 + \frac{1}{24}f^{(4)}(\mu_x)(\underline{x}-\mu_x)^4 + O((\underline{x}-\mu_x)^n)\,.\end{aligned} \tag{2.148}$$

We therefore obtain the following result: if the density function of a stochastic variable $\underline{x}$ is symmetrical, the mean for $\underline{y} = f(\underline{x})$ can be shown to be

$$\mathbb{E}(\underline{y}) = \mu_y = f(\mu_x) + \frac{1}{2}f''(\mu_x)\sigma_x^2 + \frac{1}{24}f^{(4)}(\mu_x)\mu_{4_x} + O(f^{(n)}, m_n) \qquad n > 4. \tag{2.149}$$

For normally distributed variables we take its central fourth moment $\mu_{4_x} = 3\sigma_x^4$. Using the expression $\mathbb{V}(\underline{y}) = \mathbb{E}(\underline{y}^2) - [\mathbb{E}(\underline{y})]^2$ from (2.94), p. 37 we can derive a similar expression for the variance. Restricting to even moments up to the fourth-order for Gaussian variables, we have

Exercise 2.30

$$\mathbb{V}(\underline{y}) = \left[\sigma_y^{(2)}\right]^2 = f'^2(\mu_x)\,\sigma_x^2 + \left(f'(\mu_x)f'''(\mu_x) + \frac{1}{2}f''^2(\mu_x)\right)\sigma_x^4 + O(f^{(n)}, m_n)\,. \tag{2.150}$$

Obviously the bias, i.e., the second term, depends on the variance and the higher-order derivatives: the larger the variance and the higher the curvature or the third derivative, the higher the bias. Higher-order terms again depend on derivatives and moments of order higher than 4.

expectation of function of stochastic vector

For a stochastic vector $\underline{\boldsymbol{x}}$ with symmetrical density function, the mean of the scalar function $y = f(\boldsymbol{x})$ can be shown to be

Exercise 2.31

$$\mathbb{E}(\underline{y}) = \mu_y^{(2)} = f(\boldsymbol{\mu}_x) + \frac{1}{2}\mathrm{trace}(H|_{x=\mu_x}\cdot\Sigma_{xx}) + O(f^{(n)}, m_n), \quad n \geq 3\,, \tag{2.151}$$

with the Hessian matrix $H = (\partial f^2/\partial x_i \partial x_j)$ of the function $f(\boldsymbol{x})$. This is a generalization of (2.149).

We now discuss two cases in more detail which regularly occur in geometric reasoning, the bias of a product and the bias of normalizing a vector to length 1.

Bias of a Product. The product $z = xy$ of two random variables is part of all geometric constructions when using *homogeneous coordinates* for representing geometric entities. For the product

$$\underline{z} = \underline{x}\,\underline{y} \tag{2.152}$$

of two possibly correlated normal random variables

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \right), \tag{2.153}$$

we obtain the first and second approximation for the mean value Exercise 2.32

$$\mu_z^{[1]} = \mu_x\mu_y \qquad \mu_z^{[2]} = \mu_z^{[1]} + \rho_{xy}\sigma_x\sigma_y\,. \tag{2.154}$$

Thus we obtain the *bias of the mean*,

$$b_{\mu_z} \doteq \mu_z^{[2]} - \mu_z^{[1]} = \sigma_{xy} = \rho_{xy}\sigma_x\sigma_y, \tag{2.155}$$

and the *relative bias of the mean* of the product,

$$r_{\mu_z} \doteq \frac{b_{\mu_z}}{\mu_z} = \rho_{xy}\frac{\sigma_x}{\mu_x}\frac{\sigma_y}{\mu_y}\,. \tag{2.156}$$

The relative bias of the mean is the product of the relative accuracies σ_x/μ_x and σ_y/μ_y multiplied with the correlation coefficient. The bias is zero if the random variables are uncorrelated, which is often the case when constructing a geometric entity from two others. The proof of (2.154), p. 45 uses Exercise 2.33

$$\mathbb{E}((\underline{x} - \mu_x)^2(\underline{y} - \mu_y)^2) = (1 + 2\rho_{xy})\sigma_x^2\sigma_y^2\,. \tag{2.157}$$

Similarly, we have the first- and second-order approximation for the standard deviation, Exercise 2.34

$$\sigma_z^{[1]} = \mu_y^2\sigma_x^2 + \mu_x^2\sigma_y^2 + 2\mu_x\mu_y\sigma_{xy} \qquad \sigma_z^{[2]} = \sigma_z^{[1]} + (1 + \rho_{xy}^2)\sigma_x^2\sigma_y^2\,. \tag{2.158}$$

The *bias of the variance* is

$$b_{\sigma_z^2} = \sigma_z^{2[2]} - \sigma_z^{2[1]} = \sigma_x^2\sigma_y^2 + \sigma_{xy}^2 = (1 + \rho_{xy}^2)\sigma_x^2\sigma_y^2\,, \tag{2.159}$$

and therefore the *relative bias of the variance*,

$$r_{\sigma_z^2} = \frac{b_{\sigma_z^2}}{\sigma_z^2} = \frac{(1 + \rho_{xy}^2)\sigma_x^2\sigma_y^2}{\mu_y^2\sigma_x^2 + \mu_x^2\sigma_y^2 + 2\mu_x\mu_y\sigma_{xy}}, \tag{2.160}$$

does not vanish for uncorrelated random variables.

If the variables are uncorrelated and have the same relative precision, i.e., $\sigma_x/\mu_x \approx \sigma_y/\mu_y \approx \sigma/\mu$, we obtain the relative bias

$$r_{\sigma_z^2} = \frac{b_{\sigma_z^2}}{\sigma_z^2} \approx \frac{1}{2}\left(\frac{\sigma}{\mu}\right)^2. \tag{2.161}$$

Thus, the relative bias $r_{\sigma_z^2}$ of the variance is approximately half of the square of the relative precision σ/μ.

Bias of Normalization. The normalization of an n-vector $\mathbf{x}$ to unit length, which we will apply to homogeneous coordinates regularly (Sect. 5.1, p. 195), is given by

$$\mathbf{x}^s = \frac{\mathbf{x}}{|\mathbf{x}|} \qquad \text{or} \qquad \mathbf{x}_i^s = \frac{x_i}{|\mathbf{x}|}\,. \tag{2.162}$$

We assume $\underline{\boldsymbol{x}}$ has covariance matrix $\boldsymbol{\Sigma}_{\mathrm{xx}} = \sigma_x^2 I_n$. This leads to the following expression for the mean when taking terms up to the fourth-order into account:

Exercise 2.35

$$\mathbb{E}(\underline{\mathbf{x}}^s) = \frac{\boldsymbol{\mu}_x}{|\boldsymbol{\mu}_x|}\left(1 - \frac{1}{2}\frac{\sigma_x^2}{|\boldsymbol{\mu}_x|^2}\right). \tag{2.163}$$

Here too, the relative bias, since it is identical to the bias, is approximately half of the square of the relative accuracy.

The bias of the variance behaves in a similar manner as for the product of two entities: the relative bias of the variance follows quadratically with the relative precision of the given entities; cf. (2.161).

In nearly all cases which are practically relevant when geometrically analysing images, the relative precision results from the observation process in images, which is below one pixel (see the following example). Even for wide-angle cameras, the focal length is far beyond 100 pixels. The directional uncertainty is therefore much better than one percent. As a consequence, the relative bias when determining the mean value or the variance using only the first-order approximation is significantly smaller than 0.01%.

2.7.7 On the Mean and the Variance of Ratios

Care has to be taken when deriving Euclidean coordinates, $\boldsymbol{x}$, from homogeneous ones, $\mathbf{x}$, e.g., using the ratios

$$x = \frac{u}{w} \qquad y = \frac{v}{w} \tag{2.164}$$

if the denominator w is uncertain. If $\underline{w} \sim \mathcal{N}(\mu_w, \sigma_w^2)$, the mean and the variance of $\underline{x}$ and $\underline{y}$ are not defined (cf. Hartley and Zisserman, 2000, App. 3). The reason is that with a possibly very small probability the denominator w will be zero; thus, the variable x will be infinite, making the integral $\mu_x = \int_{-\infty}^{\infty} x p(x) \mathrm{d}x$ vanish.

Exercise 2.36

However, the first-order approximation for deriving the mean $\mu_x = \mu_u/\mu_w$ and the variance is still useful due to the practical procedure of preprocessing the observed data $\mathbf{x}$: they are usually checked for outliers, and only the inliers are used in further processing. This preprocessing limits the range of possible random perturbations for the inlying observations, and would make it necessary to work with a distribution with limited support, say $\pm 4\sigma_w$:

$$\underline{w} \mid \text{inlier} \sim p_{w|inlier}(w|\text{inlier}) = \begin{cases} k \cdot g(w \mid \mu_w, \sigma_w^2), & \text{if } w \in [\mu_w - 4\sigma_w, \mu_w + 4\sigma_w] \\ 0, & \text{else} \end{cases} \tag{2.165}$$

with an adequate normalization constant k for the truncated Gaussian density g. This distribution has approximately the same first and second moments as the corresponding Gaussian but does not cause infinite mean or variance if $|\mu_w|$ is far enough from zero, i.e., $|\mu_w| > 4\sigma_w$. Therefore, the classical determination of the mean and the variance by using variance propagation is sufficiently accurate.

In order to be able to handle outliers as well, we model the causing gross error as a shift b_w of the mean,

$$\underline{w} \mid \text{outlier} \sim p_{w|inlier}(w - b_w), \tag{2.166}$$

which also allows variance propagation and is consistent with the model of classical hypothesis testing (Sect. 3.1.1, p. 62), which is the basis for outlier detection, e.g., in a RANSAC procedure (Sect. 4.7.7, p. 153).

We therefore recommend using variance propagation based only on the linearized relations. The example on p. 48 supports the recommendation.

2.7.8 Unscented Transformation

Classical variance propagation of nonlinear functions only uses the first-order terms of the Taylor series. The bias induced by omitting higher-order terms in many practical cases is irrelevant.

We now discuss a method which uses terms up to the fourth-order and in many cases yields results which are accurate up to the second-order. It is called *unscented transformation* (cf. Julier and Uhlmann, 1997).

It is based on the idea of representing the distribution of the given random N-vector $\underline{\boldsymbol{x}}$ by $2N+1$ well-selected points $\boldsymbol{x}_i$ and of deriving the weighted mean vector and the covariance matrix from the nonlinearly transformed points $\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{x}_n)$.

The selected points depend on the square root

$$\mathsf{S}_{xx} = \sqrt{\Sigma_{xx}} = [\boldsymbol{s}_n]\,, \qquad \Sigma_{xx} = \mathsf{S}_{xx}\mathsf{S}_{xx}^{\mathsf{T}} \tag{2.167}$$

of the covariance matrix of the given random variable. Its columns are $\boldsymbol{s}_n$. For numerical reasons, S_{xx} is best determined by Cholesky decomposition (Rhudy et al., 2011). Now we have

$$\Sigma_{xx} = [\boldsymbol{s}_1, ..., \boldsymbol{s}_n, ...\boldsymbol{s}_N] \begin{bmatrix} \boldsymbol{s}_1^{\mathsf{T}} \\ ... \\ \boldsymbol{s}_n^{\mathsf{T}} \\ ... \\ \boldsymbol{s}_N^{\mathsf{T}} \end{bmatrix} = \sum_{n=1}^{N} \boldsymbol{s}_n \boldsymbol{s}_n^{\mathsf{T}}\,. \tag{2.168}$$

The $2N+1$ points $\boldsymbol{x}_n$ and their weights w_n then are:

$$\begin{aligned} \boldsymbol{x}_1 &= \boldsymbol{\mu}_x\,, & w_1 &= \frac{\kappa}{N+\kappa} \\ \boldsymbol{x}_n &= \boldsymbol{\mu}_x + \sqrt{N+\kappa}\,\boldsymbol{s}_n\,, & w_n &= \frac{1}{2(N+\kappa)} \qquad n = 2, ..., N+1 \\ \boldsymbol{x}_{n+N} &= \boldsymbol{\mu}_x - \sqrt{N+\kappa}\,\boldsymbol{s}_n\,, & w_n &= \frac{1}{2(N+\kappa)} \qquad n = N+2, ..., 2N+1\,. \end{aligned} \tag{2.169}$$

They depend on a free parameter κ. The weights add to 1. For Gaussian random variables, we best use

$$\kappa = 3 - N \tag{2.170}$$

in order to obtain minimum bias. As a result, some of the weights may be negative.

Determining the mean and covariance matrix of $\underline{\boldsymbol{y}}$ is performed in three steps:

1. transforming the points

$$\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{x}_n) \qquad n = 1, ..., 2N+1\,, \tag{2.171}$$

2. determining the mean vector

$$\boldsymbol{\mu}_y = \sum_{n=1}^{2N+1} w_n \boldsymbol{y}_n\,, \tag{2.172}$$

and

3. determining the covariance matrix

$$\Sigma_{yy} = \sum_{n=1}^{2N+1} w_n (\boldsymbol{y}_n - \boldsymbol{\mu}_y)(\boldsymbol{y}_n - \boldsymbol{\mu}_y)^{\mathsf{T}} = \left(\sum_{n=1}^{2N+1} w_n \boldsymbol{y}_n \boldsymbol{y}_n^{\mathsf{T}} \right) - \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathsf{T}}\,. \tag{2.173}$$

Example 2.7.6: ***Unscented transformation of a linear function.*** In the case of a linear function $\boldsymbol{y} = \mathsf{A}\boldsymbol{x} + \boldsymbol{a}$, we obtain the same mean and covariance matrix as with the classical variance propagation.

Proof: The mean value $\boldsymbol{\mu}_y$ is obviously identical to $\boldsymbol{f}(\boldsymbol{\mu}_x)$. For the covariance matrix, we use the transformed points $\boldsymbol{y}_1 - \boldsymbol{\mu}_y = \mathbf{0}$ and $\boldsymbol{y}_n - \boldsymbol{\mu}_y = \pm\sqrt{N+\kappa}\,\mathsf{A}\boldsymbol{s}_n$. Then (2.173) yields

$$\boldsymbol{\Sigma}_{yy} = \sum_{n=1}^{N} \frac{1}{2(N+\kappa)} \left((\sqrt{N+\kappa})^2 \boldsymbol{A}\boldsymbol{s}_n \boldsymbol{s}_n^{\mathsf{T}} \boldsymbol{A}^{\mathsf{T}}) + (\sqrt{N+\kappa})^2 (-\boldsymbol{A}\boldsymbol{s}_n)(-\boldsymbol{s}_n^{\mathsf{T}} \boldsymbol{A}^{\mathsf{T}}) \right) = \boldsymbol{A}\boldsymbol{\Sigma}_{xx}\boldsymbol{A}^{\mathsf{T}} .$$

◇

Example 2.7.7: ***Square of a standard Gaussian random variable.*** Here we have $\underline{x} \sim \mathcal{N}(0,1)$ and the function $y = f(x) = x^2$. The mean and the variance can be derived from the general properties of the χ^2 distribution. For the sum $\underline{z} \sim \chi^2_N$ of N squared independent random variables $\underline{u}_n \sim \mathcal{N}(0,1)$, the mean and variance are

$$\mathbb{E}(\underline{z}^2) = N \qquad \mathbb{D}(\underline{z}^2) = 2N . \tag{2.174}$$

In our special case, $n = 1$, the mean is

$$\mathbb{E}(\underline{x}^2) = 1 , \qquad \mathbb{D}(\underline{x}^2) = 2 . \tag{2.175}$$

The classical variance propagation leads to completely wrong results $\mu_y^{(1)} = 0$ and $\sigma_y^{(1)} = 0$, as $y(0) = y'(0) = 0$.

With the unscented transformation, with $N = 1$ we use the $2N + 1 = 3$ points and weights:

$$x_1 = 0 , \quad w_1 = \frac{2}{3} , \qquad x_2 = \sqrt{3} , \quad w_2 = \frac{1}{6} , \qquad x_3 = -\sqrt{3} , \quad w_3 = \frac{1}{6} . \tag{2.176}$$

Therefore we obtain

1. the transformed points $y_1 = 0$, $y_2 = y_3 = 3$,
2. the weighted mean
$$\mu_y = \frac{2}{3} \cdot 0 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 3 = 1 , \tag{2.177}$$
3. the weighted sum of the squares $\sum_{n=1}^{3} w_n y_n^2 = 3$ and therefore the variance
$$\sigma_y^2 = \sum_{n=1}^{3} w_n y_n^2 - \mu_y^2 = 2 . \tag{2.178}$$

Comparison with (2.175) shows that the unscented transformation in this highly nonlinear case yields the correct result. ◇

2.8 Stochastic Processes

2.8.1 Notion of a Stochastic Process 48
2.8.2 Continuous Gaussian Processes 50
2.8.3 Autoregressive Processes 52
2.8.4 Integrated AR Processes 54

In this section we discuss sequences of random variables and their statistical properties. We will use such processes for modelling surface profiles in Chap. 16, p. 727. We address two types of models: (1) using (auto-) covariance functions,[3] which specify the process by its second-order statistics, and (2) using autoregressive processes, which refer to the first-order statistics. Both models allow the generation of sample processes and the estimation of the underlying parameters. They differ in the efficiency for interpolation and the ease of generalizing the concept from one to two dimensions.

2.8.1 Notion of a Stochastic Process

stochastic process

Following the introduction of random variables in Sect. 2.3, p. 24, a stochastic process associates to a certain outcome $s \in S$ of an experiment a function $\underline{x}(t, s)$ depending on the independent variable t (Papoulis and Pillai, 2002): The function

[3] This is in contrast to crosscovariance functions between two different processes.

$$\underline{x}(t) : S \to \mathbb{F} \qquad \underline{x}(t) = \underline{x}(t, s) \tag{2.179}$$

is called a stochastic process. The range $\mathbb{F}$ of functions is to be specified. This notion naturally can be generalized to more functions of more than one variable if the scalar t is replaced by a d-dimensional vector. We start with functions of one variable t as they naturally occur as time series.

Depending on whether we fix t or s, we can interpret $\underline{x}(t, s)$ as

1. a stochastic process $\underline{x}(t, s)$, if t and s are variables ,
2. a sampled function $x(t)$, if s is fixed,
3. a random variable $\underline{x}(s)$, if t is fixed and s is variable, and
4. a sampled value x at time t, if s and t are fixed.

A stochastic process is completely specified by the distribution function

$$P(x_1, ..., x_n; t_1, ..., t_n) = P(\underline{x}(t_1) \leq x_1, ..., \underline{x}(t_n) \leq x_n) \tag{2.180}$$

for arbitrary n and $t_1, ..., t_n$. A stochastic process is called *stationary in the strict sense* if the distribution function is invariant to a shift of the parameters t_n by a common delay. *strict stationarity*

We distinguish between continuous and discrete processes, depending on whether t is taken from a continuous domain $\mathcal{D} \subseteq \mathbb{R}$ or whether t is taken from a discrete domain, e.g., $\mathcal{D} \subseteq \mathbb{Z}$. If a process is discrete, we use n as an independent variable and write

$$\underline{x}(n) = \underline{x}(n, s) , \qquad n \in \mathbb{Z} \tag{2.181}$$

where x depends on a discrete parameter n. Such processes can be interpreted as sequences of random variables, e.g., $\underline{x}(n), n = 1, ..., N$.

Furthermore, we only address Gaussian processes. They are fully characterized by their first and second moments

$$\mu_x(t) = \mathbb{E}(\underline{x}(t)) \quad \text{and} \quad \sigma_{xx'}(t, t') = \mathbb{C}\text{ov}(\underline{x}(t), \underline{x}(t')) \tag{2.182}$$

$$\mu_x(n) = \mathbb{E}(\underline{x}(n)) \quad \text{and} \quad \sigma_{xx'}(n, n') = \mathbb{C}\text{ov}(\underline{x}(n), \underline{x}(n')) . \tag{2.183}$$

In the following paragraphs we refer to continuous and discrete processes using t as an argument.

A stochastic process is called *weakly stationary* if the first and second moments do not depend on time. Then we have $\mu_x(t) = \mu_x(t')$ or *weak stationarity*

$$\mu_x = \mathbb{E}_x(\underline{x}(t)) = \int x\, p(x, t)\, \mathrm{d}x \qquad \text{for all } t \tag{2.184}$$

and $\sigma(t + u, t' + u) = \sigma(t, t')$. With the difference between two variables, which is called the *lag*, *lag*

$$d = t' - t, \tag{2.185}$$

we obtain

$$\sigma_{xx'}(d) = \sigma_{xx'}(t, t + d) = \sigma_{xx'}(-d) , \tag{2.186}$$

the last relation resulting from the symmetry of the covariance of two random variables. The function $\sigma_{xx'}(d)$ is the *covariance function* of the stationary process and often written as *covariance function*

$$C_{xx}(d) = \mathbb{C}\text{ov}(\underline{x}(t), \underline{x}(t + d)) . \tag{2.187}$$

A stationary stochastic process is therefore characterized by its mean μ_x and its covariance function $C_{xx}(d)$.

We first discuss continuous processes specified by their covariance function, and then a special class of models which define the sequence of the random variables recursively.

2.8.2 Continuous Gaussian Processes

A stationary continuous Gaussian process is characterized by the mean value μ_x and the covariance function $C_{xx}(d)$. We discuss the main properties of covariance functions.

Stationary One-Dimensional Gaussian Processes. The covariance function C_{xx} needs to guarantee that, for any i, the vector $\boldsymbol{x} = [x(t_i)], i = 1, ..., I$, the covariance matrix

$$\begin{aligned}\Sigma_{xx} = \mathbb{D}(\underline{\boldsymbol{x}}) &= \begin{bmatrix} \mathbb{C}\mathrm{ov}\,(x(t_1), x(t_1)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_1), x(t_i)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_1), x(t_I)) \\ \dots & \dots & \dots & \dots & \dots \\ \mathbb{C}\mathrm{ov}\,(x(t_i), x(t_1)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_i), x(t_i)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_i), x(t_I)) \\ \dots & \dots & \dots & \dots & \dots \\ \mathbb{C}\mathrm{ov}\,(x(t_I), x(t_1)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_I), x(t_i)) & \dots & \mathbb{C}\mathrm{ov}\,(x(t_I), x(t_I)) \end{bmatrix} \\ &= \begin{bmatrix} C_{xx}(0)) & \dots & C_{xx}(t_1 - t_i) & \dots & C_{xx}(t_1 - t_I) \\ \dots & \dots & \dots & \dots & \dots \\ C_{xx}(t_i - t_1) & \dots & C_{xx}(0) & \dots & C_{xx}(t_i - t_I) \\ \dots & \dots & \dots & \dots & \dots \\ C_{xx}(t_I - t_1) & \dots & C_{xx}(t_I - t_i) & \dots & C_{xx}(0) \end{bmatrix} \end{aligned} \tag{2.188}$$

is positive semi-definite. This can be achieved if we choose a *positive semi-definite function*. Following Bochner's theorem (cf. Rasmussen and Williams, 2005, Sect. 4.2.1), a positive definite function is a function whose Fourier transform is positive, or which can be written as

positive semi-definite and positive definite functions

$$C_{xx}(d) = \sum_{k=0}^{\infty} c_k \cos(2\pi k d) \tag{2.189}$$

with

$$\sigma_x^2 = \sum_{k=0}^{\infty} c_k < \infty \quad \text{and} \quad c_k > 0, \text{ for all } k\,. \tag{2.190}$$

If the coefficients fulfil $c_k \geq 0$, the function is called positive semi-definite. Observe that the diagonal elements of the covariance matrix are identical to the variance of the process: $C_{xx}(0) = \sigma_x^2$. Similarly we have positive semi-definite correlation functions using (2.103), p. 38,

$$R_{xx}(d) = \frac{C_{xx}(d)}{C_{xx}(0)} = \frac{C_{xx}(d)}{\sigma_x^2}\,. \tag{2.191}$$

Examples of correlation functions are

$$R_1(d) = \begin{cases} 1, & \text{if } d = 0 \\ 0, & \text{else} \end{cases} \tag{2.192}$$

$$R_2(d) = \exp\left(-\frac{|d|}{|d_0|}\right) \tag{2.193}$$

$$R_3(d) = \exp\left(-\frac{1}{2}\left(\frac{d}{d_0}\right)^2\right) \tag{2.194}$$

$$R_4(d) = \frac{1}{1 + \left(\frac{d}{d_0}\right)^2} \tag{2.195}$$

with some reference distance d_0.

Linear combinations $h(d) = af(d) + bg(d)$ with positive coefficients a and b and products $h(d) = f(d)g(d)$ of two positive functions $f(d)$ and $g(d)$ again are positive definite functions.

Figure 2.9 shows three samples of a Gaussian process $x(t_k), k = 1, 2, 3$. The standard deviation of the processes is $\sigma_x = 1$. The covariance function is $C_{xx}(d) = \exp\left(-\frac{1}{2}(d/20)^2\right)$, cf. R_3 in (2.194). The method for generating such sequences is given in Sect. 2.9, p. 55.

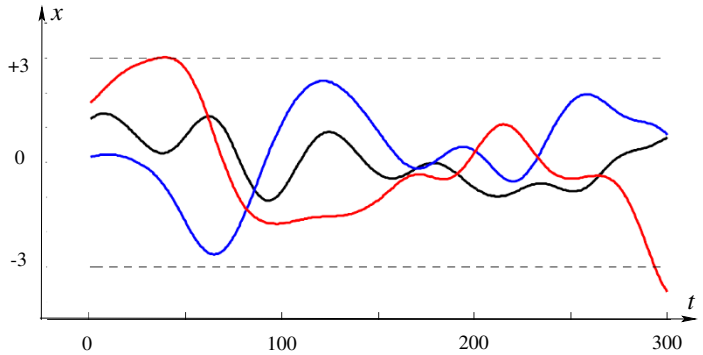


Fig. 2.9 Three samples of size 300 of a Gaussian process with mean 0, standard deviation $\sigma_x = 1$, and correlation function $R_3(d)$ with $d_0 = 20$

Homogeneous and Isotropic Higher Dimensional Gaussian Processes. The concept of stationary Gaussian processes can be generalized to functions depending on two or more variables, collected in a vector, say $\boldsymbol{u}$. They usually are applied to represent spatial stochastic processes. We refer to a two-dimensional stochastic process $\underline{x}(\boldsymbol{u}, s)$ in the following. It will be used to describe the random nature of surfaces, where x represents the height and $\boldsymbol{u} = [u, v]$ the position.

For spatial processes the concept of invariance to translation is called *homogeneity*, which is equivalent to the notion of *stationarity* for time processes. Moreover, the characteristics of spatial processes may be also invariant to rotation. A higher dimensional stochastic process is called *isotropic* if the covariance between two values $\underline{x}(\boldsymbol{u}_1)$ and $\underline{x}(\boldsymbol{u}_2)$ does not depend on a rotation of the coordinate system: $\mathbb{C}\text{ov}(\underline{x}(\boldsymbol{u}_1), \underline{x}(\boldsymbol{u}_2)) = \mathbb{C}\text{ov}(R\underline{x}(\boldsymbol{u}_1), R\underline{x}(\boldsymbol{u}_2))$ for an arbitrary rotation matrix R.

homogeneous stochastic process

isotropic stochastic process

Now, homogeneous and isotropic Gaussian processes can again be characterized by their mean μ_x and their covariance function

$$C_{xx}(d(\boldsymbol{u}, \boldsymbol{u}')) = \mathbb{C}\text{ov}(x(\boldsymbol{u}), x(\boldsymbol{u}')) \tag{2.196}$$

where the distance $d = d(\boldsymbol{u}, \boldsymbol{u}') = |\boldsymbol{u}' - \boldsymbol{u}|$ is the Euclidean distance between the positions $\boldsymbol{u}$ and $\boldsymbol{u}'$. Again, an arbitrary covariance matrix Σ_{xx} must be positive semi-definite.

Remark: If the distance $d = |\boldsymbol{u}' - \boldsymbol{u}|$ is replaced by a weighted distance, say $d = \sqrt{(\boldsymbol{u}' - \boldsymbol{u})^\mathsf{T} W (\boldsymbol{u}' - \boldsymbol{u})}$, with a constant positive definite matrix W, the stochastic process still is homogeneous, but anisotropic. Generalizing the concept to nonhomogeneous anisotropic processes is out of the scope of this book. ◇

Representing stochastic processes using covariance functions can be seen as characterizing the second moments of vectors of random variables, where the index refers to a parameter, say t, of a continuous or discrete domain. This has the advantage of generalizing the concept to more dimensions. Next we discuss a class of models for stochastic processes which are based on a generative model for the process itself, which has the advantage of leading to more efficient computational schemes.

2.8.3 Autoregressive Processes

An autoregressive model AR(P) of order P is characterized by P parameters $a_p, p = 1, ..., P$, and a variance σ_e^2. It uses a sequence $\underline{e}_n \sim \mathcal{M}(0, \sigma_e)$ of identically and independently distributed (iid) random variables. This sequence controls the stochastic development of the stochastic process $\underline{x}_n$; therefore, it is often called the driving process. Starting from a set of P random variables $\underline{x}_n$, with $\mathbb{E}(\underline{x}_n) = 0$, the elements $\underline{x}_n, n > P$, of the random sequence linearly and deterministically depend on the previous P values, $\underline{x}_{n-p}$ of the sequence and the nth element, $\underline{e}_n$, of the driving process, in the following manner:

$$\underline{x}_n = \sum_{p=1}^{P} a_p \, \underline{x}_{n-p} + \underline{e}_n \,, \quad \underline{e}_n \sim \mathcal{M}\left(0, \sigma_e^2\right) \,, \quad n > P \,. \tag{2.197}$$

Since $\mathbb{E}(\underline{e}_n) = 0$, we have

$$\mathbb{E}(\underline{x}_n) = 0 \,. \tag{2.198}$$

If this condition is not fulfilled, the process model may be modified by adding the mean value c:

$$\underline{x}_n = c + \sum_{p=1}^{P} a_p (\underline{x}_{n-p} - c) + \underline{e}_n \,, \quad \underline{e}_n \sim \mathcal{M}\left(0, \sigma_e^2\right) \tag{2.199}$$

The stochastic process is stationary if the generally complex zeros of the polynomial $1 - \sum_{p=1}^{P} a_p z^p$ are outside the unit circle (cf. Box and Jenkins, 1976). We illustrate the situation for the autoregressive model AR(1).

AR(1) Processes. An AR(1) model, using $a := a_1$ for simplicity, is given by:

$$\underline{x}_n = a\underline{x}_{n-1} + \underline{e}_n \,, \quad \underline{e}_n \sim \mathcal{M}\left(0, \sigma_e^2\right) \quad \text{and} \quad |a| < 1 \,. \tag{2.200}$$

We choose the initial value $\underline{x}_0 \sim \mathcal{M}(0, 0)$ and

$$\underline{e}_1 \sim \mathcal{M}\left(0, \frac{1}{1-a^2}\sigma_e^2\right) \tag{2.201}$$

intentionally in order to obtain a stationary process, as can be seen immediately. We recursively obtain

$$\underline{x}_1 = \underline{e}_1 \qquad \sigma_{x_1}^2 = \frac{1}{1-a^2}\sigma_e^2 \tag{2.202}$$

$$\underline{x}_2 = a\underline{e}_1 + \underline{e}_2 \qquad \sigma_{x_2}^2 = \left(\frac{a^2}{1-a^2} + 1\right)\sigma_e^2 \tag{2.203}$$

$$\underline{x}_3 = a^2\underline{e}_1 + a\underline{e}_2 + \underline{e}_3 \qquad \sigma_{x_3}^2 = \left(\frac{a^4}{1-a^2} + a^2 + 1\right)\sigma_e^2 \tag{2.204}$$

$$\ldots \qquad \ldots \tag{2.205}$$

$$\underline{x}_n = a^{n-1}\underline{e}_1 + a^{n-2}\underline{e}_2 + \ldots + \underline{e}_n \qquad \sigma_{x_n}^2 = \left(\frac{a^{2(n-1)}}{1-a^2} + a^{2(n-2)} + \ldots + 1\right)\sigma_e^2 \,. \tag{2.206}$$

As can be checked easily, we therefore have

$$\sigma_x^2 = \frac{\sigma_e^2}{1-a^2} \tag{2.207}$$

independent of n. Obviously, only values $|a| < 1$ lead to stationary sequences with limited variance:

1. For $a = 0$ we have a white noise process.
2. For $a \in (0, 1)$ the process is randomly deviating from zero while keeping close to 0.
3. For $a \in (-1, 0)$ the process is oscillating while staying close to 0.
4. For $|a| > 1$ and first increment $\underline{e}_1 \sim \mathcal{M}(0, \sigma_e^2)$ the process x_n is quickly diverging with $\sigma_{x_n}^2 = (a^{2n} - 1)/(a^2 - 1)\,\sigma_e^2$.

Furthermore, from (2.202)ff. we obtain the covariance function, i.e., the covariance between neighbouring random variables $\underline{x}_n$ and $\underline{x}_{n+d}$,

$$C_{xx}(d) = \mathbb{C}\mathrm{ov}(\underline{x}_n, \underline{x}_{n+d}) = a^d\, \sigma_{x_n}^2\,, \tag{2.208}$$

which is an exponential function of the lag d. Thus the correlation (2.55), p. 31 between neighbouring variables

$$\rho_d = \rho_{x_n, x_{n+d}} = a^d \tag{2.209}$$

decays exponentially with the distance d for $|a| < 1$. The covariance matrix of a sequence $\{x_n\}$ with N values, collected in the N-vector $\underline{\boldsymbol{x}}$, therefore is

$$\mathbb{D}(\underline{\boldsymbol{x}}) = \frac{\sigma_e^2}{1 - a^2} \begin{bmatrix} 1 & a & a^2 & \dots & a^{N-2} & a^{N-1} \\ a & 1 & a & \dots & a^{N-3} & a^{N-2} \\ a^2 & a & 1 & \dots & a^{N-4} & a^{N-3} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a^{N-2} & a^{N-3} & a^{N-4} & \dots & 1 & a \\ a^{N-1} & a^{N-2} & a^{N-3} & \dots & a & 1 \end{bmatrix} = \frac{\sigma_e^2}{1 - a^2} \left[a^{|i-j|} \right] . \tag{2.210}$$

This matrix has a special structure. Its off-diagonal elements only depend on the distance $|i - j|$ from the main diagonal. Such matrices are called *Toeplitz* matrices. *Toeplitz matrix*

Integrated White Noise Processes. For $a = 1$ we obtain a special process: It is a summed white noise process, often called an *integrated white noise process*,

$$\underline{x}_n = \underline{x}_{n-1} + \underline{e}_n\,, \quad \mathbb{D}(\underline{e}_n) = \sigma_e^2 \tag{2.211}$$

with starting value $\underline{x}_0 = 0$. The name of this process results from the sequence

$$\underline{x}_1 = \underline{e}_1 \tag{2.212}$$
$$\underline{x}_2 = \underline{e}_1 + \underline{e}_2 \tag{2.213}$$
$$\underline{x}_3 = \underline{e}_1 + \underline{e}_2 + \underline{e}_3 \tag{2.214}$$
$$\dots = \dots \tag{2.215}$$
$$\underline{x}_n = \sum_{k=1}^{n} \underline{e}_k\,. \tag{2.216}$$

Two samples for such a process with different standard deviations of the driving noise process are given in Fig. 2.10, upper row. They are generated using a random number generator for the sequence e_k (cf. .Sect. 2.9). Rewriting the generating equation in the form

$$\underline{e}_n = \underline{x}_n - \underline{x}_{n-1} \tag{2.217}$$

reveals the driving white noise sequence $\{\underline{e}_n\}$ to represent the discrete approximation of the first derivative of the discrete function x_n. The process is slowly diverging with $\sigma_n = \sqrt{n}\,\sigma_e$. It is not a stationary process.

If we apply a second summation we arrive at the second-order autoregressive process AR(2) with coefficients $a_1 = 2$ and $a_2 = -1$, a *doubly integrated white noise process*,

$$\underline{x}_n = 2\underline{x}_{n-1} - \underline{x}_{n-2} + \underline{e}_n\,, \quad \mathbb{D}(\underline{e}_n) = \sigma_e^2 \tag{2.218}$$

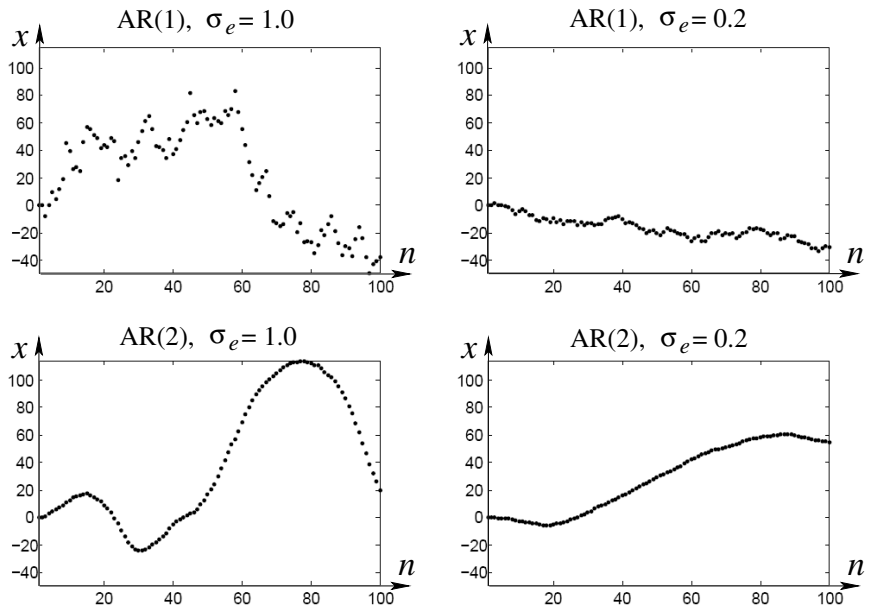


Fig. 2.10 Examples for autoregressive processes. Sequences of 100 points. Integrated and doubly integrated white noise processes (upper and lower row) with standard deviation of driving noise process $\sigma_e = 1.0$ and $\sigma_e = 0.2$ (left and right column)

with starting values values $\underline{x}_0 = \underline{x}_{-1} = 0$. Two examples for such a process are given in Fig. 2.10, lower row. Again solving for $\underline{e}_n$ yields

$$\underline{e}_n = \underline{x}_n - 2\underline{x}_{n-1} + \underline{x}_{n-2}\,. \tag{2.219}$$

Thus $\underline{e}_n$ measures the second derivative of the sequence $\underline{x}_n$ at position $n-1$. Again, as the mean value of the driving noise process $\underline{e}_n$ is zero, the variance σ_e^2 of the AR(2) process measures the smoothness of the sequence.

2.8.4 Integrated AR Processes

We have discussed two versions of an integrating process, where a white noise process drives it. This idea can be generalized to situations where the white noise process drives the first- or higher-order derivatives of the process. When the Dth derivatives of a process follow an AR(P) model, the process is called an integrated autoregressive process, and denoted by ARI(P, D).

As an example, we have an autoregressive model ARI(P,2) for the sequence of second derivatives,

$$\underline{x}_{n-1} - 2\underline{x}_n + \underline{x}_{n+1} = \sum_{p=1}^{P} a_p \underline{x}_{n-p} + \underline{e}_n\,, \tag{2.220}$$

which will turn out to be a good model for terrain profiles. Obviously, this model can be written as

$$\underline{x}_{n+1} = -(\underline{x}_{n-1} - 2\underline{x}_n) + \sum_{p=1}^{P} a_p \underline{x}_{n-p} + \underline{e}_n \tag{2.221}$$

or as an AR($P+1$)-process. It can be written as

$$\underline{x}_n = -(\underline{x}_{n-2} - 2\underline{x}_{n-1}) + \sum_{p=1}^{P} a_p \underline{x}_{n-p-1} + \bar{\underline{e}}_n \tag{2.222}$$

$$= 2\underline{x}_{n-1} + a_1\underline{x}_{n-2} + a_2\underline{x}_{n-3} + ... + a_P\underline{x}_{n-(P+1)} + \bar{\underline{e}}_n \tag{2.223}$$

$$= \sum_{q=1}^{P+1} b_q \underline{x}_{n-q} + \bar{\underline{e}}_n \tag{2.224}$$

with coefficients

$$b_1 = 2\,, \quad b_2 = a_1 - 1\,, \qquad b_q = a_{q-1} \text{ for } q = 3, ..., P+1\,, \quad \bar{\underline{e}}_n = \underline{e}_{n-1}\,. \tag{2.225}$$

2.9 Generating Random Numbers

Testing algorithms involving random variables can be based on simulated data. Here we address the generation of random variables following a certain distribution, which then can be used as input for an algorithm. Software systems provide functions to generate samples of most of the distributions given in this chapter. Visualization of the distributions can be based on scatterplots or histograms.

Take as an example a random variable $\underline{y} \sim \mathcal{N}(\mu_y, \sigma_y^2)$. We want to visualize its distribution for given μ_y and variance σ_y^2. Provided we have a routine for generating a random variable $\underline{x} \sim \mathcal{N}(0, 1)$, we can derive a sample y of a random variable $\underline{y}$ using (2.134), p. 42. We choose the linear function

$$y = \mu_y + \sigma_y\, x \tag{2.226}$$

to derive a sample y from a sample x. Repeating the generation process usually provides statistically independent samples, a property which has to be guaranteed by the random number generator. Alternatively the provided routine allows us to generate vectors or matrices of random numbers. As an example, the package MATLAB provides the function `x=randn(N,M)` to generate an $N \times M$ matrix of random variables $\underline{x}_{nm}$ which follow a standard normal distribution $\underline{x} \sim \mathcal{N}(0, 1)$.

The samples for the autoregressive processes in Fig. 2.10, p. 54 have been generated using a vector $\underline{\boldsymbol{e}}$ of normally distributed random variables $\underline{e}_n$.

A large sample of N values $\underline{x}_n$ can be taken to visualize the distribution via the histogram. The histogram takes a set of K bins $[x_k, x_{k+1})$, which are half open intervals, and counts the number N_k of samples in the bins. The bins usually are equally spaced. A useful number K for the bins is $K = \lfloor\sqrt{N}\rfloor$, as this is a balance between too narrow and too few bins. As the probability P_k that a sample value lies in a bin is $P_k = \int_{x=x_k}^{x_{k+1}} p_x(x)\mathrm{d}x$, and N_k/N is an estimate for this probability, the form of the histogram can be visually compared to the theoretical density $p_x(x)$ by overlaying the histogram by the function $N\,P_k$ using the approximation of $P(\underline{x} \in [x, x + \mathrm{d}x]) = p_x(x)\mathrm{d}x$ (cf. (2.16), p. 26, and Fig. 2.11, top right), namely

$$P_k \approx \frac{1}{2}\left(p_x(x_k) + p_x(x_{k+1})\right)\ \ (x_{k+1} - x_k)\,. \tag{2.227}$$

If we want to generate a sample of a vector of normally distributed values $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_{yy})$, we can proceed similarly. We start from a vector $\boldsymbol{x} = [x_n], n = 1, ..., N$, where the independent samples $\underline{x}_n \sim \mathcal{N}(0, 1)$ follow a standard normal distribution, thus $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{0}, I_N)$. We need the square root S_{yy} of the covariance matrix Σ_{yy} (cf. (2.167), p. 47). Then the linear function

$$\boldsymbol{y} = \boldsymbol{\mu}_y + S_{yy}\,\boldsymbol{x} \tag{2.228}$$

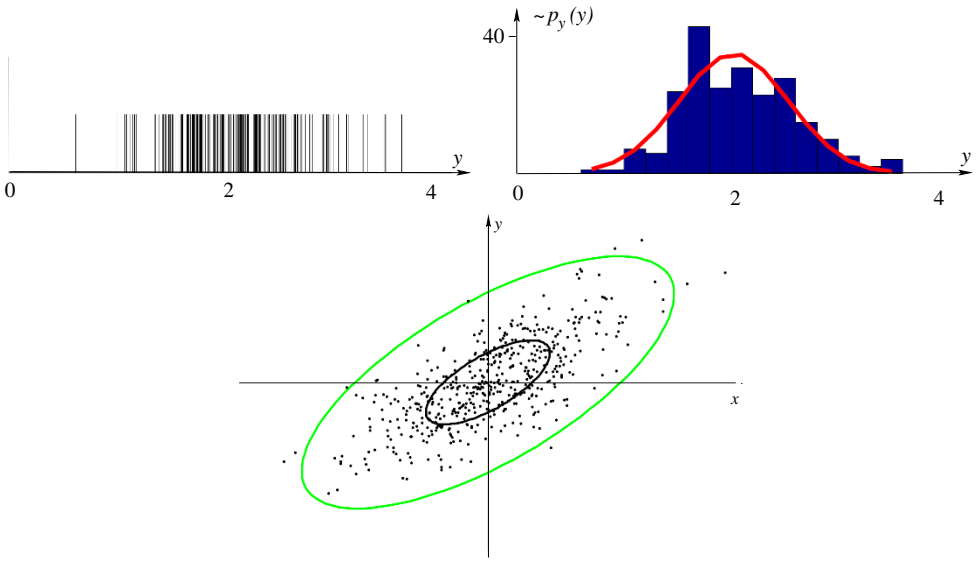


Fig. 2.11 Top row left: One-dimensional scatter plot of a sample of $N = 225$ normally distributed random variables $\underline{y} \sim \mathcal{N}(2, 0.25)$. **Top row right:** Histogram of the same sample with 15 bins, overlayed with its probability density. **Bottom:** 2D scatter plot of $N = 500$ samples of normally distributed random vectors overlayed with the standard ellipse (black) and threefold standard ellipse (green) (Fig. (2.6), p. 32). Approximately 99% of the samples lie in the threefold standard ellipse (Table 2.2, p. 32): $d = 2$, $S = 0.99$

of the sample $\boldsymbol{x}$ of the random vector $\underline{\boldsymbol{x}}$ leads to a sample vector $\boldsymbol{y}$ with distribution $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_{yy})$.

The Gaussian processes in Fig. 2.9, p. 51 have been realized by (1) specifying a regular sequence of $N = 300$ arguments $t = 1, ..., N$, (2) generating the $N \times N$ covariance matrix Σ_{xx} using the standard deviation $\sigma_x = 1$ and the correlation function $R_3(d)$, and (3) taking samples from a normally distributed vector $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{xx})$.

Exercise 2.37 Samples of other distributions can be generated using similar routines.

2.10 Exercises

The number in brackets at the beginning of each exercise indicate its difficulty, cf. Sect. 1.3.2.4, p. 16

Basics

1. (1) How could you randomly choose a month when throwing a die twice? Is the expected probability of all months the same?
2. (1) Give a probability the sun will shine tomorrow? What are the problems when giving such a number?
3. (2) Take a die and throw it repeatedly. Determine the probability of the event 1 after every sixth throw following von Mises' definition of probability. Describe how the determined probability evolves over time. When do you expect to be able to prove that the determined probability converges towards 1/6?
4. (2) You throw a die four times. What is the probability of throwing the sequence $(1, 2, 3, 4)$? What is the probability of throwing three even numbers? What is the probability of throwing 6 at least twice, if the first two throws are $(3, 6)$. What is the probability of throwing the sum 10?

5. (1) Plot the probability and the density function for throwing the numbers 1 to 6 with a die. What would change if the die did not show numbers but six different colours?
6. (2) Plot the density function of n times throwing a 6 when throwing a die $N = 3$ times. Give the density function $p(n)$ explicitly. What is the probability in this experiment of throwing a 6 at least once? Show this probability in a plot of the cumulative probability function.
7. (2) Assume the display of a range sensor can show numbers between 0.000 and 999.999. The sensor may fail, yielding an outlier. Assume the sensor shows an arbitrary number $\underline{s}$ if it fails. Describe the random variable $\underline{s}$ for the outlier. Is it a discrete or continuous random variable? How large is the difference between a discrete and a continuous model for the outlier? What is the probability that $\underline{s} \in [100, 110]$ in the discrete and the continuous model? What changes if the display shows numbers only up to one digit after the dot, i.e., in the range 0.0 to 999.0?
8. (2) Plot the density function of random variables $\underline{x}$ and $\underline{y}$ following the exponential and the Laplace distribution, respectively. Give names to the axes. Give the probability that $\underline{x} \in [-1, 2]$ and $\underline{y} \in [-1, 2]$.

Computer Experiments

9. (3) Use a program for generating M samples of a normal distribution $\mathcal{N}(0, 1)$. Determine the histogram

$$h(x_i|b) = \#(\underline{x} \in [x_i - b/2, x_i + b/2]\,, \quad x_i = ib\,, \quad b \in \mathbb{R}\,,\, i \in \mathbb{Z} \tag{2.229}$$

   from M samples. Prespecify the bin size b. Determine the probability $p(x_i|b) = h(x_i|b)/M$ that a sample falls in a certain bin centred at x_i. Overlay the plot with the density function of the normalized normal distribution $\phi(x)$. How do you need to scale the axes such that the two functions $\phi(x)$ and $p(x_i|b)$ are comparable. Vary the bin size b and the number of samples M. What would be a good bin size if M is given?
10. (2) Repeat the previous exercise for M samples y_m of a χ-square distribution with n degrees of freedom. For this generate $\underline{y}_m$ as the sum of the squares of n samples from a standard normal distribution. Also vary the degrees of freedom n. Describe the distribution for $n = 1, 2, 3$ and for large n.
11. (2) Prove that the bounding box for the standard ellipse has size $2\sigma_x \times 2\sigma_y$. *Hint:* Show the y-coordinate of the highest and lowest point of the ellipse is $\pm\sigma_y$ based on the partial derivative of $(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) = 1$ w.r.t. x, see (2.56), p. 31.
12. (3) Generate a covariance matrix V following a Wishart distribution $\underline{\mathsf{V}} \sim \mathcal{W}(n, \mathsf{I}_2)$. Plot the standard ellipse of V. Repeat the experiment and observe the variation of V. Vary $n = 5, 10, 50$ and discuss the result.
13. (2) This and the following exercise show that it is sufficient to determine the noncentral moments of basic variables, since the central moments and moments of transformed variables linearly depending on the original variables can be expressed as functions of the noncentral moments. As an example we have the relation between the second central moment μ_2 and the moments m_1 and m_2, given by $\mu_2 = m_2 - m_1^2$. This can be generalized to higher-order moments.

    Express the third central moments of a distribution $\mu_{ij}, i + k = 3$ as a function of the third moments $m_{ij}, i + j = 3$.
14. (3) Let the moments of two variables $\underline{x}$ and $\underline{y}$ be denoted by $m_x := m_{10}$, $m_y := m_{01}$, $m_{xx} := m_{20}$, etc. Derive the central second moments m_{uu}, m_{uv}, m_{vv} of the rotated variables $\underline{u}$ and $\underline{v}$,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{2.230}$$

    as a function of ϕ and the noncentral moments of $\underline{x}$ and $\underline{y}$.

15. (1) Given are two correlated random variables $\underline{x}$ and $\underline{y}$ with the same standard deviation σ. Give the standard deviations and the correlation of their sum and their difference. How does the result specialize, if (a) the two random variables are uncorrelated, (b) are correlated with 100%, and (c) are correlated with minus 100%?
16. (1) Show that the correlation coefficient ρ_{xy} between two stochastic variables $\underline{x}$ and $\underline{y}$ lies in the interval $[-1, +1]$, as the covariance matrix needs to be positive semi-definite. Show that the covariance matrix is singular if and only if $\rho = \pm 1$.
17. (1) Prove $\mathbb{E}(a\underline{x} + b) = a\mathbb{E}(\underline{x}) + b$, see (2.109), p. 38.
18. (2) Given are three stochastically independent random variables, $\underline{x} \sim \mathcal{M}(3, 4)$, $\underline{y} \sim \mathcal{M}(-2, 1)$, and $\underline{z} \sim \mathcal{M}(1, 9)$.
    a. (1) Derive the mean and the standard deviation of the two functions
    $$u = 1 + 2x - y\,, \qquad v = -3 + 2y + 3z\,. \tag{2.231}$$
    b. (1) What is the correlation coefficient ρ_{uv}?
    c. (1) Let a further random variable be $\underline{w} = \underline{u} + \underline{z}$. What is the variance of $\underline{w}$ and its correlation ρ_{xw} with $\underline{x}$?
    d. (1) What is the covariance matrix $\mathbb{C}\mathrm{ov}(\underline{u}, [\underline{v}, \underline{w}]^\mathsf{T})$?
19. (2) We want to approximate the normal distribution $\mathcal{N}(\mu, \sigma^2)$ by a uniform distribution such that the mean and the variance is identical to the normal distribution. Give the parameters a and b. Especially relate the range $r = b - a$ of the uniform distribution to the standard deviation σ of the normal distribution. Compare the result to $\sigma_r = \sqrt{1/12}$, see (2.120), p. 39.
20. (1) Given a sequence $\underline{g}(i) \sim \mathcal{M}(\mu(i), \sigma^2), i = 1, 2, 3, ...$ of random variables representing a noisy sampled signal $g(t)$, its discrete derivative can be determined from $g_t(i) = (g(i + 1) - g(i - 1))/2$. Determine the standard deviation of $\underline{g}_t(i)$.
21. (3) We say a random variable $\underline{z} \sim k\chi^2_n$ follows a $k\chi^2_n$ distribution if $\underline{z}/k \sim \chi^2_n$. Given an array $\underline{g}_{ij} \sim \mathcal{M}(\mu_{ij}, \sigma^2)$ of random variables, representing a noisy sampled function $g(x, y)$, the partial derivatives can be derived from
    $$g_x(i, j) = (g(i+1, j) - g(i-1, j))/2\,, \quad g_y(i, j) = (g(i, j+1) - g(i, j-1))/2\,. \tag{2.232}$$
    Give the standard deviations of the two partial derivatives and their covariance. What is the distribution of the squared magnitude $\underline{m}^2(i, j) := |\nabla \underline{g}(i, j)|^2 = \underline{g}^2_x(i, j) + \underline{g}^2_y(i, j)$ of the gradient $\nabla g = [g_x, g_y]^\mathsf{T}$? *Hint:* Which distribution would $\underline{m}^2$ follow if the two random variables $\underline{g}_x$ and $\underline{g}_y$ were standard normally distributed?
22. (1) Let $\underline{y} \sim \chi^2_2$ be χ-square distributed with two degrees of freedom. Determine the mean μ_y. Relate the α-percentile $\chi_{2,\alpha}$ to the mean.
23. (2) Given a random variable $\underline{x} \sim \mathcal{N}(0, 1)$, show that $\underline{x}^2 \sim \chi^2_1$.
24. (2) Given the basis b of two cameras with principal distance c and the x-coordinates x' and x'' of the two image points of a scene point, its distance Z from the camera is given by
    $$Z = \frac{bc}{x'' - x'}\,. \tag{2.233}$$
    Assume the variables, namely b, c, x', and x'', are uncertain, with individual standard deviations σ_b, σ_c, $\sigma_{x'}$, and $\sigma_{x''}$, respectively, and mutually independent. Derive the standard deviation σ_Z of $\underline{Z}$. Derive the relative precision σ_Z/μ_Z of $\underline{Z}$ as a function of the relative precision of the three variables $\underline{b}$, $\underline{c}$, and $\underline{p} = \underline{x}'' - \underline{x}'$.
25. (2) Given are two points $\boldsymbol{p} = [2, 1]^\mathsf{T}$ m and $\boldsymbol{q} = [10, 9]^\mathsf{T}$ m. Their distances to an unknown point $\boldsymbol{x} = [x, y]$ are $s = 5$ m and $t = 13$ m and have standard deviation $\sigma_s = \sigma_r = 0.1$ m.

a. (1) Prove that the two intersection points of the circles around $\boldsymbol{p}$ and $\boldsymbol{q}$ are $\boldsymbol{x}_1 = [14, 6]^\mathsf{T}$ m and $\boldsymbol{x}_2 = [7, 13]^\mathsf{T}$ m.
b. (2) Derive the covariance matrix of the intersection point $\boldsymbol{x}_1$.

26. (3) Given is the function $y = f(x) = x^4 - x^3$ and the random variable $\underline{x} \sim \mathcal{N}(0, 1)$. Derive the mean and the variance of $\underline{y} = f(\underline{x})$

    a. using variance propagation,
    b. using the unscented transformation,
    c. using 10,000 samples of $\underline{x}$ as reference,

    and compare.

Proofs

27. (1) Steiner's theorem ((2.94), p. 37) relates the noncentral second and the central second moments of a variable via the mean. Generalize the theorem to multivariate variables.
28. (1) Prove the expression (2.70), p. 34 for the χ distribution. *Hint:* Apply (2.128), p. 41 to (2.66), p. 33.
29. (1) Refer to the Wishart distribution ((2.71), p. 34) and prove that for $\boldsymbol{\Sigma} = 1$ and $V = y$ we obtain the χ^2 distribution ((2.66), p. 33).
30. (1) Prove the expression (2.150), p. 44 for the second-order approximation for the variance.
31. (1) Prove the expression (2.151), p. 44 for the second-order approximation of the mean of a function depending on a vector.
32. (1) Prove the first- and second-order approximation (2.154), p. 45 for the mean of a product.
33. (2) Prove the expression (2.157), p. 45 for the expectation of $(\underline{x} - \mu_x)^2(\underline{y} - \mu_y)^2$ of two correlated Gaussian variables. *Hint:* Assume $\mu_x = \mu_y = 0$.
34. (1) Prove the expression (2.158), p. 45 for the second-order approximation of the expectation of a random vector, which is normalized to length 1.
35. (1) Prove (2.163), p. 46. *Hint:* use (2.151), p. 44 for each component x_i of $\mathbf{x}$.
36. (1) Let the random variable $\underline{x} \sim \mathcal{N}(m, \sigma_x^2)$ with $m > 0$ be given. Let the derived random variable be $\underline{y} = 1/\underline{x}$.
    Using (2.149), p. 44 and (2.117), p. 39, derive a general expression for the odd moments of $\mathbb{E}(\underline{y})$. Show that the series for odd n begins with

$$\mathbb{E}\left(\frac{1}{\underline{x}}\right) = \frac{1}{\mu_x}\left(1 + \frac{\sigma_x^2}{\mu_x^2} + \frac{3\sigma_x^4}{\mu_x^4} + \frac{15\sigma_x^6}{\mu_x^6} + \ldots\right) \tag{2.234}$$

    Show that the series diverges.
37. (1) Given the cumulative distribution $P_x(x)$ of a random variable $\underline{x}$, show that the random variable $P_x^{-1}(\underline{y})$ has density $p_x(x)$ if $\underline{y}$ is uniformly distributed in the interval $[0, 1]$.

Chapter 3
Testing

3.1 Principles of Hypothesis Testing 61
3.2 Testability of an Alternative Hypothesis 65
3.3 Common Tests 69
3.4 Exercises 72

Hypothesis tests are valuable tools to check whether there may be a reason to assume that the used model and the acquired data are inconsistent and, if so – based on user provided hypothesis – to identify possible causes. Hypothesis tests are applied in all phases of the analysis process, especially for outlier detection, for identifying systematic deviations in the model, or just for checking the correctness of the implementation of an estimation procedure with simulated data.

On the other hand, decisions based on hypothesis testing may be incorrect. Then it is of advantage to know how large outliers or systematic model deviations need to be in order to be detectable with a certain probability.

This chapter provides the necessary tools for performing hypothesis tests, evaluates their performance and provides lower bounds for detectable deviations from a given hypothesis. For the most relevant testing tasks we collect the adequate tests. These will be used in order to evaluate estimation results applied to geometric reasoning tasks.

3.1 Principles of Hypothesis Testing

3.1.1 Classical Hypothesis Testing 62
3.1.2 Bayesian Testing 64

The goal of hypothesis testing is to evaluate assertions about the parameters of a distribution based on a sample of that distribution. For example, we would like to evaluate whether the mean of a Gaussian distribution is positive or negative.

There are two basic approaches to formalize this problem. *Classical hypothesis testing*, developed by Neyman and Pearson (1933), aims at disproving a hypothesis w.r.t. an alternative hypothesis, based on the evaluation of what is called a *test statistic*. The principle is to limit the probability of erroneously rejecting the hypothesis and to minimize the probability of erroneously rejecting the alternative hypothesis. *Bayesian hypothesis testing* aims at arriving at some objective measure of preference by comparing the posterior probability of two alternative hypotheses, based on some prior probability of them and the likelihood of the sample w.r.t. each one; thus – in contrast to classical hypothesis testing – handles both hypotheses symmetrically. We discuss both approaches.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_3

3.1.1 Classical Hypothesis Testing

The idea of classical hypothesis testing is to evaluate what is called the null hypothesis H_0 with respect to one or many possibly parametrized alternative hypotheses H_a. The goal of a test is to disprove the null hypothesis H_0 in favour of the alternatives. The outcome of the test can take two forms: either there is a reason to reject H_0 or there is not. There is conceptually no possibility to prove the validity of a hypothesis. In this sense, applying classical hypothesis tests can be seen as a sieve, where unlikely hypotheses are rejected.

hypothesis testing as sieve

As the test is based on a random sample, its outcome will be uncertain. In order to properly design such a test, we must take into account four possible outcomes and be able to evaluate the probability of their occurrence, cf. Table 3.1 and Fig. 3.1:[1]

1. The test may correctly not reject the null hypothesis when it actually holds. The probability
$$S = P(H_0 \text{ not rejected} \mid H_0) \tag{3.1}$$
of this situation should be large. The probability S is called the *significance level* of the test.

significance level

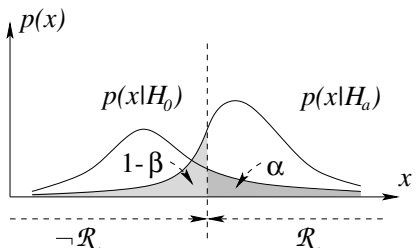


Fig. 3.1 Decisions during hypothesis testing and their probabilities. Rejection region $\mathcal{R}$ and nonrejection region $\neg\mathcal{R}$

2. The test may erroneously reject the null hypothesis when it actually holds. This erroneous decision is called an error of type I. The probability
$$\alpha = P(H_0 \text{ rejected} \mid H_0) = 1 - S\,, \tag{3.2}$$
also called the *significance number*, should be small. Obviously, neither probability depends on the chosen alternative hypotheses.

significance number

3. The test may correctly reject the null hypothesis when actually an alternative hypothesis, H_a, holds. The probability
$$\beta = P(H_0 \text{ rejected} \mid H_a) \tag{3.3}$$
should be large. It is called the *power of the test* w.r.t. the alternative hypothesis and depends on the alternative hypothesis.

power of test

4. The test may erroneously not reject the null hypothesis when an alternative hypothesis H_a actually holds. This erroneous decision is called a decision error of type II. Its probability
$$1 - \beta = P(H_0 \text{ not rejected} \mid H_a) \tag{3.4}$$
should be small.

We cannot expect to simultaneously make the probabilities of errors of types I and II arbitrarily small. But we can design a testing scheme which allows us to control the probabilities of making decision errors.

[1] There is a close but intricate link to the notions true positive, false negative, false positive, and false negative when performing classification: Here the test is designed to reject the null hypothesis, whereas in classification the two alternatives are symmetric w.r.t. each other.

Table 3.1 Probabilities for different decisions when testing hypotheses. The test statistic $\underline{T}$ may lie either in the rejection region $\mathcal{R}$ or in the nonrejection region $\neg\mathcal{R}$. The null hypothesis H_0 or some alternative hypothesis $H_a(t)$ may hold. The alternative hypothesis may be a parametrized set, depending on parameter t – not shown in Fig. 3.1

	H_0 not rejected $T \in \neg\mathcal{R}$	H_0 rejected $T \in \mathcal{R}$
H_0 true	correct decision $S = P(\underline{T} \in \neg\mathcal{R}\|H_0)$	type I error $\alpha = 1 - S = P(\underline{T} \in \mathcal{R}\|H_0)$
$H_a(t)$ true	type II error $1 - \beta(t) = P(\underline{T} \in \neg\mathcal{R}\|H_a(t))$	correct decision $\beta(t) = P(\underline{T} \in \mathcal{R}\|H_a(t))$

The basis for the decision is a test statistic, $T(\boldsymbol{x})$, which is a scalar function of the given sample. The test statistic needs to be *sufficient*, i.e., it has to contain the same information about the underlying distribution as the complete sample (cf. Fisher, 1922, for the concept of sufficiency).

The idea is to specify a *rejection region* $\mathcal{R}$. If the test statistic falls in the rejection region, and thus $T \in \mathcal{R}$, there is a reason to reject the hypothesis H_0 in favour of H_a. Otherwise, if the test statistic is not in the rejection region, and is thus in the *nonrejection region*, $T \in \neg\mathcal{R}$, the hypothesis H_0 is not rejected.[2]

If we know the distribution of the test statistic $\underline{T} \mid H_0$, provided H_0 holds, we can determine the probability α of making a type I error. This is usually no problem as in most practical circumstances we have an idea what the distribution of the sample looks like. On the other hand, if we knew the distribution of the test statistic given the validity of the alternative hypothesis, we could determine the probability $1 - \beta$ of a type II error. This is generally much harder, for two reasons:

1. If the alternative is 'a sample contains outliers', it is not at all clear how to specify the distribution of the sample.
2. The number of alternatives may be large, possibly infinite. For example, if the null hypothesis H_0 is $\mu = \mu_0$, i.e., the mean of the distribution μ is identical to some given value, the number of alternatives is infinite, namely all $\mu = t$ with $t \neq \mu_0$. We could also interpret this alternative hypothesis, $H_a(t)$, which is parametrized by t; this is a classical approach.

Theoretically, the choice of the nonrejection region $\neg\mathcal{R}$ is free, but the effects of this choice on the probability of making errors need to be taken into account. The classical testing scheme is to require that the type I errors have a prespecified probability, say α, and then determine the nonrejection regions $\neg\mathcal{R}$ such that the probability of a type II error given a set of alternatives is minimized.

For example, if the null hypothesis is $H_0 : \mu = \mu_0$, the alternative set is $H_a(t) : \mu = \mu_0 + t$, and we use the sample mean $T = \widehat{\mu}$ as test statistic, then there exists a uniformly most powerful test in the sense that the nonrejection region $\neg\mathcal{R} = \{T|T < \mu_0 + k\sigma_{\widehat{\mu}}\}$ is best for all alternatives. Here k is the critical value which depends on the prespecified probability of a type I error, in this example for $\alpha = 5\%$, leading to $k = 1.6449$.

Practically, the boundary of $\neg\mathcal{R}$ is unsharp, as the knowledge about the distribution of the sample is uncertain. Without modelling this situation in detail, it might be useful to have three regions: a nonrejection region $\neg\mathcal{R}$, where the probability of avoiding a type I error is large, a rejection region $\mathcal{R}$, where the probability of correctly rejecting the null hypothesis is large, and a doubt region $\mathcal{D}$, where it is uncertain whether the null hypothesis should be rejected or not. This leads to what could be called *traffic light decisions*, corresponding to where the test statistic lies: green if $T \in \neg\mathcal{R}$, yellow if $T \in \mathcal{D}$, and red if $T \in \mathcal{R}$ (Fig. 3.2). *traffic light decision*

[2] Often the nonrejection is called "acceptance region", a notion we avoid, since there is no test outcome which may lead to an acceptance of the hypotheses.

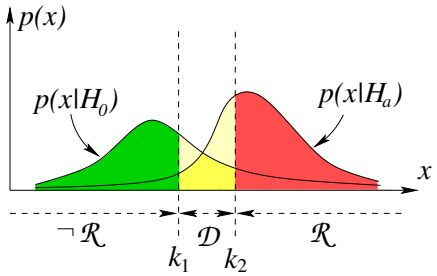


Fig. 3.2 Traffic light decision with two thresholds and three regions: The nonrejection region $\neg\mathcal{R}$, the rejection region $\mathcal{R}$ and the doubt region $\mathcal{D}$. If the test statistic falls into the doubt region, the decision on the rejection of the null hypothesis in favour of the alternative hypothesis is postponed

3.1.2 Bayesian Testing

Bayesian testing is identical to decision making. In contrast to classical hypothesis testing, here the two alternative hypotheses are treated symmetrically. Starting from prespecified prior probabilities for H_1 and H_2, posterior probabilities for the two alternatives H_1 and H_2 given the data $\boldsymbol{x}$ are determined from

$$P(H_1|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|H_1)}{P(\boldsymbol{x})}P(H_1) \quad \text{and} \quad P(H_2|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|H_2)}{P(\boldsymbol{x})}P(H_2)\,. \tag{3.5}$$

In order to eliminate the common factor $P(\boldsymbol{x})$, we can use the probability ratio or the *Bayesian factor*

$$r = \frac{P(H_1|\boldsymbol{x})}{P(H_2|\boldsymbol{x})} = \frac{P(\boldsymbol{x}|H_1)P(H_1)}{P(\boldsymbol{x}|H_2)P(H_2)}\,. \tag{3.6}$$

The output of a Bayesian test is thus probabilities or probability ratios. A decision can be made if both probabilities are different or if $r \neq 1$. That hypothesis is favoured which leads to the higher posterior probability.

As probabilities are estimates, the decision will be uncertain. However, the outcome of the test is given without any specification of errors of type I or II, for which we would need

$$P(\text{erroneously not rejecting } H_1) = P(r > 1|H_2) \tag{3.7}$$

$$P(\text{erroneously not rejecting } H_2) = P(r < 1|H_1)\,, \tag{3.8}$$

which are difficult to evaluate in practice.

The clarity of the testing scheme comes with some conceptual and practical difficulties. In case a hypothesis refers to a specific value, e.g., $H_1 : \mu = \mu_0$, the prior probability of that value, if it is continuous, should be zero. In case both hypotheses refer to a specific point, we could work with the densities; however, if one hypothesis refers to a specific value and the other to a region, specification of priors becomes difficult. No conceptual problems arise if both hypotheses refer to a region. However, practical problems arise since usually no realistic prior probability is known. Take the example of an outlier in the sample modelled as a sample value coming from a Gaussian with the same variance but a different, unknown, biased mean. Then we would need the distribution of the bias values, which can be found empirically only in very special cases.

Assuming the priors $P(H_1)$ and $P(H_2)$ in (3.5)ff. to be uniform, e.g., equal to $1/2$, then just the likelihood functions $L(H_i) = P(\boldsymbol{x}|H_i)$ are taken, and also used in the classical testing scheme. If we cannot really rely on these uniform priors for decision making, the concept of the classical testing scheme needs to be reconsidered.

If priors for the alternatives are available, e.g., from previous experiments, then Bayesian testing is to be favoured, as posterior probabilities can be derived. In our applications,

where we use testing mainly for checking the validity of geometric models or for the detection of outliers, we generally do not have priors for model variations or the existence of outliers. Therefore, in the following, we stick to the classical hypothesis testing scheme.

3.2 Testability of an Alternative Hypothesis

When we perform a statistical test according to the classical testing scheme, the power of the test w.r.t. an alternative hypothesis may not be large enough to be of practical value. Therefore, it is reasonable to identify the boundary between alternative hypotheses with a power lower or higher than a prespecified minimum power, say β_0. Only an alternative hypothesis leading to a power $\beta \geq \beta_0$ can be treated as a testable alternative hypothesis. For instance, the null hypothesis $H_0 : \mu = \mu_0$ is unlikely to be testable w.r.t. the alternative $H_a : \mu = \mu_0 + 0.5$ based on a sample from a Gaussian distribution with standard deviation $\sigma = 1$, since the power will be very low.

A similar situation arises when testing whether the sample vector $\boldsymbol{x}$ of a multi-dimensional random vector $\underline{\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0}, I)$ significantly differs from the mean, where the test statistic $\underline{X}^2 = |\underline{\boldsymbol{x}} - \boldsymbol{\mu}|^2$ follows a χ^2 distribution.

We discuss both cases, where the test statistic follows a normal distribution, and where the test statistic follows a χ^2 distribution.

3.2.1 Testability with Respect to the Mean of a Scalar

When testing the null hypothesis $H_0 : \mu = \mu_0$ w.r.t. the set $H_a(\delta)$ of alternatives,

$$H_a(\delta) : \mu = \mu_0 + \delta\sigma, \quad \delta \in \mathbb{R} \setminus 0, \tag{3.9}$$

based on a single sample x with the probability $\alpha \leq \alpha_0$ for a type I error, we use the test statistic

$$z = \frac{x - \mu_0}{\sigma} \tag{3.10}$$

with $p(z|H_0) = \phi(z)$ ((2.46), p. 30). As δ may be positive or negative, we apply a two-sided test. The nonrejection region depends on the chosen significance number α_0 and is

$$\neg\mathcal{R}(\alpha_0) = [-k(\alpha_0), k(\alpha_0)] \tag{3.11}$$

(Fig. 3.3) with

$$k(\alpha_0) = \Phi^{-1}\left(1 - \frac{\alpha_0}{2}\right) \tag{3.12}$$

and the cumulative standard normal distribution $\Phi(x)$ (cf. (2.47), p. 30).

The power of the test depends on the assumed shift δ and on the significance number α_0. The *power function* $\beta(\delta)$ of the test w.r.t. the alternative hypotheses, depending on δ, is *power function*

$$\begin{aligned}
\beta(\delta) &= P(\underline{z} \in \mathcal{R} \mid H_a(\delta)) && (3.13)\\
&= P(\underline{z} < -k \mid H_a(\delta)) + P(\underline{z} > +k \mid H_a(\delta)) && (3.14)\\
&= \Phi(-k - \delta) + \Phi(\delta - k) && (3.15)\\
&\approx \Phi(\delta - k)\ . && (3.16)
\end{aligned}$$

The function $\Phi(z - \delta)$ is the cumulative probability function of the noncentral standard normal distribution $\phi(z|\delta) = \phi(z - \delta)$ with *noncentrality parameter* δ. The first term refers to the area under $p(\underline{z} \mid H_a(\delta))$ for $z < -k$: If the noncentrality parameter $\delta > k$ this term is very small and can be neglected. *noncentrality parameter*

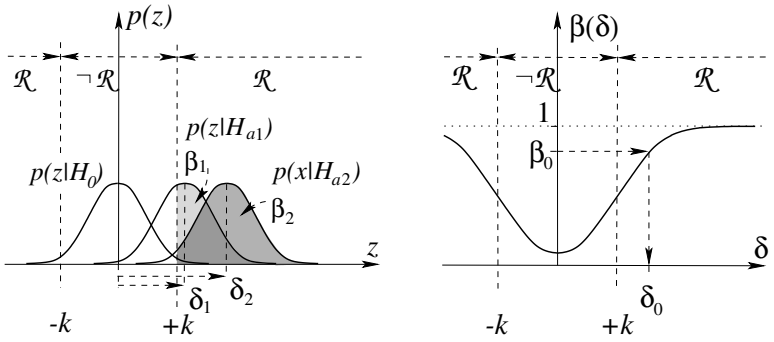


Fig. 3.3 Hypothesis testing and power of test, shown for a test statistic with normalized Gaussian distribution. **Left:** The power β of correctly rejecting an alternative hypothesis depends on the distance δ of the alternative hypothesis from the null hypothesis. **Right:** The power function $\beta(\delta)$. The lower bound δ_0 for the distance δ of the alternative from the null hypothesis, according to Baarda (1967), can be derived from a required lower bound β_0 for the probability β of correctly rejecting the null hypothesis

Following Baarda (1967), we now require that the test be performed with power

$$\beta \geq \beta_0 \,. \tag{3.17}$$

This yields a lower bound δ_0 for the noncentrality parameter δ,

$$\delta \geq \delta_0 =: \nabla_0 z, \tag{3.18}$$

with

$$\delta_0(\alpha_0, \beta_0) \approx \Phi^{-1}\left(1 - \frac{\alpha_0}{2}\right) + \Phi^{-1}(\beta_0) = k(\alpha_0) + \Phi^{-1}(\beta_0) \,. \tag{3.19}$$

The lower bound δ_0 at the same time is a lower bound $\nabla_0 z$ for the test statistic z to be rejected with power β_0. Since

$$\mathbb{E}\left(\underline{z} \mid H_a(\delta)\right) = \mathbb{E}\left(\frac{\underline{x} - \mu_x}{\sigma_x} \mid H_a(\delta)\right) = \delta > \delta_0 \,, \tag{3.20}$$

we find that the minimum deviation $\nabla_0 \mu$ of μ from μ_0 detectable by the test with the required minimum power is

$$\nabla_0 \mu = \delta_0 \sigma_x \,. \tag{3.21}$$

lower bound for detectable deviation

We characterize the detectability of a deviation from H_0 by this *lower bound for a detectable deviation*: only mean values larger than this bound can be detected with a probability larger than the required bound β_0 when using the test with significance number α_0. Thus the *detectability*, besides the statistical parameters of the test procedure, depends on the standard deviation (here of the mean), the form of the test statistic, and the significance level. Alternative hypotheses with $\delta < \delta_0$ are not *testable* in the sense defined above.

testability

Following Baarda (1967), every test should be specified by the minimum probability of a type I error and the minimum required power of avoiding a type II error, and thus by the pair (α_0, β_0). Thus, for a hypothesis against an alternative with a given distance $\Delta\mu$, the experiment needs to be designed such that it leads to a sufficiently small standard deviation of the mean, since δ_0 is fixed by the choice of (α_0, β_0). Table 3.2 contains several values for the minimum bound δ_0 as a function of the significance number α_0 and the minimum power β_0.

Table 3.2 Lower bound for noncentrality parameter δ_0 of a test (from Förstner, 1987). The critical value of the test is $k(\alpha_0)$. Observe, the power β_0 of identifying an alternative hypothesis with $\delta_0 = k$ is 50%. We often assume a value $\delta_0 = 4.13$ which corresponds to a significance number of 0.1% and a minimum power of 80% or to a significance number of 5% and a minimum power of 99%

	α_0	0.01%	0.1%	1%	5%
β_0	k	3.89	3.29	2.58	1.96
50%		3.89	3.29	2.58	1.96
70%		4.41	3.81	3.10	2.48
80%		4.73	**4.13**	3.42	2.80
90%		5.14	4.57	3.86	3.24
95%		5.54	4.94	4.22	3.61
99%		6.22	5.62	4.90	**4.29**
99.9%		6.98	6.38	5.67	5.05

3.2.2 Testability with Respect to the Mean of a Vector

We want to test whether a d-dimensional vector follows a standardized normal distribution $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}, I_d)$ or has a different mean, we have the following null hypothesis and the alternative hypothesis:

$$\underline{\boldsymbol{x}}|H_0 \sim \mathcal{N}(\boldsymbol{\mu}, I_d) \quad \text{and} \quad \underline{\boldsymbol{x}}|H_a \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{t}, I_d)\,. \tag{3.22}$$

Analysing the testability of the alternative hypothesis in this case is a little more complex.

We use the squared distance $X^2 = |\boldsymbol{x} - \boldsymbol{\mu}|^2$ of the sample vector to the hypothesized mean $\boldsymbol{\mu}$ as test statistic.[3] The two alternatives, now referring to the random variable $\underline{X}^2$, then are

$$H_0 :\ \mu = \mu_0 \quad \text{and} \quad H_a(\delta) :\ \mu = \mu_0 + \delta^2\,. \tag{3.23}$$

If the null hypothesis holds, the test statistic $\underline{X}^2$ follows a χ^2_d distribution with d degrees of freedom,

$$\underline{X}^2|H_0 \sim \chi^2_d \tag{3.24}$$

with mean $\mu_0 = d$; otherwise it follows a non-central χ^2_d distribution (Sect. 2.4.5, p. 33),

$$\underline{X}^2|H_a(\delta) \sim \chi'^2_d(\delta^2) \tag{3.25}$$

with mean $d + \delta$. We are only interested in positive deviations δ. Therefore, we perform a one-sided test. The nonrejection region is, see Fig. 3.4

$$\neg\mathcal{R}(\alpha_0) = [0, c(\alpha_0)] \tag{3.26}$$

with the critical value

$$c(\alpha_0, d) = P^{-1}_{\chi^2}(1 - \alpha_0 \mid d) \tag{3.27}$$

using the inverse $P^{-1}_{\chi^2}$ of the cumulative χ^2_d distribution $P_{\chi^2}(x|d)$.

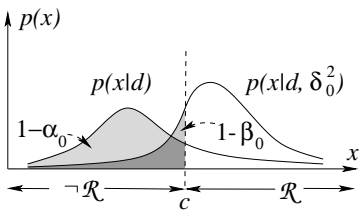


Fig. 3.4 Lower bound δ_0 for the noncentrality parameter of the density function of the test statistic $\underline{X}^2$ (when the alternative hypothesis holds) as a function of the significance number α_0 and the minimal power β_0 of the test

[3] Generally, we name a test statistic X^2, if it follows a χ^2 distribution.

The power β of the test can be derived from the noncentral χ^2 distribution $P_{\chi'^2}(x \mid d, \delta^2)$,

$$1 - \beta(\delta) = P_{\chi'^2}(c|d, \delta^2), \tag{3.28}$$

and depends on the noncentrality parameter δ. Again, requiring the power of the test β to be larger than a prespecified lower bound β_0 leads to a lower bound δ_0 of the noncentrality parameter δ. This lower bound can be derived by solving

$$1 - \beta_0 = P_{\chi'^2}\left(c(\alpha_0, d) \mid d, \delta^2\right) \tag{3.29}$$

for δ, leading to the lower bound for the noncentrality parameter δ,

$$\delta_0(d) = \delta(\alpha_0, \beta_0, d)\,. \tag{3.30}$$

The following three tables 3.3, 3.4, and 3.5 summarize the minimum bounds for the noncentrality parameter δ_0 for some representative cases.

Table 3.3 Lower bound δ_0 for the noncentrality parameter of the χ'^2_d distribution for different significance numbers α_0 and for the lower bound for the power $\beta_0 = 0.8$ of the test

$\alpha_0 \backslash d$	1	2	3	4	8	10	20	50
5%	2.8016	3.1040	3.3019	3.4547	3.8758	4.0300	4.5783	5.4958
1%	3.4175	3.7257	3.9316	4.0926	4.5426	4.7092	5.3068	6.3181
0.1%	4.1321	4.4342	4.6417	4.8063	5.2749	5.4508	6.0887	7.1837

Table 3.4 As for Table 3.3 with for $\beta_0 = 0.9$

$\alpha_0 \backslash d$	1	2	3	4	8	10	20	50
5%	3.2415	3.5572	3.7645	3.9249	4.3684	4.5312	5.1120	6.0884
1%	3.8574	4.1745	4.3872	4.5538	5.0210	5.1945	5.8182	6.8784
0.1%	4.5721	4.8803	5.0926	5.2614	5.7434	5.9247	6.5839	7.7195

Table 3.5 As for Table 3.3 for $\beta_0 = 0.95$

$\alpha_0 \backslash d$	1	2	3	4	8	10	20	50
5%	3.6048	3.9298	4.1437	4.3095	4.7690	4.9382	5.5427	6.5626
1%	4.2207	4.5442	4.7617	4.9325	5.4124	5.5909	6.2342	7.3309
0.1%	4.9354	5.2481	5.4640	5.6360	6.1280	6.3133	6.9885	8.1550

For example, if the test statistic has $d = 3$ degrees of freedom, as when testing the identity of two 3D points, and the test is performed with a significance level of $S = 0.999 = 1 - 0.001$, and thus $\alpha = 0.001$, and if the required minimum power for rejecting the alternative hypothesis is $\beta_0 = 90\%$, then the alternative hypothesis must be characterized by a noncentrality parameter of at least $\delta_0 = 5.05$. This corresponds to the statement: Two 3D points with a covariance matrix $\Sigma = \sigma^2\, I_3$ must have at least a distance of $5.05\,\sqrt{2}\,\sigma = 7.14\,\sigma$ in order to be distinguishable by the test with the required probabiltities. The factor $\sqrt{2}$ results from the fact that both points are assumed to be uncertain, and we actually test the null hypothesis $H_0 : \boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y} = \boldsymbol{0}$.

As the values for the minimum noncentrality parameter do not vary too much with the power β_0 of the test, it is recommended it be fixed for all situations. We will use $\alpha_0 = 0.001$ and $\beta_0 = 0.8$ in all examples.

We will use this line of thought to characterize the ability to identify systematic or gross errors in an estimation procedure.

The concept of detectability uses the probabilities of type I and II errors when testing for a single alternative hypothesis. In case the zero hypothesis is tested w.r.t. two or more

alternatives a type III error may occur, namely if the null hypothesis erroneously is not rejected in favour of the wrong alternative (Förstner, 1983). This yields measures for the *separability of two alternatives*, which might be useful when checking an estimation result for outliers and systematic errors. They may be only weakly separable if the correlation coefficient between the test statistics is high.

separability of alternative hypotheses

3.3 Common Tests

3.3.1 Testing the Mean of a Gaussian Variable 69
3.3.2 Testing the Variance of Gaussian Variables 70

We will now present the most common tests relevant for checking the validity of estimation models. We specify the underlying null and alternative hypothesis and give the sufficient test statistic and its distribution under the null hypothesis. This is sufficient for deriving the nonrejection regions. We will treat the concepts of testability in more detail in the next section on estimation.

3.3.1 Testing the Mean of a Gaussian Variable

Testing the Mean with Given Covariance Matrix. The test whether the mean of a vector-valued Gaussian variable is identical to a given vector is used for outlier detection or for evaluating systematic model deviations. The two hypotheses are

$$H_0 : \underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx}) \qquad H_a : \underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{d}, \Sigma_{xx}) \tag{3.31}$$

for some $\boldsymbol{d} \neq \boldsymbol{0}$. It is assumed that the covariance matrix Σ_{xx} is given. As a test statistic, we use what is called the *Mahalanobis distance* between the sample vector $\boldsymbol{x}$ and the mean, given the null hypothesis $\boldsymbol{\mu}_x|H_0$,[4]

$$\underline{X}^2(\underline{\boldsymbol{x}}) = (\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^\mathsf{T} \Sigma_{xx}^{-1} (\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)\,, \qquad \underline{X}^2|H_0 \sim \chi^2_R\,, \tag{3.32}$$

which is χ^2_R-distributed and where R is the dimension of the random vector $\underline{\boldsymbol{x}}$. If $\boldsymbol{d} \neq \boldsymbol{0}$, the test statistic follows a noncentral χ'^2 distribution $p_{\chi'^2}(x \mid R, \delta^2)$ with noncentrality parameter $\delta^2 = \boldsymbol{d}^\mathsf{T} \Sigma_{xx}^{-1} \boldsymbol{d}$. If the dimension is 1, the test statistic can be reduced to

$$\underline{z}(\underline{x}) = \frac{\underline{x} - \mu_x}{\sigma_x}\,, \qquad \underline{z}|H_0 \sim \mathcal{N}(0,1)\,. \tag{3.33}$$

Generally, we name test statistic z or X^2 if they follow a normal or a χ^2 distribution, respectively.

Testing the Mean with Unknown Variance. The testing of the mean of a Gaussian variable with unknown variance given a sample $\{x_n\}$ of size N is based on the hypotheses

$$H_0 : \underline{x}_n \sim \mathcal{N}(\mu_x, \sigma_x^2 \boldsymbol{I}_N) \text{ for all } n \tag{3.34}$$
$$H_a : \underline{x}_n \sim \mathcal{N}(\mu_x + d, \sigma_x^2 \boldsymbol{I}_N) \text{ for all } n\,. \tag{3.35}$$

We use the estimated mean,

$$\widehat{\mu}_x = \frac{1}{N} \sum_{n=1}^{N} x_n\,, \tag{3.36}$$

[4] The term Mahalanobis distance as used in the literature actually means a squared distance.

and the variance of the estimated mean,

$$\widehat{\sigma}^2_{\widehat{\mu}_x} = \frac{1}{N(N-1)} \sum_{n=1}^{N} (x_n - \widehat{\mu}_x)^2 \,. \tag{3.37}$$

The test statistic is

$$\underline{t} = \frac{\widehat{\underline{\mu}}_x - \mu_x}{\widehat{\underline{\sigma}}_{\widehat{\mu}_x}} \,, \qquad \underline{t}|H_0 \sim t(N-1) \,, \tag{3.38}$$

which under the null hypothesis follows Student's t-distribution with $N-1$ degrees of freedom. Observe, the difference with the previous approach is twofold: The mean is estimated from N values instead from only one value, and the variance of the mean is not given, instead, it is estimated from the sample.

In case of multiple random vectors, the test generalizes to Hotelling's T-test (Hotelling, 1931).

3.3.2 Testing the Variance of Gaussian Variables

independent and identically (i.i.d.) distributed variables

The test of the estimated variance $\widehat{\sigma}^2$ derived from a sample $\underline{\boldsymbol{x}} = [x_n], n = 1, ..., N$, is done for checking the validity of a model. We assume the sample values to be independent and identically distributed (i.i.d.), taken from a normal distribution: $\underline{x}_n \sim \mathcal{N}(\mu_x, \sigma_x^2)$. In all cases the alternative hypothesis states that the mean of the distribution deviates from that of the null hypothesis. There are different versions of this test, depending on whether the mean is given or estimated and whether the variance of the Gaussian is given or estimated.

Test of the Variance for a Given Mean. The most simple test is based on the two hypotheses

$$\underline{\boldsymbol{x}}|H_0 \sim \mathcal{N}(\mu_x, \sigma_x^2) \qquad \underline{\boldsymbol{x}}|H_a \sim \mathcal{N}(\mu_x, (1+\lambda)\sigma_x^2) \tag{3.39}$$

for some $\lambda > 0$. The test statistic is the Mahalanobis distance

$$\underline{X}^2(\underline{\boldsymbol{x}}) := \underline{\Omega} = \frac{\sum_{n=1}^{N} (\underline{x}_n - \mu_x)^2}{\sigma_x^2} \,, \qquad \underline{X}^2|H_0 \sim \chi^2_N \tag{3.40}$$

of the sample and the mean, i.e., the sum of the squared residuals $(x_n - \mu_x)^2$, normalized with the given variance σ_x^2 or the estimated variance of the sample,

$$\widehat{\underline{\sigma}}^2_x = \frac{\underline{\Omega}}{N} \,. \tag{3.41}$$

Then we have the alternative test statistic,

$$\underline{F}(\underline{\boldsymbol{x}}) = \frac{\widehat{\underline{\sigma}}^2_x}{\sigma_x^2} \,, \qquad \underline{F} \mid H_0 \sim F(N, \infty) \,. \tag{3.42}$$

Test of the Variance for an Unknown Mean. If the mean of the sample is unknown, we use the estimated mean

$$\widehat{\underline{\mu}}_x = \frac{1}{N} \sum_{n=1}^{N} \underline{x}_n \sim \mathcal{N}\left(\mu, \frac{\sigma_x^2}{N}\right) \,. \tag{3.43}$$

We then obtain the test statistic, which is the Mahalanobis distance of the sample from the mean,

$$\underline{X}^2(\underline{\boldsymbol{x}}) = \underline{\Omega} = \frac{\sum_{n=1}^{N} (\underline{x}_n - \widehat{\underline{\mu}}_x)^2}{\sigma_x^2} \,, \qquad \underline{X}^2|H_0 \sim \chi^2_{N-1} \,. \tag{3.44}$$

Alternatively, we can use the estimated variance of the observations,

$$\underline{\widehat{\sigma}}_x^2 = \frac{\underline{\Omega}}{N-1}, \tag{3.45}$$

and obtain the test statistic

$$\underline{F}(\underline{\boldsymbol{x}}) = \frac{\underline{\widehat{\sigma}}_x^2}{\sigma_x^2}\,, \qquad \underline{F}|H_0 \sim F(N-1,\infty)\,. \tag{3.46}$$

Test of the Variances of Two Variables with Unknown Means. Given two *independent* samples $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ of Gaussian variables with different and unknown means, we can test whether the variances are identical or different by testing the two alternatives

$$H_0:\ \begin{bmatrix}\underline{\boldsymbol{x}}\\ \underline{\boldsymbol{y}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\sigma_x^2 \boldsymbol{I}_{N_x} & \boldsymbol{0}\\ \boldsymbol{0} & \sigma_x^2 \boldsymbol{I}_{N_y}\end{bmatrix}\right) \tag{3.47}$$

$$H_a:\ \begin{bmatrix}\underline{\boldsymbol{x}}\\ \underline{\boldsymbol{y}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\sigma_x^2 \boldsymbol{I}_{N_x} & \boldsymbol{0}\\ \boldsymbol{0} & \sigma_y^2 \boldsymbol{I}_{N_y}\end{bmatrix}\right) \tag{3.48}$$

with $\sigma_x \neq \sigma_y$. This test is the basis for comparing the results of two independent estimation processes.

Taking the individual estimates of the mean values,

$$\underline{\widehat{\mu}}_x = \frac{1}{N_x}\sum_{n=1}^{N_x} \underline{x}_n \quad \text{and} \quad \underline{\widehat{\mu}}_y = \frac{1}{N_y}\sum_{n=1}^{N_y} \underline{y}_n\,, \tag{3.49}$$

and the two variances,

$$\underline{\widehat{\sigma}}_x^2 = \frac{1}{N_x-1}\sum_{n=1}^{N_x}(\underline{x}_n - \underline{\widehat{\mu}}_x)^2 \quad \text{and} \quad \underline{\widehat{\sigma}}_y^2 = \frac{1}{N_y-1}\sum_{n=1}^{N_y}(\underline{y}_n - \underline{\widehat{\mu}}_y)^2\,, \tag{3.50}$$

the sum of squared residuals are χ^2 distributed under the null hypothesis,

$$\underline{\Omega}_x = (N_x-1)\underline{\widehat{\sigma}}_x^2/\sigma_x^2\,, \qquad \underline{\Omega}_x|H_0 \sim \chi^2_{N_x-1}\,, \tag{3.51}$$

$$\underline{\Omega}_y = (N_y-1)\underline{\widehat{\sigma}}_y^2/\sigma_x^2\,, \qquad \underline{\Omega}_y|H_0 \sim \chi^2_{N_y-1}\,, \tag{3.52}$$

which yields the test statistic for testing the identity of the variances:

$$\underline{F}(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}) = \frac{\underline{\widehat{\sigma}}_x^2}{\underline{\widehat{\sigma}}_y^2} = \frac{\underline{\Omega}_x/(N_x-1)}{\underline{\Omega}_y/(N_y-1))}\,, \qquad \underline{F}|H_0 \sim F(N_x-1, N_y-1)\,. \tag{3.53}$$

Multi-dimensional Test of a Covariance Matrix. Given a set of N sample values $\{\underline{\boldsymbol{x}}_n\}$ of a vector-valued U-dimensional Gaussian distribution, we can estimate the covariance matrix. We want to test whether the sample comes from a distribution with a given covariance matrix $\boldsymbol{\Sigma}_{xx}$, i.e., we want to compare the hypotheses

$$H_0:\ \underline{\boldsymbol{x}}_n \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}) \qquad H_a:\ \underline{\boldsymbol{x}}_n \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx} + \boldsymbol{\Lambda}) \tag{3.54}$$

with some positive semi-definite matrix $\boldsymbol{\Lambda}$. This test is useful for evaluating the validity of a theoretically derived covariance matrix with respect to an empirically determined one.

Starting from the estimated mean vector

$$\widehat{\boldsymbol{\mu}}_x = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n\,, \tag{3.55}$$

we obtain the estimated covariance matrix

$$\widehat{\Sigma}_{xx} = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_x)(\boldsymbol{x}_n - \widehat{\boldsymbol{\mu}}_x)^\mathsf{T}\,. \tag{3.56}$$

The test statistic

$$\underline{X}^2(\{\boldsymbol{x}_n\}) = (N-1)\left[\ln\left(\det\boldsymbol{\Sigma}_{xx}/\det\underline{\widehat{\Sigma}}_{xx}\right) - U + \operatorname{tr}\left(\underline{\widehat{\Sigma}}_{xx}\boldsymbol{\Sigma}_{xx}^{-1}\right)\right] \tag{3.57}$$

is approximately χ^2-distributed under the null hypothesis

$$\underline{X}^2|H_0 \sim \chi^2_{U(U+1)/2} \tag{3.58}$$

with $U(U+1)/2$ degrees of freedom for specifying a $U \times U$ covariance matrix $\boldsymbol{\Sigma}_{xx}$. The derivation uses the fact that under the null hypothesis, the estimated covariance matrix is Wishart-distributed: $\underline{\widehat{\Sigma}}_{xx} \sim \mathcal{W}(N-1, \boldsymbol{\Sigma}_{xx})$ (cf. Koch, 1999, Sects. 2.8.7, 4.1.212).

The results of this section are the basis for all techniques for evaluation the validity of the used mathematical models during estimation. Generally, testing refers to alternative models, for example, in the presence of outliers or systematic errors, which will be described in more detail in Sect. 4.6, and then for evaluation and planning of multi-camera setups for bundle adjustment in Chap. 15. Tests are primarily used to identify outliers (Sect. 4.6.4), but also for checking geometric relations (Sect. 10.4). The basic idea, to invert the power function of a test for arriving at minimal detectable distances between hypotheses, is due to Baarda (1967, 1968). When applied to estimation procedures we first arrive at measures for detectability and testability of outliers and systematic errors. They lead to measures for the sensitivity of estimation results w.r.t. non-detectable and not-testable systematic errors in Sect. 4.6. In all cases we assume observations may occur as single values or as groups, which require multi-dimensional tests. Furthermore statistical tests will be used to check the correctness of estimation procedures based on simulations (Sect. 4.6, p. 115). These tests are applied to vanishing point location (Sect. 10.6.2), homography estimation (Sect. 10.6.3), and for self-calibrating bundle adjustment (Sect. 15.4.1.6, p. 684).

3.4 Exercises

1. (1) A test conceptually cannot prove a hypothesis. Why?
2. (1) Decision errors of type I sometimes are called *producers's risk*, whereas decision errors of type II are called *consumer's risk*. Take the scenario of performing an outlier test before generating the result of some data analysis and giving it to your customer and explain the two notions.
3. (3) Assume you test the null hypothesis $H_0 : \mu = 0$ against the alternative hypothesis $H_a : \mu > 0$. The best nonrejection region is $\neg\mathcal{R} = (-\infty, k(\alpha)]$, where the critical value k depends on the significance number α. Demonstrate with another choice, $\neg\mathcal{R}_1$, for the nonrejection region fulfilling $P(\underline{x} \in \neg\mathcal{R}_1|H_0) = 1-\alpha$, that the power of the test is smaller than when using the best nonrejection region. Proposal: Choose as nonrejection region the nonintuitive region $\neg\mathcal{R}_1 = (-\infty, -k_1] \cup [k_1, \infty)$. Determine k_1 and the power of the test $\beta_1 = P(\underline{x} \in \mathcal{R}_1|H_a)$.
4. (2) If we have a model for the distribution of the alternative hypothesis, we can use Bayesian testing. Choose your cheapest favourite measuring device which fails in certain situations. How would you derive a reasonable statistical model for failure situations, captured in an alternative hypothesis?
5. (1) Plot the power function of the optimal one-sided test, namely for the alternative $H_1 : \mu < 0$ on the mean of a normally distributed random variable.
6. (1) Confirm the value $\delta_0(\alpha_0, \beta_0)$ in Table 3.2, p. 67 for $\alpha_0 = 0.001$ and $\beta_0 = 0.9$. Explain the meaning of the value $\delta_0 = 4.57$ in simple words.

7. (1) Confirm the value $\delta_0(\alpha_0, \beta_0, d) = 5.09$ in Table 3.4, p. 68 for $\alpha_0 = 0.001$, $\beta_0 = 0.9$, and $d = 3$.
8. (3) For each of the tests in Sect. 2.3:
   a. Specify the null hypothesis, e.g., $H_0 : \mu = 1$.
   b. Generate a sufficiently large sample fulfilling H_0.
   c. Perform the generation repeatedly, i.e., S times, and test the hypothesis. Use $\alpha = 0.001$.
   d. Estimate the probability $\widehat{\alpha}$ of making a type I error. Compare it to α. The difference should not be significant, taking the sample size S into account.
   e. Disturb the sample by some outliers and repeat the last two steps. Discuss the outcome of the experiment.
9. (2) Given are two hypotheses: $H_0 : p(x|H_0) = \mathcal{N}(4, 9)$ and $H_a(\mu) : p(x|H_a) = \mathcal{N}(\mu, 2)$.
   a. Plot the two densities.
   b. Assume you perform a Bayesian test of H_0 versus $H_a(\mu = 7)$.
      i. Give the nonrejection region for H_0. What prior probability have you used?
      ii. What are the probabilities $P(H_0|H_a)$ and $P(H_a|H_0)$ of making a decision error of type I and type II? Answer with a complete sentence.
      iii. Now assume that H_0 has a doubly higher probability than H_a. What is the nonrejection region for a Bayesian test, again with $\mu = 7$? What are the probabilities of making a decision error of kind I and kind II?
      iv. Plot the power function $P(H_a|H_a(\mu))$ in the range $\mu \in [-6, 24]$.
      v. How large must μ be that the alternative hypothesis will be accepted with at least $\beta_0 = 80\%$.
   c. Assume you perform a classical test. The significance level is specified to be $S = 0.95$.
      i. What is the nonrejection region for a one-sided and a two-sided test?
      ii. What is the probability of correctly deciding for H_a if $\mu = 7$?
      iii. Plot the power function $P(H_a|H_a(\mu)$ for the test in the range $\mu \in [-6, 24]$.
      iv. How large must μ be so that the alternative hypothesis will be accepted with at least $\beta_0 = 80\%$.
10. (1) Assume you want to perform a traffic light decision for a one-sided test.
    a. Give the two critical values k_1 and k_2 such that the decision errors of type I are smaller than 1% if the standard deviation σ_1 in the null hypothesis $H_0 : p(x|H_0) = \mathcal{N}(4, \sigma_1^2)$ is uncertain and can be assumed to be uncertain up to a factor of 1.5.
    b. What is the range for the power of a test against the alternative hypothesis $H_a(\mu) : p(x|H_a) = \mathcal{N}(15, 2)$.
11. (2) Alternative hypotheses only can be identified with a high probability if the precision of the underlying measurement is high enough.
    a. Assume you want to be able to separate two measurements x and y with a probability larger than 90% by performing a statistical test based on their difference $d = y - x$. In what range must the standard deviation of both measurements $\sigma_x = \sigma_y$ lie if the test is performed with a significance level of $S = 0.99$?
    b. Assume somebody tells you that two 3D points at a distance of 8 cm from each other are significantly different, based on a test with a significance level of 99%.
       i. What can you conclude about the precision of the 3D point coordinates? What do you assume about the standard deviations of the coordinates and their correlations?
       ii. How does the answer change qualitatively if the two points have different precision and if the coordinates are correlated?

Chapter 4
Estimation

4.1 Estimation Theory 75
4.2 The Linear Gauss–Markov Model 81
4.3 Gauss–Markov Model with Constraints 99
4.4 The Nonlinear Gauss–Markov Model 102
4.5 Datum or Gauge Definitions and Transformations 108
4.6 Evaluation 115
4.7 Robust Estimation and Outlier Detection 141
4.8 Estimation with Implicit Functional Models 160
4.9 Methods for Closed Form Estimations 176
4.10 Estimation in Autoregressive Models 183
4.11 Exercises 185

This chapter assembles the necessary tools for performing parameter estimation from redundant measurements in the context of geometric computation within photogrammetric computer vision. The main tool is weighted least squares estimation as a special case of Bayesian estimation. We discuss the Gauss–Markov model with its variations, including the handling of crisp constraints, the estimation of variance components, and robust estimation, and generalize it to what is called the Gauss–Helmert model. Special emphasis is placed on tools for evaluating the results of parameter estimation. This covers all aspects of testing and selecting hypotheses on model deviations, namely gross and systematic deviations, the detectability of such model deviations, the sensitivity of the result with respect to non-detectable model deviations, and the acceptability of the resultant estimate with its precision, following the principal ideas of Baarda (1967, 1968, 1973).

We do not cover models with inequality constraints or models which contain discrete variables, which are treated in the second volume of this book.

4.1 Estimation Theory

4.1.1 Bayesian Estimators 76
4.1.2 Maximum Likelihood Estimators 78
4.1.3 Best Unbiased Estimators 79
4.1.4 Least Squares Estimators 79
4.1.5 Comparison of the Optimization Principles 80

Estimation theory frames the methodology for estimating unknown parameters from given observations. It starts from a *mathematical model* consisting of the functional model and the stochastical model of the observation process.

1. The *functional model* specifies the assumed relations between the observations and the unknown parameters. The set of observations can be interpreted as a sample from a set

functional model

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_4

of random variables. The functional model relates the true or the mean values of the random variables associated with the observations to the random variables associated with the parameters. If the parameters are taken as fixed values we interpret them as random variables with zero variance. As the sample values will generally not fulfil the functional model, they need to be corrected or fitted to the model, such that the corrected or fitted observational values satisfy the functional model.

When prior information about the unknown parameters is available, we can interpret it as additional observations and treat these in the same manner as the usual observations. Priors about the observations, which will be less accurate, simply can be treated as additional observations.

stochastical model

2. The *stochastical model* specifies the statistical properties of the observation process, i.e., the random variables associated with the observations and possibly of the prior information about the unknown parameters. This specification may not be complete. The stochastical model itself may contain unknown parameters which also have to be estimated from the given observations.

   Since the distribution of the stochastic variables is often unknown, it may be preferable not to specify the complete distribution, but only certain of its properties, such as the second moments, and thus variances and covariances. Even just relative weighting of the observations may be a reasonable choice. The special type of knowledge about the uncertainty of the observations has to be taken into account during estimation.

The separation of the mathematical model into the functional and the stochastical models is useful, as the functional model usually results from physical or geometrical laws which have to be fulfilled. Thus the functional model can be as complex as the physical or geometrical reality and the application requires. The stochastical model is usually much simpler, at least in the context of modelling physical and geometrical problems, even in the case of outliers or systematic errors. The situation is therefore somewhat different from problems in image analysis, where the stochastical model often is more complex than the functional model.

We will often refer to the uncertainty of observations when addressing the random variables associated with them.

In general, mathematical models are meant to support the solution of a practical problem. Therefore, they need to be as simple as possible, but not simpler (Einstein), in order to balance closeness to reality and efficiency of problem solving. They are never true, but they may or may not be *adequate* for solving a certain task, similarly to the process of choosing an optical model in physics, which – depending on the problem – may be from geometric optics, wave optics or particle optics.

adequacy of mathematical model

Based on a chosen mathematical model, parameter estimation may follow different principles. It is advisable to exploit all available information. In this case, we arrive at what is called *sufficient* statistic, which are superior to estimates which do not exploit the full information. As an example, the arithmetic mean is a sufficient statistic for the mean of the distribution, whereas the median, using only one value, is not.

Based on the available information, we can distinguish between different estimation principles, which are discussed in the following sections.

4.1.1 Bayesian Estimators

Bayesian estimation treats the unknown parameters, collected in a U-vector $\underline{\boldsymbol{x}}$, as stochastic variables for which there is some prior knowledge represented as an *a priori probability density*,

prior probability

$$\underline{\boldsymbol{x}} \sim p(\boldsymbol{x})\,. \tag{4.1}$$

Equation (4.1) allows us to differentiate between the knowledge for individual parameters x_u. Lack of knowledge can be modelled by a very broad density function $p(\boldsymbol{x})$. Observe,

we used the simplified notation $p(\boldsymbol{x})$ for the probability density function $p_x(\boldsymbol{x})$, as the naming index is identical to the argument.

The unknown parameters are connected to observations, which are collected in an N-vector $\boldsymbol{l}$. Generally this connection is represented by the conditional density

$$\underline{\boldsymbol{l}}|\boldsymbol{x} \sim p(\boldsymbol{l}|\boldsymbol{x})\,. \tag{4.2}$$

It states how probable it is to observe the vector $\boldsymbol{l}$ when the parameter vector $\boldsymbol{x}$ is given. The *likelihood function*

likelihood function

$$L(\boldsymbol{x}) = p(\boldsymbol{l}|\boldsymbol{x}) \tag{4.3}$$

is a function of $\boldsymbol{x}$ and states how likely the unknown parameters $\boldsymbol{x}$ are when the observed values $\boldsymbol{l}$ are given by some observation process.

Later we will allow the connection between the observed vales $\boldsymbol{l}$ and the parameters $\boldsymbol{x}$ to be expressed as a set of G constraints $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \boldsymbol{0}$, which in certain cases, important within geometric computation, will lead to algebraically simpler expressions.

Using the law of conditional probability, cf. (2.6), p. 23, we first obtain $p(\boldsymbol{x}, \boldsymbol{l}) = p(\boldsymbol{x}|\boldsymbol{l})\, p(\boldsymbol{l}) = p(\boldsymbol{l}|\boldsymbol{x})\, p(\boldsymbol{x})$ (cf. Fig. 4.1), and thus

$$p(\boldsymbol{x}|\boldsymbol{l}) = \frac{p(\boldsymbol{l}|\boldsymbol{x})\, p(\boldsymbol{x})}{p(\boldsymbol{l})}\,. \tag{4.4}$$

The probability density $p(\boldsymbol{x}|\boldsymbol{l})$ is the *a posteriori density* of the parameter vector $\boldsymbol{x}$ for an observed $\boldsymbol{l}$. Obviously, the denominator $p(\boldsymbol{l})$ does not depend on $\boldsymbol{x}$, thus we only need the a priori density $p(\boldsymbol{x})$ and the likelihood function $p(\boldsymbol{l}|\boldsymbol{x})$ in order to derive the a posteriori probability $p(\boldsymbol{x}|\boldsymbol{l})$ using the total probability $p(\boldsymbol{l}) = \int p(\boldsymbol{l}|\boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$ for normalization. As a result, we obtain the a posteriori probability density which can be used for evaluation and testing.

a posteriori density

Maximizing this a posteriori density leads to the *Bayesian estimate*

Bayesian estimate, maximum a posteriori estimate (MAP)

$$\widehat{\boldsymbol{x}}_B = \operatorname{argmax}_x\, p(\boldsymbol{x}|\boldsymbol{l}) = \operatorname{argmax}_x\, p(\boldsymbol{l}|\boldsymbol{x})\, p(\boldsymbol{x}) \tag{4.5}$$

of the parameter vector $\boldsymbol{x}$, for which the normalization with $p(\boldsymbol{l})$ is not necessary. Finding the global optimum generally requires searching the complete parameter space.

Remark: Some authors (cf. Li, 2000; Vaseghi, 2000) distinguish Bayesian estimation from maximum a posteriori estimation. Bayesian estimation, apart from a probabilistic model, uses cost $c(\widehat{\boldsymbol{x}}, \boldsymbol{x})$ as a function of the estimated and the unknown parameters and minimizes the expected costs and determines $\operatorname{argmax}_x \int_x c(\widehat{\boldsymbol{x}}, \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{l}) \mathrm{d}\boldsymbol{x}$ in this way. Following this convention, the estimate (4.5) is the maximum a posteriori (MAP) estimate, as seemingly no costs are taken into account, but only probabilities. However, maximum a posteriori estimation results from Bayesian estimation by using the cost function $1 - \delta(\widehat{\boldsymbol{x}} - \boldsymbol{x})$. We will not make this distinction, as in our context costs are difficult to specify. ◇

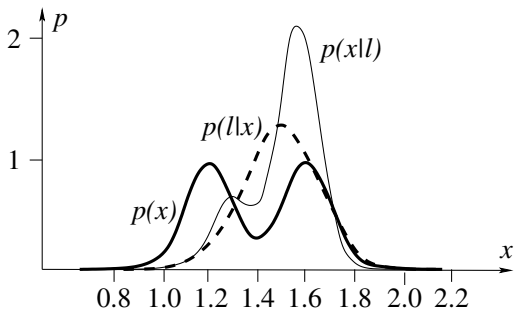


Fig. 4.1 Example for Bayesian estimation. The a priori density $p(x)$ (thick solid curve) indicates that it is not sure whether x is approximately 1.2 or 1.6. The likelihood function $p(l|x)$ (dashed curve) results from a poor measurement around 1.5. The a posteriori density $p(x|l)$ (thin solid curve), obtained with (4.4), indicates the most likely x to be $\widehat{x}_B \approx 1.57$

4.1.2 Maximum Likelihood Estimators

Maximum likelihood (ML) estimation treats $\boldsymbol{x}$ as a fixed but unknown vector. This is equivalent to the assumption that the a priori density $p(\boldsymbol{x})$ is a constant. Maximizing the likelihood function $p(\boldsymbol{l}|\boldsymbol{x})$ leads to what is called the *maximum likelihood estimate*

$$\widehat{\boldsymbol{x}}_{\mathrm{ML}} = \operatorname{argmax}_x p(\boldsymbol{l}|\boldsymbol{x}) \tag{4.6}$$

for $\boldsymbol{x}$. ML estimation is therefore a special case of Bayesian estimation (cf. Fig. 4.2).

The two principles lead to identical results if no a priori information is used. The result of the ML estimation can be evaluated using the negative logarithm of the likelihood function in the vicinity of $\widehat{\boldsymbol{x}}$, e.g., by analysing the Hessian matrix $\mathsf{H} = [\partial^2(-\log p(\boldsymbol{l}|\boldsymbol{x})/(\partial x_i \partial x_j)]$ containing the second derivatives of the negative log-likelihood function, $-\log p(\boldsymbol{l}|\boldsymbol{x})$, evaluated at $\widehat{\boldsymbol{x}}_{\mathrm{ML}}$. In the case of normally distributed observations $\underline{\boldsymbol{l}}$ and linear relations between the parameters and the means of the observations, we get

$$\underline{\boldsymbol{l}} \sim \mathcal{N}(\mathsf{A}\boldsymbol{x} + \boldsymbol{a}, \Sigma_{ll})\,. \tag{4.7}$$

The ML estimate is also normally distributed and thus can be tested easily.

Maximum likelihood estimation can be used to perform Bayesian estimation. We need two types of observations, namely $\underline{\boldsymbol{l}}$ for the likelihood term and specially chosen *fictitious observations* $\underline{\boldsymbol{y}}$ for the prior term:

fictitious observations

$$\underline{\boldsymbol{l}}|\boldsymbol{x} \sim p(\underline{\boldsymbol{l}} = \boldsymbol{l}|\boldsymbol{x})\,, \tag{4.8}$$

$$\underline{\boldsymbol{y}}|\boldsymbol{x} \sim p(\underline{\boldsymbol{y}} = \boldsymbol{y}|\boldsymbol{x})\,. \tag{4.9}$$

We assume the stochastic variables $\underline{\boldsymbol{l}}|\boldsymbol{x}$ and $\underline{\boldsymbol{y}}|\boldsymbol{x}$ to be conditionally independent given the values $\boldsymbol{x}$. We thus obtain the likelihood function

$$p(\boldsymbol{y}, \boldsymbol{l}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{l}|\boldsymbol{x})\,. \tag{4.10}$$

The maxima of the likelihood function $p(\boldsymbol{y}, \boldsymbol{l}|\boldsymbol{x})$ in (4.10) and the posterior probabilitiy $p(\boldsymbol{x}|\boldsymbol{l})$ in (4.4) are identical if we choose

$$p(\boldsymbol{x}) = k\, p(\boldsymbol{y}|\boldsymbol{x}) \tag{4.11}$$

with some arbitrary constant k. This can generally be achieved if the prior $p(\boldsymbol{x})$ depends on some parameter $\boldsymbol{y}$; thus, we have $p(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ and therefore the equality

$$p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x}) = f(\boldsymbol{x} - \boldsymbol{y})\,. \tag{4.12}$$

For normally distributed variables this can be achieved by choosing

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{y}, \Sigma_{xx}) \quad \text{and} \quad p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{x}, \Sigma_{xx})\,. \tag{4.13}$$

The equivalence of the estimates using (4.4) and (4.10) can be interpreted as follows: The prior $p(\boldsymbol{x})$ states that the random variable $\underline{\boldsymbol{y}}$ has mean $\boldsymbol{y}$. From $p(\boldsymbol{y}|\boldsymbol{x})$ we conclude that a sample value $\boldsymbol{y}$ of the stochastic variable $\underline{\boldsymbol{y}}$ can be interpreted as a direct observation of $\boldsymbol{x}$, similarly to how the sample value $\boldsymbol{l}$ of $\underline{\boldsymbol{l}}$ is an observation of $\underline{\boldsymbol{l}}|\boldsymbol{x}$. Therefore we can use some fixed value $\boldsymbol{y}$ that is a fictitious observation (a sample of $\underline{\boldsymbol{y}}$ with uncertainty Σ_{xx}) of the mean of $\underline{\boldsymbol{y}}$ together with the observational value $\boldsymbol{l}$ for performing maximum likelihood type estimation. But, due to (4.13), this observational value $\boldsymbol{y}$ needs to be the mean of $\underline{\boldsymbol{y}}$. The maximum likelihood estimation with the two relations in (4.8) and (4.9) leads to the same result as Bayesian estimation with prior density $p(\boldsymbol{x})$ in (4.13) (cf. Bishop, 2006, Sect. 2.3.3).

This method of introducing prior information by fictitious observations can be generalized when we have uncertain information not about the parameters themselves, but

about some functions $\boldsymbol{b}(\boldsymbol{x})$ of the parameters. Then the model for the maximum likelihood estimation reads

$$\underline{\boldsymbol{l}}|\boldsymbol{x} \sim p(\underline{\boldsymbol{l}} = \boldsymbol{l}|\boldsymbol{x})\,, \tag{4.14}$$

$$\underline{\boldsymbol{b}}|\boldsymbol{x} \sim p(\underline{\boldsymbol{b}} = \boldsymbol{b}|\boldsymbol{x}) \tag{4.15}$$

with $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{x})$. This shows that uncertain prior information can be treated in the same way as actual observations.

4.1.3 Best Unbiased Estimators

If only the first and the second moments of the likelihood function are known, we have the model

$$\underline{\boldsymbol{l}} \sim \mathcal{M}(\mathbb{E}(\underline{\boldsymbol{l}}), \mathbb{D}(\underline{\boldsymbol{l}}))\,, \tag{4.16}$$

since the second moments and the second central moments are related by the first moment, cf. (2.94), p. 37. Then we might require the estimate $\widehat{\boldsymbol{x}} = \boldsymbol{s}(\boldsymbol{l})$ for $\boldsymbol{x}$, a function $\boldsymbol{s}$ of the observations $\boldsymbol{l}$, to have no *bias* and have *minimal variance*, i.e., be *best*. The bias of an estimate is the difference between the expected value of the estimate and the *true value*,

true value and bias

$$\boldsymbol{b} = \mathbb{E}(\underline{\widehat{\boldsymbol{x}}}) - \tilde{\boldsymbol{x}}\,. \tag{4.17}$$

The true value may be known in simulation studies, or it may be defined by some observation process with significantly superior accuracy. An estimate is *unbiased* if the expected value of the estimated value is identical to the true value,

$$\mathbb{E}(\boldsymbol{s}(\underline{\boldsymbol{l}})) = \tilde{\boldsymbol{x}}\,. \tag{4.18}$$

An estimate has *minimum variance* if the trace $\mathrm{tr}(\mathbb{D}(\boldsymbol{s}(\underline{\boldsymbol{l}})))$ of the covariance matrix of the estimated parameters is minimal. We then define the *best unbiased estimate* as

$$\widehat{\boldsymbol{x}}_{\mathrm{BUE}} = \mathrm{argmin}_s\, \mathrm{tr}(\mathbb{D}(\boldsymbol{s}(\underline{\boldsymbol{l}})))\,, \tag{4.19}$$

taking the restriction (4.18) on the functions $\boldsymbol{s}$ into account. If the function $\boldsymbol{s}(\boldsymbol{l})$ is linear, the estimate is called the *best linear unbiased estimate (BLUE)*.

best linear unbiased estimate

Consequently, if the model (4.7) holds, the expectations of the observations are linear functions of the parameters, at the same time the ML estimate for the parameters $\boldsymbol{x}$ is the best linear unbiased estimate (cf. Koch, 1999, Sect. 3.2.4). This also holds approximately for nonlinear relations between the mean values of the observations and the unknown parameters if the variances of the observations are small compared to the second derivatives of the functions, and if approximate values are available and sufficiently close to the global optimum such that it can be reached in an iterative scheme.

When only the first two moments of the likelihood function are specified, no information about the distribution of $\widehat{\boldsymbol{x}}$ is available; thus, testing of the result is not possible. However, the best unbiased estimator for $\boldsymbol{x}$ can be evaluated by its covariance matrix. In the case of normally distributed observations, the covariance matrix is identical to the Hessian matrix of the log-likelihood function.

4.1.4 Least Squares Estimators

Given N observed values l_n and weights w_n, $n = 1, ..., N$, the *weighted least squares estimate*

weighted LS estimation

$$\widehat{\boldsymbol{x}}_{LS} = \text{argmin}_x \sum_{n=1}^{N} w_n (f_n(\boldsymbol{x}) - l_n)^2 \tag{4.20}$$

minimizes the weighted sum of the squares of corrections or residuals, $f_n(\widehat{\boldsymbol{x}}) - l_n$, where $\tilde{l}_n = f_n(\tilde{\boldsymbol{x}})$ provides the relation between the true observations $\tilde{l}_n$ and the true unknown parameters $\tilde{\boldsymbol{x}}$. Individual observations can be excluded from the estimation process simply by choosing their weights to be zero.

ordinary LS estimation

If all weights are $w_n = 1$, we obtain the *ordinary least squares estimate* in its simplest form. Using the symmetric positive semi-definite diagonal $N \times N$ weight matrix $\mathsf{W} = \text{Diag}([w_n])$, we may write (4.20) as

$$\widehat{\boldsymbol{x}}_{LS} = \text{argmin}_x (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l})^\mathsf{T} \mathsf{W} (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l}) \,. \tag{4.21}$$

general weighted LS estimation

If the weight matrix W is not diagonal, we call it a *general weighted least squares estimate.*

In the following, we will refer to least squares estimates (LS) if their optimization function is written in the form (4.21).

4.1.5 Comparison of the Optimization Principles

The four optimization principles discussed so far are collected in Fig. 4.2. They build a specialization hierarchy.

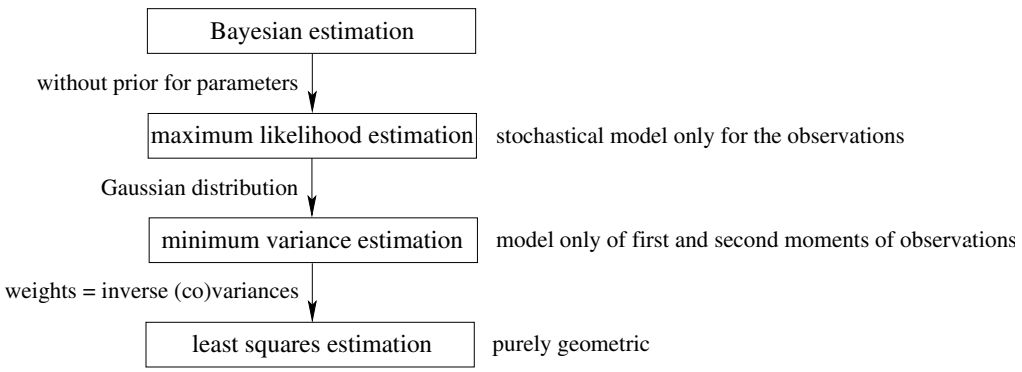


Fig. 4.2 Hierarchy of estimation principles, from the most general optimization principle, the Bayesian estimation, to the most specific estimation principle of least squares estimation. Each specialization step may be seen as imposing constraints on the previous estimation principle: Bayesian estimation reduced to ML-estimation if no priors for the parameters are available. ML-estimation reduces to minimum variance estimation if the observations follow a Gaussian distribution. Minimum variance estimation reduces to weighted least squares if the weights are chosen to be the inverse variances. Thus, least squares estimation can be interpreted purely geometrically. Minimum variance estimation does just rely on the first two moments of the distribution of the observations. Maximum likelihood estimation completely specifies the stochastical model, but only for the observation process

The least squares estimate is identical to the BLUE if the functional relation $l_n = f_n(\boldsymbol{x})$ is linear, the observations are statistically independent, and the weights are chosen to be $w_n = 1/\sigma_{l_n}^2$. No information about the distribution is available, in particular no covariance matrix, and no testing is possible.

We may argue backwards (cf. Fig. 4.2):

1. LS estimates can be interpreted as best linear unbiased estimates with variances $\sigma_{l_n}^2 = 1/w_n$, or more generally, if the weight matrix is symmetric and positive definite, as best linear unbiased estimates with the covariance matrix

$$\Sigma_{ll} = W^{-1} . \tag{4.22}$$

This relation can be used to interpret the chosen weights in weighted LS estimation. No assumption about the underlying distribution is made.

2. Best linear unbiased estimates can be interpreted as ML estimates with normally distributed observations which have the assumed covariance matrix. This can be motivated by the maximum entropy principle, which implies that a distribution for which only the mean and the covariance are given must be a Gaussian distribution (cf. Cover and Thomas, 1991, Sect. 11).
   This enables the use of statistical tests of the estimates.
3. Finally, ML estimates can be interpreted as Bayesian estimates with a constant or very broad a priori density for the unknown parameters. More specifically, in the case of a normally distributed a priori and likelihood function, Bayesian estimation leads to the weighted mean of the ML estimate and the prior.

The reverse argumentation from the bottom to the top of Fig. 4.2 is therefore useful for statistically interpreting least squares estimates or best linear unbiased estimates, for checking their plausibility, e.g., the chosen weights, and for performing perform statistical testing.

In the following Sect. 4.2 we introduce the estimation procedure using the *Gauss–Markov model*. We discuss various modifications, specializations, and generalizations of this model, which can then be transferred to other models.

In Sect. 4.6, p. 115, we present various tools for the *evaluation* of the results of an estimation. In particular, we discuss diagnostic tools for the detection of model deviations, such as outliers or systematic model errors, and the sensitivity of the result w.r.t. undetectable model deviations, as well as tools for evaluating the fulfilment of accuracy requirements.

The presence of outliers gives rise to what is called *robust estimators*, whose principal representatives are given in Sect. 4.7, p. 141. These include efficient methods to handle large percentages of large outliers, as well as methods which can handle a moderate percentage of small outliers in models with a large number of unknowns.

The last Sect. 4.8 discusses estimation models, which are based on constraints between the observations and the unknown parameters, and therefore do not allow the use of the Gauss–Markov model. These models are derived from what is called the *Gauss–Helmert model with constraints between the unknowns*. We present the estimation procedure and a robust estimation algorithm. A synopsis collects their main properties and their estimation procedures.

We refer you to the classical textbooks of Rao (1973), Mikhail and Ackermann (1976), and Koch (1999). Parts of this section follow McGlone et al. (2004, Sect. 2.2).

4.2 The Linear Gauss–Markov Model

4.2.1 The Mathematical Model 82
4.2.2 Estimation and Covariance Matrices 84
4.2.3 Gauss–Markov Model with Unknown Variance Factor 89
4.2.4 Estimation of Variance Components 91
4.2.5 Bayesian Estimation in the Gauss–Markov Model 93
4.2.6 Partitioning and Reduction of the Parameters 94
4.2.7 Groups of Observations and Sequential Estimation 96
4.2.8 Covariance Intersection 98

Regression models (Galton, 1890) are used in many applied sciences. They have the form of a Gauss–Markov model: observations are described as functions of unknown pa-

rameters allowing additive noise: $\boldsymbol{l} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{n}$. We take the Gauss–Markov model as a starting point as its derivation is straightforward. Many modifications, specializations, and generalizations refer to it as a base model.

We start with the *linear* Gauss–Markov model, as it leads to a one-step and unique solution and will be the core of iterative solutions of the nonlinear counterpart.

Given are the N observations $\boldsymbol{l} = [l_n], n = 1, ..., N$, from which the U unknown parameters $\boldsymbol{x} = [x_u], u = 1, ..., U$ are to be determined, with generally $U \leq N$.

4.2.1 The Mathematical Model

4.2.1.1 The Functional Model

The functional model, which relates the true values $\tilde{\boldsymbol{l}}$ of the observations to the true values $\tilde{\boldsymbol{x}}$ of the unknowns, reads as

$$\tilde{\boldsymbol{l}} = \underset{N\times U}{\mathsf{A}}\,\tilde{\boldsymbol{x}} + \boldsymbol{a}\,. \tag{4.23}$$

design matrix

The $N \times U$ design matrix

$$\mathsf{A} = \left[\boldsymbol{a}_n^{\mathsf{T}}\right] \tag{4.24}$$

with rows $\boldsymbol{a}_n^{\mathsf{T}}$ and the additive vector $\boldsymbol{a} = [a_n]$ are given. The notion *design matrix* results form the property that its entries numerically reflect the – presumably planned, i.e., designed – observation process. At the moment we assume the design matrix to have full rank,

$$\mathrm{rk}\mathsf{A} = \min(N, U)\,. \tag{4.25}$$

Some textbooks on estimation theory omit the constant vector $\boldsymbol{a}$ in the model by integrating it into the observation vector, replacing $\boldsymbol{l} - \boldsymbol{a}$ by $\boldsymbol{l}$. The model (4.23) allows us to use the observed values without modification. Eq. (4.23) contains, what is also called the *observation equations* of the linear Gauss–Markov model.

The number N of observations may differ from the number U of the unknown parameters. The structure of the solution depends on their difference, called the *redundancy* of the problem, which in the context of this discussion is

redundancy

$$R = N - U\,. \tag{4.26}$$

We distinguish between three cases.

1. $R < 0$: The problem is *underconstrained* if we have fewer observations than unknowns ($N < U$), and thus negative redundancy. Then we obtain a space of unknown parameters, not necessarily a single parameter or a set of parameters.
   The solution here is not unique; therefore, the problem is *ill-posed* in the sense of Hadamard (cf. Kabanikhin, 2008, Def. 4.1). According to Hadamard, a problem is well-posed if the following three conditions hold: a solution exists, the solution is unique, and the solution depends continuously on the given data. Otherwise the problem is called ill-posed and requires additional constraints to be fulfilled, a process which is called *regularization*. As an example, we can arrive at a unique solution by introducing *additional constraints*, e.g., the norm $|\boldsymbol{x}|$ of the parameters to be minimal. Generally the model cannot be evaluated.

   regularization
2. $R > 0$: The problem is *overconstrained* if we have fewer unknowns than observations ($U < N$), and thus positive redundancy. Due to deviations of the observed values $\boldsymbol{l}$ from the true ones $\tilde{\boldsymbol{l}}$ there is no solution in general. Only if we introduce additive *residuals* or *corrections* $\tilde{\boldsymbol{v}}$ to the observations $\boldsymbol{l}$ into the functional model might we achieve consistency, which is necessary for obtaining solutions:

   residuals, corrections

$$\tilde{\boldsymbol{l}} = \boldsymbol{l} + \tilde{\boldsymbol{v}} = \mathsf{A}\tilde{\boldsymbol{x}} + \boldsymbol{a}\,. \tag{4.27}$$

As these corrections $\tilde{\boldsymbol{v}}$ are unknown, we obtain an underconstrained problem which is ill-posed. A unique solution can be obtained by requiring the norm $|\boldsymbol{v}|$ of the corrections to be minimal. The residuals indicate differences between the model and the given observations which allows us to evaluate the model and the observations.
Sometimes it is useful (cf. Draper and Smith, 1998), to introduce the concept of observational deviations or *observational errors* $\tilde{\boldsymbol{e}}$, and thus express the observations $\boldsymbol{l}$ as *observational errors*

$$\boldsymbol{l} = \tilde{\boldsymbol{l}} + \tilde{\boldsymbol{e}} = \mathsf{A}\tilde{\boldsymbol{x}} + \boldsymbol{a} + \tilde{\boldsymbol{e}}\,. \tag{4.28}$$

This is a *generative model* in the sense that the model allows us to simulate (generate) observations if the model parameters, namely $(\mathsf{A}, \tilde{\boldsymbol{x}}, \boldsymbol{a})$ and the characteristics of $\tilde{\boldsymbol{e}}$, are given. Obviously, the errors are the negative corrections $\tilde{\boldsymbol{e}} = -\tilde{\boldsymbol{v}}$. *generative model*

3. $R = 0$: The minimal number of observations required to obtain a solution is $N = U$, leading to a redundancy $R = 0$. Only in this case can we expect to obtain a solution without introducing additional constraints. Generally the problem has a limited number of solutions, which can be found directly using algebraic methods. The residuals will generally be zero. There is no way to check the validity of the model.

If the rank constraint (4.25) does not hold, the solution may not be unique even in the case of positive redundancy, e.g., when fitting a plane through four or more collinear points. The design matrix then will be singular and additional independent constraints are necessary in order to arrive at unique estimates for the parameters.

Under specific conditions, the quality of the result can be predicted in all three cases. The second case is the most important one, as we can exploit the redundancy for evaluating the validity of the assumed model, e.g., for finding outliers.

4.2.1.2 The Stochastical Model

We assume the given observed values $\boldsymbol{l}$ to result from the true values $\tilde{\boldsymbol{l}}$ by additive deviations $\boldsymbol{e}$. We assume these deviations to be of random nature. The uncertain observations $\underline{\boldsymbol{l}}$ are therefore modelled by some arbitrary distribution $\mathcal{M}$ with first and second moments,

$$\underline{\boldsymbol{l}} = \tilde{\boldsymbol{l}} + \underline{\boldsymbol{e}}\,, \quad \underline{\boldsymbol{e}} \sim \mathcal{M}(\boldsymbol{0}, \Sigma_{ll})\,. \tag{4.29}$$

For the observations, we thus have the unknown residuals and the unknown observational errors

$$\Sigma_{ll} = \mathbb{D}(\underline{\boldsymbol{l}}) = \mathbb{D}(\underline{\boldsymbol{v}}) = \mathbb{D}(\underline{\boldsymbol{e}})\,. \tag{4.30}$$

We assume that the covariance matrix has full rank,

$$\mathrm{rk}(\Sigma_{ll}) = N\,, \tag{4.31}$$

which covers the case where all observations are statistically independent and have nonzero variance $\sigma_{l_n}^2 > 0$. We will regularly use the weight or *precision matrix* (Bishop, 2006) *precision matrix* *weight matrix*

$$\mathsf{W}_{ll} = \Sigma_{ll}^{-1}\,. \tag{4.32}$$

If the observations are mutually independent, the covariance matrix Σ_{ll} is diagonal, and we have

$$\Sigma_{ll} = \mathrm{Diag}\left(\left[\sigma_{l_n}^2\right]\right)\,, \qquad \mathsf{W}_{ll} = \mathrm{Diag}\left([w_{l_n}]\right)\,, \qquad w_{l_n} = \frac{1}{\sigma_{l_n}^2}\,. \tag{4.33}$$

Thus the weights w_{l_n} are the inverse variances. Weights are large if the precision is high.

Remark: Care has to be taken in distinguishing between fixed, i.e., certain, values, and uncertain values. Conceptually, observed values l are seen as a sample of the underlying stochastic variable $\underline{l}$. As such, observational values are fixed, certain values, and not uncertain. If we say an observation is uncertain, we take a view on the underlying experiment the sample is taken from, and refer to the uncertainty,

i.e., non-predictability, of the outcome of that experiment. As an example, take $l_1 = 4$ pixel as one of the observed coordinates of a point at position $\boldsymbol{x} = [l_1, l_2]^\mathsf{T}$ in a digital image. Of course, running the algorithm twice does give the same coordinate, unless the algorithm has a random component. If we refer to the coordinate $l_1 = 4$ pixel as an uncertain one with a standard deviation of, say, $\sigma_{l_1} = 0.3$ pixel, we have in mind an experiment, where we assume the image has been taken several times with slightly different illumination conditions, leading to slightly different positions due to variable sensor noise, light fluctuations or even some small local deformations of vegetation, knowing that there is no chance and also no need to explicitly model these variations. ◇

4.2.2 Estimation and Covariance Matrices

4.2.2.1 Estimation

The complete model, including the functional and the stochastical model, can compactly be written as

$$\underline{\boldsymbol{l}} + \underline{\boldsymbol{v}} = \mathsf{A}\tilde{\boldsymbol{x}} + \boldsymbol{a} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{l}}) = \mathbb{D}(\underline{\boldsymbol{v}}) = \Sigma_{ll}\,. \tag{4.34}$$

The task is to find estimates $\widehat{\boldsymbol{x}}$ for the true parameters $\tilde{\boldsymbol{x}}$ from given observations $\boldsymbol{l}$.

weighted sum of the squared residuals

As both the parameters and the residuals are unknown, the problem is underconstrained. We regularize it by requiring the *weighted sum of the squared residuals*,

$$\Omega(\boldsymbol{x}) = \boldsymbol{v}^\mathsf{T}(\boldsymbol{x})\,\Sigma_{ll}^{-1}\,\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}^\mathsf{T}(\boldsymbol{x})\,\mathsf{W}_{ll}\,\boldsymbol{v}(\boldsymbol{x}) \tag{4.35}$$

of the residuals

$$\boldsymbol{v}(\boldsymbol{x}) = \mathsf{A}\boldsymbol{x} + \boldsymbol{a} - \boldsymbol{l}\,, \tag{4.36}$$

to be minimized:

$$\widehat{\boldsymbol{x}} = \operatorname{argmin}_x \Omega(\boldsymbol{x})\,. \tag{4.37}$$

We can identify the weighted sum of the squares of the residuals Ω as the sum of the squared *Mahalanobis distances* of the residuals to the zero vector, cf. (3.32), p. 69.

A necessary condition for the estimate $\widehat{\boldsymbol{x}}$ is that the partial derivative is zero:

$$\frac{1}{2}\,\frac{\partial\Omega(\boldsymbol{x})}{\partial\boldsymbol{x}^\mathsf{T}}\bigg|_{x=\widehat{x}} = \mathsf{A}^\mathsf{T}\mathsf{W}_{ll}(\mathsf{A}\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}) = \mathsf{A}^\mathsf{T}\,\mathsf{W}_{ll}\,\widehat{\boldsymbol{v}}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}\,. \tag{4.38}$$

We use the derivative of a scalar function $\Omega(\boldsymbol{x})$ w.r.t. the vector $\boldsymbol{x}$, which is a column vector $\boldsymbol{b} = \partial\Omega/\partial\boldsymbol{x}^\mathsf{T}$ with elements $b_u = \mathrm{d}\Omega/\mathrm{d}x_u$, in order to be able to write the total derivative $\Delta\Omega = \Delta\boldsymbol{x}^\mathsf{T}\boldsymbol{b} = \boldsymbol{b}^\mathsf{T}\Delta\boldsymbol{x}$ (cf. Fackler, 2005) using the Jacobian formulation (cf. Wikipedia, 2015, Layout conventions).

normal equation system

The unknown parameters $\widehat{\boldsymbol{x}}$ appear linear in this equation. Therefore, we obtain the estimated parameters as the solution of what is called the *normal equation system*,

$$\mathsf{N}\widehat{\boldsymbol{x}} = \boldsymbol{n} \tag{4.39}$$

with the normal equation matrix and the right-hand side

$$\mathsf{N} = \mathsf{A}^\mathsf{T}\mathsf{W}_{ll}\mathsf{A}\,, \qquad \boldsymbol{n} = \mathsf{A}^\mathsf{T}\mathsf{W}_{ll}(\boldsymbol{l} - \boldsymbol{a})\,, \tag{4.40}$$

Explicitly we have the estimated parameters and the estimated residuals

$$\widehat{\boldsymbol{x}} = \left(\mathsf{A}^\mathsf{T}\mathsf{W}_{ll}\mathsf{A}\right)^{-1}\,\mathsf{A}^\mathsf{T}\mathsf{W}_{ll}(\boldsymbol{l} - \boldsymbol{a}) \qquad \widehat{\boldsymbol{v}} = \mathsf{A}\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}\,. \tag{4.41}$$

Practically the unknown parameters are derived by solving the equation system without determining the inverse of the normal equation matrix. This can be done efficiently using Cholesky decomposition; see the explanation after Theorem 5.2.4 in Golub and van Loan

(1996) and Sect. A.9, p. 776. When the observations are mutually independent, i.e., when Σ_{ll} is a diagonal matrix, the normal equation matrix N and the vector $\boldsymbol{n}$ can be written as

$$N = \sum_{n=1}^{N} w_n \boldsymbol{a}_n \boldsymbol{a}_n^\mathsf{T}, \qquad \boldsymbol{n} = \sum_{n=1}^{N} w_n (l_n - a_n)\boldsymbol{a}_n . \tag{4.42}$$

This allows us to build up the normal equations incrementally, e.g., by initiating and sequentially adding *normal equation components* $\{w_n \boldsymbol{a}_n \boldsymbol{a}_n^\mathsf{T};\, w_n(l_n - a_n)\boldsymbol{a}_n\}$,

$$N := 0 \qquad \boldsymbol{n} := \boldsymbol{0} \tag{4.43}$$

$$N := N + w_n \boldsymbol{a}_n \boldsymbol{a}_n^\mathsf{T}, \qquad \boldsymbol{n} := \boldsymbol{n} + w_n (l_n - a_n)\boldsymbol{a}_n . \tag{4.44}$$

Therefore, the design matrix does not necessarily need to be stored. When using (4.34), it can easily be proven that the estimate of the unknown parameter vector is *unbiased*, i.e., its expectation is identical to the true value, and the expectation of the estimated residuals $\widehat{\boldsymbol{v}} = A\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}$ is zero:

unbiasedness of $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{v}}$

$$\mathbb{E}(\underline{\widehat{\boldsymbol{x}}}) = \tilde{\boldsymbol{x}}, \qquad \mathbb{E}(\underline{\widehat{\boldsymbol{v}}}) = \boldsymbol{0} . \tag{4.45}$$

The optimization function (4.35) can be interpreted in two important ways, geometrically and probabilistically:

1. *Geometrically.* The mean $A\boldsymbol{x} + \boldsymbol{a}$ of the distribution $\mathcal{M}$ can be written as

$$\mathbb{E}(\underline{\boldsymbol{l}}) = \sum_u \boldsymbol{a}_u x_u + \boldsymbol{a}, \tag{4.46}$$

which spans an affine U-dimensional subspace in $\mathbb{R}^N$ with base vectors $\boldsymbol{a}_u$ and coordinates x_u. The observed vector $\boldsymbol{l}$ generally does not lie on this subspace. The task is to find the particular point $\widehat{\boldsymbol{x}}$ in the subspace where the distance $d(\boldsymbol{l}, A\boldsymbol{x}+\boldsymbol{a}) = |\boldsymbol{v}|_{W_{ll}}$ is smallest when taking the metric (represented by the weight matrix) into account

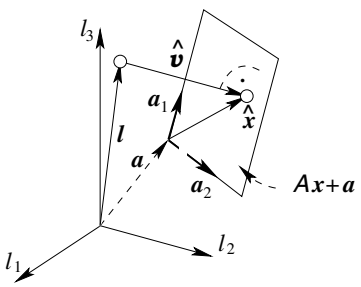


Fig. 4.3 Geometry of least squares estimation shown for $U = 2$, $N = 3$, and $W_{ll} = I_3$

(cf. Fig. 4.3). Then the correction vector $\widehat{\boldsymbol{v}} = A\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}$ is normal to the subspace, motivating the term *normal equations* and expressed by the last relation in (4.38), which can be written as the orthogonality constraint:

$$\langle A, \widehat{\boldsymbol{v}} \rangle_{W_{ll}} = \boldsymbol{0} . \tag{4.47}$$

2. *Probabilistically.* Minimizing (4.35) is equivalent to the maximum likelihood estimation of the parameters when $\mathcal{M}$ is a normal distribution. Then the likelihood function is

$$p(\boldsymbol{l}|\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_{ll}|}} \exp\left(-\frac{1}{2}(A\boldsymbol{x} + \boldsymbol{a} - \boldsymbol{l})^\mathsf{T} \Sigma_{ll}^{-1} (A\boldsymbol{x} + \boldsymbol{a} - \boldsymbol{l})\right), \tag{4.48}$$

and the negative logarithm is identical to (4.35), except for a constant factor and an additive constant.

sparse matrices

Remark: In many practical applications, the percentage of nonzero elements in the matrices A, W_{ll} and consequently also in N is very low, i.e., they are *sparse matrices*, as single observations or small groups of observations only refer to a few unknown parameters and are statistically independent.

As the inverse of a sparse matrix is generally full, it is advantageous to exploit the sparseness and use special techniques for solving the equation system, so that the computation of the inverse can be avoided (cf. Golub and van Loan, 1996), e.g., using the Cholesky decomposition of the normal equation matrix.

Moreover, it is possible to determine the elements of the inverse of the normal equation matrix where there are nonzero elements in the normal equation matrix without necessarily calculating all other elements of the inverse, which is helpful when evaluating the result.[1]

We will regularly identify steps in the estimation or evaluation where sparseness may be exploited and discuss the corresponding methods in Sect. 15.3, p. 651. ◇

4.2.2.2 Covariance Matrices of the Estimates

estimated parameters

The covariance matrix of the *estimated parameters* can easily be obtained by variance propagation ((2.136), p. 42) applied to (4.41), as the estimates $\widehat{\underline{\boldsymbol{x}}}$ are linear functions of the observations $\underline{\boldsymbol{l}}$. It is given by

$$\Sigma_{\widehat{x}\widehat{x}} = \mathsf{N}^{-1} = (\mathsf{A}^{\mathsf{T}}\mathsf{W}_{ll}\mathsf{A})^{-1}\,. \tag{4.49}$$

Therefore the precision matrix of the estimated parameters is identical to the normal equation matrix (4.40),

$$\mathsf{W}_{\widehat{x}\widehat{x}} = \mathsf{N}\,. \tag{4.50}$$

Fisher information matrix

Cramer–Rao bound

The normal equation matrix is also called the *Fisher information matrix* (Rao, 1973). The covariance matrix at the same time is what is called the *Cramer–Rao bound*: Any estimator $\widehat{x}_T$ which is unbiased, i.e., for which we have $\mathbb{E}(\widehat{\underline{\boldsymbol{x}}}_T) = \tilde{\boldsymbol{x}}$, has a covariance matrix

$$\Sigma_{\widehat{x}_T\widehat{x}_T} \geq \Sigma_{\widehat{x}\widehat{x}}\,, \tag{4.51}$$

the inequality denoting that the difference matrix $\Sigma_{\widehat{x}_T\widehat{x}_T} - \Sigma_{\widehat{x}\widehat{x}}$ is positive definite. This indicates that the estimate in (4.41) is the best in the sense that it leads to the least uncertainty among all unbiased estimators. Furthermore, the Cramer–Rao bound can be taken as the reference covariance matrix when evaluating the implementation of an estimation by simulation (cf. Sect. 4.6.8.2, p. 140).

estimated observations

The covariance matrix of the estimated or *fitted observations*

$$\widehat{\boldsymbol{l}} = \mathsf{A}\widehat{\boldsymbol{x}} + \boldsymbol{a} \tag{4.52}$$

can be derived from (4.41) by variance propagation and is given by the rank U matrix

$$\Sigma_{\widehat{l}\widehat{l}} = \mathsf{A}\Sigma_{\widehat{x}\widehat{x}}\mathsf{A}^{\mathsf{T}}\,. \tag{4.53}$$

The matrix

$$\mathsf{U} = \Sigma_{\widehat{l}\widehat{l}}\Sigma_{ll}^{-1} \tag{4.54}$$

allows us to write the fitted observations as explicit functions of the observations

$$\widehat{\boldsymbol{l}} - \boldsymbol{a} = \mathsf{U}(\boldsymbol{l} - \boldsymbol{a})\,, \tag{4.55}$$

hat matrix

and in statistics literature is called the *hat matrix*, denoted by H, as it puts the hat onto $\boldsymbol{l}$. With the nth column $\boldsymbol{u}_n$ of U, Eq. (4.55) allows us to analyse the effect of a change Δl_n in the nth observation on the fitted values $\widehat{\boldsymbol{l}}$ or onto a single fitted observation,

$$\Delta\widehat{\boldsymbol{l}} = \boldsymbol{u}_n\,\Delta l_n\,, \qquad \Delta\widehat{l}_m = U_{mn}\,\Delta l_n\,, \tag{4.56}$$

[1] Takahashi et al. (1973, cf. MATLAB-code `sparseinv.m`), and Vanhatalo and Vehtari (2008)

relations which can be used for diagnostic purposes (Sect. 4.6).

The matrix $\mathbf{U}$ is idempotent, thus $\mathbf{U}^2 = \mathbf{U}$. Therefore its rank equals its trace, and we have

$$\mathbf{U} = \sum_{n=1}^{N} u_n = \operatorname{tr}(\mathbf{U}) , \qquad \text{with} \qquad u_n = U_{nn} . \tag{4.57}$$

A look at the diagonal elements of $\mathbf{U}$ in (4.55) and the second relation in (4.56) shows that the N elements u_n indicate how a change Δl_n in the observation l_n influences the corresponding individual fitted value $\widehat{l}_n$. The variances of the fitted observations are

$$\sigma_{\widehat{l}_n}^2 = \boldsymbol{a}_n^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}} \boldsymbol{a}_n . \tag{4.58}$$

Remark: In case the vector $\boldsymbol{a}_n$ is sparse, i.e., it only contains a few nonzero elements, the calculation requires just a few elements of the covariance matrix. If the variances of the fitted observations are of interest, e.g., when evaluating the design (cf. Sect. 4.6.2, p. 117), then only a small percentage of the elements of $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ needs to be determined. This advantage can be exploited in estimation problems with a large number of unknowns, e.g., in a bundle adjustment, discussed in Sect. 15.3, p. 651. ◇

The covariance matrix of the *estimated residuals* is given by *estimated residuals*

$$\boldsymbol{\Sigma}_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll} - \boldsymbol{\Sigma}_{\widehat{l}\widehat{l}} = \boldsymbol{\Sigma}_{ll} - \mathbf{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\mathbf{A}^{\mathsf{T}} . \tag{4.59}$$

What is called the *redundancy matrix*, *redundancy matrix*

$$\mathbf{R} = \boldsymbol{\Sigma}_{\widehat{v}\widehat{v}} \boldsymbol{\Sigma}_{ll}^{-1} \tag{4.60}$$

is also idempotent with rank R, since $\mathbf{R} = \mathbf{I}_N - \mathbf{U}$, and if a matrix $\mathbf{B}$, is idempotent $\mathbf{I} - \mathbf{B}$ is also idempotent. Therefore, the matrix $\boldsymbol{\Sigma}_{\widehat{v}\widehat{v}}$ also has rank $R \leq N$; so we have

$$R = \sum_{n=1}^{N} r_n = \operatorname{tr}(\mathbf{R}) , \qquad \text{with} \qquad r_n = 1 - u_n = R_{nn} . \tag{4.61}$$

The redundancy matrix $\mathbf{R}$ allows us to write the estimated residuals as explicit functions of the observations,

$$\widehat{\boldsymbol{v}} = -\mathbf{R}(\boldsymbol{l} - \boldsymbol{a}) . \tag{4.62}$$

With the nth column $\boldsymbol{r}_n$ of $\mathbf{R}$ (4.62) makes it possible to analyse the effect of a change Δl_n in the nth observation on the residuals $\widehat{\boldsymbol{v}}$ or on a single residual $\widehat{v}_m$,

$$\Delta\widehat{\boldsymbol{v}} = \boldsymbol{r}_n \, \Delta l_n , \qquad \Delta\widehat{v}_m = R_{mn} \, \Delta l_n , \tag{4.63}$$

which also can be used for diagnostic purposes (Sect. 4.6). The variances of the estimated residuals are

$$\sigma_{\widehat{v}_n}^2 = \sigma_{l_n}^2 - \sigma_{\widehat{l}_n}^2 . \tag{4.64}$$

Again, looking at (4.63), the diagonal elements r_n of the redundancy matrix $\mathbf{R}$ indicate how a change in an observation influences the corresponding residual.

Comparing (4.54), (4.57), (4.60), and (4.61), we arrive at the symmetric relations (Förstner, 1987)

- for the idempotent matrices,
$$\mathbf{I}_N = \mathbf{U} + \mathbf{R} , \tag{4.65}$$
- for the individual contributions of each observation,
$$1 = u_n + r_n = U_{nn} + R_{nn} , \tag{4.66}$$
- for the number of observations, unknowns, and redundancy,

$$N = U + R = \mathrm{tr}U + \mathrm{tr}R = \sum_n u_n + \sum_n r_n \,, \tag{4.67}$$

- and for the covariances and variances,

$$\Sigma_{ll} = \Sigma_{\widehat{l}\widehat{l}} + \Sigma_{\widehat{v}\widehat{v}} \quad \text{and} \quad \sigma_{l_n}^2 = \sigma_{\widehat{l}_n}^2 + \sigma_{\widehat{v}_n}^2 \,. \tag{4.68}$$

  As we will see in (4.73), p. 88, the estimated parameters, and hence also the estimated observations, and the residuals are stochastically independent. Therefore, this equation shows how the uncertainty of the observations is split between the fitted observations and the residuals.

We therefore define the following two entities, which will be crucial for the evaluation.

Definition 4.2.1: *Redundancy numbers* r_n. The contribution of a single observation from a set of statistically independent observations to the total redundancy R is given by the redundancy number (Förstner, 1979)

$$r_n = R_{nn} = (\Sigma_{\widehat{v}\widehat{v}} W_{ll})_{nn} = \left(I_N - A(A^\mathsf{T} W_{ll} A)^{-1} A^\mathsf{T} W_{ll} \right)_{nn} \in [0, 1] \,. \tag{4.69}$$

◇

Exercise 4.14

If the observations are not stochastically independent, the redundancy numbers need not be in the range $[0, 1]$.

Due to (4.67) the average redundancy number is

$$\overline{r_n} = \frac{\sum_{n=1}^N r_n}{N} = \frac{R}{N} \,, \tag{4.70}$$

which we will use if we do not have access to the individual values r_n.

Consequently, we have the contribution of a single observation from a set of statistically independent observations to the number U of unknown parameters,

$$u_n = U_{nn} = \left(\Sigma_{\widehat{l}\widehat{l}} W_{ll}\right)_{nn} = \left(A(A^\mathsf{T} W_{ll} A)^{-1} A^\mathsf{T} W_{ll} \right)_{nn} \in [0, 1] \,. \tag{4.71}$$

Together with (4.56) and (4.62), we therefore have the two relations – again for uncorrelated observations

$$u_n = \frac{\sigma_{\widehat{l}_n}^2}{\sigma_{l_n}^2} = \frac{\widehat{\Delta l}_n}{\Delta l_n} \,, \qquad r_n = \frac{\sigma_{\widehat{v}_n}^2}{\sigma_{l_n}^2} = -\frac{\widehat{\Delta v}_n}{\Delta l_n} \,. \tag{4.72}$$

The numbers $u_n = 1 - r_n$ and the redundancy numbers r_n give insight into the increase of the precision, $\sigma_{\widehat{l}_n}^2 / \sigma_{l_n}^2$, of the observations and into the expected size, $\sigma_{\widehat{v}_n}^2$, of the squared residuals compared to the variances, $\sigma_{l_n}^2$, of the given observations. At the same time, they indicate which part of a change Δl_n affects the corresponding fitted observation and the estimated residual. We will use the last relation in (4.72) to derive an estimate for the size of an outlier from the estimated residual during testing (cf. Sect. 4.6.4.1, p. 124).

Finally, the *covariance of the unknown parameters and the residuals* can be derived from (4.41), p. 84 and (4.62) by variance propagation of the concatenated vector $\boldsymbol{z}^\mathsf{T} = [\widehat{\boldsymbol{x}}^\mathsf{T}, \widehat{\boldsymbol{v}}^\mathsf{T}]$. It is the zero matrix

Exercise 4.9

$$\Sigma_{\widehat{x}\widehat{v}} = \mathbb{E}((\underline{\widehat{\boldsymbol{x}}} - \mathbb{E}(\underline{\widehat{\boldsymbol{x}}}))\, \underline{\widehat{\boldsymbol{v}}}^\mathsf{T}) = \boldsymbol{0} \,. \tag{4.73}$$

unknowns and residuals are uncorrelated

Therefore the fitted observations are also statistically independent of the estimated residuals. The proof uses the orthogonality relations (4.38), p. 84 and (4.47), p. 85,

$$A^\mathsf{T} W_{ll} \widehat{\boldsymbol{v}} = \langle A, \widehat{\boldsymbol{v}} \rangle_{W_{ll}} = \boldsymbol{0} \,, \tag{4.74}$$

indicating that the estimated residuals are orthogonal to the columns of A when taking the metric W_{ll} into account. We will use the independence relation when testing within sequential estimation in Sect. 4.2.7.3, p. 98. The relation (4.74) can serve as a numerical check of the implementation.

4.2.3 Gauss–Markov Model with Unknown Variance Factor

We now generalize the Gauss–Markov model and assume we only know the ratios of the variances and the correlations of the observations, i.e., we do not know the scaling of the covariance matrix of the observations. The covariance matrix Σ_{ll} of the observations is given as the product of some approximate initial covariance matrix Σ_{ll}^a and an initial variance factor σ_0^2,

$$\Sigma_{ll} = \sigma_0^2 \Sigma_{ll}^a \quad \text{or} \quad W_{ll} = \frac{1}{\sigma_0^2} W_{ll}^a \,, \tag{4.75}$$

including the corresponding expression for the weight matrix of the observations. This initial variance factor should be chosen to be $\sigma_0 = 1$, though it could be chosen arbitrarily. This assumption about the stochastical model is equivalent to fixing the structure of the initial weight matrix

$$W_{ll}^a = (\Sigma_{ll}^a)^{-1} \,. \tag{4.76}$$

The mathematical model can then be expressed in the following manner, cf. (4.34):

$$\underline{\boldsymbol{l}} \sim \mathcal{M}(A\tilde{\boldsymbol{x}} + \boldsymbol{a}, \sigma_0^2 \Sigma_{ll}^a) \,. \tag{4.77}$$

The estimation result (4.41) obviously does not depend on the variance factor σ_0^2 but only on the approximate covariance matrix Σ_{ll}^a.

Remark: In classical textbooks on adjustment theory, the initial covariance matrix Σ_{ll}^a is often called the *weight coefficient matrix*, denoted by Q_{ll}, possibly derived from a weight matrix W_{ll} by $Q_{ll} = W_{ll}^{-1}$. We avoid weight coefficient matrices in the following, as they may result from situations where the variance factor is not chosen to be 1; thus, the weights are not defined as $w_{l_n} = 1/\sigma_{l_n}^2$. ◇

We now derive an estimate for the variance factor. The expected value of the Mahalanobis distance can be shown to be

$$\mathbb{E}(\underline{\Omega}) = \mathbb{E}\left(\underline{\widehat{\boldsymbol{v}}}^\mathsf{T} \, \sigma_0^{-2} W_{ll}^a \, \underline{\widehat{\boldsymbol{v}}}\right) = R \,. \tag{4.78}$$

This equation holds since

$$\mathbb{E}(\underline{\widehat{\boldsymbol{v}}}^\mathsf{T} W_{ll} \underline{\widehat{\boldsymbol{v}}}) = \mathbb{E}(\mathrm{tr}(\underline{\widehat{\boldsymbol{v}}}^\mathsf{T} W_{ll} \underline{\widehat{\boldsymbol{v}}})) = \mathbb{E}(\mathrm{tr}(W_{ll} \underline{\widehat{\boldsymbol{v}}} \, \underline{\widehat{\boldsymbol{v}}}^\mathsf{T})) = \mathrm{tr}(W_{ll} \mathbb{E}(\underline{\widehat{\boldsymbol{v}}} \, \underline{\widehat{\boldsymbol{v}}}^\mathsf{T})) = \mathrm{tr}(W_{ll} \Sigma_{\widehat{v}\widehat{v}}) \,. \tag{4.79}$$

In the second step, we use the relation $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, in the third step the linearity of the expectation operator, and in the last step $\mathbb{E}(\underline{\widehat{\boldsymbol{v}}}) = \boldsymbol{0}$. Finally, we use the fact that the matrix $W_{ll}\Sigma_{\widehat{v}\widehat{v}}$ is idempotent; thus, its trace is equal to its rank R, cf. (4.60).

We obtain an unbiased *estimate for the variance factor*, again using the initial weight matrix W_{ll}^a of the observations, *estimated variance factor*

$$\widehat{\sigma}_0^2 = \frac{\widehat{\boldsymbol{v}}^\mathsf{T} W_{ll}^a \widehat{\boldsymbol{v}}}{R} \,. \tag{4.80}$$

The Mahalanobis distance can also be determined from (cf. (4.40), p. 84),

$$\Omega = \widehat{\boldsymbol{v}}^\mathsf{T} W_{ll}^a \widehat{\boldsymbol{v}} = (\boldsymbol{l} - \boldsymbol{a})^\mathsf{T} W_{ll}^a (\boldsymbol{l} - \boldsymbol{a}) - \boldsymbol{n}^\mathsf{T} \widehat{\boldsymbol{x}} \,, \tag{4.81}$$

which is useful as a numerical check. Exercise 4.8

Remark: A maximum likelihood estimate of the variance factor would lead to $\widehat{\sigma}_0^2 = \widehat{\boldsymbol{v}}^\mathsf{T} W_{ll}^a \widehat{\boldsymbol{v}}/N$, which – due to the denominator N instead of R – is a *biased* estimate for the variance factor. ◇

The variance of the estimated variance $\underline{\widehat{\sigma}_0}^2$ can be given if we specify the distribution $\mathcal{M}$ up to fourth-order moments. If it has the same first four moments as the corresponding normal distribution, it is given by

$$\mathbb{D}(\underline{\widehat{\sigma}_0}^2) = \frac{2}{R} \sigma_0^4 \tag{4.82}$$

relative precision of $\widehat{\sigma_0}$

(cf. Koch, 1999, Eq. (3.295)), from which we obtain the relative precision of the estimated factor $\widehat{\sigma}_0$,

$$\frac{\sigma_{\widehat{\sigma}_0}}{\widehat{\sigma}_0} = \sqrt{\frac{1}{2R}}\,. \tag{4.83}$$

As the relative precision for the medium values of is still small (e.g., for $R = 32$ we obtain $\sigma_{\widehat{\sigma}_0}/\sigma_0 \approx 0.125$, i.e., $\sigma_{\widehat{\sigma}_0}$ is only accurate to 12.5%), we should use the estimate $\widehat{\sigma}_0$ only when the redundancy is large enough.

Using the estimated variance factor, we can derive the *estimated covariance matrix* of the observations:

$$\widehat{\Sigma}_{ll} = \widehat{\sigma}_0^2 \Sigma_{ll}^a\,. \tag{4.84}$$

Observe that the term *estimated* only refers to the variance factor, as the internal structure of the covariance matrix is not changed. Thus the estimated variance factor $\widehat{\sigma}_0^2$ tells by which factor the variances $\sigma_{l_n}^2$ of the observations need to be multiplied in order to arrive at an unbiased estimate $\widehat{\sigma}_{l_n}^2$ of the variances of the observations. Equivalently, we have, for the standard deviations,

$$\widehat{\sigma}_{l_n} = \widehat{\sigma}_0\ \sigma_{l_n}^a\,. \tag{4.85}$$

Therefore, it is very useful to report the factor $\widehat{\sigma}_0$ and the redundancy R.

We can test the result of the estimation statistically. If the mathematical model H_0,

$$\underline{\boldsymbol{l}}|H_0 \sim \mathcal{N}(\mathsf{A}\tilde{\boldsymbol{x}} + \boldsymbol{a}, \sigma_0^2 \Sigma_{ll}^a)\,, \tag{4.86}$$

is valid, i.e., (4.34) holds and the observations are normally distributed, the estimated variance factor follows a Fisher distribution with R and ∞ degrees of freedom. We have the test statistic for a global test,

$$F = \frac{\widehat{\sigma}_0^2}{\sigma_0^2} \quad \text{with} \quad \underline{F}|H_0 \sim F_{R,\infty}\,, \tag{4.87}$$

which can be used for statistically testing the mathematical model. A significant deviation of the estimate $\widehat{\sigma}_0^2$ from the initial σ_0^2 indicate deviations from the mathematical model H_0, which may be outliers, unmodelled systematic errors, neglected correlations between the observations, or some combination thereof. Further hypothesis testing would be necessary to identify possible causes for the deviations from H_0.

The test is only useful for relatively small degrees of freedom, thus for redundancies below about 200. In the case of a larger redundancy, the F-Test with (R, ∞) degrees of freedom of $\widehat{\sigma}_0$ versus $\sigma_0 = 1$ will practically always be rejected, as the confidence interval for large R is very small; e.g., for $R = 200$ and a significance level of 95%, we obtain the constraint $F < 1.17$. The reason for the small confidence interval is the assumption that σ_0 is error-free, as it is assumed to be derived from an infinite sample, represented by the second parameter of the Fisher distribution. Actually, we do not know the true value of σ_0 very precisely. We should work, therefore, with a finite number of degrees of freedom R_0 for the uncertain value $\underline{\sigma}_0 = 1$, and perform a test with statistic

$$T = \frac{\widehat{\sigma}_0{}^2}{\underline{\sigma}_0^2} \quad \text{with} \quad \underline{T}|H_0 \sim F(R, R_0)\,. \tag{4.88}$$

Starting from some assumed relative precision of σ_0, R_0 can be derived from (4.83), yielding

$$R_0 = \left\lceil \frac{1}{2\sigma_{\sigma_0}^2} \right\rceil\,. \tag{4.89}$$

For example, if $R = 10\,000$, we obtain a critical value as 95% percentile $F(10\,000, \infty, 0.95) = 1.0234$. Thus, the empirical value of $\widehat{\sigma}_0$ should not be larger than $\sqrt{1.023} = 1.0116$, i.e., it should only deviate from 1 by 1%. In contrast, if we only have a redundancy

of $R = 13$, say, the critical value is $F(10,000, 13, 0.95) = 2.2$; thus, the estimated value $\widehat{\sigma}_0$ may deviate by $\sqrt{2.2} - 1 \approx 50\%$. The critical value thus mainly depends on R_0 for $R > R_0$.

Algorithm 1 collects the essential steps for the estimation in a linear Gauss–Markov model. It assumes that all matrices are full matrices and the covariance matrix Σ_{ll}^a of the observations is a good approximation of Σ_{ll}. The estimated variance factor $\widehat{\sigma}_0^2$ only is meaningful if the redundancy is large enough, which needs to be checked outside the algorithm, see the discussion after (4.80), p. 89. The regularity check during the inversion of a matrix, cf. line 5, usually is provided by the used software package.

The algorithm yields the estimated parameters together with their covariance matrix $\{\widehat{\boldsymbol{x}}, \widehat{\Sigma}_{\widehat{x}\widehat{x}}\}$, the estimated variance factor $\widehat{\sigma}_0^2$ and the redundancy R. Since it is up to the user of the algorithm to decide what minimum redundancy is necessary for relying on $\widehat{\sigma}_0^2$, determining the estimated covariance matrix $\widehat{\Sigma}_{\widehat{x}\widehat{x}} = \widehat{\sigma}_0^2 \Sigma_{\widehat{x}\widehat{x}}$ of the estimated parameters is to be done outside the algorithm. If the design matrix is sparse and the complete inverse is not necessary, the solution of the equation system and the determination of parts of the inverse will be separated.

Algorithm 1: Estimation in the linear Gauss–Markov model.
$[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}, \widehat{\sigma}_0^2, R] =$ `GaussMarkovModelLinear`$(\boldsymbol{l}, \Sigma_{ll}, A, \boldsymbol{a})$
Input: N observed values $\{\boldsymbol{l}, \Sigma_{ll}\}$, $N \times U$-design matrix A and constant vector $\boldsymbol{a}$.
Output: parameters $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = N - U$;
2 **if** $R < 0$ **then stop**, not enough observations;
3 Weight matrix: $W_{ll} = (\Sigma_{ll})^{-1}$;
4 Normal equations: $[N, \boldsymbol{n}] = A^\mathsf{T} W_{ll} [A, (\boldsymbol{l} - \boldsymbol{a})]$;
5 Theoretical covariance matrix: **if** N is regular **then** $\Sigma_{\widehat{x}\widehat{x}} = N^{-1}$**else stop**, N is singular;
6 Estimated parameters: $\widehat{\boldsymbol{x}} = \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{n}$;
7 Estimated residuals: $\widehat{\boldsymbol{v}} = A\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}$;
8 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \widehat{\boldsymbol{v}}^\mathsf{T} W_{ll} \widehat{\boldsymbol{v}} / R$ **else** $\widehat{\sigma}_0^2 = 1$.

4.2.4 Estimation of Variance Components

When the observations consist of two or more groups or their variance depends on two or more unknown factors, we can generalize the stochastical model. Then the covariance can be assumed to have the following form, cf. (4.34):

$$\Sigma_{ll} = \sum_{j=1}^{J} \Sigma_j = \sum_{j=1}^{J} \sigma_j^2 \Sigma_j^a \quad \text{with} \quad W_{ll} = \Sigma_{ll}^{-1} , \tag{4.90}$$

where the symmetric and possibly nondiagonal $N \times N$ matrices Σ_j^a are given, and the factors σ_j^2 are unknown (simplifying the notation by omitting the indices 0 in the factors and the index l in the matrices). The factors σ_j^2 in statistical literature are called *variance components*.

Two models for variance components are common:

- For *groups of N_j observations* with different variances the matrices Σ_j have the form, when assuming two groups, *groups of observations*

$$\Sigma_1^a = \begin{bmatrix} I_{N_1} & 0 \\ 0 & 0 \end{bmatrix} , \qquad \Sigma_2^a = \begin{bmatrix} 0 & 0 \\ 0 & I_{N_2} \end{bmatrix} . \tag{4.91}$$

  The groups can be chosen freely. For example, within a bundle adjustment (see Part III), one group contains all coordinates of image points, the other all coordinates of given 3D points.

additive variance model

- For an *additive model for the variances* we would assume

$$\sigma_{l_n}^2 = \sigma_1^2 + \sigma_2^2\, s(n)\,, \qquad n = 1, \ldots, N\,. \tag{4.92}$$

The function $s(n)$ needs to be specified. For instance, if assuming that the points at the border of an image are less accurate than in the centre, we could assume the model $\sigma_{l_n}^2 = \sigma_1^2 + \sigma_2^2\, d_n^2$, and thus $s(n) = d_n^2$, where d_n is the distance of the image point from the image centre.

For such an additive variance model, the two matrices Σ_j in (4.90) read as

$$\Sigma_1^a = I_N\,, \qquad \Sigma_2^a = \mathrm{Diag}([s(n)])\,. \tag{4.93}$$

For a derivation of the corresponding estimate for the variance components σ_j, we rewrite the weighted sum of the residuals using (4.62),

$$\Omega = (\boldsymbol{l} - \boldsymbol{a})^\mathsf{T} R^\mathsf{T} W_{ll} \sum_{j=1}^{J} \Sigma_j W_{ll} R (\boldsymbol{l} - \boldsymbol{a}) \tag{4.94}$$

$$= \sum_{j=1}^{J} (\boldsymbol{l} - \boldsymbol{a})^\mathsf{T} R^\mathsf{T} W_{ll} \Sigma_j W_{ll} R (\boldsymbol{l} - \boldsymbol{a}) = \sum_{j=1}^{J} \omega_j\,, \tag{4.95}$$

and evaluate the expected values of the components ω_j,

$$\mathbb{E}(\omega_j) = \mathrm{tr}(W_{ll} \Sigma_j W_{ll} \Sigma_{\widehat{v}\widehat{v}})\,. \tag{4.96}$$

variance component estimates

We now replace Σ_j by $\sigma_j^2 \Sigma_j^a$, solve for σ_j^2, and obtain unbiased estimates for the variance components,

$$\widehat{\sigma}_j^2 = \frac{\widehat{\boldsymbol{v}}^\mathsf{T} W_{ll} \Sigma_j^a W_{ll} \widehat{\boldsymbol{v}}}{\mathrm{tr}(W_{ll} \Sigma_j^a W_{ll} \Sigma_{\widehat{v}\widehat{v}})}\,. \tag{4.97}$$

The resultant variance factors $\widehat{\sigma}_j^2$ can be used to update the approximate matrices $\Sigma_j^{(\nu)} := \Sigma_j^a$ and thus the complete covariance matrix,

$$\Sigma_{ll}^{(\nu+1)} = \sum_{j=1}^{N} \widehat{\sigma}_j^{2\,(\nu)}\, \Sigma_j^{(\nu)}\,, \tag{4.98}$$

in an iterative scheme for the estimation of the unknown parameters $\widehat{\boldsymbol{x}}$.

We now specialize (4.97) for the two variance component models in (4.91) and (4.93). If we model two different groups of statistically independent observations $\{l_{n1}\}$ and $\{l_{n2}\}$, as in (4.91), we can simplify the estimate for the variance components to

$$\widehat{\sigma}_j^2 = \frac{\sum_{n \in \mathcal{N}_j} w_{nj} \widehat{v}_{nj}^2}{\sum_{n \in \mathcal{N}_j} w_{nj} \sigma_{\widehat{v}_{nj}}^2} = \frac{\sum_{n \in \mathcal{N}_j} w_{nj} \widehat{v}_{nj}^2}{\sum_{n \in \mathcal{N}_j} r_{nj}}\,, \tag{4.99}$$

where $\mathcal{N}_j$ is the set of indices belonging to the j-group. This is intuitive, as the sum of weighted squared residuals and the redundancy is simply split into two parts corresponding to the two groups of observations. The equation is also valid for more than two groups.

If we have the additive variance model in (4.93), we obtain the variance components

$$\widehat{\sigma}_1^2 = \frac{\sum_n w_n^2\, \widehat{v}_n^2}{\sum_n w_n\, r_n}\,, \qquad \widehat{\sigma}_2^2 = \frac{\sum_n w_n^2\, \widehat{v}_n^2\, s(n)}{\sum_n w_n\, r_n\, s(n)}\,. \tag{4.100}$$

Both equations, (4.99) and (4.100), are to be read as update equations within an iterative scheme: The weights, residuals and redundancy numbers on the right-hand side are to be determined from an estimation with the variances $\Sigma_{ll}^{(\nu)}$ of the iteration ν. The

left-hand side then are the variance factors $\widehat{\sigma}_j^{2\,(\nu)}$ to be used for determining the updated covariance matrix $\Sigma_{ll}^{(\nu+1)}$ of the observations following (4.98).

4.2.5 Bayesian Estimation in the Gauss–Markov Model

We now assume that we have prior information $p(\boldsymbol{x})$ about the unknown parameters $\boldsymbol{x}$. We realize Bayesian estimation by maximum likelihood estimation with adequate fictitious observations representing the prior knowledge, as discussed in Sect. 4.1.2, p. 78. We assume our pre-knowledge can be represented by the first and second moments,

$$\underline{\boldsymbol{y}}|\tilde{\boldsymbol{x}} \sim \mathcal{M}(\tilde{\boldsymbol{x}}, \Sigma_{xx})\,. \tag{4.101}$$

With this representation we can perform a maximum likelihood estimation. Following (4.8), p. 78ff., the Gauss–Markov model reads as

$$\begin{bmatrix} \underline{\boldsymbol{l}} \\ \underline{\boldsymbol{y}} \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} A \\ I_U \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \Sigma_{ll} & 0 \\ 0 & \Sigma_{xx} \end{bmatrix} \right)\,, \tag{4.102}$$

cf. the Sect. 4.2.7, p. 96 on groups of observations. We now use the real observational values $\boldsymbol{l}$ for $\underline{\boldsymbol{l}}$ and the fictitious observation $\boldsymbol{y}$ for $\underline{\boldsymbol{y}}$ and obtain the residuals for the two types of observations, both depending on the unknown parameters $\boldsymbol{x}$,

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{l} - (A\boldsymbol{x} + \boldsymbol{a}) \quad \text{and} \quad \boldsymbol{v}_x(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{x}\,. \tag{4.103}$$

The task is to find that particular $\widehat{\boldsymbol{x}}$ which minimizes the squared (augmented) Mahalanobis distance

$$\Omega'(\boldsymbol{x}) = \boldsymbol{v}^{\mathsf{T}}(\boldsymbol{x})\Sigma_{ll}^{-1}\boldsymbol{v}(\boldsymbol{x}) + \boldsymbol{v}_x^{\mathsf{T}}(\boldsymbol{x})\Sigma_{xx}^{-1}\boldsymbol{v}_x(\boldsymbol{x})\,. \tag{4.104}$$

This leads to the Bayesian estimate for the parameters

$$\widehat{\boldsymbol{x}} = \operatorname{argmax}_x \Omega'(\boldsymbol{x}) = (A^{\mathsf{T}} W_{ll} A + W_{xx})^{-1}(A^{\mathsf{T}} W_{ll}(\boldsymbol{l} - \boldsymbol{a}) + W_{xx}\boldsymbol{y}) \tag{4.105}$$

with covariance matrix

$$\Sigma_{\widehat{x}\widehat{x}} = (A^{\mathsf{T}} W_{ll} A + W_{xx})^{-1} \tag{4.106}$$

in full equivalence to Bishop (2006, Sect. 2.3.3).

Example 4.2.8: ***Wiener Filtering.*** Assume we observe the K values $\boldsymbol{s} = [s_k], k = 1, ..., K$, of an unknown signal with mean $\boldsymbol{y} = \boldsymbol{0}$ and known regular covariance matrix Σ_{ss}, cf. Fig. 4.4. The observational K-vector $\boldsymbol{l}$ is a noisy version of the signal. The additive noise is assumed to have mean zero and the regular covariance matrix Σ_{nn}. The task is to find the statistically best estimate $\widehat{\boldsymbol{s}}$ of the signal given the observations $\boldsymbol{l}$. It is given by

$$\widehat{\boldsymbol{s}} = \Sigma_{ss}(\Sigma_{ss} + \Sigma_{nn})^{-1}\boldsymbol{l}\,. \tag{4.107}$$

The estimation of the signal $\widehat{\boldsymbol{s}}$ from its observations $\boldsymbol{l}$ is called the *Wiener filter*. It is the best linear

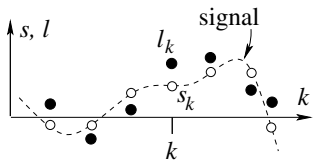


Fig. 4.4 Observations $\boldsymbol{l}_k$ of a set $\{s_k\}$ of signal values with mean zero and known covariance matrix

estimate of the signal given the observations.

Proof: The model can be written as Bayesian estimation in the Gauss–Markov model with $\boldsymbol{s}$ as unknown parameters for which we have prior information $\underline{\boldsymbol{s}}$,

$$\begin{bmatrix} \underline{\boldsymbol{l}} \\ \underline{\boldsymbol{s}} \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} I_K \\ I_K \end{bmatrix} \tilde{\boldsymbol{s}}, \begin{bmatrix} \Sigma_{nn} & 0 \\ 0 & \Sigma_{ss} \end{bmatrix} \right) . \tag{4.108}$$

The covariance matrix of the estimated parameters therefore is

$$\Sigma_{\widehat{s}\widehat{s}} = (\Sigma_{ss}^{-1} + \Sigma_{nn}^{-1})^{-1} = \Sigma_{ss}(\Sigma_{ss} + \Sigma_{nn})^{-1}\Sigma_{nn} , \tag{4.109}$$

the second expression using (A.15), p. 769. With the prior $\boldsymbol{y}$ for the signal $\underline{\boldsymbol{s}}$ we thus obtain the estimated signal

$$\widehat{\boldsymbol{s}} = (\Sigma_{ss}(\Sigma_{ss} + \Sigma_{nn})^{-1}\Sigma_{nn})(\Sigma_{nn}^{-1}\boldsymbol{l} + \Sigma_{ss}^{-1}\boldsymbol{y}) . \tag{4.110}$$

Using the specific prior $\boldsymbol{y} = \boldsymbol{0}$, this finally leads to the estimates for the signal in (4.107). ◇

4.2.6 Partitioning and Reduction of the Parameters

Often, one subset of the parameters is not relevant for further evaluation, or we are interested in reducing the number of parameters in the estimation process for efficiency reasons. This can be done conveniently when the covariance matrix of the observations is diagonal or block diagonal and each observation only relates to a few unknown parameters.

We assume the mathematical model reads

$$\underline{\boldsymbol{l}} \sim \mathcal{M}\left(C\tilde{\boldsymbol{k}} + D\tilde{\boldsymbol{p}} + \boldsymbol{a}, \Sigma_{ll} \right) ; \tag{4.111}$$

thus we have the partitionings

$$A = [C \mid D] , \qquad \boldsymbol{a}_n^{\mathsf{T}} = [\boldsymbol{c}_n^{\mathsf{T}} \mid \boldsymbol{d}_n^{\mathsf{T}}] , \qquad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{p} \end{bmatrix} . \tag{4.112}$$

(The naming of the variables originates from the classical photogrammetric bundle adjustment of images: there we have U_k coordinates $\boldsymbol{k}$ of points and U_p transformation parameters $\boldsymbol{p}$ of images.)

The normal equation system $N\widehat{\boldsymbol{x}} = \boldsymbol{n}$ reads

$$\begin{bmatrix} N_{kk} & N_{kp} \\ N_{kp}^{\mathsf{T}} & N_{pp} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{k}} \\ \widehat{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_k \\ \boldsymbol{n}_p \end{bmatrix} \tag{4.113}$$

with

$$\begin{bmatrix} N_{kk} & N_{kp} \\ N_{kp}^{\mathsf{T}} & N_{pp} \end{bmatrix} = \begin{bmatrix} C^{\mathsf{T}}W_{ll}C & C^{\mathsf{T}}W_{ll}D \\ D^{\mathsf{T}}W_{ll}C & D^{\mathsf{T}}W_{ll}D \end{bmatrix} , \quad \begin{bmatrix} \boldsymbol{n}_k \\ \boldsymbol{n}_p \end{bmatrix} = \begin{bmatrix} C^{\mathsf{T}}W_{ll}(\boldsymbol{l} - \boldsymbol{a}) \\ D^{\mathsf{T}}W_{ll}(\boldsymbol{l} - \boldsymbol{a}) \end{bmatrix} . \tag{4.114}$$

Reduction to the Transformation Parameters. When we are only interested in the transformation parameters, we can eliminate the coordinates from the estimation process, obtaining the same result as when keeping them. Solving the first equation, $N_{kk}\widehat{\boldsymbol{k}} + N_{kp}\widehat{\boldsymbol{p}} = \boldsymbol{n}_k$, in (4.113) for $\widehat{\boldsymbol{k}}$,

$$\widehat{\boldsymbol{k}} = N_{kk}^{-1}(\boldsymbol{n}_k - N_{kp}\widehat{\boldsymbol{p}}) , \tag{4.115}$$

and substituting it in the second equation, we obtain the reduced $U_p \times U_p$ normal equation system

$$\overline{N}_{pp}\widehat{\boldsymbol{p}} = \overline{\boldsymbol{n}}_p \tag{4.116}$$

with

$$\overline{N}_{pp} = N_{pp} - N_{pk}N_{kk}^{-1}N_{kp} , \qquad \overline{\boldsymbol{n}}_p = \boldsymbol{n}_p - N_{pk}N_{kk}^{-1}\boldsymbol{n}_k . \tag{4.117}$$

This allows us to determine the estimated parameters $\widehat{\boldsymbol{p}}$. The covariance matrix of the parameters $\widehat{\boldsymbol{p}}$ can be shown to be

$$\Sigma_{\widehat{p}\widehat{p}} = \overline{N}_{pp}^{-1} . \tag{4.118}$$

The estimated parameters $\widehat{\boldsymbol{k}}$ can then be derived from (4.115). The covariance matrix of the parameters $\widehat{\boldsymbol{k}}$ results from

$$\Sigma_{\widehat{k}\widehat{k}} = N_{kk}^{-1} + N_{kk}^{-1} N_{kp} \overline{N}_{pp}^{-1} N_{pk} N_{kk}^{-1} , \tag{4.119}$$

which can be determined efficiently when N_{kk} is block diagonal and the triangular decomposition of $\overline{N}_{pp}$, e.g. Cholesky, is sparse.

Reduction to the Coordinates. A similar reduction can be performed for the coordinates, leading to the normal equation system

$$\overline{N}_{kk} \widehat{\boldsymbol{k}} = \overline{\boldsymbol{n}}_k \tag{4.120}$$

with

$$\overline{N}_{kk} = N_{kk} - N_{kp} N_{pp}^{-1} N_{pk} , \qquad \overline{\boldsymbol{n}}_k = \boldsymbol{n}_k - N_{kp} N_{pp}^{-1} \boldsymbol{n}_p . \tag{4.121}$$

Reduced Design Matrix. By matrix multiplication we can show that with the design matrix reduced to the coordinates,

reduced design matrix

$$\overline{C} = C - D N_{pp}^{-1} N_{pk} = (I - D(D^{\mathsf{T}} W_{ll} D)^{-1} D^{\mathsf{T}} W_{ll})\, C , \tag{4.122}$$

the reduced normal equation system is equal to

$$\overline{C}^{\mathsf{T}} W_{ll} \overline{C}\, \widehat{\boldsymbol{k}} = \overline{C}^{\mathsf{T}} W_{ll}\, (\boldsymbol{l} - \boldsymbol{a}) \tag{4.123}$$

with the covariance matrix of the coordinates

$$\Sigma_{\widehat{k}\widehat{k}} = \left(\overline{C}^{\mathsf{T}} W_{ll} \overline{C} \right)^{-1} . \tag{4.124}$$

The form of the normal equation system (4.123) is algebraically fully equivalent to the one in the original Gauss–Markov model (4.39), p. 84ff., with the reduced design matrix $\overline{C}$ and the unknown coordinate parameters $\boldsymbol{k}$ replacing the design matrix A and the unknown parameter vector $\widehat{\boldsymbol{x}}$. Any analysis of the estimates of the coordinates $\boldsymbol{k}$ can refer to this model, without regarding the nuisance parameters $\boldsymbol{p}$.

In analogy to (4.71), p. 88, we now define the contribution $\overline{u}_{k_n}$ of a single observation l_n from a set of statistically independent observations for the determination of the coordinates $\widehat{\boldsymbol{k}}$ as

$$\overline{u}_{k_n} = \overline{U}_{k,nn} = \overline{\boldsymbol{c}}_n^{\mathsf{T}} \Sigma_{\widehat{k}\widehat{k}} \overline{\boldsymbol{c}}_n w_n , \qquad \overline{U}_k = \overline{C} (\overline{C}^{\mathsf{T}} W_{ll} \overline{C})^{-1} \overline{C}^{\mathsf{T}} W_{ll} . \tag{4.125}$$

The remaining contribution u_{p_n} of l_n to the transformation parameters $\widehat{\boldsymbol{p}}$ is

$$u_{p_n} = U_{p,nn} = \boldsymbol{d}_n^{\mathsf{T}} N_{pp}^{-1} \boldsymbol{d}_n w_n , \qquad U_p = D(D^{\mathsf{T}} W_{ll} D)^{-1} D^{\mathsf{T}} W_{ll} . \tag{4.126}$$

This leads to the three symmetric relations:

- for the idempotent matrices,

$$I_N = \overline{U}_k + U_p + R , \tag{4.127}$$

- for the contributions,

$$1 = \overline{u}_{k_n} + u_{p_n} + r_n , \tag{4.128}$$

- and for the number of observations,

$$N = U_k + U_p + R = \operatorname{tr}(\overline{U}_k) + \operatorname{tr}(U_p) + \operatorname{tr} R = \sum_n \overline{u}_{k_n} + \sum_n u_{p_n} + \sum_n r_n . \tag{4.129}$$

We will use these values for evaluating the result of an estimation process.

4.2.7 Groups of Observations and Sequential Estimation

We now give relations for parameter estimation for the case of two statistically independent groups of observations. This situation is important in sequential estimation, where the observations become available over time. We first derive the result for the estimated parameters from a joint estimation and then provide equations for a sequential estimation procedure.

4.2.7.1 Estimation with Two Groups

The linear Gauss–Markov model for the two groups $\{\boldsymbol{l}_i, \Sigma_{l_i l_i}\}, i = 1, 2,$[2] of observations can be written compactly as

$$\begin{bmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}, \sigma_0^2 \begin{bmatrix} \Sigma_{11} & \mathbf{0} \\ \mathbf{0} & \Sigma_{22} \end{bmatrix} \right) . \tag{4.130}$$

The two $N_i \times U$ design matrices $A_i, i = 1, 2$, and the approximate covariance matrices $\Sigma_{ii}^a := \Sigma_{ii}$ are assumed to be regular and to be known. For simplicity, we omit the superscript ^{a} in the approximate covariance matrices for the rest of the section. The variance factor is assumed to be initiated with $\sigma_0^2 = 1$. As the observations $\boldsymbol{l}_i$ are assumed to be mutually independent, so their covariance matrix $\Sigma_{12} = \Sigma_{21}^\mathsf{T}$ equals zero in (4.130). We already used this partitioned model for Bayesian estimation with fictitious observations, cf. Sect. 4.2.5, p. 93.

The estimated parameters result from the solution of the normal equation system

$$(A_1^\mathsf{T}\Sigma_{11}^{-1}A_1 + A_2^\mathsf{T}\Sigma_{22}^{-1}A_2)\widehat{\boldsymbol{x}} = A_1^\mathsf{T}\Sigma_{11}^{-1}(\boldsymbol{l}_1 - \boldsymbol{a}_1) + A_2^\mathsf{T}\Sigma_{22}^{-1}(\boldsymbol{l}_2 - \boldsymbol{a}_2) \tag{4.131}$$

and are identical to the solution of the normal equation system

$$\begin{bmatrix} A_1^\mathsf{T}\Sigma_{11}^{-1}A_1 & A_2^\mathsf{T} \\ A_2 & -\Sigma_{22} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} A_1^\mathsf{T}\Sigma_{11}^{-1}(\boldsymbol{l}_1 - \boldsymbol{a}_1) \\ \boldsymbol{l}_2 - \boldsymbol{a}_2 \end{bmatrix} , \tag{4.132}$$

where $\boldsymbol{\lambda}$ are Lagrangian multipliers.

The advantage of this solution is that it can handle the situation $\Sigma_{22} = 0$ rigorously, which is not possible otherwise. In this case, the additional observations are crisp constraints: thus $\boldsymbol{l}_2 = A_2\boldsymbol{x} + \boldsymbol{a}_2$ holds strictly, and $\widehat{\boldsymbol{v}}_2 = \mathbf{0}$ (see Sect. 4.3.3).

When using the normal equation system (4.132), the inverse

$$\begin{bmatrix} A_1^\mathsf{T}\Sigma_{11}^{-1}A_1 & A_2^\mathsf{T} \\ A_2 & -\Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_{\widehat{x}\widehat{x}} & S^\mathsf{T} \\ S & T \end{bmatrix} \tag{4.133}$$

contains the covariance matrix $\Sigma_{\widehat{x}\widehat{x}} = (A_1^\mathsf{T}\Sigma_{11}^{-1}A_1 + A_2^\mathsf{T}\Sigma_{22}^{-1}A_2)^{-1}$ of the unknown parameters as the upper left submatrix, which can be proven using the inverse of a block matrix; see App. (A.17), p. 769.

4.2.7.2 Sequential Estimation

If the observations $(\boldsymbol{l}_2, \Sigma_{22})$ are added after a solution with the first group only, we arrive at an update $\widehat{\boldsymbol{x}}^{(2)}$ of the previous estimates $\widehat{\boldsymbol{x}}^{(1)}$ together with their covariance matrices.

This procedure of updating is known in *Kalman filtering*; here, however, we have the simple situation that the mathematical model does not change during the updates, only its parameters. The updated values $\widehat{\boldsymbol{x}}^{(2)}$ together with their covariance matrices are identical

[2] We use the index i for groups of observations, in contrast to the index n for single observations.

to the simultaneous estimation with all observations. The update step is computationally less complex than repeating the estimation with the complete set of observations.

The idea is to use four variables to represent the current state of the estimation

$$\mathcal{S}^{(i)} = \{\boldsymbol{x}^{(i)}, \Sigma_{\widehat{x}\widehat{x}}^{(i)}, \Omega^{(i)}, R^{(i)}\}\,. \tag{4.134}$$

They contain complete information about the estimation process, as they allow the evaluation of both the parameters and the observations. The procedure

$$\mathcal{S}^{(i)} = f\left(\mathcal{S}^{(i-1)}, \{\boldsymbol{l}_i, \Sigma_{l_i l_i}\}\right) \tag{4.135}$$

is described in detail here for $i = 2$, without giving the proofs (cf. e.g., Koch, 1999, Sect. 3.2.8).

Let the solution solely be based on $\{\boldsymbol{l}_1, \Sigma_{11}\}$:

$$\Sigma_{\widehat{x}\widehat{x}}^{(1)} = (A_1^\mathsf{T}\Sigma_{11}^{-1}A_1)^{-1} \tag{4.136}$$
$$\widehat{\boldsymbol{x}}^{(1)} = \Sigma_{\widehat{x}\widehat{x}}^{(1)}A_1^\mathsf{T}\Sigma_{11}^{-1}(\boldsymbol{l}_1 - \boldsymbol{a}_1) \tag{4.137}$$
$$\widehat{\boldsymbol{v}}_1 = A_1\widehat{\boldsymbol{x}}^{(1)} + \boldsymbol{a}_1 - \boldsymbol{l}_1 \tag{4.138}$$
$$\Omega^{(1)} = \widehat{\boldsymbol{v}}_1^\mathsf{T}\Sigma_{11}^{-1}\widehat{\boldsymbol{v}}_1 \tag{4.139}$$
$$R^{(1)} = N_1 - U\,. \tag{4.140}$$

With what are called the *prediction errors*, the negative residuals, which are not the negative residuals after a *joint* estimation of both groups, *prediction errors*

$$-\widehat{\boldsymbol{v}}_2 = \boldsymbol{l}_2 - (A_2\widehat{\boldsymbol{x}}^{(1)} + \boldsymbol{a}_2) \tag{4.141}$$

and their covariance matrix,

$$\Sigma_{\widehat{v}_2\widehat{v}_2} = \Sigma_{22} + A_2\Sigma_{\widehat{x}\widehat{x}}^{(1)}A_2^\mathsf{T}\,, \tag{4.142}$$

and the matrix (in Kalman filtering called *Kalman filter gain matrix*)

$$F = \Sigma_{\widehat{x}\widehat{x}}^{(1)}A_2^\mathsf{T}\Sigma_{\widehat{v}_2\widehat{v}_2}^{-1}\,, \tag{4.143}$$

we obtain the updates Exercise 4.10

$$\widehat{\boldsymbol{x}}^{(2)} = \widehat{\boldsymbol{x}}^{(1)} - F\widehat{\boldsymbol{v}}_2 \tag{4.144}$$
$$\Delta\Sigma_{\widehat{x}\widehat{x}} = FA_2\Sigma_{\widehat{x}\widehat{x}}^{(1)} \tag{4.145}$$
$$\Sigma_{\widehat{x}\widehat{x}}^{(2)} = \Sigma_{\widehat{x}\widehat{x}}^{(1)} - \Delta\Sigma_{\widehat{x}\widehat{x}} \tag{4.146}$$
$$\Delta\Omega = \widehat{\boldsymbol{v}}_2^\mathsf{T}\Sigma_{\widehat{v}_2\widehat{v}_2}^{-1}\widehat{\boldsymbol{v}}_2 \tag{4.147}$$
$$\Omega^{(2)} = \Omega^{(1)} + \Delta\Omega \tag{4.148}$$
$$\Delta R = N_2 \tag{4.149}$$
$$R^{(2)} = R^{(1)} + \Delta R\,. \tag{4.150}$$

This sequential procedure has several important properties:

1. All four components of the estimation process are updated: $\widehat{\boldsymbol{x}}$, $\Sigma_{\widehat{x}\widehat{x}}$, Ω and R.
2. The precision of the estimate increases since the matrix $\Delta\Sigma_{\widehat{x}\widehat{x}}$ is positive semi-definite, therefore $\Sigma_{\widehat{x}\widehat{x}}^{(2)} \leq \Sigma_{\widehat{x}\widehat{x}}^{(1)}$.
3. It is possible to *undo* information, i.e., go back to a previous state by substituting $-\Sigma_{22}$ for Σ_{22} (Mikhail and Ackermann, 1976). We will exploit this possibility when deriving leave-one-out tests for evaluating an estimation w.r.t. gross errors.
4. Strong information can be easily introduced at a later stage by setting $\Sigma_{22} = \mathbf{0}$.

The recursive estimation equations are valid also if the observations are not normally distributed, but their second moments follow the model, cf. (2.136), p. 42. For testing we need to assume the observations are Gaussian distributed.

5. For normally distributed observations, the two quadratic forms $\underline{\Omega}^{(1)}$ and $\underline{\Delta\Omega}$ in (4.139) and (4.147) are statistically independent, as the observational groups $\boldsymbol{l}_i$ are independent and the estimates $\widehat{\boldsymbol{x}}^{(1)}$ are independent from the residuals $\widehat{\boldsymbol{v}}^{(1)}$ due to (4.73), p. 88.

4.2.7.3 Testing in the Second Step

testing prediction errors

Before updating the parameters, the prediction errors $\widehat{\boldsymbol{v}}_2$ can be tested in order to decide on the validity of the model $\underline{\boldsymbol{l}}_2 \sim \mathcal{N}(\mathsf{A}_2\tilde{\boldsymbol{x}} + \boldsymbol{a}_2, \Sigma_{22})$. For normally distributed observations $\boldsymbol{l}_i$, the test statistic

$$\underline{F}_2 = \frac{\underline{\Delta\Omega}}{\Delta R} \sim F(\Delta R, \infty) \tag{4.151}$$

follows a Fisher distribution $F(\Delta R, \infty)$ with ΔR and ∞ degrees of freedom if the model for the second group of observations holds. The value ΔR is the number of degrees of freedom of the quadratic form $\Delta\Omega$ if A_2 has full rank equal to N_2. The test also assumes that the two covariance matrices Σ_{ii} are consistent.

Otherwise, we can use the test statistic

$$\underline{F}'_2 = \frac{\underline{\Delta\Omega}/\Delta R}{\underline{\Omega}^{(1)}/R^{(1)}} \sim F(\Delta R, R_1)\,, \tag{4.152}$$

which follows a Fisher distribution with $\Delta R = N_2$ and $R^{(1)} = N_1 - U$ degrees of freedom if the model for both groups of observations holds. For a preset significance level $S = 1 - \alpha$, the hypothesis that the additional observations are unbiased and have the given covariance matrix will be rejected if $F'_2 > F(\Delta R, R^{(1)}, \alpha)$. The reason for $\underline{F}'_2$ to be Fisher distributed is the statistical independence of $\underline{\Delta\Omega}$ and $\underline{\Omega}^{(1)}$.

4.2.8 Covariance Intersection

Sequential procedures usually assume that the observations in subsequent steps are uncorrelated. This may not be the case, or rigorously taking the correlations into account may be computationally too expensive. As correlations change the covariance matrix of the resulting estimate, determining an *upper bound for the covariance matrix* may be of advantage. We discuss such an upper bound for the important case of determining the weighted average of two vectors, which is the core of sequential estimation. The method is called *covariance intersection* and has been proposed by Uhlmann (1995) in the context of Kalman-filtering. The geometric situation is shown in Fig. 4.5: The standard ellipse of the covariance matrix $\Sigma_{\widehat{\mu}\widehat{\mu}}$ of the mean $\widehat{\boldsymbol{\mu}}$ of two correlated vectors $\boldsymbol{x}_i, i = 1, 2$, always lies in the shaded area that is the intersection of the two standard ellipses of the two vectors touching the boundary, as indicated by the bold ellipse. The shaded area can be approximated by a standard ellipse through the four intersection points (dashed ellipse), e.g., by the covariance matrix

$$\Sigma_{\widehat{\mu}\widehat{\mu}} \leq \Sigma'_{\widehat{\mu}\widehat{\mu}} = 2(\Sigma_{11}^{-1} + \Sigma_{22}^{-1})^{-1}\,, \tag{4.153}$$

which is double the covariance matrix of the uncorrelated mean. As a consequence, if in the sequential estimation procedure the observations $\underline{\boldsymbol{l}}_1$ and $\underline{\boldsymbol{l}}_2$ are correlated by an unknown amount, the estimation of parameters $\widehat{\boldsymbol{x}}^{[2]}$ in the second step of the sequential procedure

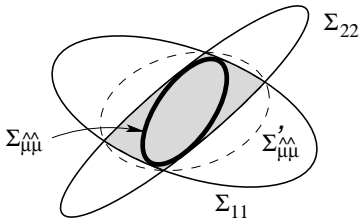


Fig. 4.5 Covariance intersection: The mean of two vectors with the same mean and covariances Σ_{11} and Σ_{22} has a covariance matrix $\Sigma_{\hat{\mu}\hat{\mu}}$ with a standard ellipse lying in the common region of the two standard ellipses. Any standard ellipse belonging to the covariance matrix $\Sigma'_{\hat{\mu}\hat{\mu}} = (\alpha\Sigma_{11}^{-1} + (1-\alpha)\Sigma_{22}^{-1})^{-1}$, $\alpha \in [0,1]$ passes through the four intersection points, encloses the shaded region, and therefore can be taken as an upper bound for the uncertainty represented by the shaded region

needs not be changed (cf. (4.144), p. 97). But the covariance matrix of the estimate a posteriori in (4.146), p. 97 has to be set to the upper bound,

$$\Sigma^{(2)}_{\hat{x}\hat{x}} = 2\,(\Sigma^{(1)}_{\hat{x}\hat{x}} - \Delta\Sigma_{\hat{x}\hat{x}})\,. \tag{4.154}$$

Neglecting possible correlations between the observations thus leads to a covariance matrix which maximally is a factor of 2 larger than the correct covariance matrix.

4.3 Gauss–Markov Model with Constraints

4.3.1 Estimation 100
4.3.2 Gauss–Markov Model with Design Matrix Not of Full Rank 101
4.3.3 Weak Constraints 102

Often the estimated parameters need to fulfil certain constraints. This may arise in the case of redundant parametrization, e.g., when representing a direction by a unit vector, when the estimated vector needs to have length 1; or when deriving scene coordinates from observed image coordinates alone, when we need to fix the position, orientation and scale of the scene coordinate system by introducing constraints on the derived coordinates. Such constraints always can be avoided by a minimal parametrization, but this may be cumbersome.

The linear Gauss–Markov model for the N observations with the H constraints can be written as

$$\underline{\boldsymbol{l}} \sim \mathcal{M}(\underset{N\times U}{\boldsymbol{A}}\,\tilde{\boldsymbol{x}} + \boldsymbol{a}, \Sigma_{ll}) \tag{4.155}$$

$$\boldsymbol{c}_h = \underset{H\times U}{\boldsymbol{H}^{\mathsf{T}}}\,\tilde{\boldsymbol{x}} \quad \text{with} \quad H < U\,, \tag{4.156}$$

with the $N \times U$ design matrix $\boldsymbol{A}$, the N-vector $\boldsymbol{a}$, the $U \times H$ constraint matrix $\boldsymbol{H}$, and the fixed H-vector $\boldsymbol{c}_h$. The number U of constraints needs to be less than the number U of parameters, which why the coefficient matrix $\boldsymbol{H}^{\mathsf{T}}$ is introduced as the transpose matrix.[3]

The matrix $\boldsymbol{H}$ must have full rank to guarantee the constraints do not contradict each other. The matrix $\boldsymbol{A}$ does not need to have full rank, but the block matrix $[\boldsymbol{A}^{\mathsf{T}}, \boldsymbol{H}]$ must have full rank in order to guarantee the estimation problem has a unique solution. Thus, we have the following rank constraints for the model (cf. Koch, 1999, Def. 1.225):

rank constraints for $\boldsymbol{H}$

[3] The convention is used that if a matrix is square or has more rows than columns, we then denote it by $\boldsymbol{A}$, and if a matrix is either square or has fewer columns than rows, we denote it by $\boldsymbol{A}^{\mathsf{T}}$. Exceptions will be indicated.

$$\mathrm{rk}(H) = H\,, \qquad \mathrm{rk}\left(\begin{bmatrix} A \\ H^\mathsf{T} \end{bmatrix}\right) = U\,. \tag{4.157}$$

redundancy

The redundancy of the estimation problem is

$$R = N + H - U\,, \qquad H < U\,, \tag{4.158}$$

as due to the constraints the parameter space is only $(U - H)$-dimensional. Therefore, the number H of constraints needs to be smaller than the number U of parameters.

4.3.1 Estimation

The optimal estimate results from minimizing the Mahalanobis distance with given constraints,

$$\widehat{\boldsymbol{x}} = \mathrm{argmin}_{x|H^\mathsf{T}x=c_h}\ \boldsymbol{v}^\mathsf{T}(\boldsymbol{x})\ \Sigma_{ll}^{-1}\ \boldsymbol{v}(\boldsymbol{x})\,. \tag{4.159}$$

Using Lagrangian multipliers $\boldsymbol{\lambda}$, we need to minimize

$$\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}(A\boldsymbol{x} + \boldsymbol{a} - \boldsymbol{l})^\mathsf{T}\ W_{ll}\ (A\boldsymbol{x} + \boldsymbol{a} - \boldsymbol{l}) + \boldsymbol{\lambda}^\mathsf{T}(H^\mathsf{T}\boldsymbol{x} - \boldsymbol{c}_h) \tag{4.160}$$

with respect to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$. The optimal estimates result from the solution of the two equation systems using $\partial \boldsymbol{a}^\mathsf{T}\boldsymbol{x}/\partial \boldsymbol{x} = \boldsymbol{a}^\mathsf{T}$,

$$\mathbf{0} = \left.\frac{\partial \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \boldsymbol{x}^\mathsf{T}}\right|_{x=\widehat{x}} = A^\mathsf{T} W_{ll}\ (A\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l}) + H\boldsymbol{\lambda} \tag{4.161}$$

$$\mathbf{0} = \left.\frac{\partial \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}^\mathsf{T}}\right|_{x=\widehat{x}} = H^\mathsf{T}\widehat{\boldsymbol{x}} - \boldsymbol{c}_h\,. \tag{4.162}$$

normal equations

This can be written as the following normal equations:

$$\begin{bmatrix} A^\mathsf{T} W_{ll} A & H \\ H^\mathsf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} A^\mathsf{T} W_{ll}(\boldsymbol{l} - \boldsymbol{a}) \\ \boldsymbol{c}_h \end{bmatrix}\,, \tag{4.163}$$

which is a linear equation system for $\widehat{\boldsymbol{x}}$ and $\boldsymbol{\lambda}$. Observe that, in all cases, introducing constraints leads to a normal equation matrix which is not positive definite. Thus when using a numerical solution based on a Cholesky factorization, you need to take care of the negative entries on the main diagonal (cf. Miccoli, 2003).

We can show that the inverse of the 2×2-block matrix contains the *covariance matrix of the estimated parameters*. With $N = A^\mathsf{T} W_{ll} A$, we have

$$\begin{bmatrix} \Sigma_{\widehat{x}\widehat{x}} & K \\ K^\mathsf{T} & M \end{bmatrix} = \begin{bmatrix} N & H \\ H^\mathsf{T} & 0 \end{bmatrix}^{-1} \tag{4.164}$$

(Koch, 1999, Eq. (3.70)) or, explicitly,

$$\Sigma_{\widehat{x}\widehat{x}} = N^{-1} - N^{-1}H(H^\mathsf{T}N^{-1}H)^{-1}H^\mathsf{T}N^{-1}\,. \tag{4.165}$$

Proof: Using (A.16), p. 769, the inverse of the 2×2 block matrix of the normal equation system in (4.164) reads as

$$\begin{bmatrix} N & H \\ H^\mathsf{T} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} N^{-1} - N^{-1}H(H^\mathsf{T}N^{-1}H)^{-1}H^\mathsf{T}N^{-1} & N^{-1}H(H^\mathsf{T}N^{-1}H)^{-1} \\ (H^\mathsf{T}N^{-1}H)^{-1}H^\mathsf{T}N^{-1} & -(H^\mathsf{T}N^{-1}H)^{-1} \end{bmatrix}\,. \tag{4.166}$$

Observe that the matrix $\Sigma_{\widehat{x}\widehat{x}}N$ is idempotent (cf. App. A.6, p. 774). Therefore, the solution for $\underline{\widehat{\boldsymbol{x}}}$ is

$$\underline{\widehat{\boldsymbol{x}}} = \Sigma_{\widehat{x}\widehat{x}}A^\mathsf{T}W_{ll}(\underline{\boldsymbol{l}} - \boldsymbol{a}) + K\boldsymbol{c}_h\,. \tag{4.167}$$

Since $\boldsymbol{c}_h$ is not random, we obtain the covariance matrix of $\underline{\widehat{\boldsymbol{x}}}$ by variance propagation and exploiting the idempotence of the matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{N}$,

$$\mathbb{D}(\underline{\widehat{\boldsymbol{x}}}) = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{W}_{ll}\,\boldsymbol{\Sigma}_{ll}\,\boldsymbol{W}_{ll}\boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}} = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{N}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}} = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\,. \tag{4.168}$$

◇

If we assume the covariance matrix of the observations is specified by an approximate covariance matrix $\boldsymbol{\Sigma}_{ll}^{a}$ and a variance factor σ_0^2, using the estimated residuals

$$\widehat{\boldsymbol{v}} = \boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{l} \tag{4.169}$$

we can arrive at an estimate for the variance factor,

$$\widehat{\sigma}_0^2 = \frac{\widehat{\boldsymbol{v}}^{\mathsf{T}}\boldsymbol{W}_{ll}^{a}\widehat{\boldsymbol{v}}}{R}\,. \tag{4.170}$$

The Mahalanobis distance Ω can be expressed in two ways, Exercise 4.11

$$\Omega = \widehat{\boldsymbol{v}}^{\mathsf{T}}\boldsymbol{W}_{ll}\widehat{\boldsymbol{v}} = (\boldsymbol{l}-\boldsymbol{a})^{\mathsf{T}}\boldsymbol{W}_{ll}(\boldsymbol{l}-\boldsymbol{a}) - (\boldsymbol{l}-\boldsymbol{a})^{\mathsf{T}}\boldsymbol{W}_{ll}\boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{c}_h^{\mathsf{T}}\boldsymbol{\lambda}\,, \tag{4.171}$$

which can be used as a numerical check of the implementation.

Again, if the mathematical model holds, the test statistic $\underline{\widehat{\sigma}_0}^2/\sigma_0^2 \sim F(R,\infty)$ follows a Fisher distribution and can be used for testing whether the data follow the assumed model.

The covariance matrix of the residuals is given by

$$\boldsymbol{\Sigma}_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll} - \boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}^{\mathsf{T}}\,, \tag{4.172}$$

using $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ from (4.164). The matrix $\boldsymbol{\Sigma}_{\widehat{v}\widehat{v}}$ is idempotent and has rank R. The redundancy matrix, as in (4.60), equals $\boldsymbol{R} = \boldsymbol{\Sigma}_{\widehat{v}\widehat{v}}\boldsymbol{W}_{ll}$.

4.3.2 Gauss–Markov Model with Design Matrix Not of Full Rank

If the observations are not sufficient to determine the parameters, the design matrix $\boldsymbol{A}$ will not have full rank, e.g., rank $Q < U$, and consequently, the normal equation matrix $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\boldsymbol{A}$ will be singular. This may happen in two cases: a) if there are not sufficient observations, e.g., when a 3D point is observed just in one image and the depth remains undefined, and b) if the parameters cannot be estimated from the observations, e.g., when estimating the coordinates of a triangle from three angles, or when estimating a homogeneous vector which is only constrained up to a scale.

A simple way to overcome this situation is to introduce additional constraints of the following form:

$$\boldsymbol{H}^{\mathsf{T}}\boldsymbol{x} = \boldsymbol{0}\,, \qquad \text{with} \qquad \boldsymbol{H} = \mathsf{null}(\boldsymbol{A})\,, \qquad \boldsymbol{H}^{\mathsf{T}}\boldsymbol{H} = \boldsymbol{I}_{U-Q}\,. \tag{4.173}$$

Then the extended normal equation matrix in (4.163), p. 100 will be regular and the covariance matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ of the resultant parameters can be derived from (4.164), p. 100. An explicit expression for the covariance matrix is

$$\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}} = \boldsymbol{N}^{+} = (\boldsymbol{N} + \boldsymbol{H}\boldsymbol{H}^{\mathsf{T}})^{-1} - \boldsymbol{H}\boldsymbol{H}^{\mathsf{T}}\,, \tag{4.174}$$

where $\boldsymbol{N}^{+}$ is what is called the *Moore–Penrose pseudo-inverse* of the matrix $\boldsymbol{N} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}^{-1}\boldsymbol{A}$ (hereafter, just pseudo-inverse) as shown in App. A.12.2, p. 779 (cf. Penrose, 1954; Moore, 1920). *Moore–Penrose pseudo-inverse*

We will come back to this model when discussing the question of choosing a coordinate system during estimation in Sect. 4.5, p. 108.

4.3.3 Weak Constraints

Constraints of the form $\boldsymbol{c}_h = \mathsf{H}^\mathsf{T}\boldsymbol{x}$ are crisp. They can also be made weak by declaring $\boldsymbol{c}_h$ a stochastic observation vector $\underline{\boldsymbol{c}}_h$ with some covariance matrix Σ_{hh}:

$$\underline{\boldsymbol{c}}_h \sim \mathcal{M}(\mathsf{H}^\mathsf{T}\tilde{\boldsymbol{x}}, \Sigma_{hh})\,. \tag{4.175}$$

Thus the function $\mathsf{H}^\mathsf{T}\boldsymbol{x} - \boldsymbol{c}_h$ may deviate from zero when referring to the covariance matrix. This can be seen as an estimation with two groups (cf. Sect. 4.2.7.1, p. 96). If $\Sigma_{hh} = \mathsf{0}$, the constraints are cris; if $\mathsf{W}_{hh} = \mathsf{0}$, the constraints are not applied.

The weak constraints can be introduced in two ways. Both start from the normal equation system for the unknown parameters:

$$\mathsf{N}\widehat{\boldsymbol{x}} = \boldsymbol{n} \qquad \text{or} \qquad \mathsf{A}^\mathsf{T}\Sigma_{ll}^{-1}\mathsf{A}\,\widehat{\boldsymbol{x}} = \mathsf{A}^\mathsf{T}\Sigma_{ll}^{-1}(\boldsymbol{l} - \boldsymbol{a})\,. \tag{4.176}$$

1. The first equation allows a continuous transition between a weak and no constraint, controlled by the parameter τ:

$$(\mathsf{N} + \tau\mathsf{H}\mathsf{W}_{hh}\mathsf{H}^\mathsf{T})\,\widehat{\boldsymbol{x}} = \boldsymbol{n} + \tau\mathsf{H}\mathsf{W}_{hh}\boldsymbol{c}_h\,. \tag{4.177}$$

   Obviously, when choosing $\tau = 1$, we introduce the original weak constraint, reducing τ towards 0 is reducing the effect of the constraint. With $\tau = 0$ the constraint is not applied. Introducing nearly crisp constraints by choosing a large value for τ leads to numerical instabilities as the second term $\tau\mathsf{H}\mathsf{W}_{hh}\mathsf{H}^\mathsf{T}$ then is dominant and singular, since the number of constraints with H is smaller than the number U of unknowns, i.e., $H < U$.
2. The second equation allows a continuous transition between a weak and a crisp constraint, controlled by the parameter κ, by solving the following system:

$$\begin{bmatrix} \mathsf{N} & \mathsf{H} \\ \mathsf{H}^\mathsf{T} & -\kappa\Sigma_{hh} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{x}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{c}_h \end{bmatrix}\,. \tag{4.178}$$

   First we see that by reducing the equation system to the parameters $\widehat{\boldsymbol{x}}$, we obtain the above-mentioned system (4.177) when choosing $\kappa = 1/\tau$. Choosing $\kappa = 1$ therefore means introducing the weak constraint. Reducing κ towards 0 increases the influence of the constraint. With $\kappa = 0$ we enforce the constraint crisply.

4.4 The Nonlinear Gauss–Markov Model

4.4.1 The Nonlinear Gauss–Markov Model Without Constraints 103
4.4.2 The Nonlinear Gauss–Markov Model with Constraints 104

If the observations $\boldsymbol{x}$ depend nonlinearly on the unknown parameters $\boldsymbol{x}$, namely

$$\tilde{\boldsymbol{l}} = \boldsymbol{f}(\tilde{\boldsymbol{x}})\,, \tag{4.179}$$

we generally do not have a direct solution. There may be multiple solutions even in case of positive redundancy. This might be acceptable in certain applications. Alternatively, we may start from approximate values, we linearize and iteratively correct the approximate values until they converge. If the approximate values are close enough to the true values, we can expect the iteration scheme to converge to the global solution, which we then take as the final estimate. The uncertainty estimates in the last iteration can then be transferred to the final estimates, which allows the application of all statistical techniques discussed so far.

There are two classical ways to handle the nonlinearity of the functions in the functional model in an iterative scheme:

1. The *Gauss–Newton method*: First linearize the model and then iteratively apply the estimation procedure for the linear Gauss–Markov model. This method is well-suited for problems with small observational variances and functions $\boldsymbol{f}$ with low curvature. *Gauss–Newton method*
2. The Newton method, also known as the *Newton–Raphson method*: First take the partial derivatives of the nonlinear optimization function and then use their linearized form for the solution (Antoniou and Lu, 2007). This method takes the second derivatives of the functions $\boldsymbol{f}$ in the into account and therefore can handle cases with larger observational variances and highly curved functions $\boldsymbol{f}(\boldsymbol{x})$. *Newton–Raphson method*

We first discuss the nonlinear model without additional constraints applying the Gauss–Newton method. We then provide the more general Newton–Raphson method, where we also present the estimation with additional nonlinear constraints $\boldsymbol{h}(\tilde{\boldsymbol{x}}) = \boldsymbol{0}$ for the unknown parameters.

4.4.1 The Nonlinear Gauss–Markov Model Without Constraints

The nonlinear Gauss–Markov model reads as

$$\underline{\boldsymbol{l}} \sim \mathcal{M}\left(\boldsymbol{f}(\tilde{\boldsymbol{x}}), \Sigma_{ll}\right) \tag{4.180}$$

for the N observations $\boldsymbol{l}$ and the U unknown parameters $\tilde{\boldsymbol{x}}$ related by some arbitrary function $\boldsymbol{f}(\boldsymbol{x})$, which we in the following assume to be differentiable twice. The task is to find the estimate,

$$\widehat{\boldsymbol{x}} = \operatorname{argmin}_x (\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}))^{\mathsf{T}}\, \Sigma_{ll}^{-1}\, (\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}))\,. \tag{4.181}$$

The Gauss–Newton method starts from approximate values $\widehat{\boldsymbol{x}}^a$ for the estimates of the unknown parameters. It linearizes the model, estimates corrections $\widehat{\Delta\boldsymbol{x}}$ for the unknown parameters within a linear Gauss–Markov model using the techniques of the previous section, and updates the approximate values in an iterative scheme. With the iteration index ν and the approximate values in the first iteration $\widehat{\boldsymbol{x}}^{(1)} = \widehat{\boldsymbol{x}}^a$ we thus have

$$\widehat{\boldsymbol{x}}^{(\nu+1)} = \widehat{\boldsymbol{x}}^{(\nu)} + \widehat{\Delta\boldsymbol{x}}^{(\nu)}\,. \tag{4.182}$$

In the case of convergence, i.e., $\widehat{\Delta\boldsymbol{x}}^{(\nu)} \to \boldsymbol{0}$, we adopt the result of the last iteration as the sought optimum.

In the following, we describe one iteration of the iteration process using the substitutions for simplifying notation, cf. Fig. 4.6:

$$\widehat{\boldsymbol{x}} := \widehat{\boldsymbol{x}}^{(\nu+1)}\,, \qquad \widehat{\boldsymbol{x}}^a := \widehat{\boldsymbol{x}}^{(\nu)}\,, \qquad \widehat{\Delta\boldsymbol{x}} := \widehat{\Delta\boldsymbol{x}}^{(\nu)} \tag{4.183}$$

and thus use the update rule

$$\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{x}}^a + \widehat{\Delta\boldsymbol{x}}\,. \tag{4.184}$$

With the linearized function

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}^a) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\widehat{x}^a} \cdot \widehat{\Delta\boldsymbol{x}} + O(|\widehat{\Delta\boldsymbol{x}}|^2) \tag{4.185}$$

we arrive at the *linear substitute model* (4.34), *linear substitute model*

$$\underline{\Delta\boldsymbol{l}} \sim \mathcal{M}(A\widehat{\Delta\boldsymbol{x}}, \Sigma_{ll})\,, \tag{4.186}$$

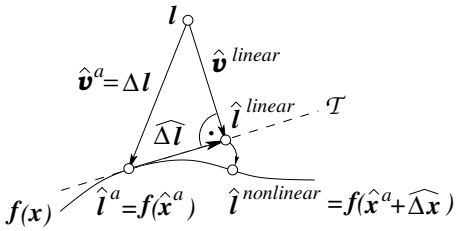


Fig. 4.6 Update in the nonlinear Gauss–Markov model. The corrections $\widehat{\Delta \boldsymbol{x}}$ for the estimated parameters are determined from the linearized model in the tangent space $\mathcal{T}$ of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at the approximated values $\widehat{\boldsymbol{x}}^a$ for the parameters $\widehat{\boldsymbol{x}}$. The difference between the linear and the nonlinear estimates for the fitted observations $\widehat{\boldsymbol{l}}$ and the estimated residuals $\widehat{\boldsymbol{v}}$ are meant to converge to zero. The vector of the nonlinear residuals $\widehat{\boldsymbol{v}}^{\text{nonlinear}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) - \boldsymbol{l}$ is not shown.

with the reduced observations in the linear substitute model, identical to approximates $\widehat{\boldsymbol{v}}^a$ for the residuals,

$$\Delta \boldsymbol{l} = \boldsymbol{f}(\widehat{\boldsymbol{x}}^a) - \boldsymbol{l} =: \widehat{\boldsymbol{v}}^a \,, \tag{4.187}$$

the design matrix, which is the Jacobian of the function evaluated at the approximate values of the unknown parameters,

$$\mathsf{A} = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|_{x=\widehat{x}^a} \,, \tag{4.188}$$

and the unknown corrections to the parameters,

$$\widehat{\Delta \boldsymbol{x}} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^a \,. \tag{4.189}$$

The estimated corrections thus can be determined from the normal equation system,

$$\mathsf{A}^\mathsf{T} \mathsf{W}_{ll} \mathsf{A} \, \widehat{\Delta \boldsymbol{x}} = \mathsf{A}^\mathsf{T} \mathsf{W}_{ll} \Delta \boldsymbol{l} \,. \tag{4.190}$$

The residuals of the linear substitute model,

$$\widehat{\boldsymbol{v}}^{\text{linear}} = \mathsf{A} \widehat{\Delta \boldsymbol{x}} - \Delta \boldsymbol{l} \,, \tag{4.191}$$

at the end of the iterations should be equal to those of the nonlinear model

$$\widehat{\boldsymbol{v}}^{\text{nonlinear}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) - \boldsymbol{l} \,. \tag{4.192}$$

This is a check on the correct linearization.

The residuals can be used to determine the estimated variance factor $\widehat{\sigma}_0^2$ according to $\Omega = \widehat{\boldsymbol{v}}^\mathsf{T} \mathsf{W}_{ll}^a \widehat{\boldsymbol{v}}$ using the approximate weight matrix $\mathsf{W}_{ll}^a = (\Sigma_{ll}^a)^{-1}$ and the redundancy $R = N - U$.

The further analysis steps, modifications and generalizations, can be performed as for the linear Gauss–Markov model.

4.4.2 The Nonlinear Gauss–Markov Model with Constraints

The nonlinear Gauss–Markov model with constraints starts from the model,

$$\underset{N\times 1}{\underline{\boldsymbol{l}}} \sim \mathcal{M}(\boldsymbol{f}(\tilde{\boldsymbol{x}}), \Sigma_{ll}) \tag{4.193}$$

$$\underset{H\times 1}{\boldsymbol{0}} = \boldsymbol{h}(\tilde{\boldsymbol{x}}) \tag{4.194}$$

for the N observations $\boldsymbol{l}$ and the U unknown parameters, $\tilde{\boldsymbol{x}}$, with the N nonlinear functions, $\boldsymbol{f}(\boldsymbol{x}) = [f_n(\boldsymbol{x})]$, and the H nonlinear constraints, $\boldsymbol{h}(\tilde{\boldsymbol{x}}) = [h_\eta(\tilde{\boldsymbol{x}})]$, $\eta = 1, \ldots, H$. The redundancy of the model is the same as for the linear Gauss–Markov model with linear constraints, $R = N + H - U$. The task is to minimize the weighted sum of residuals,

$$\Omega = \frac{1}{2}(\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x}))^{\mathsf{T}} W_{ll}(\boldsymbol{l} - \boldsymbol{f}(\boldsymbol{x})) \quad \text{such that} \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}\,, \tag{4.195}$$

using the constraints of (4.194).

4.4.2.1 Gauss–Newton Method

The Gauss–Newton method again uses approximate values for the unknown parameters and the update relations in (4.182) and arrives at the linearized substitute model,

$$\underline{\Delta \boldsymbol{l}} \sim \mathcal{M}\left(A\,\widetilde{\Delta \boldsymbol{x}}, \Sigma_{ll}\right) \tag{4.196}$$

$$\boldsymbol{c}_h = H^{\mathsf{T}}\,\widetilde{\Delta \boldsymbol{x}}\,, \tag{4.197}$$

with the reduced observations in (4.187), the Jacobian in (4.188), the residual constraints,

$$\boldsymbol{c}_h = -\boldsymbol{h}(\widehat{\boldsymbol{x}}^a)\,, \tag{4.198}$$

and the Jacobian of the constraints,

$$H = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}^{\mathsf{T}}}\right|_{x=\widehat{x}^a}\,. \tag{4.199}$$

The unknown parameters $\widehat{\Delta \boldsymbol{x}}$ of the linearized model therefore can be determined from the extended normal equation system,

$$\begin{bmatrix} A^{\mathsf{T}} W_{ll} A & H \\ H^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta \boldsymbol{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} A^{\mathsf{T}} W_{ll} \Delta \boldsymbol{l} \\ \boldsymbol{c}_h \end{bmatrix}\,. \tag{4.200}$$

Iteratively updating the parameters using (4.182), p. 103 results in the Gauss–Newton method. However, due to the nonlinearity of the constraints the update requires special care, see Sect. 4.4.2.3, p. 107 and Sect. 10.6.1, p. 415. The covariance matrix of the estimates parameters $\widehat{\boldsymbol{x}}$ can be derived from (4.164).

4.4.2.2 Newton–Raphson Iteration Scheme

The Newton–Raphson method directly refers to the original optimization problem (4.195) with constraints (cf. Griva et al., 2009, Sect. 14.6). The Lagrangian function is

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l})^{\mathsf{T}} W_{ll}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l}) + \boldsymbol{h}^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{\lambda}\,, \tag{4.201}$$

with the H-vector $\boldsymbol{\lambda}$ of Lagrangian multipliers. With the vector $\boldsymbol{p}^{\mathsf{T}} = [\boldsymbol{x}^{\mathsf{T}}, \boldsymbol{\lambda}^{\mathsf{T}}]$ the Newton–Raphson method starts from approximate values $\boldsymbol{p}^{a\mathsf{T}}$ and determines updates,

$$\boldsymbol{p} = \boldsymbol{p}^a + \Delta \boldsymbol{p} = \boldsymbol{p}^a - \left(\nabla^2 L(\boldsymbol{p}^a)\right)^{-1}\, \nabla L(\boldsymbol{p}^a)\,, \tag{4.202}$$

until convergence. The Hessian $\nabla^2 L$ and the gradient ∇L of the Lagrangian are to be evaluated at the approximate values. Expanding the elements of the Hessian and the gradient w.r.t. the corrections $\Delta \boldsymbol{x}$ and $\Delta \boldsymbol{\lambda}$ of the unknowns and the Lagrangian parameters yields the normal equation system,

$$\begin{bmatrix} \nabla^2_{xx} L(\widehat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^a) & \nabla^2_{x\lambda} L(\widehat{\boldsymbol{x}}^a) \\ \nabla^2_{\lambda x} L(\widehat{\boldsymbol{x}}^a) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\Delta \boldsymbol{x}} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla_x L(\widehat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^a) \\ \nabla_\lambda L(\widehat{\boldsymbol{x}}^a) \end{bmatrix}, \tag{4.203}$$

where the indices x and λ indicate the variable of the partial differentiation. With the $U \times U$ Hessian matrices,

$$\underset{U\times U}{\boldsymbol{P}_n}(\widehat{\boldsymbol{x}}^a) = \nabla^2_{xx} f_n(\widehat{\boldsymbol{x}}^a) \quad \text{and} \quad \underset{U\times U}{\boldsymbol{Q}_\eta}(\widehat{\boldsymbol{x}}^a) = \nabla^2_{xx} h_\eta(\widehat{\boldsymbol{x}}^a) \tag{4.204}$$

of the individual functions, f_n and h_η, to be evaluated at the approximate values we obtain the explicit form of the normal equation system (see proof below),

$$\boldsymbol{M}\Delta\boldsymbol{p} = \boldsymbol{m}: \quad \begin{bmatrix} \boldsymbol{N} & \boldsymbol{H} \\ \boldsymbol{H}^\mathsf{T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\Delta \boldsymbol{x}} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{c}_h \end{bmatrix}, \tag{4.205}$$

with

$$\boldsymbol{N} = \boldsymbol{A}^\mathsf{T}(\widehat{\boldsymbol{x}}^a)\, \boldsymbol{W}_{ll}\, \boldsymbol{A}(\widehat{\boldsymbol{x}}^a) + \sum_n (\boldsymbol{W}_{ll}\widehat{\boldsymbol{v}})_n\, \boldsymbol{P}_n(\widehat{\boldsymbol{x}}^a) + \sum_\eta \lambda_\eta \boldsymbol{Q}_\eta(\widehat{\boldsymbol{x}}^a) \tag{4.206}$$

$$\widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}^a) - \boldsymbol{l} \tag{4.207}$$

$$\boldsymbol{H} = \boldsymbol{H}(\widehat{\boldsymbol{x}}^a) \tag{4.208}$$

$$\boldsymbol{n} = -\boldsymbol{A}^\mathsf{T}(\widehat{\boldsymbol{x}}^a)\, \boldsymbol{W}_{ll}\, \widehat{\boldsymbol{v}} - \boldsymbol{H}^\mathsf{T}(\widehat{\boldsymbol{x}}^a)\boldsymbol{\lambda}^a \tag{4.209}$$

$$\boldsymbol{c}_h = -\boldsymbol{h}(\widehat{\boldsymbol{x}}^a), \tag{4.210}$$

where the scalar $(\boldsymbol{W}_{ll}\widehat{\boldsymbol{v}})_n$ is the nth element of the vector $\boldsymbol{W}_{ll}\widehat{\boldsymbol{v}}$. The iteration can be started with $\boldsymbol{\lambda}^{(0)} = \boldsymbol{0}$. The covariance matrix of the estimates parameters $\widehat{\boldsymbol{x}}$ can be derived from (4.164).

The differences between the extended normal equation system (4.205) and the one of the linear solution in (4.163), p. 100 are the additive terms, $\sum_n \boldsymbol{P}_n(\widehat{\boldsymbol{x}}^a)(\boldsymbol{W}_{ll}\widehat{\boldsymbol{v}})_n$, $\sum_\eta \lambda_\eta \boldsymbol{Q}_\eta(\widehat{\boldsymbol{x}}^a)$ and $-\boldsymbol{H}^\mathsf{T}(\widehat{\boldsymbol{x}}^a)\boldsymbol{\lambda}^a$, in the normal equation matrix, $\boldsymbol{N}$, and the right-hand sides, $\boldsymbol{n}$, respectively. They are essential for convergence if the observational noise or equivalently the difference between the approximate and the final estimates is large. This should be reflected in large standard deviations σ_l for the observations. Otherwise, due to the large weights, $\boldsymbol{W}_{ll}$ in $\boldsymbol{N}$, (4.206) and $\boldsymbol{n}$ (4.209), the first terms are dominant, and the additional second-order terms can be neglected, leading to the Gauss–Newton method (4.200).

Proof: For proving (4.205) we need the Hessian and the gradient of the Lagrangian function L. The gradient is given by

$$\nabla L = \begin{bmatrix} \nabla_x L(\boldsymbol{x}, \boldsymbol{\lambda}) \\ \nabla_\lambda L(\boldsymbol{x}, \boldsymbol{\lambda}) \end{bmatrix} = \begin{bmatrix} \nabla_x f^\mathsf{T}(\boldsymbol{x})\, \boldsymbol{W}_{ll}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l}) + \nabla_x h^\mathsf{T}(\boldsymbol{x})\, \boldsymbol{\lambda} \\ \boldsymbol{h}(\boldsymbol{x}) \end{bmatrix}, \tag{4.211}$$

The Hessian matrix $\nabla^2_{xx} L$ of the Lagrangian function L w.r.t. $\boldsymbol{x}$ is achieved by partial differentiation of $\nabla_x L$ in (4.211), using $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l}$

$$\nabla^2_{xx} L(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_n \left(\underset{U\times 1}{\nabla_x f_n(\boldsymbol{x})} \left(\boldsymbol{W}_{ll} \underset{1\times U}{\frac{\partial \boldsymbol{v}(\boldsymbol{x})}{\partial \boldsymbol{x}}} \right)_n + \underset{U\times U}{\nabla^2_{xx} f_n(\boldsymbol{x})} \underset{1\times 1}{(\boldsymbol{W}_{ll}\boldsymbol{v}(\boldsymbol{x}))_n} \right) + \sum_\eta \lambda_\eta\, \nabla^2_{xx} h_\eta(\boldsymbol{x}). \tag{4.212}$$

The mixed derivatives of L are

$$\nabla^2_{x\lambda} L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{\partial^2 L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \boldsymbol{x} \partial \boldsymbol{\lambda}} = \boldsymbol{H}(\boldsymbol{x}). \tag{4.213}$$

All gradients and Hessians need to be evaluated at the approximate values $(\widehat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^a)$. With the Jacobians,

$$\underset{N\times U}{\boldsymbol{A}(\boldsymbol{x})} = \nabla_x f(\boldsymbol{x}) = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{v}(\boldsymbol{x})}{\partial \boldsymbol{x}} \quad \text{and} \quad \underset{G\times U}{\boldsymbol{H}^\mathsf{T}(\boldsymbol{x})} = \nabla_x h(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}, \tag{4.214}$$

and the Hessians in (4.204), insertion of (4.211), (4.212), and (4.213) into (4.203) yields (4.205). ◇

4.4.2.3 Consistent Updates

Since the constraints $\boldsymbol{h}(\tilde{\boldsymbol{x}}) = \boldsymbol{0}$ are nonlinear, the update of the parameters $\widehat{\boldsymbol{x}}$ in this model needs some care. If we would follow the update (4.202)

$$\widehat{\boldsymbol{x}}^{(\nu+1)} = \widehat{\boldsymbol{x}}^{(\nu)} + \widehat{\Delta\boldsymbol{x}}\,, \tag{4.215}$$

the resulting update $\widehat{\boldsymbol{x}}^{(\nu+1)}$ will not fulfil the constraints, and thus would lead to updated approximate values, which are not consistent with the functional model. For example, if we want to estimate a normalized direction vector $\boldsymbol{x}$ having unit length, we impose the constraint $h(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^{\mathsf{T}}\tilde{\boldsymbol{x}} - 1 = 0$, i.e., the estimate is forced to lie on the unit circle. Then, due to the linearied constraint (4.205), $h(\widehat{\boldsymbol{x}}^a)+\boldsymbol{H}^{\mathsf{T}}\widehat{\Delta\boldsymbol{x}} = h(\boldsymbol{x}^a)+2\widehat{\boldsymbol{x}}^a\widehat{\Delta\boldsymbol{x}} = 0$, the correction $\widehat{\Delta\boldsymbol{x}}$ will be in the direction perpendicular to $\widehat{\boldsymbol{x}}^a$ if $\widehat{\boldsymbol{x}}^a$ fulfills the constraint. Thus, even if $\widehat{\boldsymbol{x}}^{(\nu)}$ fulfills the constraint and lies on the unit circle, the updated estimate, $\widehat{\boldsymbol{x}}^{(\nu+1)}$, will not. Generally, this may prevent convergence. In order to obtain updates which are consistent with the constraints, it is necessary to correct $\widehat{\boldsymbol{x}}^{(\nu+1)}$ such that it fulfills the constraint, in the example by normalization. Since this correction is dependent on the constraints we write it as

$$\widehat{\boldsymbol{x}}^{(\nu+1)} = u_x(\widehat{\boldsymbol{x}}^{(\nu)}, \widehat{\Delta\boldsymbol{x}})\,, \tag{4.216}$$

which in the example would be

$$\widehat{\boldsymbol{x}}^{(\nu+1)} = \mathsf{N}(\widehat{\boldsymbol{x}}^{(\nu)} + \widehat{\Delta\boldsymbol{x}})\,. \tag{4.217}$$

We will discuss various correction functions when discussing nonlinear constraints for geometric entities in Chap. 10, p. 359.

Algorithm 2 collects the essential steps for the estimation in a nonlinear Gauss–Markov model with constraints between the unknown parameters. It assumes that all matrices are full matrices. It uses the Gauss–Newton update scheme.

The algorithm requires two procedures: (1) The procedure $c_f(\widehat{\boldsymbol{x}}^a, \boldsymbol{l})$ for computing the corrections $\Delta\boldsymbol{l}$, (4.187) together with the Jacobians $\boldsymbol{A}$ and (2) the procedure $c_h(\widehat{\boldsymbol{x}}^a)$ for determining $\boldsymbol{c}_h$ and $\boldsymbol{H}$. (4.214). They contain the functions $\boldsymbol{f}$ and $\boldsymbol{h}$ and depend on the current point $\widehat{\boldsymbol{x}}^a$ of the Taylor expansion. Therefore, the estimation requires approximate values $\boldsymbol{x}^a$ for all unknown parameters. For checking the convergence an approximation $\sigma^a_{x_u}$ for the final standard deviations of the estimated parameters $\widehat{x}_u$ is to be provided.

The algorithm yields the estimated parameters together with their covariance matrix $\{\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{\Sigma}}_{\widehat{x}\widehat{x}}\}$, the estimated variance factor $\widehat{\sigma}_0^2$ and the redundancy R.

The following steps of the algorithm need some explanation:

1 The redundancy R needs to be nonnegative. If no constraints are provided, then $H = 0$.

5,15,18 The stopping variable s is starting with $s = 0$. The estimation finalizes either if the parameters do not change more than a threshold T_x or if a preset number `maxiter` of iterations is reached. Then the stopping variable is set to 2.

17 The estimated parameters are updated using the update function u_x, which depends on the approximate values $\widehat{\boldsymbol{x}}^{(\nu)}$ for the unknown parameters and the estimated corrections $\widehat{\Delta\boldsymbol{x}}$. For linear constraints we safely can use $u_x(\widehat{\boldsymbol{x}}^{(\nu)}, \widehat{\Delta\boldsymbol{x}}) = \widehat{\boldsymbol{x}}^{(\nu)} + \widehat{\Delta\boldsymbol{x}}$. When estimating transformations or geometric entities represented by homogeneous coordinates, this update, however, involves multiplications and normalizations (see Chap. 10, p. 359). For improving convergence line search or trust region methods may be necessary (cf. Nocedal and Wright, 1999).

If no constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = 0$ between the parameters are present, the setup of the normal equation system, the determination of the covariance matrix, and the determination of the estimated parameters may be simplified.

Algorithm 2: Estimation in the Gauss–Markov model with constraints and Newton method
$[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}, \widehat{\sigma}_0^2, R] =$ `GaussMarkovModell_Constraints`$(\boldsymbol{l}, \Sigma_{ll}, c_f, c_h, \widehat{\boldsymbol{x}}^a, \boldsymbol{\sigma}_{\widehat{x}}^a, T_x,$ `maxiter`$)$
Input: observed values $\{\boldsymbol{l}, \Sigma_{ll}\}$, number N, number H,
Functions for linearization $[\boldsymbol{\Delta l}, \boldsymbol{A}] = c_f(\widehat{\boldsymbol{x}}, \boldsymbol{l})$, $[\boldsymbol{c}_h, \boldsymbol{H}] = c_h(\widehat{\boldsymbol{x}})$,
Approximate values $\widehat{\boldsymbol{x}}^a$, expected uncertainty $\boldsymbol{\sigma}_{\widehat{x}}^a = [\sigma_{\widehat{x}_u}^a]$,
Thresholds for convergence T_x, `maxiter`.
Output: parameters $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

Redundancy $R = N - U + H$;
if $R < 0$ **then stop**, not enough observations;
Initiate: $\nu = 1$, $\widehat{\boldsymbol{x}}^{(\nu)} = \widehat{\boldsymbol{x}}^a$;
Stopping condition: $s = 0$;
Weight matrix: $\boldsymbol{W}_{ll} = \Sigma_{ll}^{-1}$;
repeat
  Observations and Jacobians for $\boldsymbol{f}$: $[\widehat{\boldsymbol{\Delta l}}, \boldsymbol{A}] = c_f\left(\widehat{\boldsymbol{x}}^{(\nu)}, \boldsymbol{l}\right)$ (4.187), (4.188), p. 104;
  Constraints and Jacobians for $\boldsymbol{h}$: $[\boldsymbol{c}_h, \boldsymbol{H}] = c_h\left(\widehat{\boldsymbol{x}}^{(\nu)}\right)$, (4.198), (4.199), p. 105;
  Normal equations: $\boldsymbol{M}$ and $\boldsymbol{m}$, see (4.205), p. 106;
  if $\boldsymbol{N}$ is singular **then stop**;
  Estimated corrections to parameters: $\widehat{\boldsymbol{\Delta x}}$, see (4.205), p. 106;
  Next iteration step: $\nu := \nu + 1$;
  if $\max|\widehat{\Delta x}_u|/\sigma_{\widehat{x}_u}^a < T_x$ for all u or $\nu =$ `maxiter` **then** $s = 2$;
  Estimated parameters: $\widehat{\boldsymbol{x}}^{(\nu)} = u_x\left(\widehat{\boldsymbol{x}}^{(\nu-1)}, \widehat{\boldsymbol{\Delta x}}\right)$;
until $s \equiv 2$;
Estimated residuals: $\widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}^{(\nu)}) - \boldsymbol{l}$;
Covariance matrix: $\boldsymbol{C} = \begin{bmatrix} \Sigma_{\widehat{x}\widehat{x}} & \cdot \\ \cdot & \cdot \end{bmatrix} = \boldsymbol{M}^{-1}$, see (4.164), p. 100 ;
if $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \widehat{\boldsymbol{v}}^\mathsf{T} \boldsymbol{W}_{ll} \widehat{\boldsymbol{v}}/R$ **else** $\widehat{\sigma}_0^2 = 1$.

4.5 Datum or Gauge Definitions and Transformations

4.5.1 Gauge and Datum 109
4.5.2 Gauge Constraints and Inner Precision 110
4.5.3 Imposing Gauge Constraints and Gauge Transformations 111
4.5.4 Gauge in a System Reduced to the Transformation Parameters . . . 113

The unknown parameters may not be estimable from the given observations. In this case, the resulting Jacobian $\boldsymbol{A}$ and therefore, the normal equation matrix $\boldsymbol{N}$ in (4.39) will be rank deficient, disclosing a lack of information.

There are two classical situations where this may happen:

1. There are not sufficient observations; e.g., when observing an unknown 3D point in a single image only, not all three 3D coordinates will be estimable. Leaving out a set of parameters, or adding observations, or introducing some prior information for the parameters in a Bayesian manner may solve the problem.
2. In spite of having redundant observations, some parameters may still not be estimable; e.g., if we want to derive the coordinates of the points of a triangle from the three side lengths alone, or if we want to derive the position, orientation and size of a 3D object, represented by a set of 3D coordinates, from image information alone. In this case the position, rotation and scale of the object are not determinable with respect to a reference coordinate system, as no 3D reference points in that reference system are provided and observed.

This second situation is characterized by a lack of definition of the reference system for the unknown coordinates.

The choice of the reference is free. An estimation wherein the reference coordinate system can be freely chosen is called a *free adjustment*. As the observational structure can usually be represented as a network, a network where the reference system of the coordinates is not fixed is called a *free network*. Therefore, the resulting coordinates and their uncertainty depend on the choice of the reference system. However, the form of the 3D object and its uncertainty can be determined uniquely.

free adjustment
free network

4.5.1 Gauge and Datum

The freedom of choosing the reference system is similar to setting the initial time of a calendar, i.e., fixing the *datum*, or to setting the length of a measuring rod, i.e., fixing the *gauge*. Whereas the notion *datum* is used in the geodetic-photogrammetric community as equivalent to the coordinate system within a free estimation, the notion *gauge* is common in physics when fixing the calibration of a measuring device. The choice of *datum* or *gauge* has no effect on the phenomenon to be investigated, in our context the shape of objects. We will use the notion *gauge* in the following.

datum, gauge

Not fixing a reference coordinate system will cause the normal equation matrix to be singular, indicating that scene coordinates and transformation parameters are not *estimable quantities*, as they refer to a coordinate system which is not specified.

estimable quantities

We need to distinguish between two specifications (cf. Fig. 4.7):

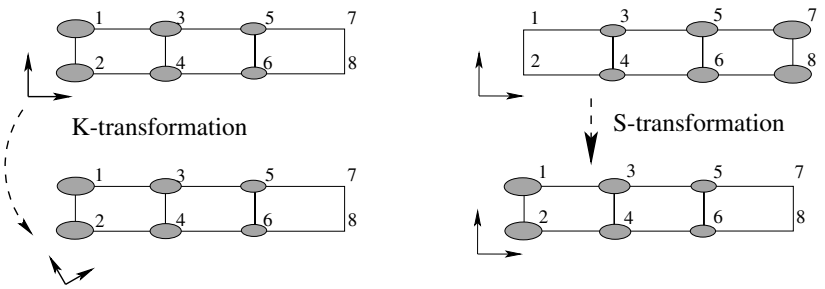


Fig. 4.7 Coordinate or K-transformation and gauge or S-transformation. **Left:** K-transformation changes the reference coordinate system of the coordinates. The gauge is not changed, which here is defined by point pair (7,8). **Right:** S-transformations changes the gauge of a covariance matrix, here from point pair (1,2) to point pair (7,8). The coordinates are not changed

1. Specifying the *gauge of the coordinate system* of the scene points or the transformation parameters means setting the origin, the direction, and the scaling of the axes of this coordinate system (cf. Molenaar, 1981).
   Changing the coordinate system will be called a *K-transformation*. Thus classical coordinate transformations are K-transformations. They are realized by performing the transformation of the coordinates with fixed transformation parameters and applying the well-known variance–covariance propagation, and are therefore not discussed in detail in the following.

   coordinate- or K-transformation
2. Specifying the *gauge of the covariance matrix* of the scene points or the transformation parameters means fixing the position, direction and scale of selected, possibly virtual scene elements such that they have standard deviation zero. Following Baarda (1973), who proposed this method, changing the gauge of a covariance matrix will be called an *S-transformation*.[4] Such an S-transformation does not change the coordinates but

   gauge- or S-transformation

[4] This refers a similarity transformation for a 2D or 3D point set where only angles or directions are observed.

only their covariance matrix. Smith et al. (1991) proposed the same method in the context of robotics.

Both specifications obviously do not coincide (cf. Fig. 4.7): Moving the coordinate system of the scene points, a K-transformation changes their coordinates, but not the reference system of their covariance matrix. Changing the gauge of the covariance matrix, on the other hand, does not change the coordinate values, but the reference system for their uncertainty (cf. Baarda, 1973, Sect. 7).

A classical case is the definition of the coordinate system for the 3D coordinates from a set of 3D points derived from images, e.g., using the techniques discussed in Part III. Here, seven gauge parameters for a similarity transformation need to be specified,[5] namely the translation (three parameters), the rotation (three parameters), and the scale (one parameter) of the point set. These cannot be derived from image information only. Being conformal, the transformation does not change the bundles of rays of the images from which the point set is derived. Therefore, if the 3D coordinates of the model points are to be derived from image measurements, we may fix the coordinate system by arbitrarily specifying seven coordinates, e.g., the 3D coordinates of two points, leaving the rotation around the line connecting the two points unspecified, and one additional coordinate of a third point (not collinear to the previous two) to fix the rotation. We often stipulate these coordinates are identical to the approximate coordinate values with which the estimation is started.

Gauge or S-transformations are necessary when comparing results of free adjustments following the procedure discussed in Sect. 4.6.2.3, p. 120, if the results of the two free adjustments refer to a different gauge, or when their gauge is not known. A classical situation of this type arises when the documentation of a software package does not provide this information. Then both results first need to be brought into the same coordinate system and the same gauge. Afterwards, all scene coordinates or all transformation parameters not defining the gauge can be compared using the test in (4.264), p. 121.

In the following, we will discuss how to impose gauge constraints to achieve an estimation result in a prespecified gauge, and how to impose gauge constraints onto the coordinates $\widehat{\boldsymbol{k}}$ in the case in which the parameters are reduced to the transformation parameters.

4.5.2 Gauge Constraints and Inner Precision

The result of a free adjustment represented by $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}\}$ is fixed up to a transformation $\mathcal{G}(\boldsymbol{g})$ depending on D parameters; for instance, the result of a bundle adjustment from calibrated images alone is unique up to a seven-parameter transformation. Both parameters, $\widehat{\boldsymbol{x}}$ and their covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$, depend on the chosen coordinate system. However, all entities $\underline{\boldsymbol{z}}$ which can be directly derived from the observations $\boldsymbol{l}$ via $\underline{\boldsymbol{z}} = \boldsymbol{z}(\underline{\boldsymbol{l}})$ are invariant w.r.t. the chosen coordinate system, namely their values $\boldsymbol{z}$ and their covariance matrices Σ_{zz}.

Invariants of a spatial similarity are (among others) angles or distance ratios between arbitrary point pairs. These are functions of the form of the geometric point configuration. The covariance matrix contains parts which represent the uncertainty of the form. It also contains parts which depend on the uncertainty of the specified coordinate system. We refer to that part of the geometric configuration which is independent of the coordinate system as the *inner geometry* of the configuration. If we restrict the specification of the coordinate system to a spatial motion, this would be the form of the configuration, since the form of an object is invariant to a motion. Correspondingly, we will refer to the *inner precision* as the uncertainty of the form, which is that part of the uncertainty of a free adjustment which is invariant to the gauge.

inner geometry and precision

[5] Provided the calibration of the used cameras is known.

For simplicity, we restrict the discussion to fixing the gauge by imposing constraints onto the coordinates only. However, the method can be transferred to fixing the gauge using the transformation parameters.

4.5.3 Imposing Gauge Constraints and Gauge Transformations

A simple way to fix the gauge would be to use the given approximate parameter values as observations for the parameters; this would eliminate the singularity.[6] However, the observations would influence the parameters and thus enforce some constraints on the geometric configuration even if they had low weights.

Therefore, it is better to fix the gauge, either by specifying a minimal set of parameters or by imposing adequate gauge constraints onto the estimated parameters. Though they differ in implementation, they are equivalent. We will discuss gauge constraints here, as they are easier to implement and more flexible in use.

The H constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \mathbf{0}$ on the estimated coordinates $\widehat{\boldsymbol{x}}$ are to be selected such that the extended normal equation matrix (4.163) in the Gauss–Markov model becomes regular. The $U \times H$ matrix $\mathsf{H} = \partial \boldsymbol{h}/\partial \boldsymbol{x}^\mathsf{T}$ must span the U-dimensional null space of the design matrix A, thus $\mathsf{AH} = \mathsf{0}$ (cf. (4.157), p. 100 and Koch (1999, Eq. (1.227)). This ensures that the form of the object is not changed by the constraints. The number of constraints depends on the geometry of the estimation problem. This type of estimation has already been discussed in Sect. 4.3.2, p. 101.

We now discuss how to choose appropriate constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = 0$ or, equivalently, an appropriate matrix H.

minimum trace solution of free adjustment

We can fix the gauge based on the approximate values for the parameters, specifically the coordinates $\boldsymbol{k}$, in a fully symmetric manner. For this we determine the parameters of the differential similarity transformation $\mathcal{S}$ between the unknown estimated coordinates $\widehat{\boldsymbol{k}}_i$ and the given approximate values $\boldsymbol{k}_i^a$ and apply $\mathcal{S}$ to all coordinates. The residuals $\widehat{\boldsymbol{k}} - \mathcal{S}(\boldsymbol{k}^a)$ are treated as the entities carrying the uncertainty of the coordinates ${}^c\widehat{\boldsymbol{k}}$ in the coordinate system c; thus, they have the desired covariance matrix, as they do not contain the gauge parameters,

$$ {}^c\Sigma_{\widehat{k}\widehat{k}} := \mathbb{D}\left(\widehat{\underline{\boldsymbol{k}}} - \mathcal{S}(\boldsymbol{k}^a)\right) . \tag{4.218}$$

It can be shown that this definition of the gauge of the covariance matrix of the parameters minimizes the trace.

nonlinear gauge constraints

More specifically, the nonlinear gauge constraint is the best similarity between the approximate and the unknown estimated coordinate parameters. Let the model for the differential similarity transformation be

$$ \widehat{\boldsymbol{k}}_i + \boldsymbol{v}_{k_i} = \lambda \mathsf{R} \boldsymbol{k}_i^a + \boldsymbol{T} , \qquad {}^c\mathsf{W}_{ii} = w_i \mathsf{I}_3 , \qquad \text{for all } i = 1, ..., I \tag{4.219}$$

with the rotation matrix R, the translation vector $\boldsymbol{T}$ and the scale parameter λ. The matrix $\mathrm{Diag}(\{\mathsf{W}_{ii}\})$ is the weight, of the point $\boldsymbol{k}_i$ for defining the gauge parameters. The linearized model reads

$$ \widehat{\Delta \boldsymbol{k}} + \boldsymbol{v}_k = \mathsf{H}[\Delta \boldsymbol{T}^\mathsf{T}, \Delta \boldsymbol{R}^\mathsf{T}, \Delta\lambda]^\mathsf{T} , \qquad {}^c\mathsf{W} = \mathrm{Diag}(\{w_i \mathsf{I}_3\}) \tag{4.220}$$

with the $3I \times 7$ Jacobian

[6] This is very similar to the regularization using Marquart–Levenberg iterations, where singularities in the normal equations due to the lack of information are eliminated by adding a k-fold unit matrix to the normal equation matrix, with k decreasing during the iterations. Lack of information may be caused by missing observations or by an insufficiently specified reference system.

$$H = \left[\begin{array}{ccc|ccc|c} 1&0&0&0&-Z_1&Y_1&X_1 \\ 0&1&0&Z_1&0&-X_1&Y_1 \\ 0&0&1&-Y_1&X_1&0&Z_1 \\ \hline .. & .. & .. & .. & .. & .. & .. \\ \hline 1&0&0&0&-Z_i&Y_i&X_i \\ 0&1&0&Z_i&0&-X_i&Y_i \\ 0&0&1&-Y_i&X_i&0&Z_i \\ \hline .. & .. & .. & .. & .. & .. & .. \\ \hline 1&0&0&0&-Z_I&Y_I&X_I \\ 0&1&0&Z_I&0&-X_I&Y_I \\ 0&0&1&-Y_I&X_I&0&Z_I \end{array}\right]^a \tag{4.221}$$

depending on the approximate values $\boldsymbol{k}_i^a = [X_i^a, Y_i^a, Z_i^a]^\mathsf{T}$ of the 3D coordinates. Here the unknown parameters now are a small rotation vector $\Delta\boldsymbol{R}$, a small translation vector $\Delta\boldsymbol{T}$ and a small scale change $\Delta\lambda$. We obtain the parameters from the best similarity transformation minimizing the weighted sum of the squared residuals $\sum_i w_i|\boldsymbol{v}_{k_i}|^2$ in (4.219), p. 111. The estimated residuals are

$$-\boldsymbol{v}_k = {}^c\mathsf{S}\,\widehat{\Delta\boldsymbol{k}} \tag{4.222}$$

S-matrix

with what is called the *S-matrix*

$${}^c\mathsf{S} = \mathsf{I} - \mathsf{H}(\mathsf{H}^\mathsf{T}{}^c\mathsf{W}\mathsf{H})^{-1}\mathsf{H}^\mathsf{T}{}^c\mathsf{W} \tag{4.223}$$

(cf. Baarda, 1973, and (4.62), p. 87). We now apply variance propagation, starting from the given covariance matrix $\Sigma_{\widehat{kk}}$ of the coordinates, and obtain the covariance matrix of the coordinates in the system c

singular gauge transformation, singular S-transformation

$${}^c\Sigma_{\widehat{kk}} = {}^c\mathsf{S}\,\Sigma_{\widehat{kk}}\,{}^c\mathsf{S}^\mathsf{T}\,. \tag{4.224}$$

This is the desired gauge transformation. Since the resulting covariance matrix is singular, it is called a *singular S-transformation*. It transforms the covariance matrix from a set of coordinates into a gauge, specified by the constraint matrix H and the weight matrix ${}^c\mathsf{W}$ without changing the inner geometry, i.e., the geometric relation between the coordinates, including their covariance structure. This is valid even when the gauge of the given coordinates is not known and the original covariance matrix is regular. Observe, the constraint matrix H is the null space of ${}^c\mathsf{S}$, thus ${}^c\mathsf{S}\mathsf{H} = \mathsf{0}$.

If a minimum number of parameters is chosen to fix the gauge, the $(U-H)\times(U-H)$ covariance matrix of the remaining parameters is regular and can be used for evaluation of the covariance matrix. For simplicity, let us assume the first H parameters are used to fix the gauge. Then $U\times U$ weight matrix ${}^c\mathsf{W} = \mathrm{Diag}([w_u])$ with $w_u = 1, u = 1, ..., H$, and 0 otherwise. Then the first H rows and columns of the covariance matrix ${}^c\Sigma_{\widehat{kk}}$ are zero. The remaining $(U-H)\times(U-H)$ matrix can be directly derived from the *regular gauge transformation* or *regular S-transformation*

regular gauge transformation, regular S-transformation

$${}^c\Sigma_{\widehat{kk},r} = {}^c\mathsf{S}_r\,\Sigma_{\widehat{kk}}\,{}^c\mathsf{S}_r^\mathsf{T}\,, \quad \text{with} \quad {}^c\mathsf{S}_r = [\mathsf{0}_{U-H\times H}\,,\,\mathsf{I}_{U-H}]\;{}^c\mathsf{S}\,. \tag{4.225}$$

It is the covariance matrix of those residuals

$$-\boldsymbol{v}_{k,r} = {}^c\mathsf{S}_r\widehat{\Delta\boldsymbol{k}}\,, \tag{4.226}$$

which are stochastic. Observe, ${}^c\mathsf{S}_r^\mathsf{T}$ is the nullspace of H^T and $[{}^c\mathsf{S}_r^\mathsf{T}, \mathsf{H}]$ has full rank.[7]

linear gauge constraints

We can also achieve estimates in the envisaged coordinate system by observing that due to $\mathsf{A}^\mathsf{T}\mathsf{W}\widehat{\boldsymbol{v}} = \mathbf{0}$, we can impose what are called the *gauge constraints* onto the coordinates,

[7] There is an intimate relation between the stochastic part of the residuals in (4.226) and what we call reduced coordinates for estimating homogeneous entities in Sect. 10, p. 359, cf. (10.22), p. 370 since both capture the estimable component of a constrained random vector, cf. Förstner (2016).

$$H^{\mathsf{T}}\,{}^{c}W\,\Delta\widehat{\boldsymbol{k}} = \mathbf{0}\,. \tag{4.227}$$

These gauge constraints can be added in all estimation problems where the gauge is not defined by the observations and needs to be specified.

The gauge constraints for the similarity transformation have an intuitive geometric meaning. They require

1. the centroid of the estimated coordinates to be identical to the centroid of the approximate values; using weights $w_i \in \{0, 1\}$ to be able to choose a subset of the coordinates, we can write the constraint as

$$\sum_{i=1}^{I} w_i(\widehat{\boldsymbol{k}}_i - \boldsymbol{k}_i^a) = \mathbf{0}\,, \tag{4.228}$$

2. the average rotation between all estimated scene points to the approximate scene points to be zero, again possibly using a weighting

$$\sum_{i=1}^{I} w_i(\widehat{\boldsymbol{k}}_i \times \boldsymbol{k}_i^a) = \mathbf{0}\,, \tag{4.229}$$

and

3. the average square distance of all scene points from their centroid and the average distance of all approximate points from their centroid to be the same,

$$\sum_{i=1}^{I} w_i(|\widehat{\boldsymbol{k}}_i|^2 - |\boldsymbol{k}_i^a|^2) = 0\,. \tag{4.230}$$

Each 3×3 weight matrix $W_i = w_i I_3$ indicates the weights of the individual coordinates of each point for fixing the gauge. If ${}^{c}W = I_{3I}$, all points are weighted equally and the centroid, the principal axes, and the average distance of the approximate coordinates fix the gauge.

Two estimates $\widehat{\boldsymbol{v}}$ for the residuals resulting from two different specifications of ${}^{c}W$ will coincide, since the constraints are orthogonal to the linearized observation equations, i.e., $AH = \mathbf{0}$. This guarantees that the normal equation system is regular and the constraints have no influence on the form of the point cloud.

However, the two covariance matrices $\Sigma_{\widehat{k}\widehat{k}}$ of the resulting coordinates depend on the gauge definition, and thus are different for different W. A comparison of covariance matrices can be performed only if they refer to the same gauge. However, the estimable quantities – in this example these are angles and distance ratios – can be compared directly, since their variances are independent of the choice of the coordinate system (Koch, 1999, Sect. 3.3.3).

The matrix H may be specialized if less than seven parameters are necessary to define the gauge. For example, when specifying the gauge for a 2D similarity transformation, we only need the first two rows for each point in the matrix H in (4.221), p. 112 and only columns 1, 2, 6, and 7 to specify translation, rotation and scale.

4.5.4 Gauge in a System Reduced to the Transformation Parameters

In many cases, the estimation is performed after reduction to the transformation parameters. This is the usual procedure, as in most practical applications the number of transformation parameters is much smaller than the number of coordinates.

Imposing the constraints when working on a reduced parameter set needs some care. It is straightforward if we reduce to that parameter set which the constraints address. We restrict ourselves to the case of reduction to the transformation parameters.

Here we discuss two conditions, namely when the gauge is defined by constraining the transformation parameters, and when the gauge is defined by constraining the coordinates. The first case is relevant if we want to enforce the same gauge without having access to the 3D points. This happens when comparing the result of two automatic bundle estimations which are based on different image measurements and thus on different scene points (cf. Dickscheid et al., 2008). The second case is the classical one, as already discussed above, but it is technically more complex.

Constrained Transformation Parameters. When fixing the transformation parameters using constraints $H_p^\mathsf{T}\widehat{\Delta\boldsymbol{p}} = \mathbf{0}$, the resulting normal equation system reads

$$\begin{bmatrix} N_{kk} & N_{kp} & 0 \\ N_{pk} & N_{pp} & H_p \\ 0 & H_p^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{k}} \\ \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \boldsymbol{n}_k \\ \boldsymbol{n}_p \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} . \tag{4.231}$$

The upper left 2×2 matrix has a rank deficiency of d in general. Its null space is

$$\mathsf{null} \begin{bmatrix} N_{kk} & N_{kp} \\ N_{pk} & N_{pp} \end{bmatrix} = \begin{bmatrix} 0 \\ H_p \end{bmatrix} ; \tag{4.232}$$

reduced normal equations for free adjustment

therefore, the complete normal equation matrix is regular. The reduced system reads

$$\begin{bmatrix} N_{pp} - N_{pk}N_{kk}^{-1}N_{kp} & H_p \\ H_p^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \overline{\boldsymbol{n}}_p \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.233}$$

with $\overline{\boldsymbol{n}}_p$ from (4.117), p. 94. The introduction of the constraints on the transformation parameters is straightforward. We will discuss the fixation of the gauge through the transformation parameters of a bundle adjustment, namely the special structure of $H_p^\mathsf{T}\Delta\boldsymbol{p} = \mathbf{0}$, in Part III.

Constrained Coordinates. If we have constraints $H_k^\mathsf{T}\widehat{\Delta\boldsymbol{k}} = \mathbf{0}$ on the scene points, we still can use the normal equation system reduced to the transformation parameters and obtain

$$\begin{bmatrix} N_{pp} - N_{pk}N_{kk}^{-1}N_{kp} & -N_{pk}N_{kk}^{-1}H_k \\ -H_k^\mathsf{T}N_{kk}^{-1}N_{kp} & -H_k^\mathsf{T}N_{kk}^{-1}H_k \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \boldsymbol{n}_p - N_{pk}N_{kk}^{-1}\boldsymbol{n}_k \\ -H_k^\mathsf{T}N_{kk}^{-1}\boldsymbol{n}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} . \tag{4.234}$$

Proof: When fixing the coordinate system with the coordinates of the scene points, with $H_k = H^c$ we have the normal equation system

$$\begin{bmatrix} N_{kk} & N_{kp} & H_k \\ N_{pk} & N_{pp} & 0 \\ H_k^\mathsf{T} & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{k}} \\ \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \widehat{\boldsymbol{h}}_k \\ \widehat{\boldsymbol{h}}_p \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} . \tag{4.235}$$

Solving for the coordinates $\widehat{\Delta\boldsymbol{k}}$ leads to $\widehat{\Delta\boldsymbol{k}} = N_{kk}^{-1}(\boldsymbol{n}_k - N_{kp}\widehat{\Delta\boldsymbol{p}} - H_k\boldsymbol{\lambda})$. The second and third equations in (4.235) are $N_{pk}N_{kk}^{-1}(\boldsymbol{n}_k - N_{kp}\widehat{\Delta\boldsymbol{p}} - H_k\boldsymbol{\lambda}) + N_{pp}\widehat{\Delta\boldsymbol{p}} - \boldsymbol{n}_p = \mathbf{0}$, or

$$\left[N_{pp} - N_{pk}N_{kk}^{-1}N_{kp} \mid -N_{pk}N_{kk}^{-1}H_k\right] \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - (\boldsymbol{n}_p - N_{pk}N_{kk}^{-1}\boldsymbol{n}_k) = \mathbf{0} , \tag{4.236}$$

and $H_k^\mathsf{T}(N_{kk}^{-1}(\boldsymbol{n}_k - N_{kp}\widehat{\Delta\boldsymbol{p}} - H_k\boldsymbol{\lambda})) = \mathbf{0}$, or

$$\left[-H_k^\mathsf{T}N_{kk}^{-1}N_{kp} \mid -H_k^\mathsf{T}N_{kk}^{-1}H_k\right] \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\lambda} \end{bmatrix} - (-H_k^\mathsf{T}N_{kk}^{-1}\boldsymbol{n}_k) = \mathbf{0} . \tag{4.237}$$

Collecting (4.236) and (4.237) finally leads to (4.234). ◇

4.6 Evaluation

4.6.1 Precision, Bias and Accuracy 116
4.6.2 Effect of Random Errors 117
4.6.3 Modelling Systematic and Gross Errors 122
4.6.4 Evaluation with Respect to Gross Errors 124
4.6.5 Evaluation with Respect to Systematic Errors 133
4.6.6 Effect of Errors in the Covariance Matrix 135
4.6.7 Model Selection 138
4.6.8 Checking the Implementation of the Estimation 139

The result of an estimation needs to be evaluated. Such an evaluation has two aspects:

1. Checking the adequateness of the mathematical model using diagnostic tools.
2. Checking the correctness of the implemented estimation procedure.

Diagnostics according to Huber (1991) aims at finding and identifying deviations from the model assumptions. This is in contrast to *robustness*, whose purpose is to have safeguards against deviations from the assumptions, which is the topic of Sect. 4.7, p. 141.

The estimation procedures rest on the assumption that the mathematical model reflects the observation process to a sufficient degree. Deviations may result from a simplified model or from unexpected errors in the observations. Diagnostic tools may indicate such deviations and identify their causes. Deviations may refer to the functional and the stochastical model, in the simplest case to the first and second moments. We will distinguish between internal diagnostics, which are solely based on the observations used in the estimation process, and external diagnostics, which use additional information.

Internal diagnostics includes the *testing* of certain hypotheses to identify certain model deviations, the evaluation of the *testability* of the parameter estimation with respect to certain hypotheses, i.e., the ability to identify possible causes by using such tests, and the*sensitivity* of the result, i.e., the effect of model errors which cannot be identified. In principle, internal diagnostics cannot differentiate between deficiencies in the mathematical model and errors in the observations.

Measures for internal diagnostics for regression problems are provided in classical data analysis packages, such as R or SPSS (Jacobi, 2005; Weinberg and Abramowitz, 2006). Here we derive more general tools which can be used also for analysing groups of observations and systematic errors and for planning purposes.

External diagnostics uses reference data, which – following the convention in remote sensing – often is called *ground truth*. External diagnostics aims at checking the validity of the predicted uncertainty of the estimates and the efficiency and robustness of the estimation. As the statistical properties of the ground truth are assumed to be known, external diagnostics allows a distinction between model deviations and errors in the observations. Though external information may be integrated into the estimation process and tested, for psychological reasons, users of an estimation procedure prefer to rely on their own reference data, which, by intent, are therefore often not made available to the estimation procedure.

The *correctness of the method* needs to be checked. Besides numerical checks using a toy problem or a set of simulated data verifying the correctness of the estimates we need to guarantee that the estimated parameters actually follow their theoretical distribution. This requires empirical testing based on simulated data. It especially addresses the unbiasedness of the estimates, i.e., the identity of the mean estimate and the true values, and the identity of the covariance matrix of the estimates and their theoretical covariance matrix.

4.6.1 Precision, Bias and Accuracy

We will distinguish between precision and accuracy, which are related by the bias. For explicitness, we refer to a sample $\boldsymbol{x}_n, n = 1, ..., N$, of 2D points.

1. *Precision* describes the deviation of repeated trials from their estimated mean. It is sometimes called *internal precision*. It is measured by the empirical covariance matrix,

internal precision

$$\widehat{\Sigma}_{xx}^{\text{int}} = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n - \widehat{\boldsymbol{x}})(\boldsymbol{x}_n - \widehat{\boldsymbol{x}})^{\mathsf{T}}\,, \tag{4.238}$$

using the estimated mean,

$$\widehat{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\,. \tag{4.239}$$

The *precision of the mean* is

$$\Sigma_{\widehat{x}\widehat{x}} = \frac{1}{N}\widehat{\Sigma}_{xx}^{\text{int}}\,. \tag{4.240}$$

2. The *bias* describes the deviation of the estimated mean from the true value. The bias is sometimes used to measure the *correctness* of the observation process. As in the preceding sections, we assume we have access to the true value $\widetilde{\boldsymbol{x}}$, e.g., by an observation process of significantly higher quality. Simulated data start from the true values. Thus the bias is given by

correctness

$$\boldsymbol{b}_x = \widehat{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}\,. \tag{4.241}$$

3. *Accuracy* describes the deviation of repeated trials from the true value. It is sometimes called *external precision*. It can be measured by the empirical covariance matrix provided the true value of the mean is taken as reference:

external precision

$$\widehat{\Sigma}_{xx}^{\text{ext}} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}})(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}) \tag{4.242}$$

The accuracy and the precision are related by

$$\widehat{\Sigma}_{xx}^{\text{ext}} = \frac{N-1}{N}\,\widehat{\Sigma}_{xx}^{\text{int}} + \boldsymbol{b}_x\boldsymbol{b}_x^{\mathsf{T}} \tag{4.243}$$

Exercise 4.13

using Steiner's theorem (cf. (2.94), p. 37). The *accuracy of the mean* is

$$\widehat{\Sigma}_{\widehat{x}\widehat{x}}^{\text{ext}} = \frac{1}{N}\widehat{\Sigma}_{xx}^{\text{int}} + \boldsymbol{b}_x\boldsymbol{b}_x^{\mathsf{T}}\,. \tag{4.244}$$

Therefore we may encounter four types of observation processes (cf. Fig. (4.8), p. 116):

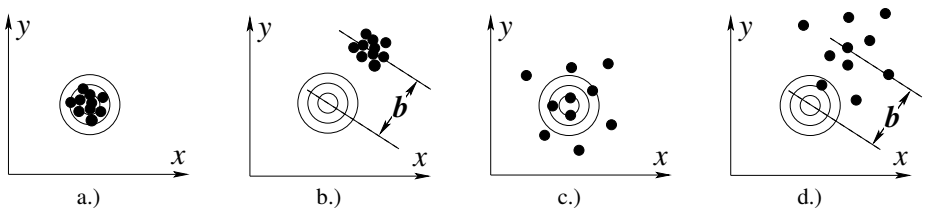


Fig. 4.8 Precision, accuracy and bias $\boldsymbol{b}$. Circles indicate iso-lines of the true distribution. a.) High precision, no bias, high accuracy. b.) high precision, large bias, low accuracy. c.) low precision, no bias, low accuracy. d.) low precision, high bias, low accuracy

- high precision and no bias. Here the internal and the external precision coincide. As can be seen from (4.244), increasing the sample size N makes it possible to increase the accuracy of the mean.
- high precision and bias. Here repeatability of the observation process is high, but due to systematic errors in the observation process the result is biased. The bias may dominate the precision; thus, the low accuracy is mainly caused by noncompensated systematic errors in the observation process.
- low precision and no bias. The observation process has low repeatability; thus, the accuracy of the mean will also be low. But as there is no bias, the accuracy of the mean can be improved by increasing the sample size. Observe, the low precision might result from a high uncertainty of the identification or definition of the entity (cf. Sect. 12.2.1, p. 490).
- low precision and bias. Here the accuracy is low and, due to the bias, cannot be increased by increasing the sample size.

We will now discuss how to evaluate precision and accuracy.

4.6.2 Effect of Random Errors

In this section we discuss methods for evaluating the precision of the result, i.e., the effect of random errors in the observations on the result.

Evaluating the result with respect to random errors may answer the following questions:

- *Theoretical precision:* Based on the planned observation scheme, what is the expected effect of random errors on the resultant parameters?
- *Empirical precision:* Based on the given observations, what is the effect of random errors on the resultant parameters?
- *Acceptability of the precision:* Given a prespecified tolerance for the precision of the parameters, is the achieved precision within the required bounds?
- *Empirical accuracy:* Based on additional observations or reference data, to what extent does the assumed mathematical model hold?
- *Checkability of the parameters:* Given reference values for the parameters, is it possible to check the identity of the estimated parameters with the reference values? We will discuss this notion in Sect. 4.6.5.2, p. 133, on evaluating additional parameters.

The theoretical precision describes the *expected* effect of random errors $\boldsymbol{e} = -\boldsymbol{v}$ of the observations $\boldsymbol{l}$ on the estimated parameters $\widehat{\boldsymbol{x}}$ based on the assumed observational process, represented by the function $\tilde{\boldsymbol{l}} = \boldsymbol{f}(\tilde{\boldsymbol{x}})$, and the assumed precision Σ_{ll} of the observations. The theoretical precision does not refer to actual observation, and can thus be used for planning purposes. It is a measure for internal diagnostics.

It can be described by the theoretical covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ of the estimated parameters, which can be derived by variance propagation. It is determined using (4.49) or (4.164).

Individual variances of a parameter x_u can be derived from

$$\sigma^2_{\widehat{x}_u} = (\Sigma_{\widehat{x}\widehat{x}})_{uu} \,. \tag{4.245}$$

The correlation between two parameters $\widehat{x}_u$ and $\widehat{x}_v$ is an indication of the degree of linear dependency and can be determined from

$$\rho_{\widehat{x}_u\widehat{x}_v} = \frac{\sigma_{\widehat{x}_u\widehat{x}_v}}{\sigma_{\widehat{x}_u}\sigma_{\widehat{x}_v}}. \tag{4.246}$$

High correlations, i.e., correlation coefficients close to -1 or $+1$, indicate relative instability: Some functions of the unknown parameters are determinable with much less precision

than others. Also, numerical instability may occur when solving the normal equations. As an example, the condition number of the 2×2 matrix,

$$N = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}, \tag{4.247}$$

is

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{1 + |\rho|}{1 - |\rho|}, \tag{4.248}$$

which is large if $|\rho|$ is close to 1, indicating that the matrix is badly conditioned.

The relevance of knowing the theoretical precision is the following: It can be shown, if the noise level is assumed correctly, that the real variances $\tilde{\Sigma}_{\widehat{x}\widehat{x}}$ of parameters derived by some unbiased estimation procedure are always larger than the theoretical variances $\Sigma_{\widehat{x}\widehat{x}}$,

$$\tilde{\Sigma}_{\widehat{x}\widehat{x}} \geq \Sigma_{\widehat{x}\widehat{x}}, \tag{4.249}$$

as the theoretical covariance matrix derived this way is the *Cramer–Rao bound* (cf. Rao, 1973, and Eq. (4.2.2.2), p. 86). The covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ supplies the basis for evaluating the empirical precision, which compares the empirically found covariance matrix ((4.238), p. 116) with the Cramer–Rao bound (cf. Sect. (4.6.2.2), p. 118).

4.6.2.1 Empirical Precision

The empirical precision indicates the effect of random errors of the observations on the estimated parameters; however, it only takes the estimated variance factor $\widehat{\sigma}_0^2$ into account. Thus the theoretical variances have to be multiplied with the estimated variance factor $\widehat{\sigma}_0^2$ in order to obtain the empirical variances. It also is a measure of internal diagonstics.

The *empirical covariance matrix* is

$$\widehat{\Sigma}_{\widehat{x}\widehat{x}} = \widehat{\sigma}_0^2 \Sigma_{\widehat{x}\widehat{x}}. \tag{4.250}$$

The *empirical standard deviation* of individual parameters therefore is

$$\widehat{\sigma}_{\widehat{x}_u} = \widehat{\sigma}_0 \, \sigma_{\widehat{x}_u} = \widehat{\sigma}_0 \sqrt{\Sigma^a_{\widehat{x}_u \widehat{x}_u}}. \tag{4.251}$$

Analogously, the *empirical standard deviation of the observations* is

$$\widehat{\sigma}_{l_n} = \widehat{\sigma}_0 \, \sigma^a_{l_n} = \widehat{\sigma}_0 \sqrt{\Sigma^a_{l_n l_n}}, \tag{4.252}$$

indicating the meaning of the estimated variance factor: The assumed standard deviations $\sigma^a_{l_n}$ of the observations need to be multiplied by the estimated factor $\widehat{\sigma}_0$.

Observe that the factor σ_0 is the same for all observations, without distinguishing between observations of different types, e.g., image coordinates and angles, or even between individual observations. This would require estimating an individual variance factor for each type of observation, as discussed in Sect. 4.2.4, p. 91 on variance component estimation.

4.6.2.2 Empirical Accuracy

The evaluation of the covariance matrices as discussed so far is only an internal evaluation relying on the internal redundancy of the observation process. Certain systematic errors, which may not have an influence on the residuals, but may deteriorate the estimated parameters, are not taken into account.

Evaluating the empirical accuracy of the estimated parameters therefore requires reference values – or ground truth – for the parameters, a subset of parameters, e.g., only the estimated coordinates, or some functions of the parameters, e.g., some distances between estimated points. This way we arrive at measures for external diagnostics.

Let us therefore assume that reference values $\boldsymbol{y}_r$ for some functions $\boldsymbol{y}(\widehat{\boldsymbol{x}})$ of the estimated parameters are available together with their covariance matrices $\Sigma_{y_r y_r}$. Then we can directly analyse the differences

$$\Delta\boldsymbol{y} = \boldsymbol{y}(\widehat{\boldsymbol{x}}) - \boldsymbol{y}_r \,. \tag{4.253}$$

In some practical cases, these differences may already be sufficient for an evaluation, e.g., by using their extrema and the histogram. This, of course, assumes that the reference values have at least the same accuracy as the estimated values.

In order to check whether the accuracy potential of the observations is fully exploited, we need to compare these differences with their standard deviations and determine the ratios

$$\underline{z}_i = \frac{\underline{\Delta y}_i}{\sigma_{\Delta y_i}} \sim \mathcal{N}(0,1) \tag{4.254}$$

with

$$\Sigma_{\Delta s \Delta s} = \Sigma_{yy} + \Sigma_{y_r y_r} \tag{4.255}$$

and

$$\sigma_{\Delta y_i} = \sqrt{\Sigma_{\Delta y_i \Delta y_i}} \,, \tag{4.256}$$

where the covariance matrix Σ_{yy} of the function values $\boldsymbol{y}$ needs to be determined from $\Sigma_{\widehat{x}\widehat{x}}$ by variance propagation.

These ratios are individual indicators and can be statistically tested individually. A histogram of these values is a valuable tool for evaluation. However, they are usually correlated, which is not taken into account when evaluating the individual ratios.

A combined test may be useful to check the complete set. It uses the Mahalanobis distance

$$\underline{X}^2 = (\underline{\boldsymbol{y}} - \underline{\boldsymbol{y}}_r)^\mathsf{T} (\Sigma_{yy} + \Sigma_{y_r y_r})^{-1} (\underline{\boldsymbol{y}} - \underline{\boldsymbol{y}}_r) \sim \chi^2_I \tag{4.257}$$

as a test statistic, which is χ^2_I-distributed with I degrees of freedom. This is the number of functions y_i if we can assume normally distributed reference data, so that $\underline{\boldsymbol{y}}_r \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{y_r}, \Sigma_{y_r y_r})$, and provided the mathematical model holds. If for the a priori specified significance level S the test statistic $\underline{X}^2$ exceeds the critical value, i.e., $X^2 > \chi^2_{I,S}$, there is a reason to reject the hypothesis, then we can conclude that the differences are not explainable by random deviations with the specified precision.

Determining the covariance matrix may be numerically complex. Therefore, often the following value is used

$$X^{*2} = (\boldsymbol{y} - \boldsymbol{y}_r)^\mathsf{T} (\boldsymbol{y} - \boldsymbol{y}_r) = I \cdot \mathrm{MSE}^2 \,, \tag{4.258}$$

which is I times the mean square error (MSE),

$$\mathrm{MSE} = \frac{1}{I} \sum_{i=1}^{I} (y_i - y_{r_i})^2 \,. \tag{4.259}$$

Often the *root mean square error* ($\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$) is reported, as it can be interpreted as an empirical standard deviation. Unfortunately the value $\underline{X}^{*2}$, and thus also the RMSE, has disadvantages:

root mean square error (RMSE)

- Its distribution is not known. Of course simulations could be used to derive it.
- It is a nonsufficient test statistic, as it does not use all information which in principle would be available. As a consequence of using a unit matrix instead of the correct

weight matrix, a value for X_a^* of one experiment compared to the value X_2^* of a second experiment cannot be used as an argument that the first experiment leads to better results than the second, see Fig. 4.9.

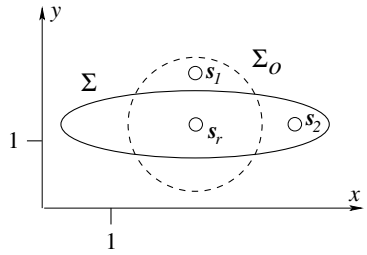


Fig. 4.9 Comparison of two experiments with suboptimal test statistic. Shown is the result $\boldsymbol{y}_i = [x;y]_i, i = 1,2$ of two experiments and the reference value $\boldsymbol{y}_r$ together with the covariance matrix $\boldsymbol{\Sigma} = \mathrm{Diag}([4,1/4])$ of the differences $\boldsymbol{y}_1 - \boldsymbol{y}_r = [3/4,0]^\mathsf{T}$, $\boldsymbol{y}_2 - \boldsymbol{y}_r = [0,3/2]^\mathsf{T}$ and the approximating covariance matrix $\boldsymbol{\Sigma}_o = \boldsymbol{I}_2$. The lengths $d_1 = X_1^* = |\boldsymbol{y}_1 - \boldsymbol{y}_r| = 3/4$ and $d_2 = X_2^* = |\boldsymbol{y}_2 - \boldsymbol{y}_r| = 3/2$ of the two difference vectors $\boldsymbol{y}_i - \boldsymbol{y}_{r,i=1,2}$ suggest the result $\boldsymbol{y}_1$ of the first experiment to be better than the result $\boldsymbol{y}_2$ of the second experiment. This corresponds to taking the covariance matrix $\boldsymbol{\Sigma}_o = \boldsymbol{I}$. However, taking into account the covariance matrix $\boldsymbol{\Sigma}$ of the difference vectors $\boldsymbol{y}_i - \boldsymbol{y}_r, i = 1,2$ clearly indicates the second experiment leads to a better result, since $X_1 = 3/2$ and $X_2 = 3/4$. The reason simply is, that the uncertainty of the difference in the x-direction is larger than in the y-direction. The weight matrix can be interpreted as the metric for measuring distances, see the discussion after (10.3), p. 361. Using a wrong weight matrix simply leads to false conclusions

Determining the empirical accuracy and reporting the RMSE is necessary, but however, not sufficient, for performing comparisons.

4.6.2.3 Acceptability of the Precision

Users may require a certain precision to be achieved by the observation process. We assume they specify this precision by some reference or *criterion matrix* $\boldsymbol{C} := \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{\mathrm{ref}}$. Then we need to compare the achieved theoretical covariance $\boldsymbol{\Sigma} := \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ with the a priori specified covariance matrix (Baarda, 1973). We first discuss the comparison and then methods for specification.

Comparing Two Covariance Matrices. The comparison of two covariance matrices is only meaningful if they refer to the same gauge. Otherwise, they need to be transformed into the same gauge, cf. Sect. 4.5, p. 108.

We may require that the achieved precision is consistently better than the reference or that it is similar to it on average.

Acceptability. The acceptability of the achieved precision may be based on the individual empirical standard deviations, requiring

$$\sigma_{\widehat{x}_u} \le \sigma_{\widehat{x}_u}^{\mathrm{ref}} \qquad \text{for a prespecified reference set of } u\,, \tag{4.260}$$

where the reference standard deviations are taken from the diagonal elements of $\boldsymbol{C}$. Obviously, the mutual dependencies of the parameters are not taken into account in this comparison.

Following Baarda (1973), we therefore could require any function (value) $y(\widehat{\boldsymbol{x}})$ of the parameters to be more precise when determined with the covariance matrix $\boldsymbol{\Sigma}$ than when determined with the criterion matrix $\boldsymbol{C}$. This can be formally written as

$$\sigma_y^{(\Sigma)} \le \sigma_y^{(C)}\,. \tag{4.261}$$

This is visualized in Fig. 4.10, left. With the Jacobian $\boldsymbol{e} = \partial f/\partial \widehat{\boldsymbol{x}}$, this leads to the requirement $\boldsymbol{e}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{e} \le \boldsymbol{e}^\mathsf{T}\boldsymbol{C}\boldsymbol{e}$ or to

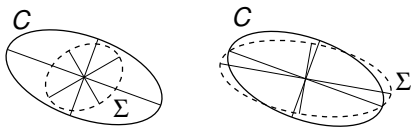


Fig. 4.10 Comparing a covariance matrix Σ with a criterion matrix C: the standard ellipsoid of the covariance matrix Σ is required to lie completely in the standard ellipsoid of the criterion matrix C, as in the left figure, or is required to be close to C, as in the right figure

$$r(\boldsymbol{e}) = \frac{\boldsymbol{e}^\mathsf{T}\Sigma\boldsymbol{e}}{\boldsymbol{e}^\mathsf{T}C\boldsymbol{e}} \leq 1\,. \tag{4.262}$$

Therefore the maximal eigenvalue $\lambda_{\max}$ of the generalized eigenvalue problem

$$\Sigma\boldsymbol{e} = \lambda C\boldsymbol{e} \tag{4.263}$$

needs to be less than 1,

$$\lambda_{\max}(C^{-1}\Sigma) \leq 1\,. \tag{4.264}$$

The analysis can also be performed on a sub-vector of $\widehat{\boldsymbol{x}}$. If the parameters are constrained, a regular S-transformation (4.225), p. 112 needs to be performed to arrive at two regular and comparable covariance matrices.

Distance of two covariance matrices. We can also determine the average distance of two $U \times U$ covariance matrices using the eigenvalues λ_u of $C^{-1}\Sigma$, which can be interpreted as the ratios of variances determined from Σ and C, respectively, and averaging their deviations from 1. By taking logarithms, we arrive at the average deviation of the ratios of the variances

$$d^2(\Sigma, C) = \frac{1}{U}\sum_{u=1}^{U} \log^2 \lambda_u(\Sigma C^{-1}) \geq 0 \tag{4.265}$$

from 1. This can be shown to be a metric between two covariance matrices (Förstner and Moonen, 1999). From this we can determine the average deviation $d/2$ of the ratio of the standard deviations from 1, a value of e.g., 0.1 indicating the standard deviations differ by 10% on average.

Whereas $\lambda_{\max}$ in (4.264) tells the worst case, the squared distance d^2 in (4.265) tells the average logarithm of the ratio of two variances determined with Σ instead of with C. However, if the role of Σ and C are exchanged, the maximum eigenvalue will be replaced by the minimum eigenvalue, but the distance d^2 remains invariant.

Specifying a Reference or Criterion Matrix. Specifying a criterion matrix can be done in several ways, depending on the context. Care has to be taken if we have to expect strong but acceptable – or unavoidable – correlations, e.g., when deriving a 3D point cloud from two images, where the distance between cameras and 3D points is quite large compared to the distances among the cameras.

Criterion matrix from a reference design. If the domain of possible designs is not very large, we may specify a representative reference design, i.e., the parameters $\tilde{\boldsymbol{x}}$, the functions $\boldsymbol{f}(\tilde{\boldsymbol{x}})$ and the uncertainty Σ_{ll}, derive the expected theoretic covariance matrix of the parameters $\Sigma_{\hat{x}\hat{x}}^{\text{ref}}$, and use it as the reference covariance matrix.

Criterion matrix of point clouds from covariance functions. If we want to specify the covariance matrix of the coordinates of a point cloud with given coordinates $\boldsymbol{X}_i, i = 1, ..., I$, we often require that they have homogeneous and isotropic uncertainty, i.e., all points have the same precision and rotating the coordinate system does not change the covariance matrix. Then the point cloud can be interpreted as a sample of a stochastic process where the covariance between the coordinates is dependent on their distance (cf. Sect. 2.8.2, p. 50). Specifically, we define the covariance function

homogeneous and isotropic uncertainty

$$C(d_{ij}) = C(d(\boldsymbol{X}_i, \boldsymbol{X}_j)) = \sigma^2 R(d_{ij}) \tag{4.266}$$

as a product of a given variance and a distance-dependent correlation function $R(d_{ij})$.

The reference covariance matrix of the coordinates then can be specified to be $\Sigma_{XX} = \sigma^2 R$, or, explicitly,

$$\Sigma_{XX} = \begin{bmatrix} \Sigma_{X_1X_1} & \dots & \Sigma_{X_1X_i} & \dots & \Sigma_{X_1X_I} \\ \dots & \dots & \dots & \dots & \dots \\ \Sigma_{X_iX_1} & \dots & \Sigma_{X_iX_i} & \dots & \Sigma_{X_iX_I} \\ \dots & \dots & \dots & \dots & \dots \\ \Sigma_{X_IX_1} & \dots & \Sigma_{X_IX_i} & \dots & \Sigma_{X_IX_I} \end{bmatrix} \tag{4.267}$$

$$= \sigma^2 \begin{bmatrix} R(0)\, I_3 & \dots & R(d_{1i})\, I_3 & \dots & R(d_{1I})\, I_3 \\ \dots & \dots & \dots & \dots & \dots \\ R(d_{i1})\, I_3 & \dots & R(0)\, I_3 & \dots & R(d_{iI})\, I_3 \\ \dots & \dots & \dots & \dots & \dots \\ R(d_{I1})\, I_3 & \dots & R(d_{Ii})\, I_3 & \dots & R(0)\, I_3 \end{bmatrix} . \tag{4.268}$$

If it is desirable to specify a covariance matrix which reflects an inhomogeneous situation, e.g., described by a spatially varying function $\sigma_i(\boldsymbol{X}_i)$ for the standard deviations and a distance-dependent correlation function $\rho(d_{ij})$, the covariance may be chosen using $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$ by

$$\Sigma_{XX} = \mathrm{Diag}\left([\sigma_i\left(\boldsymbol{X}_i\right)]\right)\ R\ \mathrm{Diag}\left([\sigma_i\left(\boldsymbol{X}_i\right)]\right) \tag{4.269}$$

with the same correlation matrix R as in (4.268), using some correlation function as in (2.192), p. 50ff.

4.6.3 Modelling Systematic and Gross Errors

This section provides tools for evaluating the result of an estimation with respect to gross and systematic errors and thus provide an essential diagnostic tool.

Systematic errors are deviations from the assumed model which are common to a large number or even all observations. They may be caused by the imperfectness of the calibration of the mensuration instrument or the lack of knowledge about the physical properties of the observation process. Gross errors are deviations in individual observational values or in small groups of observational values. Gross errors usually are significantly larger than the standard deviation σ_l and may be small (e.g., up to 20 σ_l), medium (up to 10% of the size of the observation), or large.

Evaluating the result with respect to systematic or gross errors requires answering the following questions (Förstner, 1987):

- *Testing:* How can systematic or gross errors be detected? How large are these errors? Such testing is necessary in order to prove that there is no reason to assume there are no such errors in the final result. The estimated size may be used to identify the error source, e.g., when two measurements are mistakenly exchanged by some automatic procedure for finding correspondences.
- *Detectability:* How large do systematic or gross errors have to be in order to be detectable? This type of information is useful for planning the design of a measurement procedure if it has the goal to analyse its physical properties with respect to certain systematic effects.
- *Sensitivity:* If the detection fails, how large is the effect of nondetectable and nondetected systematic or gross errors on the result? This type of information is also useful for planning the design of a measurement procedure if it has the goal that the resultant parameters are insensitive to nondetectable errors in the assumed mathematical model.

How easy it is to answer these questions depends on the method of modelling the systematic and gross errors. In principle, there is no way to distinguish between errors in the observations and errors in the model, but it is easiest to treat both gross and systematic errors as errors in the model. Modelling of systematic and gross errors therefore can be done in various ways: modifying the functional model, modifying the standard deviations, or modifying the distribution of the observations. Modifying the functional model is the simplest and most effective way of modelling systematic and gross errors, and has been extensively studied (cf. Baarda, 1967, 1968; Cook and Weisberg, 1982).

We start from a Gauss–Markov model as the null hypothesis. The alternative hypothesis assumes that the values of the observations deviate from this model. In the following we use the symbol ∇ to indicate *errors in the model*. As an example $\nabla \boldsymbol{l}$ is a – gross or systematic – error in the observational vector $\boldsymbol{l}$. The notation goes back to Baarda (1967) and should not be confused with the gradient operator. We assume that the deviation $\nabla \boldsymbol{l}$ depends on a parameter vector, leading to the hypotheses

the symbol ∇ indicates model errors

$$H_0 : \boldsymbol{l} + \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x}) \tag{4.270}$$

$$H_a : \boldsymbol{l} + \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{l} \tag{4.271}$$

with

$$\nabla \boldsymbol{l} = \boldsymbol{H}\, \nabla \boldsymbol{s}\,. \tag{4.272}$$

The effect $\nabla \boldsymbol{l}$ of the systematic or gross errors on the observations $\boldsymbol{l}$ depends linearly on the parameter vector $\nabla \boldsymbol{s}$ containing P elements, and is characterized by the influence matrix $\boldsymbol{H}$ of size $N \times P$. We formally cannot distinguish between gross and systematic errors. However, gross and systematic errors can be distinguished practically by the structure of the influence matrix: gross errors only influence one or a small set of observations, whereas systematic errors influence a large set of observations.

To simplify expressions, we do not follow this unifying framework but handle systematic errors in a slightly different way than gross errors.

Gross errors in several observations can be modelled with the null and the alternative hypotheses,

$$H_0 : \boldsymbol{l} + \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x})\,, \tag{4.273}$$

$$H_{a_i} : \boldsymbol{l}_i + \boldsymbol{v}_i = \boldsymbol{f}_i(\boldsymbol{x}) + \nabla \boldsymbol{l}_i\,. \tag{4.274}$$

We here refer to $i = 1, 2, ..., I$ groups of observations, where a group e.g., consists of the two or three coordinates of one 2D or 3D point; we assume that in the observational group $\boldsymbol{l}_i$, an error of size $\nabla \boldsymbol{l}_i$ is made and all other observations are free of gross errors. Thus we have the influence matrix of the ith group, containing d_i observations,

$$\boldsymbol{H}^{\mathsf{T}} = [\boldsymbol{0}, \ldots, \underbrace{\boldsymbol{I}_{d_i}}_{\boldsymbol{H}_i}, \ldots, \boldsymbol{0}]\,. \tag{4.275}$$

If only single observations are addressed, we use the index n, as before.

Systematic errors, more simply, are modelled as *additional parameters* s in an extended Gauss–Markov model. We start from the model

$$\boldsymbol{l} + \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s}), \tag{4.276}$$

including P additional parameters $\boldsymbol{s} = [s_p], p = 1, \ldots, P$, in the extended functional model $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s})$ for describing the systematic errors. We then have the null and the alternative hypotheses

$$H_0 : \boldsymbol{s} = \boldsymbol{0} \tag{4.277}$$

$$H_a : \boldsymbol{s} = \nabla \boldsymbol{s}\,. \tag{4.278}$$

The effect of systematic error $\nabla \boldsymbol{s}$ on the observations is

$$\nabla \boldsymbol{l} = H \nabla \boldsymbol{s} := \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s})}{\partial \boldsymbol{s}} \nabla \boldsymbol{s} \,. \tag{4.279}$$

Thus the null hypothesis assumes the additional parameter $\boldsymbol{s}$ has value $\mathbf{0}$ or the systematic error does not exist. We do not discuss the case where only a subset of the additional parameters is analysed, as this case can easily be derived; however, it leads to cumbersome expressions.

In the following we first discuss the evaluation w.r.t. outliers, starting from single outliers, then generalizing to groups of outliers. Analogously we discuss the evaluation w.r.t. systematic errors. In all cases we provide the estimated size of a possible error, the test statistic, a lower bound for the error to be detectable, and the sensitivity of the result w.r.t. a possible error and to nondetectable errors.

4.6.4 Evaluation with Respect to Gross Errors

The next two sections address the evaluation of the estimation w.r.t. gross and systematic errors. Starting from estimating the expected size of outliers and systematic errors, we derive hypothesis tests for identifying outliers and systematic errors, investigate the ability of the tests to detect such errors, and analyse the sensitivity of the estimates w.r.t. gross and systematic errors. The methods can be used for reliably analysing the results of estimates or for planning the observation process. We start the analysis w.r.t. single outliers and generalize to groups of outliers and systematic errors.

4.6.4.1 Evaluation with Respect to Single Gross Errors

Testing the observations with respect to gross errors can be based on the residuals $\widehat{\boldsymbol{v}}$. Due to the general relation (4.62), p. 87,

$$\nabla \widehat{\boldsymbol{v}} = -R \nabla \boldsymbol{l} \,, \qquad R = \Sigma_{\widehat{v}\widehat{v}} W_{ll} \,, \tag{4.280}$$

the effect of a gross error $\nabla \boldsymbol{l}_n$ on the corresponding residual $\widehat{\boldsymbol{v}}_n$ is given by

$$\nabla \widehat{v}_n = -r_n \nabla l_n \tag{4.281}$$

with the redundancy number $r_n = R_{nn}$ from (4.69), p. 88. We will use this relation in the following procedures.

estimated size of gross error

Estimated Size of a Single Gross Error. An estimate for the gross error in observation l_n together with its standard deviation is (cf. p. 128, Table 4.1, row 3, left)

$$\widehat{\nabla l}_n = -\frac{\widehat{v}_n}{r_n} \,, \qquad \sigma_{\widehat{\nabla} l_n} = \frac{\sigma_{l_n}}{\sqrt{r_n}} \,. \tag{4.282}$$

Equation (4.282) can be derived from (4.281) by setting $\nabla \widehat{v}_n = \widehat{v}_n$: Thus, if the observation is changed by $\widehat{\nabla} l_n$, the corresponding residual will be identical to the expected value, zero, of the residual.

A *leave-one-out test* would yield the same estimate for the gross error: it is the difference between the predicted value for the observation $\widehat{l}_n^{(n)}$ in an estimation without the nth observation and the observation l_n,

$$\widehat{\nabla l}_n = l_n - \widehat{l}_n^{(n)} \,. \tag{4.283}$$

Thus an expensive leave-one-out test is not necessary, but can be replaced by (4.282). Exercise 4.17

Proof: Equation (4.283) can be proven using the procedure for estimation in groups in Sect. 4.2.7.1, p. 96. We set $\boldsymbol{A}_2 = \boldsymbol{a}_n^\mathsf{T}$, $\boldsymbol{l}_2 = l_n$ and $\boldsymbol{\Sigma}_{22} = -\sigma_{l_n}^2$, thus by choosing the negative variance in the second step of the estimation we delete this observation. We use (4.144), p. 97 to determine the effect on the estimates when leaving the observation l_n out: $\nabla\widehat{\boldsymbol{x}} := \widehat{\boldsymbol{x}}^{(1)} - \widehat{\boldsymbol{x}}^{(2)} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^{(n)} = \boldsymbol{F}\widehat{\boldsymbol{v}}_2 = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\widehat{v}_n/(-\sigma_{l_n}^2 + \boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n) = -\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\widehat{v}_n/\sigma_{\widehat{v}_n}^2$ using $\sigma_{\widehat{v}_n}^2 = \sigma_{l_n}^2 - \boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n$ (cf. (4.64), p. 87).

With the effect $\nabla\widehat{l}_n^{(n)} = \boldsymbol{a}_n^\mathsf{T}\nabla\widehat{\boldsymbol{x}}$ of leaving the observation l_n out, we obtain $\widehat{\nabla}l_n = (l_n - \widehat{l}_n) + (\widehat{l}_n - \widehat{l}_n^{(n)}) = -\widehat{v}_n + \nabla\widehat{l}_n^{(n)} = -\widehat{v}_n - \boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\,\widehat{v}_n/\sigma_{\widehat{v}_n}^2 = -\widehat{v}_n - u_n/r_n\widehat{v}_n = -\widehat{v}_n(1 + u_n/r_n) = -\widehat{v}_n/r_n$. ◇

Test Statistic for a Single Residual. For single uncorrelated observations we obtain the optimal test statistic $\underline{z}_n$ for gross errors (Baarda, 1967, 1968) – cf. Table 4.1, p. 128, row 1, left – often called the *standardized residual*, possibly with opposite sign, *standardized residual*

$$\underline{z}_n = \frac{\widehat{\nabla}l_n}{\sigma_{\widehat{\nabla}l_n}} = \frac{-\widehat{v}_n}{\sigma_{\widehat{v}_n}} \sim \mathcal{N}(0,1)\,. \tag{4.284}$$

Observe, the test statistic for testing the residual and for testing the estimated size of the gross error coincide. If the null hypothesis holds, the test statistic $\underline{z}_n$ follows a standard normal distribution. If the test statistic exceeds the critical value, we have a reason to reject the null hypothesis in favour of the alternative hypothesis and thus assume that the observation contains an error.

As we usually do not know a priori which observation is erroneous, the test is performed for all observations or observational groups, see below. The observation or observational group where the absolute value of the test statistic is maximal and exceeds the critical value can then be assumed to be erroneous. This decision is not possible in the case of observations with residuals which are correlated by 100%, and thus for $\rho_{\widehat{v}_i\widehat{v}_j} = \pm 1$, indicating that the observations check each other. The decision is likely to be incorrect in the case of multiple gross errors, cf. Förstner (1983). The decision also is likely to fail if the gross error is close to its detectable limit; this phenomenon will be discussed below.

Detectability of Single Gross Errors. We now determine the minimum size $\nabla_0 l_n$ of a gross error ∇l_n in the observation l_n, which can be detected with the above-mentioned test (4.284). If the test is performed with a significance level α_0 and the gross error needs to be detected with a minimum probability $\beta > \beta_0$, we obtain the *lower bound* $\nabla_0 l_n$ for a *detectable gross error* (Baarda, 1967, 1968),

$$\nabla_0 l_n = \delta_0\,\sigma_{\widehat{\nabla}l_n} = \delta_0\frac{\sigma_{l_n}}{\sqrt{r_n}}\,, \tag{4.285}$$

with $\delta_0 = \delta_0(\alpha_0, \beta_0)$; cf. Sect. 3.2, p. 65 and Table 4.1, p. 128, row 3, right.

The lower bound for a detectable gross error depends on three factors:

- The structure of the performed test, especially the significance number α_0 and the required power β_0, is compressed in the lower bound δ_0 for the noncentrality parameter of the test statistic z_n.
- The measurement design via the redundancy number r_n. Obviously, the detectability of gross errors depends on the redundancy number and is higher with a larger redundancy number. We use the *detectability factor* *detectability factor*

  $$\mu_n' = \frac{1}{\sqrt{r_n}} \geq 1 \tag{4.286}$$

  for characterizing the measurement design w.r.t. the detectability of a gross error in observation l_n.
- The precision σ_{l_n} of the observation.

Therefore, we can also write the lower bound for detectable gross errors as

$$\nabla_0 l_n = \delta_0 \, \mu'_n \, \sigma_{l_n} \, . \tag{4.287}$$

suboptimal test

If the a priori covariance matrix of the observations is used instead of the covariance matrix of the residuals, i.e.,

$$z_n^* = -\widehat{v}_n / \sigma_{l_n} \, , \tag{4.288}$$

the test is suboptimal and thus less powerful. As a consequence, the size of detectable gross errors is

$$\nabla_0^* l_n = \delta_0 \, \sigma_{l_n} / r_n \, , \tag{4.289}$$

and thus increases by the factor $\mu'_n = \sqrt{1/r_n}$ compared to $\nabla_0 l_n$. The reason for the suboptimality simply is the following: The variance of $\underline{z}^*$ is r_n but the test erroneously assumes the variance to be 1.

Example 4.6.9: ***Detectability with optimal and suboptimal test statistic.*** For weak measurement designs with redundancy numbers $r_i = 0.1$ and using $\delta_0 \approx 4$, gross errors need to be larger than $\nabla_0 l_n \approx 12\sigma_n$ when using the optimal test statistic z_n. If we use the suboptimal test statistic z_n^*, gross errors need to be larger than $\nabla_0^* l_n \approx 40\sigma_n$ to be detectable. Thus, small outliers may not be detectable at all. ◇

Sensitivity of Estimated Parameters with Respect to Single Observations. We now want to investigate the effect of observations and nondetectable outliers on the result of the estimation. This refers to two different aspects:

1. One observation is not used in the estimation. Then the result will change. This type of leave-one-out test gives insight into the sensitivity of the result with respect to deleting outliers in the observations.
2. The observations have been tested for outliers. Then the test may fail to detect them, especially if they are below the aforementioned lower bound for detectable errors. Such errors will distort the result of the estimation without being noticeable.

Actually, both types of effects can be determined without explicitly repeating the estimation. They can also be determined for a subset of the parameters which are relevant for the specific application in mind.

We first investigate the effect of a single observation and a single nondetectable error on all parameters, generalize to groups of observations, and then specialize to subsets of estimated parameters.

The effect of arbitrary, gross or systematic, errors $\nabla \boldsymbol{l}$ in the observations on the result can be directly determined. In the case of the Gauss–Markov model, we have

$$\nabla \widehat{\boldsymbol{x}} = (\mathsf{A}^\mathsf{T} \mathsf{W}_{ll} \mathsf{A})^{-1} \mathsf{A}^\mathsf{T} \mathsf{W}_{ll} \nabla \boldsymbol{l} \, . \tag{4.290}$$

This expression is only recommended for small numbers U and N of unknowns and observations, respectively, as the matrix involved is of size $U \times N$. It assumes the observational errors $\nabla \boldsymbol{l}$ to be known, which is useful in simulation studies.

Therefore, we derive a scalar measure, namely an upper bound for the influence on an arbitrary function $y = \boldsymbol{d}^\mathsf{T} \widehat{\boldsymbol{x}}$ of the estimated parameters, e.g., distances or volumes derived from coordinates together with the standard deviation σ_y.

Empirical sensitivity with respect to one observation. The effect on a function $y(\widehat{\boldsymbol{x}})$ of leaving observation l_n out of the estimation is bounded by

$$|\nabla_n y(\widehat{\boldsymbol{x}})| \leq |z_n| \, \mu_n \, \sigma_y \tag{4.291}$$

sensitivity factor

(Baarda, 1967, 1968; Förstner, 1987) with the *sensitivity factor*

$$\mu_n = \sqrt{\frac{u_n}{r_n}} \geq 0 \, , \tag{4.292}$$

using u_n from (4.57), p. 87 and z_n from (4.284), p. 125. There is a close relation to *Cook's distance* $D_i = z_n^2\,\mu_n^2/U$ which measures the effect of leaving out one observation on the complete parameter vector $\widehat{\boldsymbol{x}}$ related to its covariance matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$, Cook (cf. 1977, Eq. (8)) and Cook and Weisberg (1982).

The sensitivity factor μ_n also measures the loss in precision when leaving out the observation l_n. We obtain

$$\mu_n^2 = \lambda_{\max}\left((\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{(n)} - \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}})\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{-1}\right) . \tag{4.293}$$

The covariance matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{(n)}$ of the parameters estimated without the observation is larger than that determined with all observations. The maximum relative increase is identical to the sensitivity factor.

Due to their possibly large influence on the parameters, observations with low redundancy numbers, below about 0.1, are also called *leverage points*, a term which originally referred to linear regression. Observations with $r_n = 0.1$, $u_n = 1 - r_n$, and thus $\mu_n = 3$, have an expected influence on a function $y(\widehat{\boldsymbol{x}})$, which is three times larger than its standard deviation σ_y. *leverage point*

Proof: Equation (4.291) can be determined from the procedure for estimation in groups in Sect. 4.2.7.1, p. 96, starting with the result of the proof of (4.283), p. 124, especially $\nabla\widehat{\boldsymbol{x}} = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\widehat{v}_n/\sigma_{\widehat{v}_n}^2$. The effect on y is $\nabla_n y = \boldsymbol{d}^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\widehat{v}_n/\sigma_{\widehat{v}_n}^2$. A bound for this effect can be derived using Cauchy's inequality; thus, we have $\boldsymbol{d}^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n \le \sqrt{\boldsymbol{d}^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{d}}\sqrt{\boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n} = \sigma_y\sigma_{\widehat{l}_n}$ and $u_n/r_n = \sigma_{\widehat{l}_n}^2/\sigma_{\widehat{v}_n}^2$ (cf. (4.72), p. 88), finally leading to $|\nabla_n y| \le \sigma_y\sigma_{\widehat{l}_n}\,|\widehat{v}_n|/\sigma_{\widehat{v}_n}^2 = |z_n|\,\mu_n\,\sigma_y$.

Equation (4.293) directly follows from (4.146), p. 97. We have $(\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{(n)} - \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}})\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{-1} = \boldsymbol{F}\boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{-1} = \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n\boldsymbol{a}_n^\mathsf{T}/\sigma_{\widehat{v}_n}^2$. Using (A.64), p. 773, the maximal eigenvalue is identical to that of $\boldsymbol{a}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{a}_n/\sigma_{\widehat{v}_n} = \mu_n^2$. ◇

Theoretical sensitivity with respect to one observation. The maximum effect of a nondetectable gross error in one of the observations on the result can be used to characterize its theoretical sensitivity with respect to gross errors. In a similar manner, the effect of nondetectable errors is bounded by (cf. Baarda, 1967, 1968; Förstner, 1987),

$$\nabla_{0,n}y(\widehat{\boldsymbol{x}}) \le \delta_0\;\mu_n\;\sigma_y\,. \tag{4.294}$$

The theoretical upper bound is proportional to the precision of y and increases with decreasing redundancy number r_n. The factor $\delta_0 = \delta_0(\beta_0, \alpha_0)$ again depends on the test characteristics. The value $\delta_0{\cdot}\mu_n{\cdot}\sigma_y$ measures the theoretical sensitivity of the result with respect to outliers and does not depend on actual observations; thus, it may be used for planning purposes.

Sensitivity of a Subset of Parameters. We now evaluate the sensitivity of the estimation if we are only interested in a subset of parameters. We assume that the parameters $\boldsymbol{x}$ are partitioned into coordinates $\boldsymbol{k}$ and transformation parameters $\boldsymbol{p}$: $\boldsymbol{x} = [\boldsymbol{k}^\mathsf{T}, \boldsymbol{p}^\mathsf{T}]^\mathsf{T}$, and we are only interested in the coordinates as discussed in Sect. 4.2.6, p. 94. We start from the model

$$\underline{\boldsymbol{l}} \sim \mathcal{N}(\overline{\boldsymbol{C}}\boldsymbol{k} + \boldsymbol{a}, \boldsymbol{\Sigma}_{ll}) \tag{4.295}$$

and follow the same line of thought as in the section before. The only difference lies in the design matrix. Therefore, we only need to change the definition of the sensitivity factor. For a single observation l_n, we obtain (cf. (4.292), p. 126)

$$\mu_{nk} = \sqrt{\frac{\overline{u}_{k_n}}{r_n}} \tag{4.296}$$

with the contribution (cf. (4.125), p. 95)

$$\overline{u}_{k_n} = \frac{\overline{\boldsymbol{c}}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{k}\widehat{k}}\overline{\boldsymbol{c}}_n}{\sigma_{\widehat{v}_n}^2} = \frac{\overline{\boldsymbol{c}}_n^\mathsf{T}\boldsymbol{\Sigma}_{\widehat{k}\widehat{k}}\overline{\boldsymbol{c}}_n\,w_n}{r_n} \tag{4.297}$$

of the nth observation to the unknown coordinates and the second index k of the sensitivity factor μ_{nk} indicates that it refers to the coordinates.

The measures for evaluation of the result of an estimation w.r.t. single gross errors are collected in Table 4.1.

Table 4.1 Diagnostic measures for the evaluation of a single outlier in the observation l_n. **Left column:** empirical values used for evaluating the result of an estimation procedure. **Right column:** theoretical values to be used for planning purposes. The second expression for the estimated size $\widehat{\nabla l}_n$ in row 3, left can easily be derived from the corresponding lower bound in row 3, right.

	Empirical	Theoretical
1	Test statistic $z_n = \frac{-\widehat{v}_n}{\sigma_{\widehat{v}_n}} =: \delta_n$ (4.284), p. 125	Standardized distance between H_0 and H_{a_n} $\delta_0 := \nabla_0 z_n$ (3.18), p. 66
2	Detectability factor $\mu'_n = \sqrt{\frac{1}{r_n}} = \frac{\sigma_{\widehat{\nabla} l_n}}{\sigma_{l_n}}$ (4.306), p. 129	
3	Estimated size of error $\widehat{\nabla} l_n = \frac{-\widehat{v}_n}{r_n} = z_n\, \mu'_n\, \sigma_{l_n}$ (4.282), p. 124	Lower bound for detectable error $\nabla_0 l_n = \delta_0\, \mu'_n\, \sigma_{l_n}$ (4.287), p. 126
4	Sensitivity factor w.r.t. coordinates $\mu_{nk} = \sqrt{\frac{\overline{u}_{k_n}}{r_n}}$ (4.296), p. 127	
5	Actual influence of observation l_n $\lvert\nabla_n y(\widehat{\boldsymbol{k}})\rvert \le \lvert z_n\rvert\, \mu_{nk}\, \sigma_y$ (4.291), p. 126	Theoretical influence of undetectable outlier $\lvert\nabla_{0n} y(\widehat{\boldsymbol{k}})\rvert \le \delta_0\, \mu_{nk}\, \sigma_y$ (4.294), p. 127

4.6.4.2 Evaluation with Respect to a Group of Gross Errors

The relations easily can be generalized to groups of observations which are mutually independent. Groups are indicated by the index i.

Estimated Size of a Group i of Gross Errors. The effect of gross errors $\nabla \boldsymbol{l}_i$ on the corresponding residual $\widehat{\boldsymbol{v}}_i$ is given by

$$\nabla \widehat{\boldsymbol{v}}_i = -R_{ii} \nabla \boldsymbol{l}_i \tag{4.298}$$

with the diagonal block of the redundancy matrix R,

$$R_{ii} = \Sigma_{\widehat{v}_i \widehat{v}_i} W_{l_i l_i} \quad \text{with} \quad 0 \le \lambda(R_{ii}) \le 1\,, \tag{4.299}$$

Exercise 4.15

referring to the ith observational group, a relation which only holds if the observational group $\boldsymbol{l}_i$ is uncorrelated to all other observations. In the case of one observation l_n only, R_{ii} reduces to the redundancy number r_n. An estimate for the gross errors in observation group i together with its covariance matrix is (cf. p. 131, Table 4.2, row 3, left)

$$\widehat{\nabla \boldsymbol{l}}_i = -R_{ii}^{-1}\, \widehat{\boldsymbol{v}}_i\,, \qquad \Sigma_{\widehat{\nabla} l_i \widehat{\nabla} l_i} = \Sigma_{l_i l_i} \Sigma^{-1}_{\widehat{v}_i \widehat{v}_i} \Sigma_{l_i l_i}\,, \tag{4.300}$$

provided the matrix $\Sigma_{\widehat{v}_i \widehat{v}_i}$ is regular, as then R_{ii} is also regular. A leave-one-out test would yield the same estimate for the gross error: it is the difference between the predicted value

for the observation in an estimation without the ith observation group and the observations

$$\widehat{\nabla \boldsymbol{l}}_i = \widehat{\boldsymbol{l}}_i^{(i)} - \boldsymbol{l}_i \,. \tag{4.301}$$

Thus, an expensive leave-one-out test is not necessary and can be replaced by (4.300). Exercise 4.17

Test Statistic for a Group of Residuals. For uncorrelated observational groups, we obtain the optimal test statistic $\underline{X}_i^2$ with its distribution for a given null hypothesis (cf. Table 4.2, p. 131, row 1, left),

$$\underline{X}_i^2 = \underline{\widehat{\boldsymbol{v}}}_i^{\mathsf{T}} \, \Sigma_{\underline{\widehat{v}}_i \underline{\widehat{v}}_i}^{-1} \, \underline{\widehat{\boldsymbol{v}}}_i \sim \chi^2(d_i) \,, \tag{4.302}$$

where the size, d_i, of the observational group is the degrees of freedom of the χ^2 distribution, and $\Sigma_{\underline{\widehat{v}}_i \underline{\widehat{v}}_i}$ is the covariance matrix of the residuals of the ith group, which is assumed to be regular (cf. Stefanovic, 1978). The test statistic can also be derived from the estimated size of the gross errors as

$$\widehat{\nabla \boldsymbol{l}}_i^{\mathsf{T}} \, \Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i}^{-1} \, \widehat{\nabla \boldsymbol{l}}_i = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \, \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \, \widehat{\boldsymbol{v}}_i \,. \tag{4.303}$$

Detectability of a Group of Gross Errors. In a similar manner, we can analyse groups $\boldsymbol{l}_i$ of d_i observations with respect to the detectability of gross errors in that group. A gross error is least detectable if it is parallel to the eigenvector of $\Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i}$ in (4.300), which corresponds to its largest eigenvalue. The minimum size of a detectable error in a group $\boldsymbol{l}_i$ of observations results from

$$|\nabla_0 \boldsymbol{l}_i|^2 \le \delta_0^2(\alpha_0, \beta_0, d_i) \, \lambda_{\max} \left(\Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i} \right) \,. \tag{4.304}$$

The lower bound $\delta_0(\alpha_0, \beta_0, d_i)$ refers to the noncentrality parameter of the noncentral χ^2 distribution $P_{\chi'^2}(x; d, \delta^2)$ of the test statistic $\underline{X}_i^2$ in (4.302), provided the alternative hypothesis is true, cf. (3.30), p. 68.

For making easier to see the different causes of detectable gross errors, in the following we use the *detectability factor* *detectability factor*

$$\mu_i'^{\,2} = \lambda_{\max}(R_{ii}^{-1}) = \max \lambda \left(\Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i} \Sigma_{l_i l_i}^{-1} \right) \ge 1 \,, \tag{4.305}$$

(cf. Table 4.2, p. 131, row 2), which in the case of one observation l_n reduces to (cf. Table 4.1, p. 128, row 2)

$$\mu_n' = \sqrt{\frac{1}{r_n}} = \frac{\sigma_{\widehat{\nabla l}_n}}{\sigma_{l_n}} \,. \tag{4.306}$$

Since $\lambda_{\max}(AB) \le \lambda_{\max}(A) \lambda_{\max}(B)$, and since for statistically independent groups of observations we have $\Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i} = R_{ii}^{-1} \Sigma_{l_i l_i}$ (4.300), we obtain the lower bound for a detectable error in group $\boldsymbol{l}_i$ (cf. Table 4.2, row 3 right),

$$|\nabla_0 \boldsymbol{l}_i| \le \delta_0(\alpha_0, \beta_0, d_i) \, \mu_i' \, \sqrt{\lambda_{\max} \left(\Sigma_{l_i l_i} \right)} \,. \tag{4.307}$$

This bound is less tight than the one in (4.304) if the eigenvalues of $\Sigma_{l_i l_i}$ are different. For single observations it specializes to (4.287).

Obviously, the minimum size $|\nabla_0 \boldsymbol{l}_i|$ for detectable errors depends on

1. the precision of the observations, namely on $\Sigma_{l_i l_i}$ or σ_{l_i},
2. the strength of the design of the mensuration process pooled in the detectability factor μ_i' or μ_n', and
3. the specifications for the statistical test, namely the significance number α_0 and the required minimum power β_0, pooled in the lower bound δ_0 for the noncentrality parameter of the statistical test.

Remark: The upper bound in (4.307) may be not representative of all gross errors occurring in a group $\boldsymbol{l}_i$ if the eigenvalues of the matrix $\Sigma_{\widehat{\nabla l}_i \widehat{\nabla l}_i}$ differ greatly. A more detailed analysis provides the size of a detectable gross error in a group as a function of the direction $\nabla \boldsymbol{l}_i / |\nabla \boldsymbol{l}_i|$ of the expected outlier, which leads to easily interpretable detectability ellipses/ellipsoids, cf. Förstner (1983). ◇

Sensitivity w.r.t. a Group of Gross Errors. Again we distinguish between empirical and theoretical sensitivities.

Empirical Sensitivity with Respect to a Group of Gross Errors. The measures can be generalized to multi-dimensional tests (cf. Förstner, 2001). The influence of the estimates on a function $y(\widehat{\boldsymbol{x}}) = \boldsymbol{d}^\mathsf{T} \widehat{\boldsymbol{x}}$ when leaving out a group $\boldsymbol{l}_i$ of observations is bounded by the relation

$$|\nabla_i y(\widehat{\boldsymbol{x}})| \leq X_i \, \mu_i \, \sigma_y \tag{4.308}$$

sensitivity factor

with the test statistic X_i from (4.302), the sensitivity factor

$$\mu_i^2 = \lambda_{\max} \left((\Sigma_{\widehat{x}\widehat{x}}^{(i)} - \Sigma_{\widehat{x}\widehat{x}}) \Sigma_{\widehat{x}\widehat{x}}^{-1} \right) \geq 0 \,, \tag{4.309}$$

and the standard deviation σ_y of the function y. The sensitivity factor can be determined from

$$\mu_i^2 = \lambda_{\max} \left(\Sigma_{\widehat{l}_i \widehat{l}_i} \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \right) \tag{4.310}$$

for each observation group $\boldsymbol{l}_i$ without needing to repeat the estimation.

Proof: We again perform a stepwise estimation, cf. Sect. 4.2.7.2, p. 96, leaving out the observational group in the second step. Thus, we set the $d_i \times U$ matrix $\boldsymbol{A}_2 = \boldsymbol{A}_i^\mathsf{T}$, the observational group $\boldsymbol{l}_i$ and the covariance matrix $\Sigma_{22} = -\Sigma_{l_i l_i}$. The change of the parameters is $\nabla \widehat{\boldsymbol{x}} = \widehat{\boldsymbol{x}}^{(1)} - \widehat{\boldsymbol{x}}^{(2)} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^{(i)} = \boldsymbol{F} \widehat{\boldsymbol{v}}_2 = \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i$. The effect $\nabla_i y = \boldsymbol{d}^\mathsf{T} \nabla \widehat{\boldsymbol{x}} = \boldsymbol{d}^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i$ is bounded by

$$\begin{aligned}
\left| \boldsymbol{d}^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i \right|^2 &\leq \boldsymbol{d}^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{d} \cdot \widehat{\boldsymbol{v}}_i^\mathsf{T} \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \boldsymbol{A}_i^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i && (4.311) \\
&\leq \sigma_y^2 \cdot \widehat{\boldsymbol{v}}_i^\mathsf{T} \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i \cdot \lambda_{\max} \left(\boldsymbol{A}_i^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \right) && (4.312) \\
&= X_i^2 \cdot \mu_i^2 \cdot \sigma_y^2 \,, && (4.313)
\end{aligned}$$

which proves (4.308) using (4.310). The identity of (4.310) and (4.309) results from (4.146), p. 97, as $\mu_i^2 = \lambda_{\max}(\boldsymbol{F} \boldsymbol{A}_i^\mathsf{T}) = \lambda_{\max}(\Sigma_{\widehat{x}\widehat{x}} \boldsymbol{A}_i \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \boldsymbol{A}_i^\mathsf{T}) = \lambda_{\max}(\Sigma_{\widehat{l}_i \widehat{l}_i} \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1})$, using (A.64), p. 773. ◇

Theoretical sensitivity with respect to a group of observations. In analogy to the reasoning about detectability, we obtain a bound for the maximum effect of a set of nondetectable outliers on the result,

$$|\nabla_{0i} y(\widehat{\boldsymbol{x}})| \leq \delta_0(\alpha_0, \beta_0, d_i) \, \mu_i \, \sigma_y \,. \tag{4.314}$$

where the noncentrality parameter is the same as in (4.304), p. 129. This is a measure for the theoretical or expected sensitivity of the result w.r.t. a group of outliers.

Sensitivity of a Subset of Parameters. For groups of observations, we obtain the (square of the) sensitivity factor w.r.t. the coordinates

$$\mu_{ik}^2 = \lambda_{\max} \left(\overline{\boldsymbol{C}}_i^\mathsf{T} \Sigma_{\widehat{k}\widehat{k}} \overline{\boldsymbol{C}}_i \, \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \right) = \lambda_{\max} \left((\Sigma_{\widehat{k}\widehat{k}}^{(i)} - \Sigma_{\widehat{k}\widehat{k}}) \Sigma_{\widehat{k}\widehat{k}}^{-1} \right) . \tag{4.315}$$

using the model (4.295), p. 127

Proof: In the reduced model we have $(\Sigma_{\widehat{k}\widehat{k}}^{(i)} - \Sigma_{\widehat{k}\widehat{k}}) \Sigma_{\widehat{k}\widehat{k}}^{-1} = \boldsymbol{F} \overline{\boldsymbol{C}}_i^\mathsf{T} \Sigma_{kk} \Sigma_{\widehat{k}\widehat{k}}^{-1} = \Sigma_{\widehat{k}\widehat{k}}^{-1} \overline{\boldsymbol{C}}_i \Sigma_{\widehat{v}\widehat{v}}^{-1} \overline{\boldsymbol{C}}_i^\mathsf{T} = \Sigma_{\widehat{k}\widehat{k}}^{-1} \overline{\boldsymbol{C}}_i^\mathsf{T} \Sigma_{\widehat{v}\widehat{v}}^{-1} \overline{\boldsymbol{C}}_i$, which due to (A.64), p. 773 has the same eigenvalues as $\overline{\boldsymbol{C}}_i^\mathsf{T} \Sigma_{\widehat{k}\widehat{k}}^{-1} \overline{\boldsymbol{C}}_i \Sigma_{\widehat{v}\widehat{v}}^{-1}$. ◇

Remark: The upper bounds in (4.314) may not be representative of the effect of arbitrary nondetectable errors on the parameters. A more detailed analysis (Förstner, 1983) allows the derivation of easily interpretable sensitivity ellipsoids as a function of the direction $\nabla \boldsymbol{l}_i / |\nabla \boldsymbol{l}_i|$ of the expected gross errors. ◇

The measures for evaluating the result of an estimation w.r.t. outliers in groups of observations are summarized in Table 4.2.

Table 4.2 Diagnostic measures for the evaluation w.r.t. a group i of d_i gross errors. **Left column**: empirical values used for evaluating the result of an estimation procedure. **Right column**: theoretical values to be used for planning purposes. The sensitivity values refer to a subgroup $\boldsymbol{k}$ of the parameters $\boldsymbol{x}$

	Empirical	Theoretical
1	Test statistic $X_i^2 = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i =: \delta_i^2$ (4.302), p. 129	Standardized distance between H_0 and H_{a_i} $\delta_0(\alpha_0, \beta_0, d_i) := \nabla_0 X_i$ (3.30), p. 68
2	Detectability factor $\mu_i'^{\,2} = \max \lambda(\boldsymbol{R}_{ii}^{-1}) = \lambda_{\max} \left(\boldsymbol{\Sigma}_{\widehat{\nabla} l_i \widehat{\nabla} l_i} \boldsymbol{\Sigma}_{l_i l_i}^{-1} \right)$ (4.305), p. 129	
3	Estimated size of error $\widehat{\nabla \boldsymbol{l}}_i = -\boldsymbol{R}_{ii}^{-1} \widehat{\boldsymbol{v}}_i$ (4.300), p. 128 $\lvert \widehat{\nabla \boldsymbol{l}}_i \rvert \le X_i \sqrt{\lambda_{\max} \left(\boldsymbol{\Sigma}_{\widehat{\nabla} l_i \widehat{\nabla} l_i} \right)}$	Lower bound for detectable error $\lvert \nabla_0 \boldsymbol{l}_i \rvert = \delta_0(\alpha_0, \beta_0, d_i) \sqrt{\lambda_{\max} \left(\boldsymbol{\Sigma}_{\widehat{\nabla} l_i \widehat{\nabla} l_i} \right)}$ (4.304), p. 129
4	Sensitivity factor w.r.t. coordinates $\mu_{ik}^2 = \lambda_{\max} \left(\overline{\boldsymbol{C}}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{k}\widehat{k}} \overline{\boldsymbol{C}}_i \boldsymbol{\Sigma}_{\widehat{v}_i \widehat{v}_i}^{-1} \right)$ (4.315), p. 130	
5	Actual influence of observation $\lvert \nabla_i y(\widehat{\boldsymbol{k}}) \rvert \le X_i \, \mu_{ik} \, \sigma_y$ (4.308), p. 130	Maximal influence of undetectable outliers $\lvert \nabla_{0i} y(\widehat{\boldsymbol{k}}) \rvert \le \delta_0(\alpha_0, \beta_0, d_i) \, \mu_{ik} \, \sigma_y$ (4.314), p. 130

Together with the measures for the evaluation w.r.t. outliers in single observations, we can summarize the result of this section as follows:

- The test statistic leads to optimal tests for single observations or single groups of observations, thus referring to single alternative hypotheses. They are usually applied to all observations or observational groups. The test statistic for different hypotheses is generally correlated. This leads to smearing effects, as one erroneous observation may influence the test statistic of several other observations.
  The noncentrality parameter δ_0 refers to the test statistic z_n or X_i. It depends on the prespecified significance level $S = 1 - \alpha_0$, the required minimum power β_0 of the test, and in the case of observational groups on their size d_i. It is useful to fix these probabilities for all applications in order to yield comparable results across projects.
  The tests discussed so far rely on the given standard deviation of the observations, and thus assume the variance factor to be $\sigma_0 = 1$. If the tests use the estimated variance factor $\widehat{\sigma}_0$, we will have a different test, namely a t-test with $R - d_i$ degrees of freedom or a Fisher test with d_i and $R - d_i$ degrees of freedom. This changes both the critical values and the noncentrality parameter δ_0. The difference is negligible for large redundancies, say beyond $R > 100$.
- The detectability factors μ_n' or μ_i' indicate the largest influence of the observational design on the estimated size and the lower bound for detectable errors. The detectability factors are also the standard deviations of the estimated size of the errors, derived from the residuals, related to the standard deviations of the observations. Thus the detectability factor is relevant for the first and second moments of the estimated size of the gross errors.
- The size of a possible gross error in a single observation or an observational group can be determined from the residuals using the main diagonals or the diagonal blocks of the redundancy matrix $\boldsymbol{R}$. Therefore, we do not need the off-diagonals of this matrix. The redundancy numbers r_n or the diagonal $d_i \times d_i$ blocks $\boldsymbol{R}_{ii}$ can be determined efficiently if the normal equation matrix is sparse.

The *lower bound for the size of gross detectable errors* using the test specified by α with a minimum power of β_0 in an intuitive manner depends on the noncentrality parameter δ_0, the detectability factor, and the assumed standard deviation of the observations.

- The *sensitivity factors* μ_n or μ_i indicate the maximum influence of the observational design on the estimated parameters. At the same time they measure the loss in precision when an observation or a group of observations is left out. Again, the sensitivity factors are relevant for the first and second moments of the effects of gross errors. The sensitivity factors can be related to a subgroup of unknown parameters, e.g., when we are only interested in the coordinates and not in the transformation or even calibration parameters.
- The effect of leaving one observation or one observational group out of the estimation can be determined. An upper bound for this effect $\nabla_n y$ can be determined and given for an arbitrary function $y(\widehat{\boldsymbol{x}})$ or $y(\widehat{\boldsymbol{k}})$ of the parameters or a subset of the parameters, such as the coordinates. It intuitively depends on the size of the test statistic, the sensitivity factor, and the standard deviation of the function.

  If a statistical test is performed, gross errors may stay undetected and distort the result. The maximum effect on the estimated parameters or a subset thereof can be given and depends on the noncentrality parameter δ_0, instead of on the test statistic.

Example 4.6.10: ***Sensitivity of planar similarity transformation w.r.t. the configuration of points.*** Fig. 4.11 shows the result of a sensitivity analysis.

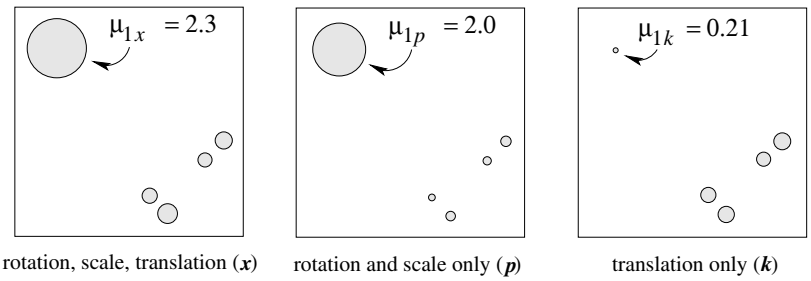


Fig. 4.11 Sensitivity of similarity transformation with five points. Shown are their sensitivity factors, which are the radii of the circles. As an example for interpreting the value μ_{1x} in the left figure: If the similarity transformation is performed with all five points, a statistical test is to identify outliers, a nondetected outlier of up to μ_{1x} in point 1 (top left) may deteriorate the estimated parameters $\widehat{\boldsymbol{x}}$ up to the $\delta_0\,\mu_{ix} \approx 8$-fold standard deviation of the estimated parameters. See text for explanation

Given are five corresponding points in two coordinate systems, with the same standard deviation of the coordinates. They are assumed to be related by a planar similarity, with four parameters $\boldsymbol{x}$, namely two translations, a rotation, and a scale. The sensitivity of the estimated parameters w.r.t. possible outliers in the measured coordinates is given by the sensitivity factors $\mu_{ix} := \mu_i$ from (4.314), the second index x indicating that the effect on all parameters is addressed, cf. Fig. 4.11, left, where the sensitivity factors are the radii of the shown circles. The isolated point in the upper left corner has the major influence on the result, as is to be expected, as rotation and scale are mainly determined by its coordinates. If we are only interested in the effect on either translation or rotation and scale, we partition the four unknown parameters $\boldsymbol{x}$ into two groups, say the 2-vector $\boldsymbol{k}$ representing the translation and the 2-vector $\boldsymbol{p}$ representing rotation and scale. We then obtain the influence factors μ_{ip} on rotation and scale, determined by (4.315), and μ_{ik} on translation, determined by the corresponding expression exchanging the role of $\boldsymbol{k}$ and $\boldsymbol{p}$, shown in the middle and the right of the figure. Observe, the four clustered points at the lower right do not have a high influence on the rotation and scale, as they are closer to the centroid of the five points. The right figure shows that outliers in the isolated point (top left corner) have nearly no influence on the translation, as they mainly influence rotation and scale. The example clearly demonstrates the usefulness of a detailed sensitivity analysis, see also the analysis of the 3D similarity transformation in Sect. 10.5.4.3, p. 408. ◇

4.6.5 Evaluation with Respect to Systematic Errors

We now transfer the evaluation of gross errors to the evaluation of systematic errors, which are treated as errors in the functional model. We start with the testing procedure.

4.6.5.1 The Test Statistic

We want to test whether additional parameters of the functional model (4.276), p. 123 significantly deviate from zero. This implies that parameter values 0 mean that there are no systematic errors. The test can be based on the estimated parameters. It directly gives an estimate $\widehat{\boldsymbol{s}} = [\widehat{s}_p], p = 1, \ldots, P$, for the size of the systematic errors,

$$\{\widehat{\boldsymbol{s}}, \boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}\} \,, \tag{4.316}$$

where the covariance matrix is provided by the estimation process.

The test statistic for testing the null hypothesis H_0 versus the alternative H_a reads as

$$\underline{X}^2 \;=\; \underline{\widehat{\boldsymbol{s}}}^{\mathsf{T}}\, \boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}^{-1}\, \underline{\widehat{\boldsymbol{s}}} \,, \sim \chi^2(P) \tag{4.317}$$

which is χ^2-distributed if the null hypothesis holds. If the value of the test statistic exceeds the critical value, we have a good reason to reject the null hypothesis (there is no systematic error) in favour of the alternative hypothesis (that there are systematic errors modelled by parameter $\boldsymbol{s}$).

If we want to test the significance of individual parameters, the test statistic specializes to

$$\underline{z} = \frac{\widehat{s}}{\sigma_{\widehat{s}}} \sim \mathcal{N}(0,1)\,. \tag{4.318}$$

We recommend that the parameters which are used for modelling systematic errors be chosen in a manner such that their correlation $\rho_{s_i s_j}$ taken from $\boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}$ is low. This has the advantage that tests on the individual parameters are then almost independent. Otherwise, we cannot safely distinguish between systematic effects modelled by parameters s_i and s_j during testing. We will find this reasoning again when modelling image distortions in Sect. 15.4.3, p. 687 and Sect. 15.5.2, p. 699.

4.6.5.2 Checkability of Parameters

When the observations are made in order to check whether the parameters $\boldsymbol{s}$ are equal to some reference parameters $\boldsymbol{s}_r$, in our case assumed to be zero, we can determine the *checkability* of these parameters. As an example, we might want to identify specific systematic effects of a lens in order to improve construction of the lens. Then it is necessary to be able to check the result of the estimation w.r.t. the parameters describing the specific systematic effects.

The checkability can be measured by the minimum deviation $\nabla_0 s$ from the reference value $s_{\text{ref}} = 0$ which can be detected by the above-mentioned test (4.318), provided that the test is performed with a significance number α_0, and the power β of the decision is larger than β_0. The lower bound $\nabla_0 s$ is then given by Förstner (1980):

$$\nabla_0 s = \delta_0(\alpha_0, \beta_0)\, \sigma_{\widehat{s}}\,. \tag{4.319}$$

Assuming $\alpha_0 = 0.001$ and $\beta_0 = 0.8$, and thus $\delta_0 \approx 4.13$, cf. Table 3.2, p. 67, the parameter describing a systematic effect must be larger than the 4.13-fold of the standard deviation of the estimated parameter for the effect to be identifiable. Obviously, the smaller the standard deviation of the parameters, the better their checkability.

The individual evaluation of the checkability can be replaced by an evaluation of the complete vector of additional parameters based on the multi-dimensional test in (4.317). We then arrive at that least checkable combination $\nabla_0 \boldsymbol{s}$ of systematic errors. The size of a checkable systematic effect is bounded by

$$|\nabla_0 \boldsymbol{s}| \leq \delta_0(\alpha_0, \beta_0, P)\ \sqrt{\lambda_{\max}(\boldsymbol{\Sigma}_{\widehat{s}\widehat{s}})}\,. \tag{4.320}$$

It obviously refers to that combination of effects which is given by the eigenvector of $\boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}$ belonging to its largest eigenvalue. This again is plausible, as this is the combination with the largest parameter uncertainty. The noncentrality parameter $\delta_0(\alpha_0, \beta_0, P)$ needs to be taken from the cumulative noncentral χ^2 distribution $P_{\chi'^2}(x, d, \delta^2)$ with $d = P$ degrees of freedom, cf. (3.30), p. 68.

4.6.5.3 Effect of Systematic Errors on the Result

We will now derive measures for the sensitivity of the estimation w.r.t. the inclusion of additional parameters for modelling systematic errors.

The effect of arbitrary systematic errors $\nabla \boldsymbol{s}$ on the result can be directly determined from the explicit relations between observations and estimated parameters. In the case of the Gauss–Markov model, we have

$$\nabla \widehat{\boldsymbol{x}} = (\boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{A})^{-1} \boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{H} \nabla \boldsymbol{s}\,. \tag{4.321}$$

Again, this expression is useful only for small numbers N and U of observations and unknowns respectively, and helpful in simulation studies, as the systematic errors and thus the parameters $\boldsymbol{s}$ need to be known.

A scalar measure for the effect of running the estimation process with and without the parameters s is

$$\nabla \widehat{\boldsymbol{x}}^\mathsf{T}(\widehat{\boldsymbol{s}})\, \boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{-1}\, \nabla \widehat{\boldsymbol{x}}(\widehat{\boldsymbol{s}}) = X_s^2\ \tau^2(\widehat{\boldsymbol{s}}), \tag{4.322}$$

with X_s, which is the test statistic from (4.317). The factor τ^2 in (4.322) results from

$$\tau^2(\widehat{\boldsymbol{s}}) = \frac{\widehat{\boldsymbol{s}}^\mathsf{T} \boldsymbol{H}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{A} (\boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{A})^{-1} \boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{H} \widehat{\boldsymbol{s}}}{\widehat{\boldsymbol{s}}^\mathsf{T}\ \boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}^{-1}\ \widehat{\boldsymbol{s}}} = \frac{\widehat{\boldsymbol{s}}^\mathsf{T}\, (\boldsymbol{W}_{\widehat{s}\widehat{s}}^{(-)} - \boldsymbol{W}_{\widehat{s}\widehat{s}})\, \widehat{\boldsymbol{s}}}{\widehat{\boldsymbol{s}}^\mathsf{T}\, \boldsymbol{W}_{\widehat{s}\widehat{s}}\, \widehat{\boldsymbol{s}}}\,, \tag{4.323}$$

where

$$\boldsymbol{W}_{\widehat{s}\widehat{s}}^{(-x)} = \boldsymbol{H}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{H} \tag{4.324}$$

is the weight or precision matrix of the additional parameters if they are estimated *without* simultaneously estimating the unknowns $\boldsymbol{x}$.

This effect is only zero if

$$\boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{H} = \boldsymbol{0}\,, \tag{4.325}$$

and thus if the vector of systematic errors $\boldsymbol{h}_p$ (from $\boldsymbol{H}$) are orthogonal to the columns $\boldsymbol{a}_u$ of the design matrix. Then the additional parameters are generally estimable; they reduce the residuals but do not change the estimated parameters $\widehat{\boldsymbol{x}}$.

Empirical sensitivity with respect to additional parameters. The effect of leaving the parameter $\boldsymbol{s}$ out of the estimation process on an arbitrarily specified function $y = \boldsymbol{d}^\mathsf{T} \widehat{\boldsymbol{x}}$ is identical to the effect $|\nabla_s y|$ of setting the parameters to $\mathbf{0}$,

$$|\nabla_s y| = \boldsymbol{d}^\mathsf{T} (\boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{A})^{-1} \boldsymbol{A}^\mathsf{T} \boldsymbol{W}_{ll} \boldsymbol{H} \nabla \boldsymbol{s}\,. \tag{4.326}$$

This effect is bounded (cf. Baarda (1967, 1968); the proof is given below),

$$|\nabla_s y(\widehat{\boldsymbol{x}})| \leq X_s\ \mu_s\ \sigma_y\,. \tag{4.327}$$

It depends on the test statistic X_s, the sensitivity factor μ_s calculated from

$$\mu_s^2 = \lambda_{\max}\left(\left(W_{\widehat{s}\widehat{s}}^{(-x)} - W_{\widehat{s}\widehat{s}}\right) W_{\widehat{s}\widehat{s}}^{-1}\right) = \lambda_{\max}\left(\left(\Sigma_{\widehat{x}\widehat{x}}^{(+s)} - \Sigma_{\widehat{x}\widehat{x}}\right)\Sigma_{\widehat{x}\widehat{x}}^{-1}\right), \tag{4.328}$$

which is the maximum of $\tau^2(\widehat{\boldsymbol{s}})$ in (4.323), and the precision $\sigma_y = \sqrt{\boldsymbol{d}^\mathsf{T}\Sigma_{\widehat{x}\widehat{x}}\boldsymbol{d}}$ of the chosen function y. The matrix $\Sigma_{\widehat{x}\widehat{x}}^{(+s)}$ denotes the covariance matrix of the parameters when performing the estimation with the additional parameters s, and $\Sigma_{\widehat{x}\widehat{x}} = (A^\mathsf{T} W_{ll} A)^{-1}$ is the covariance matrix when estimating without the additional parameters, cf. the theorem (A.70), p. 773 and its proof.

The sensitivity factor μ_s in (4.328) measures the loss in precision of the additional parameters when simultaneously estimating the unknowns, and – remarkably – the loss in precision of the unknowns when simultaneously estimating the additional parameters, cf. Sect. A.5.2, p. 773.

For a single systematic error, the empirical sensitivity is given by

$$|\nabla_s y(\widehat{\boldsymbol{x}})| \le |z|\ \mu_s\ \sigma_y\,, \tag{4.329}$$

which is a specialization of (4.327).

Thus, the effect of the found systematic errors on an arbitrary function is less than or equal to $|z|\ \mu$ times the standard deviation of that function of the parameters. The value $|z|\ \sigma_y$ measures the *empirical sensitivity* of the result with respect to the systematic errors modelled by the parameter s. In particular, using $y = \widehat{x}_u$, the effect of the found systematic errors on a particular unknown parameter $\widehat{x}_u$ is limited by

$$\nabla_s \widehat{x}_u \le z\ \mu_s\ \sigma_{\widehat{x}_u}\,. \tag{4.330}$$

Proof: Equation (4.327) holds, since

$$\boldsymbol{d}^\mathsf{T}\Sigma_{\widehat{x}\widehat{x}}A^\mathsf{T}W_{ll}H\nabla\boldsymbol{s} = (\boldsymbol{d}^\mathsf{T}\Sigma_{\widehat{x}\widehat{x}})\cdot\Sigma_{\widehat{x}\widehat{x}}^{-1}\cdot(\Sigma_{\widehat{x}\widehat{x}}A^\mathsf{T}W_{ll}H\nabla\boldsymbol{s}) \le \sqrt{\boldsymbol{d}^\mathsf{T}\Sigma_{\widehat{x}\widehat{x}}\boldsymbol{d}}\,\tau(\widehat{\boldsymbol{s}})\ X_s \le X_s\,\mu_s\,\sigma_y$$

due to the Cauchy–Schwarz identity. ◇

Theoretical sensitivity with respect to additional parameters. The effect of a noncheckable systematic error $\nabla_0\boldsymbol{s}$ on an arbitrarily specified function $y = \boldsymbol{d}^\mathsf{T}\widehat{\boldsymbol{x}}$ is bounded by

$$\nabla_{0s} y(\widehat{\boldsymbol{x}}) \le \delta_0(\alpha_0, \beta_0, P)\ \mu_s\ \sigma_y \tag{4.331}$$

with the value $\delta_0(\alpha_0, \beta_0, P)$ specifying the assumed test, here a χ^2-Test. Parameters which are not checkable therefore have an influence on an arbitrary function y which is up to $\delta_0(P)\cdot\mu_s$ times the standard deviation of that function. Therefore the factor $\delta_0(\alpha_0, \beta_0, P)\cdot\mu_s$ measures the theoretical sensitivity of the result with respect to the parameters $\boldsymbol{s}$.

In the case of a single systematic error the theoretical sensitivity is measured by the bound

$$\nabla_{0s} y(\widehat{\boldsymbol{x}}) \le \delta_0\ \mu_s\ \sigma_y\,, \tag{4.332}$$

with the lower bound for the noncentrality parameter $\delta_0(\alpha_0, \beta_0)$ from (3.19), p. 66.

The most important values for evaluating the result and planning the measurement design w.r.t. systematic errors are given in Tables 4.3 and 4.4.

4.6.6 Effect of Errors in the Covariance Matrix

Systematic errors may be caused by wrong assumptions in the stochastical model. Here we give the effect of errors in the covariance matrix.

Table 4.3 Diagnostics for a single systematic error. **Left column:** empirical values used for evaluating the result of an estimation procedure. **Right column:** theoretical values to be used for planning purposes

	Empirical	Theoretical
1	Test statistic $z = \dfrac{\widehat{s}}{\sigma_{\widehat{s}}} =: \delta$ (4.318), p. 133	Standardized distance between H_0 and H_a $\delta_0 := \nabla_0 z$ 3.18, p. 66
2	Estimated size of error $\widehat{s} = z\,\sigma_{\widehat{s}}$ (4.318), p. 133	Lower bound for detectable error $\nabla_0 s = \delta_0\,\sigma_{\widehat{s}}$ (4.319), p. 133
3	Sensitivity factor $\mu_s^2 = \dfrac{w_{\widehat{s}}^{(-x)} - w_{\widehat{s}}}{w_{\widehat{s}}}$ (4.328), p. 135	
4	Actual influence of additional parameter $\lvert\nabla_s y(\widehat{\boldsymbol{x}})\rvert \le \lvert z\rvert\,\mu_s\,\sigma_y$ (4.329), p. 135	Maximal influence of undetectable systematic errors $\lvert\nabla_{0s} y(\widehat{\boldsymbol{x}})\rvert \le \delta_0\,\mu_s\,\sigma_y$ (4.332), p. 135

Table 4.4 Multi-dimensional diagnostics for a set of P systematic errors. **Left column:** empirical values used for evaluating the result of an estimation procedure. **Right column:** theoretical values to be used for planning purposes

	Empirical	Theoretical
1	Test statistic between H_0 and H_a $X^2 = \widehat{\boldsymbol{s}}^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{s}\widehat{s}}^{-1} \widehat{\boldsymbol{s}} =: \delta_i^2$ (4.317), p. 133	Standardized distance $\delta_0(P) := \nabla_0 X$ (3.30), p. 68
2	Estimated size of error $\widehat{\boldsymbol{s}}$ (4.316), p. 133	Lower bound for detectable error $\lvert\nabla_0 \boldsymbol{s}\rvert \le \delta_0(P)\,\sqrt{\lambda_{\max}(\boldsymbol{\Sigma}_{\widehat{s}\widehat{s}})}$ (4.320), p. 134
3	Sensitivity factor $\mu_s^2 = \lambda_{\max}\left((\boldsymbol{W}_{\widehat{s}\widehat{s}}^{(-x)} - \boldsymbol{W}_{\widehat{s}\widehat{s}})\boldsymbol{W}_{\widehat{s}\widehat{s}}^{-1}\right)$ (4.328), p. 135	
4	Actual influence of additional parameters $\lvert\nabla_s y(\widehat{\boldsymbol{x}})\rvert \le X\,\mu_s\,\sigma_y$ (4.327), p. 134	Maximal influence of undetectable systematic errors $\lvert\nabla_{0s} y(\widehat{\boldsymbol{x}})\rvert \le \delta_0(P)\,\mu_s\,\sigma_y$ (4.331), p. 135

The null hypothesis and the alternative hypothesis then are

$$H_0: \quad \mathbb{E}(\underline{\boldsymbol{l}}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{a}\,, \qquad \mathbb{D}(\underline{\boldsymbol{l}}) = \boldsymbol{\Sigma}_{ll}\,, \tag{4.333}$$

$$H_a: \quad \mathbb{E}(\underline{\boldsymbol{l}}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{a}\,, \qquad \mathbb{D}(\underline{\boldsymbol{l}}) = \boldsymbol{\Sigma}_{ll} + \nabla\boldsymbol{\Sigma}_{ll}\,. \tag{4.334}$$

We have the following results:

- The effect of errors in the covariance matrix on the estimate is small. A wrong covariance matrix still leads to unbiased estimated parameters $\widehat{\boldsymbol{x}}$.
- The estimated variance factor is biased.
- The covariance matrices of the estimated parameters and the residuals will be influenced strongly.
- For specific cases the result is not changed by using a wrong covariance matrix.

Effect on the Parameters. If the error $\nabla\boldsymbol{\Sigma}_{ll}$ in the covariance matrix is small, we can use the approximation $(\boldsymbol{\Sigma}_{ll} + \nabla\boldsymbol{\Sigma}_{ll})^{-1} \approx \boldsymbol{W}_{ll} - \boldsymbol{W}_{ll}\nabla\boldsymbol{\Sigma}_{ll}\boldsymbol{W}_{ll}$ with

$$\nabla\boldsymbol{W}_{ll} := -\boldsymbol{W}_{ll}\nabla\boldsymbol{\Sigma}_{ll}\boldsymbol{W}_{ll}\,. \tag{4.335}$$

Then its influence on the estimated parameters is given by Koch (1999, Eq. (3.108)):

$$\nabla\widehat{\boldsymbol{x}} \approx (A^{\mathsf{T}}W_{ll}A)^{-1}A^{\mathsf{T}}\nabla W_{ll}\cdot(I - A(A^{\mathsf{T}}W_{ll}A)^{-1}A^{\mathsf{T}}W_{ll})(\boldsymbol{l}-\boldsymbol{a})\,. \tag{4.336}$$

Since $\mathbb{E}(\boldsymbol{l}-\boldsymbol{a}) = A\boldsymbol{x}$, the bias induced by a wrong covariance matrix is zero, independent of the magnitude of ∇W_{ll}, as the right factor vanishes. The expected total effect on the parameters is

$$d^2 = \mathbb{E}(\nabla\underline{\widehat{\boldsymbol{x}}}^{\mathsf{T}}\Sigma^{-1}_{\widehat{x}\widehat{x}}\nabla\underline{\widehat{\boldsymbol{x}}}) = \mathrm{tr}(\nabla W_{ll}\Sigma_{\widehat{l}\widehat{l}}\nabla W_{ll}\Sigma_{\widehat{v}\widehat{v}})\,, \tag{4.337}$$

with the covariance matrices $\Sigma_{\widehat{l}\widehat{l}}$ and $\Sigma_{\widehat{v}\widehat{v}}$ of the fitted observations $\widehat{\boldsymbol{l}}$ and the residuals $\widehat{\boldsymbol{v}}$, cf. (4.53), p. 86 and (4.59), p. 87. This expression can be simplified to a quadratic form using the rules for Kronecker products, cf. (A.95), p. 775,

$$d^2 = \mathrm{vec}^{\mathsf{T}}(\nabla W_{ll})(\Sigma_{\widehat{l}\widehat{l}}\otimes\Sigma_{\widehat{v}\widehat{v}})\mathrm{vec}(\nabla W_{ll})\,. \tag{4.338}$$

If the change ∇W_{ll} of the weight matrix is diagonal, we finally obtain the simple expression

$$d^2 = \mathrm{diag}^{\mathsf{T}}(\nabla W_{ll})(\Sigma_{\widehat{l}\widehat{l}}\odot\Sigma_{\widehat{v}\widehat{v}})\mathrm{diag}(\nabla W_{ll}) \tag{4.339}$$

(cf. (A.98), p. 776), where $\mathrm{diag}^{\mathsf{T}}(\nabla W_{ll})$ is the vector containing only the diagonal elements of the matrix ∇W_{ll} and $\odot$ is the elementwise or Hadamard product of two matrices.

Example 4.6.11: ***Effect of a single weight error on the result.*** The effect of errors in the observations on the result is generally very small. Let us assume $W_{ll} = I$. Then a change ∇w_n in a single observation leads to $d_n = \sqrt{r_n(1-r_n)}\nabla w_n \le \nabla w_n/2$, with $r_n \in [0,1]$, which is the redundancy number of the observation. Obviously the influence is very small, as weight errors usually are below a factor of 2. ◇

Bias of the Estimated Variance Factor. With the redundancy R and using $\boldsymbol{x}^{\mathsf{T}}A\boldsymbol{x} = \mathrm{tr}(A\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}})$, the expectation of the variance factor results from

$$\begin{aligned} R\mathbb{E}(\widehat{\sigma}_0^2) &= \mathbb{E}(\underline{\widehat{\boldsymbol{v}}}^{\mathsf{T}}(\Sigma_{ll}+\nabla\Sigma_{ll})^{-1}\underline{\widehat{\boldsymbol{v}}}) = \mathrm{tr}\mathbb{E}((\Sigma_{ll}+\nabla\Sigma_{ll})^{-1}\underline{\widehat{\boldsymbol{v}}}\underline{\widehat{\boldsymbol{v}}}^{\mathsf{T}}) && (4.340)\\ &= \mathrm{tr}\left[(\Sigma_{ll}+\nabla\Sigma_{ll})^{-1}\ \left(\Sigma_{ll} - \Sigma_{ll}A(A^{\mathsf{T}}\Sigma_{ll}^{-1}A)^{-1}A^{\mathsf{T}}\Sigma_{ll})\right)\right] && (4.341)\\ &\neq R\,. && (4.342) \end{aligned}$$

The sign of the bias $\mathbb{E}(\widehat{\sigma}_0^2) - 1$ is not definite.

Covariance of Estimates Using a Wrong Covariance Matrix. If the estimation is performed with $\Sigma_{ll} = \Sigma$ but the true covariance matrix of the observations is $\widetilde{\Sigma_{ll}} = \widetilde{\Sigma}$, then the covariance matrix of the estimated parameters is

$$\Sigma_{\widehat{x}\widehat{x}} = (A^{\mathsf{T}}\Sigma^{-1}A)^{-1}A^{\mathsf{T}}\Sigma^{-1}\ \widetilde{\Sigma}\ \Sigma^{-1}A(A^{\mathsf{T}}\Sigma^{-1}A)^{-1}\,, \tag{4.343}$$

which follows from $\underline{\widehat{\boldsymbol{x}}} = (A^{\mathsf{T}}\Sigma^{-1}A)^{-1}A^{\mathsf{T}}\Sigma^{-1}(\underline{\boldsymbol{l}}-\boldsymbol{a})$. Observe, only if $\Sigma = \widetilde{\Sigma}$ do we obtain the classical result $\Sigma_{\widehat{x}\widehat{x}} = (A^{\mathsf{T}}\widetilde{\Sigma}^{-1}A)^{-1}$. Equation (4.343) can be used to investigate the effect of choosing a simplified stochastical model, e.g., when using $\Sigma_{ll} = \sigma^2 I_N$ instead of $\widetilde{\Sigma}$. The derivation of the covariance matrix of parameters resulting from an algebraic optimization (Sect. 4.9.2.4, p. 180) uses (4.343). Examples for the effect of using an incorrect covariance matrix of observations is given in Example 10.6.2.1, p. 419.

Invariance of the Estimate with Respect to Changes in the Covariance Matrix. Certain deviations of the used covariance matrix from the correct one do not have an influence on the estimation. This following result is important when a set of otherwise uncorrelated observations is correlated due to some common effect $\boldsymbol{e} = A\boldsymbol{x}$, and this effect actually is determined in the estimation process. For example, this occurs when image coordinates are correlated due to a common uncertainty in the camera calibration, e.g., a common shift. Then the correlation induced by this effect, shown in the covariance matrix $\mathrm{Diag}([\sigma_i^2]) + A\Sigma_{xx}A^{\mathsf{T}}$, does not influence the estimation using $\underline{\boldsymbol{l}} \sim \mathcal{M}(A\boldsymbol{x}, \mathrm{Diag}([\sigma_i^2]) + A\Sigma_{xx}A^{\mathsf{T}})$. Exercise 4.18

We have the following lemma by Rao (1967, Lemma 5a and corollary): Let the Gauss–Markov model be $\underline{\boldsymbol{l}} \sim \mathcal{M}(A\tilde{\boldsymbol{x}} + \boldsymbol{a}, \Sigma_0)$. Let Z be a $N \times R$ matrix of rank $R = N - \mathrm{rk}(A)$ such that $Z^\mathsf{T} A = 0$ and let $\mathcal{S}$ be the set of matrices of the form

$$\Sigma = AXA^\mathsf{T} + \Sigma_0 ZYZ^\mathsf{T}\Sigma_0 + \Sigma_0 \,, \tag{4.344}$$

where the symmetric matrices X and Y are arbitrary. Then the necessary and sufficient condition that the least squares estimator of $\boldsymbol{x}$ in the model $\underline{\boldsymbol{l}} \sim \mathcal{M}(A\tilde{\boldsymbol{x}} + \boldsymbol{a}, \Sigma)$ is the same as that for the special choice $\mathbb{D}(\boldsymbol{l}) = \sigma^2\Sigma_0$, is $\Sigma \in \mathcal{S}$.

4.6.7 Model Selection

Up to now we assumed the mathematical model is given. We now want to address the problem of choosing one model out of several alternative models based on a given set of observations. We may address this problem in a Bayesian manner: from among alternative models $\mathcal{M}_m$ we want to choose that model where the posterior probability $P(\mathcal{M}|\boldsymbol{l})$ is largest. Then we arrive at the following optimization problem:

$$\widehat{\mathcal{M}} = \mathrm{argmax}_m \; p(\boldsymbol{l}|\mathcal{M}_m)P(\mathcal{M}_m) \,. \tag{4.345}$$

Thus we need to know the prior probabilities $P(\mathcal{M}_m)$ for the different models $\mathcal{M}_m$. This poses a practical problem, as these probabilities generally are not known.

However, we may argue that more complex models are less likely than less complex models, which require us to specify the complexity of a model. In the most simple case this complexity may be measured by the number U_m of the parameters $\boldsymbol{x}_m$ of the model $\mathcal{M}_m$. Thus we arrive at the following simplified optimization problem:

$$\widehat{\mathcal{M}} = \mathrm{argmax}_m \; p(\boldsymbol{l}|\boldsymbol{x}_m, \mathcal{M}_m)\, p(\boldsymbol{x}_m|\mathcal{M}_m)P(\mathcal{M}_m) \,, \tag{4.346}$$

with

$$P(\mathcal{M}_m) = P(U_m) \,. \tag{4.347}$$

This setup appears to be reasonable, as the product of the first two terms is the likelihood $p(\boldsymbol{l}|\mathcal{M}_m)$ of the data, given a specific model $\mathcal{M}_m$ having U_m parameters. This likelihood generally will be larger for a larger number of parameters, which can be compensated for by a lower probability $P(U_m)$ of choosing a model with U_m parameters. Taking negative logarithms, we arrive at an equivalent setup:

$$\widehat{\mathcal{M}} = \mathrm{argmin}_m \; -\log p(\boldsymbol{l}|\mathcal{M}_m) - \log P(U_m) \,. \tag{4.348}$$

If we assume the data to be normally distributed, the first term, up to an additive constant, is

$$-\log p(\boldsymbol{l}|\mathcal{M}_m) = \frac{1}{2}\widehat{\boldsymbol{v}}^\mathsf{T} W_{ll}\widehat{\boldsymbol{v}} \,. \tag{4.349}$$

There are different arguments to arrive at the last term, $-\log P(U_m)$. We mention two of them:

1. The Akaike information criterion (AIC, cf. Akaike, 1974). Akaike argues, that the model, which is represented by the U parameters $\widehat{\boldsymbol{x}}$, is uncertain, characterized by the covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ which increases with increasing U. He proposes the following selection criterion:

$$\widehat{\mathcal{M}}_{\mathrm{AIC}} = \mathrm{argmax}_m \; -\log p(\boldsymbol{l}|\mathcal{M}_m) + U_m \,. \tag{4.350}$$

This intuitively corresponds to assuming $P(U) = \exp(-U)$, preferring small numbers U of parameters.

2. The *Bayesian information criterion* (BIC, cf. Schwarz, 1978). Schwarz follows a Bayesian approach with a general assumption about the distribution of the parameters, and arrives at the following selection criterion:

$$\widehat{\mathcal{M}}_{\text{BIC}} = \text{argmax}_m \; -\log p(\boldsymbol{l}|\mathcal{M}_m) + \frac{1}{2} U_m \log N \,, \tag{4.351}$$

with the number N of the observations included. Observe, the additive term can also be written as $U_m \log \sqrt{N}$: The factor $\log \sqrt{N}$ increases the effect of choosing a larger number of parameters. This appears reasonable, as when the number of observations is large, it should be less easy to increase the number of parameters.

Both criteria have been investigated for various estimation tasks and been modified, as their reasoning is too general for covering specific modelling situations. As soon as some prior knowledge about alternative models is available, this knowledge can be used to follow (4.345).

Remark: The model selection problem is simplified if the parametric model $\mathcal{M}_1(\boldsymbol{x}_1)$ is a special case of the model $\mathcal{M}_2(\boldsymbol{x}_2)$ in the sense that

$$\boldsymbol{x}_2 = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{s} \end{bmatrix} ; \tag{4.352}$$

thus the more general model $\mathcal{M}_2$ just has some additional parameters $\boldsymbol{s}$, compared to model $\mathcal{M}_1$. Then we can test the estimated parameters $\widehat{\boldsymbol{s}}$ for significance, cf. the previous Sect. 4.6.5, p. 133 and (4.317), p. 133. If the test statistic X_s suggests rejecting the null hypothesis, i.e., the additional parameters are 0, this can be interpreted as: the model $\mathcal{M}_1$ does not hold in favour of the more general model $\mathcal{M}_2$. Observe, no prior probabilities for the two models are used, implicitly favouring the more specific model $\mathcal{M}_1$, and assuming a flat prior for the additional parameters, see the discussion on classical and Bayesian testing in Sect. 3.1.2, p. 64. ◇

4.6.8 Checking the Implementation of the Estimation

Before using the implementation of an estimation procedure we need to check whether it yields correct results. This refers to (1) the estimated parameters, (2) their covariance matrix, and (3) the estimated variance factor. The estimated parameters should be unbiased, the covariance matrix should reflect the sensitivity of the estimated parameters w.r.t. random perturbations of the observations, characterized by the stochastical model, especially the covariance matrix of the observations; and the estimated variance factor should not significantly deviate from 1.

If the implementation is correct, small perturbations in the observations following the stochastical model should lead to small perturbations in the variance factor and in the estimated parameters, where they also should follow the predicted covariance matrix. In the case of larger perturbations, effects of the linearization of a nonlinear model will be visible.

Such an evaluation is based on simulated data, since we then have access to the true values. This also has the advantage that no access to the source code is necessary; the check can be based on the output $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}, \widehat{\sigma_0}^2\}$.

Based on given true values $\tilde{\boldsymbol{x}}$ for the parameters, a given observational design, represented by the function $\boldsymbol{f}(\boldsymbol{x})$ and a stochastical model $\mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll}$, we can simulate K samples of observations $\boldsymbol{l}_k$ from

$$\underline{\boldsymbol{l}}_k = \boldsymbol{f}(\tilde{\boldsymbol{x}}) - \underline{\boldsymbol{v}}_k \,, \; k = 1, ..., K \qquad \underline{\boldsymbol{v}} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{ll}) \,, \tag{4.353}$$

leaving the model $\{\boldsymbol{f}(\boldsymbol{x}), \Sigma_{ll}\}$ and the true parameters $\tilde{\boldsymbol{x}}$ fixed.

The estimation leads to K vectors $\widehat{\boldsymbol{x}}_k$ of estimated parameters, to K estimates $\widehat{\sigma}_{0k}^2$ of the variance factor, and – provided the relative accuracy $\sigma_l/\mathbb{E}(l)$ of the observations is below 1% – a sufficiently accurate covariance matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$. In order to be able to check the validity of the model a sufficiently large number K of samples is necessary, which should be larger than the number of elements of the largest covariance matrix which is to be checked.

In the case of Gaussian noise, the evaluation can be based on well-established statistical tests. If one of these tests fails, there are good reasons to doubt whether the program code is a reliable realization of the envisaged estimation model. However, without further tests there are no clues to the source of the discrepancy; it may be the implementation of the envisaged model or of the simulation. This may require more detailed testing.

We now discuss three tests concerning the noise level, the bias, and the validity of the theoretical covariance matrix. They should be performed on a set of representative estimation tasks before using the estimation procedure in a real application.

4.6.8.1 Correctness of the Estimated Noise Level

The correctness of the estimated noise level can be reduced to check the validity of the variance factor. The validity of the estimated variance factor can be based on the mean of the K variance factors derived from the K simulations,

$$s^2 = \frac{1}{K}\sum_{k=1}^{K}\widehat{\sigma}_{0k}^2\,. \tag{4.354}$$

When the implemented model, which is the null hypothesis H_0, holds, the test statistic

$$F = \frac{s^2}{\sigma_0^2}\,, \quad \underline{F}|H_0 \sim F_{KR,\infty} \tag{4.355}$$

is Fisher distributed with KR and ∞ degrees of freedom, where R is the redundancy of the estimation task, cf. (3.42), p. 70. If for a specified significance level S, the test statistic $F > F_{KR,\infty;S}$, then the estimated variance factor indicates deviations from the assumed model – possibly caused by implementation errors. In this case, it might be useful to analyse the histogram in order to find possible sources of the deviations.

Observe, this test does not require the theoretical covariance matrix $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ of the estimated parameters.

4.6.8.2 Correctness of the Covariance Matrix

To make sure we can rely on the theoretical covariance matrix provided by the implemented estimation procedure, we compare it with the empirical covariance matrix of the simulation sample. It is given by

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{K-1}\sum_{k=1}^{K}(\widehat{\boldsymbol{x}}_k - \widehat{\boldsymbol{m}}_{\widehat{x}})(\widehat{\boldsymbol{x}}_k - \widehat{\boldsymbol{m}}_{\widehat{x}})^{\mathsf{T}} \tag{4.356}$$

with the estimated mean

$$\widehat{\boldsymbol{m}}_{\widehat{x}} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\boldsymbol{x}}_k\,. \tag{4.357}$$

When the model holds as implemented and the theoretical precision $\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}$ is correct, the test statistic

$$\underline{X}^2 = (K-1)\left[\ln\left(\det\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}/\det\underline{\widehat{\boldsymbol{\Sigma}}}\right) - U + \operatorname{tr}\left(\underline{\widehat{\boldsymbol{\Sigma}}}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}^{-1}\right)\right] \sim \chi^2_{U(U+1)/2} \tag{4.358}$$

is approximately χ^2-distributed with $U(U+1)/2$ degrees of freedom (cf. Koch, 1999, Eq. (2.205), and (3.57), p. 72). If for a prespecified significance level S the test statistic X^2 is larger than $\chi^2_{U(U+1)/2,S}$, then there is reason to assume the theoretical covariance matrix, as it results from the implemented model, does not reflect the covariance matrix of $\underline{\widehat{\boldsymbol{x}}}$ sufficiently well. In this case, it might be useful to visualize the covariance matrix in order to identify possible causes for the found deviation.

It is sufficient to take one of them as reference, though the theoretical covariances of the K samples vary slightly, as the variance propagation is performed not using the true mean, but the estimated parameters. However, as the relative size of this variation is a second-order effect, it can be neglected.

4.6.8.3 Bias in the Estimates

To check the unbiasedness of the estimated parameters we determine their empirical mean.

If the mathematical model holds, the implementation is correct, and higher-order terms during linearization are negligible; the estimated mean of the estimated parameters is Gaussian distributed according to

$$\underline{\widehat{\boldsymbol{m}}}_{\widehat{x}} \sim \mathcal{N}\left(\tilde{\boldsymbol{x}}, \frac{1}{K}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\right) . \tag{4.359}$$

Under these conditions, the test statistic, the Mahalanobis distance,

$$\underline{X} = K(\underline{\widehat{\boldsymbol{m}}}_{\widehat{x}} - \tilde{\boldsymbol{x}})^{\mathsf{T}}\, \boldsymbol{\Sigma}^{-1}_{\widehat{x}\widehat{x}}\, (\underline{\widehat{\boldsymbol{m}}}_{\widehat{x}} - \tilde{\boldsymbol{x}}) \sim \chi^2_U\,, \tag{4.360}$$

is χ^2-distributed with U degrees of freedom (cf. (3.32), p. 69). If $X > \chi^2_{U,S}$ for the test statistic and a prespecified significance level S, we have reasons to reject the hypothesis that the model, including the approximations, actually holds as implemented. In this case it might be useful to visualize the bias in order to find possible causes for the rejection of the model.

If these statistical tests are passed on a set of representative simulation data sets, the statistical tests, when applied to real data, can be used as diagnostic tools for identifying discrepancies between the data and the assumed mathematical model.

4.7 Robust Estimation and Outlier Detection

4.7.1 Outlier Model 143
4.7.2 Preconditions for Robust Estimation Procedures 144
4.7.3 Robust Estimation of the Variance Factor 145
4.7.4 Robust Maximum Likelihood-Type Estimation 147
4.7.5 L_1-Norm Minimization 150
4.7.6 Complete Search for Robust Estimation 151
4.7.7 Random Sample Consensus (RANSAC) 153
4.7.8 Robust Estimation by Clustering 157
4.7.9 Rules for Choosing Robust Estimation Technique 158

All previous sections assume either the validity of the mathematical model or some knowledge about model deviations. Confronted with real data, these assumptions often do not hold. There will always be some discrepancies a priori not known between the data and the chosen model.

When addressing this situation, there are two possible views: Erroneously assuming the model to hold leads to the concept of outliers, while erroneously assuming the data to

fulfil the model leads to the concept of model deviations. There is no way to decide on the validity of one or the other view. However, both views can be justified, and – for efficiency – lead to different procedures, as already shown in the last section. There, however, we only treated single outliers or groups of outliers, or systematic errors which can be modelled as small deviations in the functional model. This is certainly sufficient for systematic errors, but not for outliers. Therefore, in this section, we assume outliers of arbitrary size and number.

We may address the problem of estimation in the presence of outliers according to two views:

outlier / blunder detection

1. If we erroneously assume the model holds, we need to classify the observations into good and bad ones, i.e., inliers and outliers (or blunders). This view is the motivation for outlier detection schemes.

robust estimators

2. If we erroneously assume the data fulfils the model, we need to be able to find estimates in the presence of outliers. This view is the motivation for using robust estimators. According to Huber (1991), the purpose of *robustness* is to have safeguards against deviations from the model assumptions. This is in contrast to *diagnostics*, discussed in the previous section, whose purpose is to find and identify deviations from the model assumptions.

As both views are not distinguishable conceptually, we can interpret estimation methods including outlier detection schemes as robust estimators and, alternatively, use robust estimation schemes for outlier detection.

Eliminating blunders is a difficult problem:

- The algorithmic complexity is high: given N observations there are up to 2^N hypotheses for sets of good and bad values. This makes an exhaustive search for the optimized solution impossible, except for problems with few observations. It indicates the other solutions to be suboptimal in general.
- Suboptimal solutions are difficult to obtain since generic simplifications of nonlinear problems are not available.
- All variations of "Murphy's Law" (Bloch, 1978) occur:
  - outliers cluster and support each other,
  - outliers mimic good results,
  - outliers hide behind configuration defects,
  - outliers do not show their causes, making proper modelling difficult or impossible,
  - outliers are indistinguishable from other deficiencies in the model, like systematic errors.

As a consequence, a large number of methods for robust estimation have been developed; none of them is optimal.

Since outlier detection procedures may erroneously eliminate inliers, the distribution of the resulting set of inliers will be truncated. Similarly, since it is impossible to track the joint distribution of in- and outliers through the robust estimation scheme, the distribution of the estimated parameters is unknown (cf. Huber, 2009, Sects. 10 and 11). Both situations prevent a rigorous evaluation of the results as discussed in the last section.

robust estimation as oracle

A way out is to interpret robust procedures as an *oracle* in the sense of an oracle machine proposed by Turing (1939) (Feferman, 2006).[8] Such an oracle is meant to perfectly separate in- and outliers, Fig. 4.12. Of course, this can only be realized approximately.

Assuming this decision is perfect, a final estimation step using the inliers now can be accompanied with a subsequent evaluation, as the distribution of the inlying observations and the finally estimated parameters can be assumed to be sufficiently close to a Gaussian distribution, neglecting truncation effects or the nonnormality of the distribution of the estimated parameters.

[8] http://en.wikipedia.org/wiki/Oracle_machine

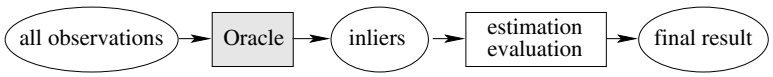


Fig. 4.12 Robust estimation as oracle machine: The oracle is assumed to give a perfect decision on in- and outliers. This allows us to perform classical estimation and statistical evaluation afterwards. Robust estimators or outlier detectors can be evaluated w.r.t. their ability to correctly separate in- and outliers

The solution to the problem of robust estimation or outlier detection thus is equivalent to developing algorithms which approximate the oracle machine for identifying inliers.

Methods for finding maximal sets of consistent observations conceptually are the first choice and under current investigation, cf. e.g., Li (2009); Yang et al. (2014); Chin et al. (2015). Since the computational complexity of these algorithms is high, they yet cannot compete with available classical methods.

This section will collect techniques from robust statistics (Hampel et al., 1986; Rousseeuw and Leroy, 1987; Koch, 1996) useful for efficiently eliminating or compensating for outliers in the data by adapting it to the presumed model or by using a modified model.

In the following, we again refer to the Gauss–Markov model, $l_n + v_n = f_n(\boldsymbol{x})$, where we assume statistically independent observations, with individual standard deviations σ_{v_n} of the measurement deviations $e_n = -v_n$. The methods can be transferred to groups $\{\boldsymbol{l}_i, \Sigma_{l_i l_i}\}$ of correlated observations. We thus will refer to either single observations l_n or observational groups $\boldsymbol{l}_i$, the index making the distinction clear.

Robust estimation with models other than the Gauss–Markov can be done similarly by referring to the basic relations $\boldsymbol{g}_j(\boldsymbol{l}, \boldsymbol{x}) = 0$, and then referring to single or vector-valued constraints to be evaluated (cf. Sect. 4.8.1, p. 162).

4.7.1 Outlier Model

Most methods for outlier detection model outliers as large deviations of the observations from their true value, and thus explicitly or implicitly use a modified probability density function as the likelihood function. The following model of a mixed distribution of the measurement deviations e_n or the residuals $v_n = -e_n$ is common:

$$p(v_n) = (1 - \varepsilon)\ \phi(e_n) + \varepsilon\ .h(e_n) \tag{4.361}$$

The distribution of the residuals is described by their (v_n). We assume a small percentage ε of outliers, with a broad distribution $h(x)$. They are collected in the set $\mathcal{B}$ of bad observations. The $1-\varepsilon$ inliers are good observations following a well-behaved distribution $\phi(x)$, usually a Gaussian. They are collected in the set $\mathcal{G}$ (Fig. 4.13). Maximizing the likelihood function $p(\boldsymbol{e}) = p(\boldsymbol{l} \mid \boldsymbol{x})$ for the given data $\boldsymbol{l}$, possibly including prior knowledge of the unknowns $\boldsymbol{x}$, is explicitly or implicitly used as the optimization criterion. Equivalently, we

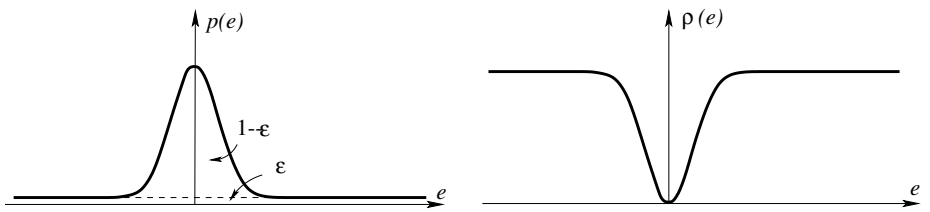


Fig. 4.13 Left: Distribution of the errors e modelled as a mixture of good and bad errors: good errors with probability $1 - \varepsilon$ follow a Gaussian distribution, bad errors with a probability ε follow a uniform distribution, valid in a limited range. **Right:** The negative logarithm $\rho(e) = -\log p(e)$ of the likelihood function is nonconvex

may minimize

$$-\log p(\boldsymbol{v}) = -\log p(\boldsymbol{l} \mid \boldsymbol{x}) \overset{\text{independence}}{=} -\sum_n \log p(v_n)\,. \tag{4.362}$$

Assuming independent observations and referring to the *normalized residuals*

$$y_n = \frac{v_n}{\sigma_{l_n}} = v_n\sqrt{w_n} \tag{4.363}$$

and defining the function ρ as

$$\rho(y_n) = -\log p(y_n)\,, \tag{4.364}$$

often this minimization function is replaced by

$$\Omega = \sum_n \rho(y_n)\,. \tag{4.365}$$

For the outlier distribution in Fig. 4.13, left, the function ρ will increase with increasing absolute value of the normalized residuals y_n until it becomes flat, shown in Fig. 4.13, right. It is a nonconvex function, indicating the optimization generally to be computationally hard.

pseudo-likelihood function

Observe, the density $p(e_n)$ is only a proper density if the uniform distribution of the outliers has a bounded domain, e.g., $[-10^6, +10^6]$. Therefore $p(v_n \mid \boldsymbol{x})$ is often called a *pseudo-likelihood* function, as w.r.t. the optimization procedure it behaves like a likelihood function. If we start with the optimization function $\rho(y_n)$ with no bounds on the domain, normalization of the derived density $p(y_n) \propto \exp(-\rho(y_n))$ will lead to an improper density.

If we have independent groups $\boldsymbol{l}_i$ of observations, the optimization function can be written as

$$\Omega = \sum_{i=1}^{I} \rho(y_i) \tag{4.366}$$

with the normalized squared residual

$$y_i^2 = \boldsymbol{v}_i^{\mathsf{T}} \Sigma_{l_i l_i}^{-1} \boldsymbol{v}_i\,, \tag{4.367}$$

which can be specialized to (4.363) if the group only contains a single observation.

4.7.2 Preconditions for Robust Estimation Procedures

The procedures for handling outliers differ significantly. Unfortunately, none of the known methods leads to optimal results in all cases. We present five procedures which seem to be representative in order to perform a comparative evaluation and propose a generic strategy for outlier detection:

1. maximum likelihood-type estimation (Huber, 1981; Hampel et al., 1986),
2. L_1-norm minimization,
3. complete search,
4. random sample consensus (RANSAC) (Fischler and Bolles, 1981), and
5. clustering.

Their feasibility and efficiency heavily depend on a number of characteristic features of the estimation problem to be solved:

a) *Invertibility of the functional model.* The functional model is called invertible if for a minimum number of given observations there is a direct solution for the parameters.

Directly determining the unknowns $\boldsymbol{x}$ from a minimal set $\boldsymbol{l}$ in the case of the *Gauss–Markov model* $\mathbb{E}(\boldsymbol{l}) = \boldsymbol{f}(\boldsymbol{x})$ requires that $\boldsymbol{f}$ be invertible, i.e., $\boldsymbol{x}(\boldsymbol{l}) = \boldsymbol{f}^{-1}(\boldsymbol{l})$, as in the case of a 3D plane, where the three plane parameters can be determined from three given points.

In the *Gauss–Helmert model* we need to be able to solve $\boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x}) = \boldsymbol{0}$ for $\boldsymbol{x}$, as e.g., in the case of a 2D circle (Sect. 4.9), where three constraints on three points are sufficient to determine the three parameters.

The solution of $\boldsymbol{f}^{-1}(\boldsymbol{l})$ may not be unique. The method may be algebraic but may also include the determination of the zeros of a high-order polynomial, as they do not need approximate values and are computationally stable, though iterative in nature.

b) *Existence and quality of approximate values.* If $\boldsymbol{f}(\boldsymbol{l})$ is not invertible, we need approximate values for $\boldsymbol{x}$ in order to solve $\boldsymbol{x} = \boldsymbol{f}^{-1}(\boldsymbol{l})$ by some iterative scheme. A similar argument holds for the case when $\boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x}) = \boldsymbol{0}$ is not solvable for $\boldsymbol{x}$. The quality of the approximate values directly influences the number of iterations. The knowledge of good approximate values in all cases may drastically reduce the complexity of the procedures.

c) *Percentage of gross errors.* The percentage of gross errors may range from $< 1\%$, specifically in large data sets derived manually, up to more than 75%, e.g., in automatic matching problems. Many procedures cannot cope with a very high percentage of errors, while some are especially suited for problems with high outlier percentages.

d) *Size of gross errors.* Only a few procedures work for any size of gross error. Large gross errors may lead to leverage points, i.e., to locally weak geometry, and such errors may not be detectable at all. If we relate the size of the errors to the size of the observed value, then relative errors less than 1 are usually detectable by all procedures.

e) *Relative redundancy.* The relative redundancy, measured by the redundancy numbers r_i (4.69), influences the detectability of errors. The theoretical results from robust statistics, especially referring to ML-type estimation, are only valid for relative redundancies above 0.8, i.e., when the number of observations is larger than five times the number of unknown parameters.

f) *Number of unknowns.* The number of unknowns directly influences the algorithmic complexity, i.e., the increase of the computing time with the number of unknowns.

The efficiency of the methods can be characterized by the following two measures:

1. *Computational complexity* as a function of the size of the input data, the percentage of outliers, and the *model complexity*.
2. *Breakdown point.* What is called the *breakdown point* is the maximal percentage of outliers above which the method yields an arbitrarily wrong estimate. For example, the breakdown point of least squares estimation is zero, as a single outlier may lead to arbitrarily wrong results, whereas the breakdown point of the median for estimating the mean is 50%, as it can find a correct solution from data with up to 50% outliers.

breakdown point

All methods rely on the knowledge of the variance factor ${\sigma_0}^2$, i.e., on a proper scaling of the weights of the observations. Therefore, we discuss robust methods for estimating the variance factor first.

4.7.3 Robust Estimation of the Variance Factor

The estimation of the variance factor σ_0^2 is based on the model $\boldsymbol{\Sigma}_{ll} = \sigma_0^2 \boldsymbol{\Sigma}_{ll}^a$, where the relative weighting is given by the approximate covariance matrix $\boldsymbol{\Sigma}_{ll}^a$. The classical nonrobust estimate

$$\widehat{\sigma}_0^2 = \frac{\Omega}{N-U} = \frac{N}{N-U}\,\widehat{\sigma}_y^2 \tag{4.368}$$

essentially depends on the root mean square error (RMSE), the standard deviation of the normalized residuals (cf. (4.363), p. 144),

$$\widehat{\sigma}_y^2 = \frac{\Omega}{N} = \frac{\sum_n y_n^2}{N} \,. \tag{4.369}$$

The factor $N/(N-U)$ in (4.368) is necessary to arrive at an unbiased estimator for the variance factor.

In the case of outliers, it is useful to replace the RMSE by a robust estimate, assuming the residuals are derived by some robust estimator or are based on adequate approximate values. Assuming an outlier percentage below 50%, we can take the median of the absolute normalized residuals and obtain a robust estimate for the standard deviation of the normalized residuals (cf. (4.363), p. 144),

robust estimator of standard deviation of normalized residuals

$$\widehat{\sigma}_{y,N/2} = 1.4826 \,\mathrm{med}(|y_n|) \,, \tag{4.370}$$

where the factor $1/\Phi^{-1}(0.75) \approx 1.4826$ compensates for the fact that the median of a centred Gaussian distributed random variable is smaller than the standard deviation by a factor $\Phi^{-1}(0.75) \approx 0.6745$. The estimator $\mathrm{med}(|y_n|)$ is the median absolute difference, cf. (2.124), p. 40.

In the case of a number $K \leq N/2$ of inliers, instead of the 50% point of the histogram of the absolute normalized residuals, we can use the K/N-point,

$$\widehat{\sigma}_{y,K} = \frac{\{|y_n|\}_K}{\Phi^{-1}\left((1+K/N)/2\right)} \,, \tag{4.371}$$

for robustly estimating the standard deviation of the normalized residuals, where $\{|y_n|\}_K$ is the Kth largest absolute normalized residual. Obviously, (4.370) is a special case of (4.371), setting $K = N/2$. The Kth largest element of a list of N values can be determined in time proportional to N (Wirth, 1978). Specifying a critical value c, e.g., $c = 3$, i.e., residuals $|y_n| < c\,\widehat{\sigma}_{y,K}$ can be regarded as inliers, yields an estimate $\widehat{K}$ for the size of the set

$$\widehat{\mathcal{G}} = \{n \mid |y_n| < c\,\widehat{\sigma}_{y,K}\} \,, \qquad \widehat{K} = |\widehat{\mathcal{G}}| \tag{4.372}$$

of good observations, i.e., inliers. We then finally arrive at a robust estimate for the variance factor, only using the $\widehat{K}$ inliers,

robust estimator of variance factor

$$\widehat{\sigma}_{0,K}^2 = \frac{1}{\widehat{K}-U} \sum_{n\in\widehat{\mathcal{G}}} \frac{y_n^2}{\widehat{\sigma}_{y,K}^2} \,. \tag{4.373}$$

This estimator for the variance factor is called *adaptive least Kth-order squares* estimator by Lee et al. (1998). The estimated robust variance factor may be used to iteratively update σ_y^2, and thus the inlier set $\widehat{\mathcal{G}}$ as the denominator in (4.371) refers to a sample with no outliers.

Alternative robust methods are the mode estimator of Chernoff (1964), which takes the centre of that interval of given length which contains the largest number of data and the half sample mode estimator of Bickel and Fruehwirth (2006). This estimate can be generalized to higher dimensions, leading to what is called the minimum-volume ellipse estimate, cf. Rousseeuw and Leroy (1987); Jolion et al. (1991).

Within iterative estimation procedures, the robust estimate $\widehat{\sigma}_y$ will be applied after each iteration if appropriate approximate values are also available before the first iteration.

The five procedures can now easily be characterized.

4.7.4 Robust Maximum Likelihood-Type Estimation

Procedures of robust maximum likelihood-type estimation replace the optimization function $\Omega = \sum_n y_n^2 = \sum_n w_n v_n^2$ of the weighted least squares solution by the sum $\Omega = \sum_n \rho(y_n)$ of a less increasing function $\rho(y_n)$ of the normalized residuals.

4.7.4.1 Modified Weights and Influence Function

The simplest way to realize this is to iteratively modify the weights as a function of the residuals, namely reducing the weight of observations which got large residuals in the previous iteration.

The reweighting scheme can be formally derived. The necessary conditions for a minimum of (4.365) are

$$\frac{\partial \Omega}{\partial x_u} = \sum_n \frac{\partial \rho(y_n)}{\partial x_u} = \sum_n \underbrace{\frac{\partial \rho(y_n)}{\partial y_n}}_{\rho'(y_n)} \frac{\partial v_n}{\partial x_u} \sqrt{w_n} \cdot \frac{v_n \sqrt{w_n}}{v_n \sqrt{w_n}} = \sum_n \underbrace{\frac{\rho'(y_n)}{y_n} w_n}_{} \cdot \frac{\partial v_n}{\partial x_u} v_n = 0 \quad (4.374)$$

If we chose $\rho(y) = y^2/2$,[9] the first factor $\rho'(y_n)/y_n = 1$ would disappear, and we would have the classical condition for the weighted LS-estimation (cf. (4.38), p. 84) with $\partial v_n/\partial x_u = a_{nu}$ and $\partial \Omega/\partial x_u = 2\boldsymbol{a}_u^{\mathsf{T}} \boldsymbol{W}_{ll} \boldsymbol{v}(\boldsymbol{x}) = 0$.

Thus in (4.374) the scalars $\rho'(y_n)/y_n\, w_n$ can be interpreted as modified weights. This gives rise to a simple update scheme. The weight $w_n^{(\nu+1)}$ in the next iteration $\nu+1$ depends on the original weight $w_n^{(0)}$ and the normalized residual $y_n^{(\nu)}$ of the previous iteration, leading to the modified weights

modified weights

$$w_n^{(\nu+1)} = w\left(y_n^{(\nu)}\right)\ w_n^{(0)}\,. \quad (4.375)$$

The weight factor

$$w\left(y_n^{(\nu)}\right) = \frac{\rho'\left(y_n^{(\nu)}\right)}{y_n^{(\nu)}} \quad (4.376)$$

depends on the normalized residuals $y_n^{(\nu)} = v_n^{(\nu)} \sqrt{w_n^{(0)}}$ in an intuitive manner:

1. The weight increases with the slope ρ' of ρ. The function

$$\psi(y) = \rho'(y) \quad (4.377)$$

   is what is called the *influence function* (Hampel et al., 1986). For large residuals, the influence function should be bounded or even zero.

   influence function

2. The weight decreases with the size of the residual. Therefore, *maximum likelihood-type estimation* using $\rho(y)$-functions other than y^2 can be realized by an LS-estimation technique with modified weights.

This *method of modified weights* can therefore be implemented in any computer program for least squares estimation by a slight modification, namely by reducing the weights of all observations after each iteration.

Depending on the assumptions about the outlier distribution and the type of approximation, we arrive at classical choices for the minimization function, cf. Table 4.5, p. 149:

[9] When using $\rho(y) = y^2/2$ the sum $\sum_n \rho(y_n)$ is only half the weighted sum of the squares of the residuals. This must be taken into account when using Ω for estimating the variance factor.

1. L_2: LS-estimation uses the minimization function

$$\rho_{L_2}(y) = \frac{1}{2}y^2 = -\log(p(y))\,. \tag{4.378}$$

As a result, the influence function $\rho'_{L_2}(y) = y$ is not bounded and the weight function $w_{L_2}(y) = 1$ is constant, independent of the size of the residuals, cf. the first row in Table 4.5.

2. L_1: The simplest outlier model assumes that all observations follow a doubly exponential distribution or *Laplace distribution*, $p(y) = \exp(-|y|)$. Since this density function decays less than the Gaussian, large deviations of the observations from their mean are more likely. This leads to the minimization function (4.364), p. 144,

$$\rho_{L_1}(y) = |y| \tag{4.379}$$

effectively minimizing the L_1-norm of the residuals. The optimization function is convex, which guarantees finding a global minimum if the function y is linear in the unknown parameters. The influence function $\rho'_{L_1}(y) = \operatorname{sign}(y)$ is bounded, but the weight function $w_{L_1}(y) = 1/|y|$ is not bounded at y approximately zero, which causes numerical difficulties when estimating with the method of modified weights, cf. the second row of Table 4.5. In the case of a linear model, L_1-norm minimization can be realized by linear programming, cf. below.

3. L_{12}, $L_{12\mathrm{s}}$: Assuming inliers and outliers are separable, the size of inliers is bounded by 1, the inliers are Gaussian distributed within ± 1, the outliers are Laplacian distributed outside this interval, and the joint density is smooth, we arrive at what is called the *Huber estimator* with

Huber estimator

$$\rho_{L_{12}}(y) = \begin{cases} \frac{1}{2}y^2, & \text{if } |y| \le 1 \\ -1/2 + |y|, & \text{if } |y| \ge 1 \end{cases}\,. \tag{4.380}$$

Its influence function is linear around 0 and constant for larger values. The weight function is bounded, but also decays with $|y|$, cf. the third row of Table 4.5:

$$w_{L_{12}}(y) = \min(1, 1/|y|)\,. \tag{4.381}$$

In order to arrive at a closed expression, this function can be replaced by the nonpartitioned smoothed function

$$\rho_{L_{12\mathrm{s}}}(y) = \sqrt{1+y^2} - 1\,, \tag{4.382}$$

which has the same curvature at $y = 0$ and also increases linearly for large $|y|$. The function is again convex, cf. fourth row of Table 4.5. The smooth weight function

$$w_{L_{12\mathrm{s}}}(y) = \frac{1}{\sqrt{1+y^2}} \tag{4.383}$$

decays with $|y|$ for large residuals. However, large observations still have an influence on the result, as the influence function is not converging to zero for large residuals.

4. $L_{2\mathrm{t}}$, $L_{\exp}$: Assuming inliers and outliers to be separable, the size of inliers to be bounded by $c\sigma_n$, the inliers to be Gaussian distributed within $\pm c\sigma_n$, the outliers to be *uniformly distributed* outside this interval, and the joint density to be continuous, we arrive at

$$\rho_{L_{2\mathrm{t}}}(y) = \min\left(\frac{1}{2}y^2, \frac{1}{2}c^2\right) \tag{4.384}$$

for the truncated L_2-norm without referring to the outlier percentage. As discussed above, this is rigorously equivalent to an ML-estimation only if the outliers are bounded, as for positive c the uniform distribution symmetric to $y = 0$ is bounded. In

contrast to the two previous models, the penalty for outliers does not depend on their size, cf. fifth row of Table 4.5 with $c = 1$. Observe, the influence function $\psi(y) = \rho'(y)$ is not monotonously increasing but redescending. Moreover, it has steps at ± 1, which may cause oscillations within the iteration sequence. Consequently the optimization function is not convex, which significantly increases the complexity of the optimization problem.

Again we can replace the function by a nonpartitioned one using an exponential function already proposed by Krarup et al. (1980),

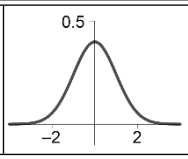
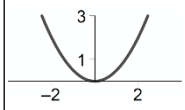
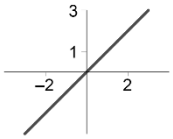
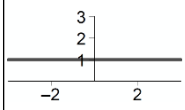
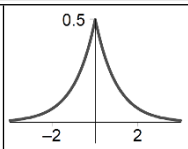
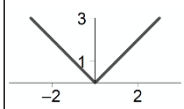
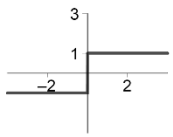
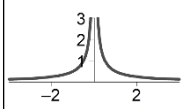
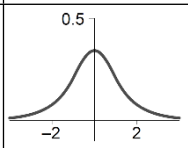
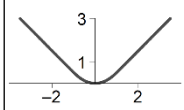
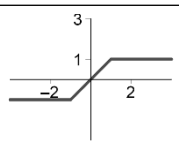
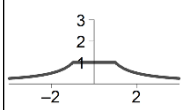
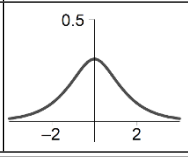
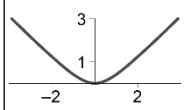
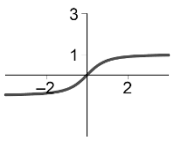
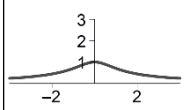
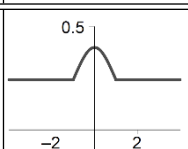
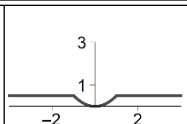
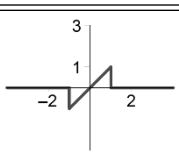
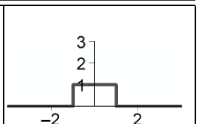
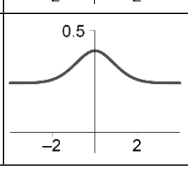
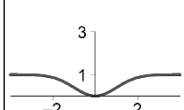
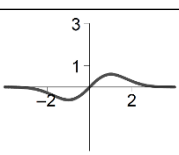
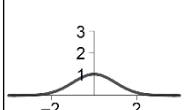
$$\rho_{\exp}(y) = 1 - \exp\left\{-\frac{1}{2}\frac{y^2}{c^2}\right\} . \tag{4.385}$$

The smooth redescending influence function and the simple weight function

$$w_{\exp}(y) = \exp\left\{-\frac{1}{2}\frac{y^2}{c^2}\right\} \tag{4.386}$$

are shown in row six of Table 4.5. The function $\rho_{\exp}(y)$ – except for scaling – is an approximation of the negative logarithm of the mixture density (4.361).

Table 4.5 Functions for robust estimators. **Columns:** Likelihood/pseudo-likelihood function $p(y)$, negative log-likelihood/pseudo log-likelihood function $-\log p(y)$, influence function $\psi(y) = \rho'(y)$, weight function $w(y)$. **Rows 1 to 6:** (1) least squares L_2-norm, (2) least absolute residuals L_1-norm, (3) mixture L_{12} of L_1 and L_2-norm, (4) smooth version L_{12s} of L_{12}, (5) truncated L_2-norm L_{2t}, (6) exponential approximation $L_{\exp}$ of L_{2t}. The last two robust estimators cannot be derived from a proper density function. The graphs are shown with $c = 1$

		$p(y)$	$\rho(y) \propto -\log p(y)$	$\psi(y) = \rho'(y)$	$w(y) = \rho'(y)/y$
1	L_2				
2	L_1				
3	L_{12}				
4	L_{12s}				
5	L_{2t}				
6	$L_{\exp}$				

4.7.4.2 Iteration Scheme

With good approximate values, the following iteration scheme is recommended. Before each iteration, the variance factor needs to be estimated robustly following Sect. 4.7.3. The original weights $w_n^{(0)}$ or weight matrices $\mathbf{W}_i^{(0)}$ for groups of observations are determined from

$$w_n^{(0)} = \frac{\widehat{\sigma}_{0,K}^2}{\sigma_{l_n}^{a,2}}, \qquad \text{or} \qquad \mathbf{W}_i^{(0)} = \widehat{\sigma}_{0,K}^{-2}\, \boldsymbol{\Sigma}_{l_i l_i}^{a,-1}, \tag{4.387}$$

for some suitably chosen number K of inliers. The choice of K is uncritical as the number of outliers is estimated, cf. (4.372). The initial variances $\sigma_{l_n}^{a,2}$ or covariances $\boldsymbol{\Sigma}_{l_i l_i}^{a}$ are not changed during the iteration process, just the weights, cf. (4.375), p. 147:

$$w_n^{(\nu+1)} = w\left(y_n^{(\nu)}\right)\, w_n^{(0)}, \quad \text{or} \quad \mathbf{W}_i^{(\nu+1)} = w\left(y_i^{(\nu)}\right)\, \mathbf{W}_i^{(0)}. \tag{4.388}$$

In the first iteration, the residuals depend on the given approximate values.

1. The first iterations use the Huber estimator ($w_{L_{12}}(y)$ or its smooth counterpart $w_{L_{12s}}(y)$ in (4.383). For linear problems, convergence to the global optimum is guaranteed.
2. The next iterations use the exponential weighting function $w_{\exp}(y)$ in (4.386), with $c \in [2, 3]$ until convergence. This guarantees large outliers have no influence on the result anymore.
3. In a final iteration, all observations with sufficiently small factor $w_{\exp}(y)$ (say, larger than 0.5) are used in an ML-estimation with the original weights, assuming a Gaussian distribution for the measurement deviations.

The procedure thus identifies inliers in the first iterations. This first step can be interpreted as the result of an oracle in the sense of the oracle machine. The last iteration takes this decision as given, performs a standard estimation procedure with inliers only, and when performing statistical tests assumes they are normally distributed.

Experience shows that the method of modified weights can be expected to work well if the outliers are not too large (say not beyond 20%) beyond their size, the percentage of outliers is small, such that outliers do not interfere, and the design is homogeneous, i.e., there is a sufficiently large local redundancy, thus without leverage points, say with redundancy numbers above 0.1. Zach (2014) reports about an improvement of finding inliers by regularizing the adapted weights.

The advantage of robust ML-type estimation is its favourable computational complexity, which is of the order $O(U^3 + NU^2)$ in the worst case. This allows us to use it for a large number U of parameters, where sparse techniques may be applied to further reduce complexity.

If ML-type estimation is integrated into a Gauss–Newton-type iteration scheme, it is not able to handle large outliers, since already the first iteration usually leads to corrections of the approximate values which prevent convergence. Here the method of conjugate gradients is favourable (Gründig, 1975): Theoretically, it requires U iterations for solving a linear equation system. Therefore, each iteration step only leads to small corrections of the parameters. Reweighting thus can be performed much earlier within the complete estimation procedure, which allows us to capture much larger outliers than with a Gauss–Newton scheme.

4.7.5 L_1-Norm Minimization

The minimizing of the L_1-norm $\sum_n |v_n/\sigma_{l_n}|$ can be realized by linear programming if the model is linear in the unknown parameters (Boyd and Vandenberghe, 2004).

A linear program is an optimization task of the form

$$\boldsymbol{y}^* = \text{argmax}_y \{\boldsymbol{c}^\mathsf{T}\boldsymbol{y} \mid \mathsf{B}\boldsymbol{y} \le \boldsymbol{b}\}\,, \tag{4.389}$$

where the inequalities are read rowwise and the unknown parameters only occur linearly in the optimization function and in the constraints. The optimal value $\boldsymbol{y}^*$ can be found if the inequality constraints do not contradict, e.g., for $y < 0$ and $y > 1$. The solution may be infinite, e.g., for $y > 10$, or not unique, e.g., for $y \in [0, 1]$.

The transfer of the L_1-norm minimization problem into a linear program can be realized in the following manner. For simplicity, let us minimize

$$\Omega = |\boldsymbol{v}|_1 = \sum_n |v_n| \quad \text{given} \quad \boldsymbol{l} + \boldsymbol{v} = \mathsf{A}\boldsymbol{x} \tag{4.390}$$

with given vector $\boldsymbol{l}$, matrix A, and unknown vectors $\boldsymbol{x}$ and $\boldsymbol{v}$. We first represent each residual by a pair of nonnegative numbers

$$[s,t]_n = \begin{cases} [+v_n, 0] \text{ if } v_n \ge 0 \\ [0, -v_n] \text{ if } v_n < 0 \end{cases}. \tag{4.391}$$

Thus the two vectors $\boldsymbol{s} = [s_n] \ge 0$ and $\boldsymbol{t} = [t_n] \ge 0$ are positive and yield

$$\boldsymbol{v} = \boldsymbol{s} - \boldsymbol{t}\,. \tag{4.392}$$

We now define the vector of unknown parameters,

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{t} \\ \boldsymbol{x} \end{bmatrix}. \tag{4.393}$$

Then the optimization function can be written as

$$\Omega = \sum_n |s_n - t_n| = |\boldsymbol{s} - \boldsymbol{t}|_1 = \underbrace{[\mathbf{1}_N^\mathsf{T}, \mathbf{1}_N^\mathsf{T}, \mathbf{0}_U^\mathsf{T}]}_{:=\boldsymbol{c}} \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{t} \\ \boldsymbol{x} \end{bmatrix} = \boldsymbol{c}^\mathsf{T}\boldsymbol{y}\,. \tag{4.394}$$

The N-equality $\mathsf{A}\boldsymbol{x} = \boldsymbol{l} + \boldsymbol{v}$ with (4.392) can be written as two inequality constraints, namely

$$\mathsf{A}\boldsymbol{x} \le \boldsymbol{l} + \boldsymbol{s} - \boldsymbol{t} \quad \text{and} \quad \boldsymbol{l} + \boldsymbol{s} - \boldsymbol{t} \le \mathsf{A}\boldsymbol{x}\ , \tag{4.395}$$

or together with the positivity constraints for $\boldsymbol{s}$ and $\boldsymbol{t}$,

$$\underbrace{\begin{bmatrix} -\mathsf{I} & \mathsf{I} & \mathsf{A} \\ \mathsf{I} & -\mathsf{I} & -\mathsf{A} \\ -\mathsf{I} & \mathsf{0} & \mathsf{0} \\ \mathsf{0} & -\mathsf{I} & \mathsf{0} \end{bmatrix}}_{\mathsf{B}} \underbrace{\begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{t} \\ \boldsymbol{x} \end{bmatrix}}_{\boldsymbol{y}} \le \underbrace{\begin{bmatrix} \boldsymbol{l} \\ -\boldsymbol{l} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\boldsymbol{b}}. \tag{4.396}$$

Simplifications are possible depending on the software package used. In all cases it is recommended to exploit the sparsity of matrix B.

4.7.6 Complete Search for Robust Estimation

Complete search determines the unknown parameters for all *minimal* combinations of observations using a direct method. The comparison of the different solutions can be based on the truncated L_{2t}-norm applied to *all* observations, and thus on the optimization

function

$$\Omega = \sum_n \rho_{2t}\left(\frac{v_n}{\sigma_{l_n}}\right) = \sum_n \min\left(\frac{1}{2}\left(\frac{v_n}{\sigma_{l_n}}\right)^2, \frac{1}{2}c^2\right) \tag{4.397}$$

for some critical value c for the normalized residuals, e.g., $c = 3$. Observe, if we would only evaluate the function for the inliers, the minimum would be achieved for an arbitrary minimal set of observations, as then the residuals would be zero. The constant $\frac{1}{2}c^2$ ensures that outliers are counted.

For example, if we have a procedure for the direct solution of the parameters $\boldsymbol{x} = \boldsymbol{d}(\boldsymbol{l}_s)$ from a minimal subset $\boldsymbol{l}_s$ of observations, a procedure for minimizing (4.397) thus could be the following:

1. Initiate the optimization function $\Omega_{\min} := \infty$ and the best solution $b_{\min} :=$ none.
2. For all minimal sets $\boldsymbol{l}_s$ of observations determine $\boldsymbol{x}_s$, and $\boldsymbol{v}_s = \boldsymbol{f}(\boldsymbol{x}_s) - \boldsymbol{l}$ of all observations and Ω_s. If $\Omega_s < \Omega_{\min}$ then set $\Omega_{\min} := \Omega_s$ and the best solution $b_{\min} := \{\boldsymbol{l}_s, \boldsymbol{x}_s\}$.
3. Report the best solution $b_{\min} = \{\boldsymbol{l}_s, \boldsymbol{x}_s\}$ and the corresponding $\Omega_{\min}$.

Complete search guarantees an optimal solution for the outlier problem. Instead of a direct method also an iterative method could be used if good enough approximate values are available to ensure convergence to the optimum.

However, the optimum solution is obtained at the expense of high algorithmic complexity. Complete or exhaustive search checks all possible configurations of good and bad observations $l_n, n = 1, ..., N$, to find the optimal solution. For example, to determine the mean of three observations, we have $2^3 - 1 = 7$ alternatives for in an outliers, (000), (001), (010), (011), (100), (101), (110), where a 0 indicates a good and a 1 indicates a bad observation. However, only those combinations of good observations are admissible where the number G of good observations is at least equal to the number U of unknown parameters, as with less than U observations the parameters cannot be determined. This is equivalent to the situation where the number B of bad observations is larger than or equal to the redundancy $R = N - U$.

With the number N of observations and the maximum number $B \in [0, R]$ of expected bad observations, the number of trials therefore is

$$\sum_{B=0}^{N-U} \binom{N}{B} < 2^N . \tag{4.398}$$

We thus check all sets of bad observations with $B \in [0, R]$ or all sets of good observations with $G \in [U, N]$.

Table 4.6 Number of trials (4.398) for complete search as a function of the number U of unknowns and the number N of observations

t | N =	2	3	4	5	6	7	8	9	10	11	12
U = 1	3	7	15	31	63	127	255	511	1023	2047	4095
2	1	4	11	26	57	120	247	502	1013	2036	4083
3	–	1	5	16	42	99	219	466	968	1981	4017
4	–	–	1	6	22	64	163	382	848	1816	3797
5	–	–	–	1	7	29	93	256	638	1486	3302
6	–	–	–	–	1	8	37	130	386	1024	2510
7	–	–	–	–	–	1	9	46	176	562	1586
8	–	–	–	–	–	–	1	10	56	232	794

For values $N = 2, ..., 12$ and $U = 1, ..., 8$, the number of trials are given in Table 4.6. For example, only seven out of the eight alternatives are admissible for estimating the mean ($U = 1$) from three ($N = 3$) observations, as at least one observation is required. Assuming all three observations to be bad, the alternative (111), would leave no observation for determining the mean value; thus, this set would not be admissible, and the situation need not to be tested.

The number of trials rapidly increases with the number of observations, namely exponentially. This limits the applicability of complete search to problems with a small number of observations and with a direct solution for the parameters.

4.7.7 Random Sample Consensus (RANSAC)

Random Sample Consensus (Fischler and Bolles, 1981; Raguram et al., 2013), similarly to complete search, also aims at finding a set of good observations. It relies on the fact that the likelihood of hitting a good configuration by *randomly choosing* a small set of observations is large. The number of necessary trials conceptionally does not depend on the number of observations N, which is why the method is favourable especially for large N. This advantage, however, is obtained at the expense of not being able to guarantee a solution free of gross errors. The procedure assumes that the following information is available:

- a set of observations or observational groups $\boldsymbol{l}_i, i \in \mathcal{I} = \{1, ..., I\}$, called observations in the following. Such an observation may, e.g., be a set of coordinate pairs for determining a best fitting line.
- a method $\widehat{\boldsymbol{x}} = \boldsymbol{d}(\{\boldsymbol{l}_i\})$ for the direct solution of the U unknown parameters $\boldsymbol{x}$ from a minimum or a small set $\{\boldsymbol{l}_i\}$ of selected observations.
- a criterion for checking the consistency of the other observations $\boldsymbol{l}_j, j \neq i$, with a parameter vector $\boldsymbol{x}$.
- the maximum number t of trials.

The procedure consists of the following steps, which we will discuss in more detail:

1. Randomly choose a minimal subset $\{\boldsymbol{l}_s\}, s \in \mathcal{S} \subset \mathcal{I}$, of S observations $\boldsymbol{l}_s, s \in \mathcal{I}$, necessary to derive the U parameters directly. For example, when determining a circle from points, we choose the minimum of $S = 3$ points for directly determining the circle parameters.
2. Determine the unknown parameters from this chosen set *directly* using a direct solution of the parameters $\widehat{\boldsymbol{x}} = \boldsymbol{d}(\{\boldsymbol{l}_s\})$ from the given set of observations. It might yield several solutions e.g., when determining a 2D point from two distances from given points.
3. Check the consistency of each solution with the other observations. This might be achieved by checking the prediction error $\widehat{\boldsymbol{v}}_j = \boldsymbol{l}_j - \boldsymbol{f}_j(\widehat{\boldsymbol{x}})$ for all observations $\boldsymbol{l}_j, j \in (\mathcal{I} \backslash \mathcal{S})$, not in the chosen set.
4. If consistency is achieved, then the procedure terminates with success. Otherwise, if the number t of trials is reached, then the procedure terminates with failure. Otherwise continue with step 1.

The probability that this procedure provides a solution depends on the possibility of avoiding false decisions such as accepting configurations with outliers and rejecting configurations without outliers. This depends on 1.) the noise level, since in the case of noise-free data, the separation of correct and false decisions would be perfect, 2.) the outlier rate, since high outlier rates increase the likelihood of accepting wrong samples, and 3.) the configuration of the data points, as outliers may mimic good configurations.

The breakdown point of RANSAC is 50% since outliers may mimic an inlier set. However, if the outlier distribution is uniform, then RANSAC may be successful if more than 50% outliers are present.

We now discuss each of the steps.

4.7.7.1 Sampling Observations in RANSAC

The number S of observations which are checked for outliers should be as small as possible since for larger S the likelihood of hitting a good sample with only inliers decreases. Observe, the sample size S is smaller than the number of unknowns if the observation gives rise to more than one constraint on the unknown parameters. For example, when estimating a 2D homography, which has $U = 8$ degrees of freedom, only $S = 4$ points are necessary, as each point gives rise to two constraints.

It might be useful to control the choice in order to achieve different sets in each trial. This is only necessary for small $\mathcal{I}$ when we, in the extreme case, may use all, i.e., $\binom{I}{S}$, possible, S-tuples from all I observations, instead of randomly sampling. For large I the likelihood for randomly drawing the same sample is very small.

If some quality measure for the observations is available which indicates whether an observation is likely to be an inlier, this can be used to sort the samples accordingly (Chum and Matas, 2005).

It is also advisable to avoid unstable configurations. For example, when fitting a straight line through given points, we may not allow point pairs which are too close together. An alternative is binning the observations, and then taking samples from different or even nonneighbouring bins in order to avoid the generation of unstable sets. Obviously, this type of selection is application-dependent.

In both cases, the necessary number of trials for getting a solution will be smaller since it is more likely to hit a good sample; thus, both remedies help to increase the likelihood of finding a solution.

4.7.7.2 Solution for Parameters

Solving for the parameters may yield multiple solutions. All are treated equally and passed on to the next step.

At this point, the configuration of the chosen set may be checked using the acceptability of the achieved precision $\Sigma_{\widehat{x}\widehat{x}}$ of the parameters $\widehat{\boldsymbol{x}}$ with respect to a specified criterion matrix C. For example, if the set of observations is minimal, the covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ can be determined by implicit variance propagation, cf. (2.7.5), p. 43. The reference precision, specified by a criterion matrix C (cf. Sect. 4.6.2.3, p. 120), may be derived from a realistic desired configuration and taking the assumed variances of the observations. This check on acceptability therefore does *not* require prior knowledge about the actual precision of the observations, as only the relative precision is checked. In addition, the parameters $\widehat{\boldsymbol{x}}$ may be checked if prior knowledge about their range is available.

If no direct solution is available, we may take an iterative procedure, provided that the space of solution can be covered with approximate values for the parameters such that an iterative solution is guaranteed to converge to the global optimum starting at least at one set of the approximate values.

It is advantageous to augment this step by deriving nonminimal sets of inliers from the current best parameter and perform an optimal estimation in order to reach more stable solutions, a method proposed by Chum et al. (2003).

4.7.7.3 Check for Consistency

It is advantageous to apply variance propagation to test the residuals using the test statistic

$$\underline{y}_j^2 = \boldsymbol{v}_j^\mathsf{T} \Sigma_{v_j v_j}^{-1} \boldsymbol{v}_j \sim \chi_{n_j}^2 \,, \quad \Sigma_{\widehat{v}_j \widehat{v}_j} = \Sigma_{l_j l_j} + \mathsf{A}_j^\mathsf{T} \Sigma_{\widehat{x}\widehat{x}} \mathsf{A}_j \tag{4.399}$$

using the n_j-vector $\boldsymbol{v}_j = \boldsymbol{f}_j(\widehat{\boldsymbol{x}}) - \boldsymbol{l}_j$ of the prediction errors and applying variance propagation with the Jacobian $\mathsf{A}_j^\mathsf{T} = \partial \boldsymbol{f}_j / \partial \boldsymbol{x}$ (Raguram et al., 2009).

This requires knowledge about the uncertainty $\Sigma_{l_j l_j}$ of the observations, which can be provided by the calling routine. Alternatively, in the case of sufficiently high redundancy the uncertainty may be robustly estimated from the residuals. Assuming an outlier rate of less than ε, a robust unbiased estimate of the variance factor σ_0^2 can be derived from (4.371), p. 146.

Since for large data sets this verification step is computationally costly, it is recommendable to determine the consistency with an increasing number of observations and determine whether the parameters are likely to result from an inlier set as proposed by Matas and Chum (2005).

4.7.7.4 Stopping Criterion for Sampling in RANSAC

Consistency can be defined in several ways: A certain minimum percentage of consistency tests may have to be accepted, or equivalently, a certain maximum percentage of observations may be allowed to be outliers. Alternatively, in a second step an estimation may be performed with all observations accepted in the first step and the result needs to have a certain quality. One possibility and is to apply the maximum likelihood-type evaluation function for all N observations as proposed by Torr and Zisserman (2000, Eq. (17)), using the truncated L_2-norm $L_{2\mathrm{t}}$ from (4.384), p. 148,

$$\sum_{n=1}^{N} \rho(e_n)\,, \qquad \text{with} \qquad \rho(x) = \min(x^2/2, c^2/2)\,, \tag{4.400}$$

where e_n is (1) the residual $\widehat{v}_n$, (2) the normalized residual $\widehat{v}_n/\sigma_n$ using the standard deviation of the observation or (3) the standardized residual $\widehat{v}_n/\sigma_{\widehat{v}_n}$ using the standard deviation of the residual from (4.399), the choice depending on the rigour of the implementation. The method can easily be transferred to observations $\boldsymbol{l}_i$, which are vector-valued.

Alternatively, the best solution out of all $t_{\min}$ solutions may be chosen, similarly to a complete search, as discussed in the previous Sect. 4.7.6.

The minimum number $t_{\min}$ of trials can be determined if we specify a minimum $P_{\min}$ for the probability P of finding at least one good set of observations in t trials, assuming a certain percentage ε of observations to be erroneous. This probability is Exercise 4.16

$$P = 1 - (1 - (1 - \varepsilon)^S)^t \tag{4.401}$$

(Fischler and Bolles, 1981). The minimum number of trials therefore is (cf. Table 4.7)

$$t_{\min}(P_{\min}, \varepsilon, S) = \frac{\ln(1 - P_{\min})}{\ln(1 - (1 - \varepsilon)^S)}\,. \tag{4.402}$$

Table 4.7 Minimum number $t_{\min}$ of trials for RANSAC, cf. (4.402), as a function of the number S of observations $\boldsymbol{l}_i$, the expected percentage $100\varepsilon\%$ of outliers and for probability $P_{\min} = 99\%$ to hit at least one correct data set

$t_{\min}$ | $\varepsilon =$	10%	20%	30%	40%	50%	60%	70%	80%	90%
$S = 1$	2	3	4	6	7	10	13	21	44
2	3	5	7	11	17	27	49	113	459
3	4	7	11	19	35	70	169	574	4 603
4	5	9	17	34	72	178	567	2 876	46 050
5	6	12	26	57	146	448	1 893	14 389	460 515
6	7	16	37	97	293	1 123	6 315	71 954	4 605 168
7	8	20	54	163	588	2 809	21 055	359 777	46 051 700
8	9	26	78	272	1 177	7 025	70 188	1 798 893	460 517 017

For example, for $S = 3$ when fitting a circle or a plane, and an outlier rate of 50%, at least $t_{\min} = 35$ trials are necessary if the probability $P_{\min}$ of finding at least one correct triple is to be larger than 99%.

The number of trials, though independent of the number N of observations, rapidly increases with the number U of unknown parameters.

Experience shows that the relation (4.402) is too optimistic (Tordoff and Murray, 2002) for several reasons.

1. The relation does not refer to the stability of the configurations. Bad configurations usually refer to bad samples. In this case, when applying a statistical test for evaluating the consensus, the large variance of the parameters leads to small test statistics, which are likely to be erroneously accepted. Therefore it is necessary to eliminate samples with a bad configuration when performing a statistical test. Eliminating bad configurations is equivalent to increasing the number of trials as proposed by Scherer-Negenborn and Schaefer (2010).
2. The relation does not take the observational noise into account. This has two effects: occasionally good examples are rejected and often bad samples are accepted. The effect cannot be compensated for by increasing the number of trials. The noise variance could be used to predict an upper limit on probability of success. How to perform such a prediction is an open question.

Often the outlier rate is not known in advance. Then we can start with a large ε, say 50%, and, if fewer outliers are identified, we can use this new smaller outlier rate to reduce the number of required samples, as proposed in Hartley and Zisserman (2000, p. 120).

Algorithm 3 consists of the following steps:

Algorithm 3: Random sample consensus
$[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}] = \texttt{RANSAC}(\mathcal{D}, P_{\min}, \alpha, \Sigma_{\text{ref}}, f_{\max})$
Input: observations $\mathcal{D} = \{\boldsymbol{l}_i, \Sigma_{l_i l_i}\}, i = 1, ..., I$, $\boldsymbol{l}_i \in \mathbb{R}^d$,
minimum probability $P_{\min}$ for success, significance number α,
reference covariance matrix Σ_{ref}, upper bound $f_{\max}$ for acceptance factor.
Assumption: function $\boldsymbol{x} = \texttt{direct_solution}(\mathcal{s})$ from set $\mathcal{s}$ of size S available.
Output: best solution $x_{\text{best}} = [\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}]$.

Outlier rate: $\varepsilon = 0.5$, variance factor $\widehat{\sigma}_0^2 = 1$;
Number of trials: $T = \ln(1 - P_{\min})/\ln(1 - (1 - \varepsilon)^S)$;
Samples: $\mathcal{S} = \emptyset$, Trial: $t = 1$, Best solution: $x_{\text{best}} = [\mathbf{0}, 0]$, number of inliers $G_{\text{best}} = 0$;
while $t \leq T$ **do**
  repeat Draw sample $\mathcal{s}$ **until** $\mathcal{s} \notin \mathcal{S}$ to guarantee not to draw same sample;
  Samples: $\mathcal{S} = \mathcal{S} \cup \mathcal{s}$;
  Determine set of solutions: $\mathcal{X} = \{[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}]\} = \texttt{direct_solution}(\mathcal{s})$;
  for $x \in \mathcal{X}$ **do**
    Acceptance factor: $f = \sqrt{\lambda_{\min}(\Sigma_{\widehat{x}\widehat{x}}\, \Sigma_{\text{ref}}^{-1})}$;
    if $f < f_{\max}$ **then**
      Trials: $t = t + 1$;
      Prediction errors: $\{y_j^2\} = \{\widehat{\boldsymbol{v}}_j^{\mathsf{T}} \Sigma_{\widehat{v}_j \widehat{v}_j}^{-1} \widehat{\boldsymbol{v}}_j | \boldsymbol{l}_j \in \mathcal{D} \setminus \mathcal{s})\}$, see (4.399), p. 154;
      Variance factor: $\widehat{\sigma}_0^2 = \text{median}(y_j^2)/P_{\chi^2}^{-1}(0.5, d)$;
      Count inliers: $G = \#(y_j^2/\widehat{\sigma}_0^2 < \chi_{d,1-\alpha}^2)$;
      Best solution: **if** $G > G_{best}$ **then** $G_{\text{best}} = G$, $x_{\text{best}} = x$;
      Outlier rate: $\varepsilon = \min(\varepsilon, 1 - G/I)$;
      Number of trials: $T = \ln(1 - P_{\min})/\ln(1 - (1 - \varepsilon)^S)$.
    end
  end
end

1-2 We assume the outlier rate ε is 50% at maximum, as this is the breakdown point of the algorithm if no further knowledge is available. The variance factor is assumed to be 1.

3 The algorithms is initiated with the empty set $\mathcal{S} = \emptyset$ of chosen samples and the minimum number $G_{\text{best}} = 0$ of good observations.

4 The number of trials t is compared to the maximum number T, which possibly is adapted in line [17].

5-6 Only new samples are accepted.

7 The direct solution may lead to a set $\mathcal{X}$ of parameters with their covariance matrix.

8 Each parameter vector of the direct solution together with its uncertainty is taken as a sample.

10 Only samples with a sufficiently good configuration are analysed. The upper bound $f_{\max}$ for the acceptance ratio can be chosen in the range 5-30.

12 The normalized squared prediction errors y_j^2 are determined using some prior knowledge about the precision $\Sigma_{l_i l_i}$ of the observations using the current estimate for the variance factor. This covariance matrix $\Sigma_{l_i l_i}$ needs to be known up to a possibly unknown scale, which can be estimated if the number of observational groups is large enough, say $I > 30$. The prediction errors $\underline{y}_i^2/\widehat{\sigma}_0^2$ are χ_d^2-distributed if $\boldsymbol{s}$ is an inlier set and $\boldsymbol{l}_{l_j}$ is an inlier.

13 The robust estimation of the variance factor $\widehat{\sigma}_0^2$ is reliable only if the number of observations is large. Since the median of a χ_d^2-distributed variable is at the 50% point of the inverse cumulative χ_d^2 distribution $P_{\chi^2}^{-1}(0.5, d)$, we have to compensate for this factor.

14-15 The algorithm assumes the best solution to be the one with the largest number G of inliers. This criterion can be replaced by a search for the solution with the smallest robust sum $\sum_j \rho(y_j)$ of residuals.

16-17 The outlier rate, and consequently the maximum number of trials, is adapted.

Summarizing, the technique again requires the invertibility of the model or approximate values and is only suitable for small S. A universal framework together with an implementation containing the above-mentioned refinements is given by Raguram et al. (2013).

In spite of its power, RANSAC has the drawbacks that the algorithm does not guarantee to provide a solution, even if the number of samples is enlarged. This is in contrast to consensus set maximization algorithms which give this guarantee and in general yield larger consensus sets than RANSAC, however, at the cost of – hitherto – significantly larger computation times (Chin et al., 2015).

4.7.8 Robust Estimation by Clustering

Clustering consists of determining the a posteriori probability density function $p(\boldsymbol{x} \mid \boldsymbol{l})$ of the parameters $\boldsymbol{x}$ under the assumption that the data $\boldsymbol{l}$ are representative of the complete sample. The mode, i.e., the maximum of $p(\boldsymbol{x} \mid \boldsymbol{l})$, is used as an estimate. This is approximated by $p(\boldsymbol{x}|\boldsymbol{l}) \propto \prod_i p(\boldsymbol{l}_i|\boldsymbol{x})$ where the product is taken over all, or at least a large enough set $\{\boldsymbol{l}_i\}$ of S, observations, implicitly assuming these observations are independent. Technically, clustering can be realized using an accumulator consisting of bins, i.e., by partitioning the space of the parameters $\boldsymbol{x}$.

Example: For estimating a translation in 2D between two point sets of unknown correspondence, the accumulator consists of $N_x \times N_y$ bins, covering the space of expected translations in x and y directions, each representing a certain range in the xy space of translations. The accumulator is initiated with zero. Each subset then votes for a certain translation. The bin with the maximum number of votes gives an estimate for the translation.

For example, let us assume that we are given two sets $\boldsymbol{x}_i$ and $\boldsymbol{u}_j$ of 2D points, as in Table 4.8, see Fig. 4.14. Some points in set one are assumed to correspond to points in set two, but differ by an unknown translation, and the correspondence is not known. The task is to find an estimate for the translation. In this case, the number of points is small and we can determine all possible translations $\boldsymbol{t}_{ij} = \boldsymbol{x}_i - \boldsymbol{u}_j$. We find the translation $(2, 3)$ three times, whereas all other translations occur less than three times. Thus we take $(2, 3)$ as an estimate for the translation. The Hough transformation (cf. also Sect. (6.6), p. 282) is a

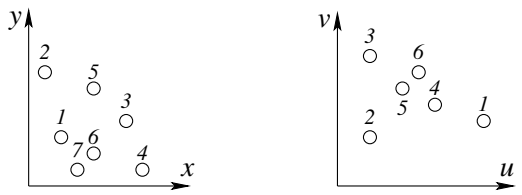


Fig. 4.14 Clustering two sets of points

Table 4.8 Data for point clustering example. Optimal translation (2,3). Matched points $(i, j) \in \{(1, 5), (4, 1), (6, 4)\}$

	x_i	y_i	u_j	v_j
1	2	3	9	4
2	1	7	2	3
3	6	4	2	8
4	7	1	6	5
5	4	6	4	6
6	4	2	5	7
7	3	1	-	-

classical example of this technique (Duda and Hart, 1972; Illingworth and Kittler, 1988; Kiryati et al., 1990). Clustering of determining the position of the projection of a 3D model in an aerial image based on straight line segments is used in Sester and Förstner (1989). Clustering can be used for pose determination (Stockman, 1987) or for object recognition (Ilson, 1997).

Clustering is recommended for problems with few unknowns or a high percentage of gross errors, and in the cases in which enough data can be expected to support the solution (high relative redundancy).

4.7.9 Rules for Choosing Robust Estimation Technique

Without discussing the individual techniques for robust estimation in detail, which would uncover a number of variations and modifications necessary for implementation, the techniques obviously are only applicable under certain more or less precisely known conditions.

The qualitative knowledge about the five robust estimation techniques discussed in the previous section is collected in Table 4.9. It shows the recommendations for each technique depending on eight types of preconditions or requirements. The preconditions or requirements refer to:

- necessary prerequisites:
  1. approximate values are available,
  2. a method for a direct solution for the unknown parameters is available. For L_1-norm minimization, the functional model must be linear in the unknown parameters;

- likelihood of success:
  3. the number of observations is large, say above 50, or above three times the number of unknown parameters,
  4. the reliability is high, i.e., the redundancy numbers are $r_n > 0.1$,
  5. the percentage of errors is high, say above 5%;
- computational complexity:
  6. the number of parameters is low, say below ten,
  7. high speed is irrelevant, and
  8. there exist large outliers, say with a size of above 20% of the size of the observations.

Depending on the preconditions we can evaluate the usefulness of each method as:

(i) "impossible". When the necessary preconditions or requirements are not fulfilled, the technique cannot be used.

(b) "bad". When it is not impossible to apply the technique, but only a few of the preconditions or requirements are fulfilled, the technique shows unfavourable properties, such as unreliability or excessive complexity.

(g) "good". When all required and recommended preconditions or requirements are fulfilled the technique can be used to its full capacity and is usually best.

Table 4.9 Preconditions for five techniques of robust estimation. For each technique, three recommendations may be given: (g) the technique is highly recommendable, (b) the technique might work, but badly, mostly due to high computational complexity, and (i) it is impossible to use the technique since a necessary prerequisite is not fulfilled. Possible preconditions are listed in the first column. Each of the 15 recommendations depends on (1) whether the preconditions with $'+'$ are fulfilled and (2) whether the preconditions with $'-'$ are not fulfilled. Otherwise the criterion is *not decisive* for the recommendation in that column. For example, we have for ML-type, b: If the configuration is not highly reliable ($'-'$), there are large errors ($'+'$) and the error rate is high ($'+'$), then ML-type estimation might work, but badly

	precondition/requirement	complete search g b i	RANSAC g b i	clustering g b i	ML-type estimation g b i	L_1-norm minimization g b i
1	approximate values	−	−	+ −	+ −	
2	direct solution	− −	+ − −	+ − −		+ −
3	many observations	− +	+	+ −	+	+ −
4	few parameters	+ −	+ −	+ −		
5	high reliability		+		+ −	+ −
6	large errors	+	+	+	− +	+
7	high error rate	+		+	− +	+
8	speed unimportant	+ −		+ −	−	−

Obviously, the qualitative reasoning about the estimation methods can be made more precise in a specific context:

- The number of observations (few, many) is actually known in a special situation. It influences the density of the cells in clustering, the relative redundancy, specifically the homogeneity of the design, and the likelihood of finding a good set in RANSAC.
- The number of unknowns (few, many) is also known in a specific situation and can be used to predict the computational effort quite precisely.
- The homogeneity of the design can usually be determined approximately if the number of observations is much higher than the number of the unknowns.
- The size and the percentage of the errors to be expected can be predicted from previous data sets.
- The required speed can usually be derived from the specification of the application and related quite rigorously to the available resources. An example for such a performance prediction in the context of recognition tasks is given by Chen and Mulgaonkar (1990).

The final goal of a formalization for selecting the effective estimation tools would be to leave the choice to the program, which of course requires the standardization of the input/output relation for the procedures. We discuss a representative example which demonstrates that different steps during outlier detection require different procedures.

Example 4.7.12: ***Determination of a homography close to the unit transformation.*** Let us assume two images of a facade are taken with the same camera from two close positions without significantly changing the orientation, and a set $(\mathcal{x}, \mathcal{x}')_i$ of approximately 200 corresponding points in the two images are found. The transformation then is a homography close to a pure translation. We expect a large percentage of small and large outliers due to the specific appearance of the facade. To identify the inliers we proceed step by step.

Step 1: First we estimate an approximate translation (two parameters) in order to obtain a reduced set of candidate matches between image points.

Clustering and RANSAC are strongly recommended. ML-type estimation is not recommended as large errors are to be expected. Complete search is not recommended as the number of observations is large.

Step 2: The second step aims at quickly finding good approximate values for the eight parameters of the homography. Since computational speed is essential in this step and the percentage of errors is large, only RANSAC is advisable. This allows us to eliminate large outliers.

Step 3: The final cleaning of the observations aims at identifying small outliers, which can be assumed to be a small percentage. Therefore, the robust ML-type estimation is highly recommendable. ◇

4.8 Estimation with Implicit Functional Models

4.8.1 Algebraic Structure of the Functional Models 161
4.8.2 Estimation in the Gauss–Helmert Model with Constraints 163
4.8.3 Overview on Estimation Procedures 171
4.8.4 Filtering, Prediction and Collocation 174

So far, we have assumed that the functional model expresses the observed values as an explicit function of the unknown parameters. This situation does not cover all practically relevant cases. As an example, assume a set of observed 3D points with coordinates $\boldsymbol{p}_i = [x, y, z]_i^\mathsf{T}, i = 1, \ldots, I$, to be on an unknown plane represented by its normalized normal vector $\boldsymbol{n} = [k, l, m]^\mathsf{T}$ and its distance d to the origin. Then for an individual point we obtain a single constraint, namely that it is on the plane, expressed as

$$\boldsymbol{p}_i^\mathsf{T}\boldsymbol{n} - d = 0\,, \qquad i = 1, \ldots, I\,, \tag{4.403}$$

together with the normalization constraint

$$|\boldsymbol{n}| = 1\,. \tag{4.404}$$

There is no simple way to find an explicit expression for the observed coordinates x_i, y_i and z_i of $\boldsymbol{p}_i$ as a function of the unknown parameters $\boldsymbol{n}$ and d, in order to arrive at a functional model in the form of a Gauss–Markov model (4.23), p. 82. It would be desirable to find optimal estimates $\widehat{\boldsymbol{n}}$ and $\widehat{d}$ for the plane parameter such that the fitted observations $\boldsymbol{p}_i + \widehat{\boldsymbol{v}}_i$ together with the estimated parameters fulfil the constraints and the weighted sum $\Omega = \sum_i \widehat{\boldsymbol{v}}_i^\mathsf{T} \Sigma_{p_i p_i}^{-1} \widehat{\boldsymbol{v}}_i$ of residuals $\widehat{\boldsymbol{v}}_i$ is minimal, and thereby assume that the given coordinates do have covariance matrices $\Sigma_{p_i p_i}$.

The Gauss–Markov model allows us to easily generate observations, useful for simulations, since the observations are functions of the unknown parameters. However, we also need models which allow us to handle constraints between observations and unknown parameters, i.e., functional relations of the general form $\boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x}) = \boldsymbol{0}$. The base model solving this situation is what is called the *Gauss–Helmert model* (Helmert, 1872).

Gauss–Helmert model, total least squares, errors-in-variables

Since the stochastical model, e.g., $\mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll}$, remains unchanged, this leads to a set of function models which we will now classify depending on their structure. Due to the close relations between the different ways of defining an optimization function, we again describe them as weighted least squares problems; however, when testing, we exploit the

fact that they are (possibly approximate) ML estimates leading to a posterior distribution which is (approximately) Gaussian. The Gauss–Helmert model is closely related to what is called the *total least squares* method or *errors-in-variables* method developed in the area of statistics assuming a stochastic design matrix $\underline{A}$. If the design matrix is perturbed with white noise or with noise having a specific covariance structure, the problem can be transformed into an eigenvalue problem (cf. Golub and van Loan (1996), the review by Gillard (2006), and Mühlich and Mester (1999)). However, it can be expressed as a Gauss–Helmert model, then allowing for arbitrarily correlated perturbations of the design matrix, cf. Schaffrin and Snow (2010). Exercise 4.6

4.8.1 Algebraic Structure of the Functional Models

All functional models for parameter estimation can formally be described as follows: We have three types of constraints between the observations $\boldsymbol{l}$ and the unknowns $\boldsymbol{x}$: G constraints of the form $\boldsymbol{g}(\boldsymbol{l}) = \boldsymbol{0}$ or $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \boldsymbol{0}$ and H constraints of the form $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$. They should hold for the true values $\tilde{\boldsymbol{l}} = \boldsymbol{l} + \tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{x}}$ as well as for the estimated values, namely the fitted observations $\widehat{\boldsymbol{l}} = \boldsymbol{l} + \widehat{\boldsymbol{v}}$ and the estimated parameters $\widehat{\boldsymbol{x}}$,

$$\begin{aligned} \underset{G\times 1}{\boldsymbol{g}}(\tilde{\boldsymbol{l}}) = \boldsymbol{0} \quad &\text{or} \quad \underset{G\times 1}{\boldsymbol{g}}(\widehat{\boldsymbol{l}}) = \boldsymbol{0}\,, \\ \underset{G\times 1}{\boldsymbol{g}}(\boldsymbol{l} + \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{x}}) = \boldsymbol{0} \quad &\text{or} \quad \underset{G\times 1}{\boldsymbol{g}}(\boldsymbol{l} + \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{x}}) = \boldsymbol{0}\,, \\ \underset{H\times 1}{\boldsymbol{h}}(\tilde{\boldsymbol{x}}) = \boldsymbol{0} \quad &\text{or} \quad \underset{H\times 1}{\boldsymbol{h}}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}\,. \end{aligned}$$

In addition, we want to include the special situation where the N observations are an explicit function of the unknowns,

$$\boldsymbol{l} + \tilde{\boldsymbol{v}} = \underset{N\times 1}{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) \quad \text{or} \quad \boldsymbol{l} + \widehat{\boldsymbol{v}} = \underset{N\times 1}{\boldsymbol{f}}(\widehat{\boldsymbol{x}})\,,$$

in order to include the Gauss–Markov model as discussed above.

Observe, we do not address estimation problems with inequalities such as $x_u \leq x_{u0}, u = 1, \ldots, U$. Such problems are discussed in Boyd and Vandenberghe (2004).

Generally the constraints are nonlinear, which leads to procedures where approximate values are iteratively improved to arrive at the final estimate, as in the nonlinear Gauss–Markov model. We therefore start from approximate values for all unknowns, namely $\widehat{\boldsymbol{l}}^a$ for the estimated parameters and $\widehat{\boldsymbol{x}}^a$ for the fitted observations, and derive linear substitute models. In the linearized models, we therefore estimate the corrections to these approximate values,

$$\widehat{\Delta\boldsymbol{x}} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^a \quad \text{and} \quad \widehat{\Delta\boldsymbol{l}} = \widehat{\boldsymbol{l}} - \widehat{\boldsymbol{l}}^a\,. \tag{4.405}$$

For the linearization we need the following Jacobians: for models A and B with generally $U \leq N$,

$$\underset{N\times U}{\mathsf{A}} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\widehat{x}^a}\,; \tag{4.406}$$

for models D and E with generally $U \leq G$,

$$\underset{G\times U}{\mathsf{A}} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{l=\widehat{l}^a, x=\widehat{x}^a}\,; \tag{4.407}$$

for models C, D and E with generally $G \leq N$,

$$\underset{N\times G}{B} = \left(\frac{\partial \boldsymbol{g}(\boldsymbol{l},\boldsymbol{x})}{\partial \boldsymbol{l}}\right)^{\mathsf{T}}\Bigg|_{l=\widehat{l}^a, x=\widehat{x}^a}; \tag{4.408}$$

and for models B and E with constraints between the unknowns with generally $H \leq U$,

$$\underset{U\times H}{H} = \left(\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^{\mathsf{T}}\Bigg|_{x=\widehat{x}^a}. \tag{4.409}$$

All Jacobians are matrices where the number of rows is at least the number of columns. This is the reason why B and H are introduced as transpose Jacobians.

We now characterize the five types of functional models A to E.

A: The Gauss–Markov Model. The already discussed Gauss–Markov model starts from N observations l_n for the U unknown parameters x_u,

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) \quad \text{or} \quad \widehat{\boldsymbol{l}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}). \tag{4.410}$$

The linearized Gauss–Markov model reads

$$\widehat{\Delta \boldsymbol{l}} = \Delta \boldsymbol{l} + \widehat{\boldsymbol{v}} = A\widehat{\Delta \boldsymbol{x}} \tag{4.411}$$

with

$$\Delta \boldsymbol{l} = \boldsymbol{l} - \boldsymbol{f}(\widehat{\boldsymbol{x}}^a), \tag{4.412}$$

$$\widehat{\Delta \boldsymbol{x}} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^a. \tag{4.413}$$

B: Gauss–Markov Model with Constraints. The Gauss–Markov model with constraints starts from N observations l_n for U unknown parameters x_u with H constraints, h_h, between the unknowns:

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) \quad \text{or} \quad \widehat{\boldsymbol{l}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}), \tag{4.414}$$

$$\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}. \tag{4.415}$$

The linearized Gauss–Markov model with constraints between the unknowns therefore is

$$\widehat{\Delta \boldsymbol{l}} = \Delta \boldsymbol{l} + \widehat{\boldsymbol{v}} = A\ \widehat{\Delta \boldsymbol{x}}, \tag{4.416}$$

$$-\boldsymbol{h}(\widehat{\boldsymbol{x}}^a) = H^{\mathsf{T}}\widehat{\Delta \boldsymbol{x}}, \tag{4.417}$$

with $\Delta \boldsymbol{l}$ and $\widehat{\Delta \boldsymbol{x}}$ in (4.412) and (4.413).

C: Model of Constraints Between the Observations Only. The model of constraints between the observations specifies G constraints, g_g, among N observations l_n:

$$\boldsymbol{g}(\boldsymbol{l} + \widehat{\boldsymbol{v}}) = \boldsymbol{0} \quad \text{or} \quad \boldsymbol{g}(\widehat{\boldsymbol{l}}) = \boldsymbol{0}. \tag{4.418}$$

This model is useful for enforcing constraints on an estimate resulting from an unconstrained estimation: This result is then taken as an observation, together with its covariance matrix, in a second estimation step. The linearized model with constraints between observations only hence reads

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + B^{\mathsf{T}}\widehat{\Delta \boldsymbol{l}} = \boldsymbol{0}. \tag{4.419}$$

D: Gauss–Helmert Model. The Gauss–Helmert model specifies G constraints, g_g, among N observations l_n and U unknown parameters x_u:

$$\boldsymbol{g}(\boldsymbol{l}+\widehat{\boldsymbol{v}},\widehat{\boldsymbol{x}})=\boldsymbol{0} \qquad \text{or} \qquad \boldsymbol{g}(\widehat{\boldsymbol{l}},\widehat{\boldsymbol{x}})=\boldsymbol{0}\,. \tag{4.420}$$

The linearized Gauss–Helmert model reads

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a,\widehat{\boldsymbol{x}}^a)+\boldsymbol{A}\widehat{\Delta\boldsymbol{x}}+\boldsymbol{B}^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}}=\boldsymbol{0}\,. \tag{4.421}$$

E: Gauss–Helmert Model with Constraints. The Gauss–Helmert model with constraints between the unknown parameters starts from G constraints, g_g, among N observations l_n and U unknown parameters x_u with additional H constraints, h_h, among the unknowns:

$$\boldsymbol{g}(\boldsymbol{l}+\widehat{\boldsymbol{v}},\widehat{\boldsymbol{x}})=\boldsymbol{0}\ , \qquad \text{or} \qquad \boldsymbol{g}(\widehat{\boldsymbol{l}},\widehat{\boldsymbol{x}})=\boldsymbol{0}\,, \tag{4.422}$$

$$\boldsymbol{h}=\boldsymbol{0} \qquad\qquad . \tag{4.423}$$

The linearized Gauss–Helmert model with constraints between the unknowns therefore is

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a,\widehat{\boldsymbol{x}}^a)+\boldsymbol{A}\ \widehat{\Delta\boldsymbol{x}}+\boldsymbol{B}^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}}=\boldsymbol{0}\,, \tag{4.424}$$

$$\boldsymbol{h}(\widehat{\boldsymbol{x}}^a)+\boldsymbol{H}^{\mathsf{T}}\widehat{\Delta\boldsymbol{x}}\qquad\qquad=\boldsymbol{0}\,. \tag{4.425}$$

The first two models are used most frequently, because the algebraic derivation for the estimates is the most simple one. The Gauss–Helmert model with constraints, model E, is the most general one. All other models can be derived from E by specialization:

- model A is obtained with $\boldsymbol{g}(\widehat{\boldsymbol{l}},\widehat{\boldsymbol{x}})=-\widehat{\boldsymbol{l}}+\boldsymbol{f}(\widehat{\boldsymbol{x}})$ and no constraints, or in the linearized model by setting $\boldsymbol{B}=-\boldsymbol{I}$,
- model B is obtained from A with the added constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}})=\boldsymbol{0}$ or in the linearized model by setting $\boldsymbol{B}=-\boldsymbol{I}$,
- model C is obtained from $\boldsymbol{g}(\widehat{\boldsymbol{l}},\widehat{\boldsymbol{x}})=\boldsymbol{g}(\widehat{\boldsymbol{l}})$ or in the linearized model C by setting $\boldsymbol{A}=\boldsymbol{0}$, and
- model D is obtained from E by omitting the constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}})=\boldsymbol{0}$.

4.8.2 Estimation in the Gauss–Helmert Model with Constraints

As the Gauss–Helmert model contains the other models as special cases we consider it helpful to derive the parameter estimation in this model in more detail.

4.8.2.1 The Model and Its Linearization

The nonlinear model consists of the G and H constraints

$$\boldsymbol{g}(\tilde{\boldsymbol{l}},\tilde{\boldsymbol{x}})=\boldsymbol{0}\,, \qquad \boldsymbol{h}(\tilde{\boldsymbol{x}})=\boldsymbol{0}\,, \qquad \mathbb{D}(\underline{\boldsymbol{l}})=\boldsymbol{\Sigma}_{ll} \tag{4.426}$$

for the U unknown parameters $\tilde{\boldsymbol{x}}$ and the N observations $\tilde{\boldsymbol{l}}$.

We assume that approximate values $\widehat{\boldsymbol{l}}^a$ and $\widehat{\boldsymbol{x}}^a$ for the estimated values $\widehat{\boldsymbol{l}}$ and $\widehat{\boldsymbol{x}}$ are available, in the first iteration achieved by some preprocessing, in the later iteration steps as the result of the previous iteration. Then the estimates are, cf. Fig. 4.15

$$\widehat{\boldsymbol{l}}=\widehat{\boldsymbol{l}}^a+\widehat{\Delta\boldsymbol{l}}=\boldsymbol{l}+\widehat{\boldsymbol{v}}\,, \qquad \widehat{\boldsymbol{x}}=\widehat{\boldsymbol{x}}^a+\widehat{\Delta\boldsymbol{x}}\,. \tag{4.427}$$

Thus we arrive at the mathematical model, including the stochastical model,

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a+\widehat{\Delta\boldsymbol{l}},\widehat{\boldsymbol{x}}^a+\widehat{\Delta\boldsymbol{x}})=\boldsymbol{0} \qquad \boldsymbol{h}(\widehat{\boldsymbol{x}}^a+\widehat{\Delta\boldsymbol{x}})=\boldsymbol{0}\,, \qquad \mathbb{D}(\boldsymbol{l})=\boldsymbol{\Sigma}_{ll}\,. \tag{4.428}$$

We assume the covariance matrix $\boldsymbol{\Sigma}_{ll}$ of the observations is regular.

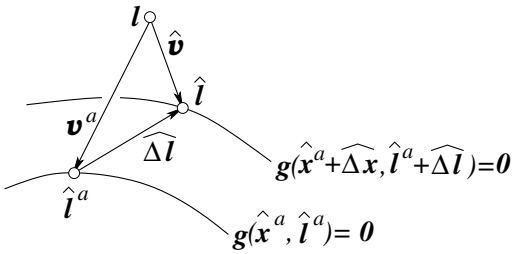


Fig. 4.15 Update of observations and unknowns in the Gauss–Helmert model. The corrections $\widehat{\Delta \boldsymbol{l}} = \widehat{\boldsymbol{l}} - \widehat{\boldsymbol{l}}^a = \widehat{\boldsymbol{v}} - \widehat{\boldsymbol{v}}^a$ of the fitted observations and the estimated residuals are meant to converge to zero

We now linearize at $\widehat{\boldsymbol{l}}^a$ and achieve (up to first-order terms) the linear substitute model

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}, \widehat{\boldsymbol{x}}) = \boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + A\widehat{\Delta \boldsymbol{x}} + B^{\mathsf{T}}\widehat{\Delta \boldsymbol{l}} = \boldsymbol{0} \tag{4.429}$$

$$\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{h}(\widehat{\boldsymbol{x}}^a) + H^{\mathsf{T}}\widehat{\Delta \boldsymbol{x}} = \boldsymbol{0}\,. \tag{4.430}$$

4.8.2.2 The Optimization Function

The goal is to minimize

$$\Omega = \widehat{\boldsymbol{v}}^{\mathsf{T}}\, \Sigma_{ll}^{-1}\, \widehat{\boldsymbol{v}} \tag{4.431}$$

under the constraints (4.429) and (4.430). Due to (4.427), the residuals $\widehat{\boldsymbol{v}}$ are

$$\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}}^a - \boldsymbol{l} + \widehat{\Delta \boldsymbol{l}}\,. \tag{4.432}$$

We use the approximate residuals

$$\widehat{\boldsymbol{v}}^a = \widehat{\boldsymbol{l}}^a - \boldsymbol{l} \tag{4.433}$$

to simplify notation. Then the optimization can be written as

$$\Omega = (\widehat{\boldsymbol{v}}^a + \widehat{\Delta \boldsymbol{l}})^{\mathsf{T}}\, \Sigma_{ll}^{-1}\, (\widehat{\boldsymbol{v}}^a + \widehat{\Delta \boldsymbol{l}}) \tag{4.434}$$

under the constraints (4.429) and (4.430).

4.8.2.3 The Solution of the Linearized Substitute Problem

Using Lagrangian multipliers we need to minimiss

$$\begin{aligned} \boldsymbol{\Phi}(\widehat{\Delta \boldsymbol{l}}, \widehat{\Delta \boldsymbol{x}}) = \frac{1}{2}&(\widehat{\boldsymbol{v}}^a + \widehat{\Delta \boldsymbol{l}})^{\mathsf{T}}\, \Sigma_{ll}^{-1}\, (\widehat{\boldsymbol{v}}^a + \widehat{\Delta \boldsymbol{l}}) + \\ &+ \boldsymbol{\lambda}^{\mathsf{T}}(\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + A\widehat{\Delta \boldsymbol{x}} + B^{\mathsf{T}}\widehat{\Delta \boldsymbol{l}}) + \\ &+ \boldsymbol{\mu}^{\mathsf{T}}(\boldsymbol{h}(\widehat{\boldsymbol{x}}^a) + H^{\mathsf{T}}\widehat{\Delta \boldsymbol{x}})\,. \end{aligned} \tag{4.435}$$

The four partials are to be set to $\boldsymbol{0}$. As $\partial \boldsymbol{a}^{\mathsf{T}}\boldsymbol{b}/\partial \boldsymbol{a}^{\mathsf{T}} = \left(\partial \boldsymbol{b}^{\mathsf{T}}\boldsymbol{a}/\partial \boldsymbol{a}\right)^{\mathsf{T}} = \boldsymbol{b}$, this yields

$$\frac{\partial \boldsymbol{\Phi}}{\partial \widehat{\boldsymbol{\Delta l}}^{\mathsf{T}}} = \Sigma_{ll}^{-1}(\widehat{\boldsymbol{v}}^a + \widehat{\boldsymbol{\Delta l}}) + B\boldsymbol{\lambda} = \mathbf{0}\,, \tag{4.436}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial \widehat{\boldsymbol{\Delta x}}^{\mathsf{T}}} = A^{\mathsf{T}}\boldsymbol{\lambda} + H\boldsymbol{\mu} = \mathbf{0}\,, \tag{4.437}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\lambda}^{\mathsf{T}}} = \boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + A\widehat{\boldsymbol{\Delta x}} + B^{\mathsf{T}}\widehat{\boldsymbol{\Delta l}} = \mathbf{0}\,, \tag{4.438}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\mu}^{\mathsf{T}}} = \boldsymbol{h}(\widehat{\boldsymbol{x}}^a) + H^{\mathsf{T}}\widehat{\boldsymbol{\Delta x}} = \mathbf{0}\,. \tag{4.439}$$

In the following, we eliminate the variables $\widehat{\boldsymbol{v}}^a$ and $\boldsymbol{\lambda}$. We first solve (4.436) for $\widehat{\boldsymbol{\Delta l}}$ as a function of $\boldsymbol{\lambda}$. By multiplication with Σ_{ll} from the left, we have

$$\widehat{\boldsymbol{v}}^a + \widehat{\boldsymbol{\Delta l}} + \Sigma_{ll}B\boldsymbol{\lambda} = \mathbf{0}\,, \tag{4.440}$$

from which we obtain

$$\widehat{\boldsymbol{\Delta l}} = -\Sigma_{ll}B\boldsymbol{\lambda} - \widehat{\boldsymbol{v}}^a\,. \tag{4.441}$$

Inserting into (4.429) yields

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + A\widehat{\boldsymbol{\Delta x}} + B^{\mathsf{T}}(-\Sigma_{ll}B\boldsymbol{\lambda} - \widehat{\boldsymbol{v}}^a) = \mathbf{0} \tag{4.442}$$

or

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + A\widehat{\boldsymbol{\Delta x}} - B^{\mathsf{T}}\widehat{\boldsymbol{v}}^a - B^{\mathsf{T}}\Sigma_{ll}B\boldsymbol{\lambda} = \mathbf{0}\,. \tag{4.443}$$

We use abbreviations for the residuals of the constraints,

$$\boldsymbol{c}_g = -\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + B^{\mathsf{T}}\widehat{\boldsymbol{v}}^a\,, \qquad \boldsymbol{c}_h = -\boldsymbol{h}(\widehat{\boldsymbol{x}}^a)\,, \tag{4.444}$$

and solve for $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda} = W_{gg}(-\boldsymbol{c}_g + A\widehat{\boldsymbol{\Delta x}})\,, \tag{4.445}$$

with the weight matrix of the constraints,

$$W_{gg} = (B^{\mathsf{T}}\Sigma_{ll}B)^{-1}\,. \tag{4.446}$$

We combine (4.437) and (4.439) and use (4.445) and (4.446) to obtain the normal equation system for the Gauss–Helmert model,

$$M\boldsymbol{p} = \boldsymbol{m}: \qquad \begin{bmatrix} A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}A & H \\ H^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\Delta x}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}\boldsymbol{c}_g \\ \boldsymbol{c}_h \end{bmatrix}\,. \tag{4.447}$$

This equation system can be built if the matrix $B^{\mathsf{T}}\Sigma_{ll}B$ is regular, and solved for the corrections $\widehat{\boldsymbol{\Delta x}}$, if the matrix M is regular; we will assume this in the following. The updates $\widehat{\boldsymbol{\Delta l}}$ for the observations can be derived by inserting (4.445) into (4.441),

$$\widehat{\boldsymbol{\Delta l}} = \Sigma_{ll}B(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}(\boldsymbol{c}_g - A\widehat{\boldsymbol{\Delta x}}) - \widehat{\boldsymbol{v}}^a\,, \tag{4.448}$$

with the updated observations and parameters from (4.427) and the residuals from

$$\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}} - \boldsymbol{l} = \widehat{\boldsymbol{v}}^a + \widehat{\boldsymbol{\Delta l}}\,. \tag{4.449}$$

Using the new estimates for the observations and the parameters as approximate values, we arrive at an iteration scheme.

The redundancy R of the system is

$$R = G + H - U\,. \tag{4.450}$$

The estimated variance factor is

$$\widehat{\sigma}_0^2 = \frac{\Omega(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}})}{R} \tag{4.451}$$

with

$$\Omega(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}}) = \widehat{\boldsymbol{v}}^{\mathsf{T}} W_{ll} \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{c}}_g^{\mathsf{T}} W_{gg} \widehat{\boldsymbol{c}}_g = -\widehat{\boldsymbol{c}}_g^{\mathsf{T}} \widehat{\boldsymbol{\lambda}} \,. \tag{4.452}$$

The last two relations can be derived using (4.441), (4.445), and (4.444) at the point of convergence where $\widehat{\Delta \boldsymbol{l}} = \boldsymbol{0}$, $\widehat{\Delta \boldsymbol{x}} = \boldsymbol{0}$, and $\boldsymbol{g}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}}) = \boldsymbol{0}$, thus

$$\widehat{\boldsymbol{v}} = -\Sigma_{ll} B \widehat{\boldsymbol{\lambda}} \,, \qquad \widehat{\boldsymbol{\lambda}} = -W_{gg} \widehat{\boldsymbol{c}}_g \,, \qquad \widehat{\boldsymbol{c}}_g = B^{\mathsf{T}} \widehat{\boldsymbol{v}} \,. \tag{4.453}$$

Remark: If each observational group is involved in only one – possibly vector-valued – constraint, say $\boldsymbol{g}_i$, and the weight matrix $W_{gg} = (B^{\mathsf{T}} \Sigma_{ll} B)^{-1}$ is block diagonal, cf. (4.43), p. 85, then the normal equation matrix can be built up sequentially using normal equation components,

$$N = A^{\mathsf{T}} (B^{\mathsf{T}} \Sigma_{ll} B)^{-1} A = \sum_{i=1}^{I} A_i^{\mathsf{T}} (B_i^{\mathsf{T}} \Sigma_{l_i l_i} B_i)^{-1} A_i \,. \tag{4.454}$$

Otherwise W_{gg} is not block diagonal, and for efficiency reasons the normal equations require the solution of a set of equation systems $B^{\mathsf{T}} \Sigma_{ll} B \; F = A$ for the columns of the matrix $F = (B^{\mathsf{T}} \Sigma_{ll} B)^{-1} A$ such that $N = A^{\mathsf{T}} F$. If the matrix $B^{\mathsf{T}} \Sigma_{ll} B$ is sparse, this is significantly more efficient than inverting this matrix. ⋄

We can show that the covariance matrix of the estimated parameters is contained in the inverse of the normal equation matrix (cf. the proof of (4.164), p. 100),

$$\begin{bmatrix} A^{\mathsf{T}} (B^{\mathsf{T}} \Sigma_{ll} B)^{-1} A & H \\ H^{\mathsf{T}} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_{\widehat{x}\widehat{x}} & U \\ U^{\mathsf{T}} & V \end{bmatrix} . \tag{4.455}$$

Exercise 4.19

The covariance matrix of the residuals is given by

$$\Sigma_{\widehat{v}\widehat{v}} = \Sigma_{ll} B W_{gg} (B^{\mathsf{T}} \Sigma_{ll} B - A \Sigma_{\widehat{x}\widehat{x}} A^{\mathsf{T}}) W_{gg} B^{\mathsf{T}} \Sigma_{ll} \,. \tag{4.456}$$

In the Gauss–Helmert model the evaluation w.r.t. outliers in the observations is equivalent to the evaluation w.r.t. outliers in the constraints. This can be seen as follows: In (4.429) we assume $\widehat{\boldsymbol{l}}^a = \boldsymbol{l}$ and set $\widehat{\boldsymbol{v}}_g = -B^{\mathsf{T}} \widehat{\Delta \boldsymbol{l}}$ and hence with (4.432) arrive at the Gauss-Markov model

$$\boldsymbol{g}(\boldsymbol{l}, \widehat{\boldsymbol{x}}^a) + \widehat{\boldsymbol{v}}_g = -A \widehat{\Delta \boldsymbol{x}} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{v}}_g) := \Sigma_{gg} = W_{gg}^{-1} = (B^{\mathsf{T}} \Sigma_{ll} B)^{-1} \,, \tag{4.457}$$

which yields the same parameters $\widehat{\boldsymbol{x}}$ as when starting from the original Gauss–Helmert model. It assumes the matrix Σ_{gg} has full rank. Then the evaluation w.r.t. outliers in the constraints in the Gauss–Helmert model is identical to the evaluation w.r.t. outliers in observations, if we replace the individual observations or the groups of observations by individual constraints or groups of constraints together with their variances or covariance matrices respectively. For example, if we have no constraints $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$ between the parameters, the redundancy matrix for the G constraints $\boldsymbol{g}(\boldsymbol{l}, \widehat{\boldsymbol{x}}^a)$ reads $R = I_G - A(A^{\mathsf{T}} W_{gg} A)^{-1} A^{\mathsf{T}} W_{gg}$, cf. (4.60) with (4.59), p. 87. This allows us to evaluate the effect of individual constraints or groups of constraints on the parameters or on a subset of the parameters. The evaluation of the effect of the individual observations or of groups of observations onto the estimates is more involving.

Exercise 4.20

4.8.2.4 The Iterative Solution

The solution of the linearized substitute problem leads to corrections for the approximate values. Thus we obtain

$$\widehat{\boldsymbol{l}}^{(\nu+1)} = \widehat{\boldsymbol{l}}^{(\nu)} + \widehat{\Delta \boldsymbol{l}}^{(\nu)} \qquad \widehat{\boldsymbol{x}}^{(\nu+1)} = \widehat{\boldsymbol{x}}^{(\nu)} + \widehat{\Delta \boldsymbol{x}}^{(\nu)} \,. \tag{4.458}$$

Convergence is achieved if the corrections are small compared to their standard deviation,

$$\left|\frac{\widehat{\Delta l}_n^{(\nu)}}{\sigma_{l_n}}\right| \leq T_c\,, \qquad \left|\frac{\widehat{\Delta x}_u^{(\nu)}}{\sigma_{x_u}}\right| \leq T_c\,, \tag{4.459}$$

using a threshold T_c, e.g., 0.01, thus requiring the corrections of all unknown parameters to be less than 1% of their standard deviation. This requires that σ_{x_u} be known. It can either be given by the user or derived from the normal equations.

4.8.2.5 The Algorithm

We give Algorithm 4 for estimation in the Gauss–Helmert model with constraints. It assumes that observed values $\boldsymbol{l}$ with covariance matrix $\boldsymbol{\Sigma}_{ll}$ are given and related to the unknown parameters by constraints $\boldsymbol{c}_g$.

The algorithm is provided for the situation where we have stochastically independent groups $\boldsymbol{l}_i$ of observations which take part in one constraint or a small set of constraints $\boldsymbol{g}_i$, also indexed with i. Reweighting can then be applied to the groups of constraints or the single constraint. This is a useful approach for many applications addressed in our context. In the general case, robust estimation with a reweighting scheme conceptually is difficult, as each observation could take part in all constraints, and thus all constraints would be effected by an outlier in one observation. Therefore, the simplifying assumption of independent groups of observations is reasonable.

groups of observations only take part in a small set of constraints

The parameters $\widehat{\boldsymbol{x}}$ have to fulfil constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$. If no additional constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ are present then the algorithm slightly simplifies with $H = 0$.

The algorithm requires procedures, named c_g and c_h, for determining the residuals $\boldsymbol{c}_g$ and $\boldsymbol{c}_h$ and the corresponding Jacobians A, B, and H, which include the functions $\boldsymbol{g}$ and $\boldsymbol{h}$; they depend on the point of the Taylor expansion. The estimation requires approximate values $\boldsymbol{x}^a$ for all unknown parameters. An approximation σ_{x_u} for the final standard deviations of the parameters is to be provided and used for checking convergence. The iteration process ends either if the parameters do not change more than a threshold T_x or if a preset number `maxiter` of iterations will be reached. The robust iteration is taken as an oracle (in the sense of an oracle machine) which provides all inlying constraints for use in one final nonrobust estimation step (cf. the introduction to this section and Fig. (4.12), p. 143)

robust estimation as oracle for the final nonrobust estimation

The steps of the algorithm are the following:

1–2 The redundancy of the system must be nonnegative; otherwise, no estimation is possible.

3 The iteration sequence is initiated with $\nu = 1$; the approximate values $\widehat{\boldsymbol{x}}^a$ for the estimated parameter, and thus $\widehat{\boldsymbol{x}}^{(1)} = \widehat{\boldsymbol{x}}^a$; and the *given* observations $\boldsymbol{l}$ as first approximations for the *fitted* observations, and hence $\widehat{\boldsymbol{l}}^{(1)} = \boldsymbol{l}$.

3,12,20 The stopping variable s controls the finalization of the iteration scheme. Starting with $s = 0$, the iteration loop is performed until $s = 2$, cf. line 20. First the estimation is iterated until convergence or the maximum number of iterations is reached. Then, if a robust estimation is required, the variable is set to $s = 1$, otherwise to $s = 2$, cf. line 12-15.

5–6 The iteration scheme starts with initializing the residuals of the constraints $\boldsymbol{c}_g$ and $\boldsymbol{c}_h$, based on the approximate values for the estimated parameters and the estimated observations. The functions c_g and c_h therefore contain the functions $\boldsymbol{g}$ and $\boldsymbol{h}$ of the Gauss–Helmert model.

The Jacobians are determined from the estimate of the parameters and the fitted observations of the current iteration. *Only in the first iteration* are the given observations used for the Taylor expansion.

7 The weight matrix of the constraints is determined for all constraints. Since we assume that an observation is only taking part in a small group of constraints, this matrix is

Algorithm 4: Robust estimation in the Gauss–Helmert model with constraints.
$[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}, \widehat{\sigma}_0^2, R] =$
`GaussHelmertModell_E_robust`$(\boldsymbol{l}, \Sigma_{ll}, c_g, c_h, \boldsymbol{x}^a, \boldsymbol{\sigma}^a_{\widehat{x}}, T_x, \texttt{maxiter}, \texttt{robust}, k_X, T_w)$
Input: observed values $\{\boldsymbol{l}, \Sigma_{ll}\}$, number N,
constraint functions $[\boldsymbol{c}_g, A, B] = c_g(\boldsymbol{l}, \widehat{\boldsymbol{l}}, \widehat{\boldsymbol{x}})$, number G,
constraints $[\boldsymbol{c}_h, H] = c_h(\boldsymbol{x})$, number H,
approximate values $\widehat{\boldsymbol{x}}^a_u$, possibly $\sigma^a_{\widehat{x}_u}$,
option `robust`$\in \{\mathsf{true}, \mathsf{false}\}$ for robust estimation,
parameters T_x, `maxiter` for controlling convergence and k_X, T_w robust weighting.
Output: estimated parameters $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = G + H - U$;
2 **if** $R < 0$ **then stop**, not enough constraints;
3 Initiate: iteration $\nu = 0$, approximate values $\widehat{\boldsymbol{l}}^{(\nu)} = \widehat{\boldsymbol{l}}^a = \boldsymbol{l}$, $\widehat{\boldsymbol{x}}^{(\nu)} = \boldsymbol{x}^a$, stopping variable: $s = 0$;
4 **repeat**
5 Residuals and Jacobians for constraints $\boldsymbol{g}$: $[\boldsymbol{c}_g, A, B] = c_g(\boldsymbol{l}, \widehat{\boldsymbol{l}}^{(\nu)}, \widehat{\boldsymbol{x}}^{(\nu)})$, see (4.444), (4.407), (4.408);
6 Residuals and Jacobians for constraints $\boldsymbol{h}$: $[\boldsymbol{c}_h, H] = c_h(\widehat{\boldsymbol{x}}^{(\nu)})$, see (4.444), (4.409);
7 Weight of constraints: $W_{gg} = (B^\mathsf{T}\Sigma_{ll}B)^{-1}$;
8 **if** `robust` **then** $[W_{gg}, R] =$ `ReweightingConstraints`$(R, s, \boldsymbol{c}_g, W_{gg}, k_X, T_w)$;
9 Normal equation system: $[M, \boldsymbol{m}]$ (4.447);
10 **if** M is singular **then stop**: normal equation matrix is singular;
11 Solution and updates of parameter vector: $\widehat{\Delta\boldsymbol{x}}$, see (4.447);
12 **if** $s \equiv 1$ **then** $s = 2$, no new iteration;
13 Set iteration: $\nu := \nu + 1$;
14 **if** $\max|\widehat{\Delta x}_u|/\sigma^a_{\widehat{x}_u} < T_x$ or $\nu =$ `maxiter` **then**
15 **if** `robust` and $s \equiv 0$ **then** $s = 1$ **else** $s = 2$
16 **end**
17 Corrections for fitted observations: $\widehat{\Delta\boldsymbol{l}}$, see (4.448);
18 Update parameters: $\widehat{\boldsymbol{x}}^{(\nu)} = \widehat{\boldsymbol{x}}^{(\nu-1)} + \widehat{\Delta\boldsymbol{x}}$;
19 Update fitted observations: $\widehat{\boldsymbol{l}}^{(\nu)} = \widehat{\boldsymbol{l}}^{(\nu-1)} + \widehat{\Delta\boldsymbol{l}}$;
20 **until** $s \equiv 2$;
21 Covariance matrix of estimated parameters: $\Sigma_{\widehat{x}\widehat{x}}$ (4.455);
22 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \boldsymbol{c}_g^\mathsf{T} W_{gg} \boldsymbol{c}_g / R$ **else** $\widehat{\sigma}_0^2 = 1$.

a block diagonal matrix. **Remark:** This weight matrix reveals the difference to the algebraic minimization procedures, which assume $W_{gg} = I$.

8 If a robust estimation is required, the constraints are reweighted using Algorithm 5. This is done before the first setup of the normal equation system, which allows us to exploit possible discrepancies between observations and approximate values. Alternatively, the reweighting could be performed after the end of the current iteration.

9 The normal equations are set up. If no additional constraints $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ are present, the normal equations slightly change. Also, if the system is large, it is recommendable to exploit the sparsity of the normal equation matrix. If a reduction of the normal equation system (cf. e.g., (4.116), p. 94) is possible, this takes place here.

10–11 Checking the singularity of the normal equation matrix usually is performed during the solution of the normal equation system. The corrections $\widehat{\Delta\boldsymbol{x}}$ to the parameters are determined from the normal equation system, possibly by exploiting the sparse structure.

12 If the robust iterations converge or reach the maximum number ($s = 1$), a final iteration with the inliers and the original weights is performed and the variable is set to $s = 2$.

13–19 The iteration number is increased and the estimated parameters and the fitted observations are updated.

14 The stopping criterion for the parameters is best made dependent on the standard deviation of the parameters. As rigorously determining the precision of the parameters in each iteration usually is not advisable for efficiency reasons, it is useful to refer to some approximate standard deviation $\sigma^a_{\widehat{x}_u}$ of the estimated parameters $\widehat{x}_u$.

17 The corrections $\widehat{\Delta \boldsymbol{l}}$ and $\widehat{\Delta \boldsymbol{x}}$ for the fitted observations and parameters in the case of convergence will be close to zero.

21 The covariance matrix can be determined from the normal equation system. For small problems, this directly results from the inversion of the normal equation matrix. For large systems, the diagonal or block diagonal elements of the covariance matrix can be determined without necessarily determining all elements. In the case of reduction to a subset of the parameters (cf. Sect. 4.2.6, p. 94), additional effort is involved at this point, as discussed in the section on general parameter estimation.

122 The variance factor is determined for positive redundancy based on the residuals of the constraints. The determination of the empirical covariance matrix $\widehat{\Sigma}_{\widehat{l}\widehat{l}} = \widehat{\sigma}_0^2 \Sigma_{\widehat{l}\widehat{l}}$ is to be done outside the algorithm.

The procedure for reweighting the constraints is given in Algorithm 5.

Algorithm 5: Reweighting constraints.
$[\boldsymbol{W}_{gg}, R] =$ `ReweightingConstraints`$(R, s, \boldsymbol{c}_g, \boldsymbol{W}_{gg}, k_X, T_w)$
Input: redundancy R, stopping variable s,
residuals of constraints $\{\boldsymbol{c}_g, \boldsymbol{W}_{gg}\}$,
parameters k_X, T_w controlling robust weighting.
Output: adapted weights $\boldsymbol{W}_{gg}$ and redundancy R.

---

for all constraint groups i **do** test statistic: $X_{g_i} = \sqrt{\boldsymbol{c}_{g_i}^\mathsf{T} \boldsymbol{W}_{g_i g_i} \boldsymbol{c}_{g_i}}$;
Determine robust average: $\mathrm{MED}_X =$ median(X_{g_i});
if $s \equiv 1$ **then** initiate outlier counter $G_{\text{out}} = 0$;
for all constraint groups i **do**
  Weight factor: $w_{g_i} = \rho_s\,(X_{g_i}/(k_X \cdot \mathrm{MED}_X))$;
  if $s \equiv 0$ **then**
    $\boldsymbol{W}_{g_i g_i} := w_{g_i}\, \boldsymbol{W}_{g_i g_i}$
  else
    if $w_{g_i} < T_w$ **then** $\boldsymbol{W}_{g_i g_i} := \boldsymbol{0}$, increment G_{out} by number of constraints in group i
  end
end
if $s \equiv 1$ **then** Redundancy with only inliers $R := R - G_{\text{out}}$.

The algorithm has the following steps:

1 The constraints are assumed to occur in groups indexed with i where no two groups depend on the same observations; thus, they can be treated as statistically independent. The reweighting essentially depends on the Mahalanobis distance of the constraint $\boldsymbol{c}_{g_i}$ from the expected values $\mathbf{0}$, namely the test statistic

test statistic for constraints

$$X_{g_i} = \sqrt{\boldsymbol{c}_{g_i}^\mathsf{T} \boldsymbol{W}_{g_i g_i} \boldsymbol{c}_{g_i}}\,. \tag{4.460}$$

2 In order to adapt to the average size of the test statistic we normalize the arguments of the reweighting function ρ by a robust estimate of the test statistic.

3,12 For the last iteration, we need to determine the redundancy, which depends on the number G_{out} of outlying constraints.

4–12 If a robust estimation is required, the weight matrices of the constraints $\boldsymbol{c}_g$ are reweighted.

The ML-type estimation uses a V-shaped ρ-function, named ρ_0 here as $s = 0$, as discussed in Sect. 4.7 on robust estimation (e.g., the smoothed $L_1 - L_2$-norm optimization function $\rho_{L_{12\mathrm{s}}}$ from (4.382), p. 148).

5,7 The argument $X_{g_i}/(k_x \cdot \mathrm{MED}_X)$ for the ρ_0-function ($s = 0$ here) for updating the weights takes care of the average test statistic MED_X and a critical value k_X, to be specified by the user.
The test statistic corresponds to taking the normalized residual $y_i/k_X = (v_i/\sigma_{l_i})/k_X$.

12 In the last iteration ($s = 1$), strongly reweighted constraints are eliminated. Their number is G_{out}. It may be defined as the number of constraints where the weights w_{g_i} are small enough, say below $T_w = 0.1$. The redundancy needs to be adapted accordingly, cf. line 12.

4.8.2.6 Estimation with Constraints Between the Observations Only

The iterative procedure for estimating the parameters in the Gauss–Markov model with constraints between the parameters can be specialized if only constraints between the observations are present. As we do not have unknown parameters, the procedure slightly changes.

Starting from the nonlinear model $\boldsymbol{g}(\tilde{\boldsymbol{l}}) = \boldsymbol{0}$ and approximate values $\widehat{\boldsymbol{l}}^a$ for the fitted observations, we arrive at the linearized model, cf. (4.429), p. 164,

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}) = \boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + \boldsymbol{B}^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}} = \boldsymbol{0}\,. \tag{4.461}$$

Minimizing Ω from (4.434), p. 164 under these constraints, we have the two partials w.r.t. $\widehat{\Delta\boldsymbol{l}}$ and $\boldsymbol{\lambda}$ (cf. (4.436) and (4.438), p. 165),

$$\frac{\partial \boldsymbol{\Phi}}{\partial \widehat{\Delta\boldsymbol{l}}^{\mathsf{T}}} = \boldsymbol{\Sigma}_{ll}^{-1}(\widehat{\boldsymbol{v}}^a + \widehat{\Delta\boldsymbol{l}}) + \boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{0}\,, \qquad \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\lambda}^{\mathsf{T}}} = \boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + \boldsymbol{B}^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}} = \boldsymbol{0} \tag{4.462}$$

The normal equations for the Lagrangian multipliers $\boldsymbol{\lambda}$ read, cf. (4.445), p. 165

$$\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\boldsymbol{B}\,\boldsymbol{\lambda} = -\boldsymbol{c}_g\,, \tag{4.463}$$

with

$$\boldsymbol{c}_g = -\boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + \boldsymbol{B}^{\mathsf{T}}\widehat{\boldsymbol{v}}^a \tag{4.464}$$

from (4.444), p. 165. This leads to the corrections of the fitted observations, cf. (4.448), p. 165,

$$\widehat{\Delta\boldsymbol{l}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{c}_g - \widehat{\boldsymbol{v}}^a\,, \tag{4.465}$$

and the estimated residuals from (4.449), p. 165. Writing the estimated residuals as a function of the original observations $\boldsymbol{l}$

$$\widehat{\boldsymbol{v}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}(-\boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + \boldsymbol{B}^{\mathsf{T}}\widehat{\boldsymbol{l}}^a - \boldsymbol{B}^{\mathsf{T}}\boldsymbol{l}) \tag{4.466}$$

yields their covariance matrix

$$\boldsymbol{\Sigma}_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}\,. \tag{4.467}$$

The iteration scheme is given in Algorithm 6. In contrast to all previous estimation algorithms we do not need approximate values, as the iteration may start at the observation values $\boldsymbol{l}$. Convergence is achieved if the corrections to the fitted observations are small enough (compared to the standard deviations of the observations) or some prespecified maximum iteration number is reached.

Algorithm 6: Estimation in the model with constraints between the observations only.
$[\widehat{\boldsymbol{l}}, \Sigma_{\widehat{ll}}, \widehat{\sigma}_0^2, R] = \texttt{Model_ConstraintsBetweenObservationsOnly}(\boldsymbol{l}, \Sigma_{ll}, c_g, T_l, \texttt{maxiter})$
Input: N observed values $\boldsymbol{l}$ and Σ_{ll},
constraint functions $[\boldsymbol{c}_g, \boldsymbol{B}] = c_g(\boldsymbol{l}, \widehat{\boldsymbol{l}})$, number of constraints G,
thresholds for convergence T_l, `maxiter`.
Output: parameters $\{\widehat{\boldsymbol{l}}, \Sigma_{\widehat{ll}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

Redundancy $R = N - G$;
if $R < 0$ **then stop**, not enough constraints;
Initiate: iteration $\nu = 1$, approximate values $\widehat{\boldsymbol{l}}^{(\nu)} = \widehat{\boldsymbol{l}}^a = \boldsymbol{l}$, stopping variable: $s = 0$;
repeat
  Residuals and Jacobians for constraints $\boldsymbol{g}$: $[\boldsymbol{c}_g, \boldsymbol{B}] = c_g(\boldsymbol{l}, \widehat{\boldsymbol{l}}^{(\nu)})$ (4.464), (4.408) ;
  Normal equation matrix: $\boldsymbol{N} = \boldsymbol{B}^\mathsf{T}\Sigma_{ll}\boldsymbol{B}$;
  if $\boldsymbol{N}$ is singular **then stop**: $\boldsymbol{N}$ is singular, constraints are linearly dependent;
  Correction for fitted observations: $\widehat{\Delta\boldsymbol{l}} = \Sigma_{ll}\boldsymbol{B}\boldsymbol{N}^{-1}\boldsymbol{c}_g - \widehat{\boldsymbol{l}}^{(\nu)} + \boldsymbol{l}$;
  Set iteration: $\nu := \nu + 1$;
  if $\max|\widehat{\Delta\boldsymbol{l}}_n|/\sigma_{l_n} < T_l$ for all n or $\nu =$ `maxiter` **then** $s = 2$;
  Updates of fitted observations: $\widehat{\boldsymbol{l}}^{(\nu)} = \widehat{\boldsymbol{l}}^{(\nu-1)} + \widehat{\Delta\boldsymbol{l}}$;
until $s \equiv 2$;
Covariance matrix of fitted observations: $\Sigma_{\widehat{ll}} = \Sigma_{ll}\boldsymbol{B}\boldsymbol{N}^{-1}\boldsymbol{B}^\mathsf{T}\Sigma_{ll}$;
if $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \boldsymbol{c}_g^\mathsf{T}\boldsymbol{W}_{gg}\boldsymbol{c}_g/R$ **else** $\widehat{\sigma}_0^2 = 1$.

4.8.3 Overview on Estimation Procedures

This section gives an overview of the estimation procedures using the different functional models, and by way of an example discusses their pros and cons.

4.8.3.1 Estimation with Different Functional Models

Table 4.10 summarizes the main ingredients for the five types of models, especially for the model D (the Gauss–Helmert model without constraints between the unknown parameters), since they have not yet been made explicit.

The table presents

1. the nonlinear model,
2. the linear substitute model used in the iterative estimation scheme, including the corrections $\widehat{\Delta\boldsymbol{l}}$ for the observations, the estimated corrections $\widehat{\Delta\boldsymbol{x}}$ for the parameters, and the residuals $\boldsymbol{c}_g$ and $\boldsymbol{c}_h$ for the constraints,
3. the normal equations for determining the estimated parameters or the Lagrangian parameters and the estimated residuals of the observations, and
4. the redundancy R of the estimation problem, which is needed for the estimated variance factor.

Models B, C, and E have been discussed explicitly, model A only in its linear version. Model D results from model E by setting $\boldsymbol{H} = \boldsymbol{0}$.

Following the derivation for the estimation procedure for model E in the last section, a robust or nonrobust estimation procedure can be developed for the other models. The discussion on the reduction of the normal equation system to a subset of unknown parameters using the Schur complement in the context of the Gauss–Markov model can easily be transferred to the other models. Though a bit more complex, the techniques for evaluating the results are also applicable in all mentioned model types.

Table 4.10 Functional models for estimation: Gauss–Markov (A), Gauss–Markov with constraints (B), constraints between observations only (C), Gauss–Helmert (D) and Gauss–Helmert with constraints (E)

	nonlinear	linear	normal equations, correction $\widehat{\Delta l}$, residuals $\widehat{v}$, and $\Sigma_{\widehat{v}\widehat{v}}$	R
A	$\boldsymbol{l}+\widehat{\boldsymbol{v}} = \underset{N\times 1}{\boldsymbol{f}}(\widehat{\boldsymbol{x}})$ Sect. 4.4, p. 102	$\boldsymbol{l}+\widehat{\boldsymbol{v}} = \underset{N\times U}{\boldsymbol{A}}\widehat{\Delta\boldsymbol{x}}$ $\Delta\boldsymbol{l} = \boldsymbol{l} - \boldsymbol{f}(\widehat{\boldsymbol{x}}^a)$ $\widehat{\Delta\boldsymbol{x}} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^a$	$\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\boldsymbol{A}\,\widehat{\Delta\boldsymbol{x}} = \boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\Delta\boldsymbol{l}$ $\Sigma_{\widehat{x}\widehat{x}} = (\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\boldsymbol{A})^{-1}$ $\widehat{\boldsymbol{v}} = \boldsymbol{A}\widehat{\Delta\boldsymbol{x}} - \Delta\boldsymbol{l} = -(\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1})\Delta\boldsymbol{l}$ $\Sigma_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll} - \boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}$	$N-U$
B	$\boldsymbol{l}+\widehat{\boldsymbol{v}} = \underset{N\times 1}{\boldsymbol{f}}(\widehat{\boldsymbol{x}})$ $\underset{H\times 1}{\boldsymbol{h}}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ Sect. 4.4.2, p. 104	$\boldsymbol{l}+\widehat{\boldsymbol{v}} = \underset{N\times U}{\boldsymbol{A}}\widehat{\Delta\boldsymbol{x}}$ $\underset{N\times U}{\boldsymbol{H}^\mathsf{T}}\widehat{\Delta\boldsymbol{x}} = \boldsymbol{c}_h$ $\boldsymbol{c}_h = -\boldsymbol{h}(\widehat{\boldsymbol{x}}^a)$	$\begin{bmatrix}\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\boldsymbol{A} & \boldsymbol{H}\\ \boldsymbol{H}^\mathsf{T} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\widehat{\Delta\boldsymbol{x}}\\ \boldsymbol{\mu}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}^\mathsf{T}\boldsymbol{\Sigma}_{ll}^{-1}\Delta\boldsymbol{l}\\ \boldsymbol{c}_h\end{bmatrix}$ $\Sigma_{\widehat{x}\widehat{x}}$: if $\boldsymbol{H} = \mathsf{null}(\boldsymbol{N})$ (4.174), p. 101, else (4.165), p. 100 $\widehat{\boldsymbol{v}} = \boldsymbol{A}\widehat{\Delta\boldsymbol{x}} - \Delta\boldsymbol{l}$ $\Sigma_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll} - \boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}$	$N+H-U$
C	$\underset{G\times 1}{\boldsymbol{g}}(\boldsymbol{l}+\widehat{\boldsymbol{v}}) = \boldsymbol{0}$ Sect. 4.8.1, p. 162	$\underset{G\times N}{\boldsymbol{B}^\mathsf{T}}\widehat{\Delta\boldsymbol{l}} = \boldsymbol{c}_g$ $\boldsymbol{c}_g = -\boldsymbol{g}(\widehat{\boldsymbol{l}}^a) + \boldsymbol{B}^\mathsf{T}(\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$	$\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B}\,\boldsymbol{\lambda} = -\boldsymbol{c}_g$ $\widehat{\Delta\boldsymbol{l}} = -\boldsymbol{\Sigma}_{ll}\boldsymbol{B}\boldsymbol{\lambda} - (\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$ $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}}^a + \widehat{\Delta\boldsymbol{l}} - \boldsymbol{l}$ $\Sigma_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}$	G
D	$\underset{G\times 1}{\boldsymbol{g}}(\boldsymbol{l}+\widehat{\boldsymbol{v}},\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ Sect. 4.8.1, p. 162	$\underset{G\times U}{\boldsymbol{A}}\widehat{\Delta\boldsymbol{x}} + \underset{G\times N}{\boldsymbol{B}^\mathsf{T}}\widehat{\Delta\boldsymbol{l}} = \boldsymbol{c}_g$ $\boldsymbol{c}_g = -\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + \boldsymbol{B}^\mathsf{T}(\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$	$\boldsymbol{A}^\mathsf{T}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{A}\,\widehat{\Delta\boldsymbol{x}} = \boldsymbol{A}^\mathsf{T}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{c}_g$ $\Sigma_{\widehat{x}\widehat{x}} = (\boldsymbol{A}^\mathsf{T}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{A})^{-1}$ $\widehat{\Delta\boldsymbol{l}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}(\boldsymbol{c}_g - \boldsymbol{A}\widehat{\Delta\boldsymbol{x}}) - (\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$ $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}}^a + \widehat{\Delta\boldsymbol{l}} - \boldsymbol{l}$ $\Sigma_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}\boldsymbol{W}_{gg}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B} - \boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}^\mathsf{T})\boldsymbol{W}_{gg}\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}$	$G-U$
E	$\underset{G\times 1}{\boldsymbol{g}}(\boldsymbol{l}+\widehat{\boldsymbol{v}},\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ $\underset{H\times 1}{\boldsymbol{h}}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$ Sect. 4.8.2, p. 163	$\underset{G\times U}{\boldsymbol{A}}\widehat{\Delta\boldsymbol{x}} + \underset{G\times N}{\boldsymbol{B}^\mathsf{T}}\widehat{\Delta\boldsymbol{l}} = \boldsymbol{c}_g$ $\underset{H\times U}{\boldsymbol{H}^\mathsf{T}}\widehat{\Delta\boldsymbol{x}} = \boldsymbol{c}_h$ $\boldsymbol{c}_g = -\boldsymbol{g}(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) + \boldsymbol{B}^\mathsf{T}(\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$ $\boldsymbol{c}_h = -\boldsymbol{h}(\widehat{\boldsymbol{x}}^a)$	$\begin{bmatrix}\boldsymbol{A}^\mathsf{T}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{A} & \boldsymbol{H}\\ \boldsymbol{H}^\mathsf{T} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\widehat{\Delta\boldsymbol{x}}\\ \boldsymbol{\mu}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}^\mathsf{T}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}\boldsymbol{c}_g\\ \boldsymbol{c}_h\end{bmatrix}$ $\widehat{\Delta\boldsymbol{l}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B})^{-1}(\boldsymbol{c}_g - \boldsymbol{A}\widehat{\Delta\boldsymbol{x}}) - (\widehat{\boldsymbol{l}}^a - \boldsymbol{l})$ $\Sigma_{\widehat{x}\widehat{x}}$ (4.455), p. 166 $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}}^a + \widehat{\Delta\boldsymbol{l}} - \boldsymbol{l}$ $\Sigma_{\widehat{v}\widehat{v}} = \boldsymbol{\Sigma}_{ll}\boldsymbol{B}\boldsymbol{W}_{gg}(\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}\boldsymbol{B} - \boldsymbol{A}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}}\boldsymbol{A}^\mathsf{T})\boldsymbol{W}_{gg}\boldsymbol{B}^\mathsf{T}\boldsymbol{\Sigma}_{ll}$	$G+H-U$

4.8.3.2 On the Choice of the Functional Model

The following example illustrates that the functional model for estimation can freely be chosen, a long as it is algebraically equivalent to some geometrical or physical model. But the resulting models lead to representations of different complexity concerning both the algebraic structure as well as the computational effort.

If the same stochastical model is used, the results of the corresponding parameter estimation are fully equivalent, in both mean and distribution. This especially holds for the estimated observations, which can be given in all cases. Thus we are free to choose the model which best fits the application. The freedom of selection should be exploited in order to arrive at simple and efficient solutions for the estimation problem.

As a representative example, we assume three 2D points $\boldsymbol{p}_i, i = 1, 2, 3$, to be observed. They are supposed to be on a straight line $\boldsymbol{\ell}$, see Fig. 4.16. Thus our geometrical model is

$$p_i \in \ell\,, \qquad i = 1,2,3\,; \tag{4.468}$$

in other words the three points are collinear.

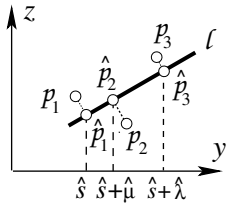


Fig. 4.16 On the equivalence of models. Three points p_i which are assumed to lie on line ℓ or equivalently to be collinear. The geometric model holds for the fitted points $\widehat{p}_i$. We assume both coordinates are uncertain

In order to arrive at comparable functional models, we represent the points p_i by their Cartesian coordinates (y_i, z_i). [10] We only discuss the case where both coordinates are perturbed.

The observations are collected in the vector $\boldsymbol{l} := [y_1, y_2, y_3, z_1, z_2, z_3]^\mathsf{T}$. We then obtain the five algebraically equivalent models.

A In order to arrive at the structure of the *Gauss–Markov model*, we represent the three foot points (s, t_1), $(s+\lambda, t_2)$, and $(s+\mu, t_3)$ explicitly via the parameters m and k of the straight line and the parameters s, λ, and μ; thus, we have the unknown parameters $\boldsymbol{x} := [m, k, s, \lambda, \mu]^\mathsf{T}$. We obtain the six nonlinear observation equations

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) : \qquad \begin{bmatrix} y_1 + \widehat{v}_{y_1} \\ y_2 + \widehat{v}_{y_2} \\ y_3 + \widehat{v}_{y_3} \\ z_1 + \widehat{v}_{z_1} \\ z_2 + \widehat{v}_{z_2} \\ z_3 + \widehat{v}_{z_3} \end{bmatrix} = \begin{bmatrix} \widehat{s} \\ \widehat{s} + \widehat{\lambda} \\ \widehat{s} + \widehat{\mu} \\ \widehat{m}\widehat{s} + \widehat{k} \\ \widehat{m}(\widehat{s} + \widehat{\lambda}) + \widehat{k} \\ \widehat{m}(\widehat{s} + \widehat{\mu}) + \widehat{k} \end{bmatrix}. \tag{4.469}$$

Though we are able to give a model with the structure of a Gauss–Markov model, it appears too complex.

B The *Gauss–Markov model with constraints* can be represented using the coordinates $\boldsymbol{x} := [s_1, s_2, s_3, t_1, t_2, t_3]^\mathsf{T}$ as unknown parameters. We have the observation equations

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{x}} : \qquad y_i + \widehat{v}_{y_i} = \widehat{s}_i \qquad z_i + \widehat{v}_{z_i} = \widehat{t}_i\,, \quad i = 1,2,3 \tag{4.470}$$

and the nonlinear collinearity constraint

$$h(\widehat{\boldsymbol{x}}) = 0 : \quad \begin{vmatrix} \widehat{s}_1 & \widehat{s}_2 & \widehat{s}_3 \\ \widehat{t}_1 & \widehat{t}_2 & \widehat{t}_3 \\ 1 & 1 & 1 \end{vmatrix} = 0\,. \tag{4.471}$$

The model appears quite symmetric in all variables.

C The *model with constraints between the observations only*:

$$g(\boldsymbol{l} + \widehat{\boldsymbol{v}}) = 0 : \quad \begin{vmatrix} y_1 + \widehat{v}_{y_1} & y_2 + \widehat{v}_{y_2} & y_3 + \widehat{v}_{y_3} \\ z_1 + \widehat{v}_{z_1} & z_2 + \widehat{v}_{z_2} & z_3 + \widehat{v}_{z_3} \\ 1 & 1 & 1 \end{vmatrix} = 0\,. \tag{4.472}$$

[10] We chose the ordinate axis as the y-axis in order to have the variable name x free for the unknown parameters in the functional models.

This model has $N = 6$ observations and $G = 1$ constraint, thus is the most effective model for this specific problem if there is no need to determine the parameters and their covariance matrix.

D The *Gauss–Helmert model* can use the Hesse representation with the unknown parameters $\boldsymbol{x} = [\phi, d]^\mathsf{T}$ to advantage:

$$g_i(\boldsymbol{l} + \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{x}}) = 0 : \quad (y_i + \widehat{v}_{y_i}) \cos\widehat{\phi} + (z_i + \widehat{v}_{z_i}) \sin\widehat{\phi} - \widehat{d} = 0 \qquad i = 1, 2, 3\,. \tag{4.473}$$

This model appears to be the most transparent one, both concerning the algebraic structure as well as the computational complexity. It is the most relevant for estimating geometric entities. We will also find it sufficient when estimating homogeneous entities.

E The *Gauss–Helmert model with constraints* can be used if we represent a line with the implicit equation $ay_1 + by_1 + c = 0$. Then the three unknown parameters $\boldsymbol{x} = [a, b, c]^\mathsf{T}$ need to be constrained, as only their ratio $a : b : c$ is relevant. Thus we arrive at the model

$$g_i(\boldsymbol{l} + \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{x}}) = 0 : \quad (y_i + \widehat{v}_{y_i})\widehat{a} + (z_i + \widehat{v}_{z_i})\widehat{b} + \widehat{c} = 0 \qquad i = 1, 2, 3 \tag{4.474}$$

with the following nonlinear constraint between the unknown parameters:

$$h(\widehat{\boldsymbol{x}}) := \widehat{a}^2 + \widehat{b}^2 + \widehat{c}^2 - 1 = 0\,. \tag{4.475}$$

All these models are nonlinear. Also, they lead to the same estimates for the fitted observations and the estimated parameters if the stochastical model is the same and provided the linearization does not induce significant bias.

The discussion can be transferred to any functional model which can be represented by equality constraints between observations and unknown parameters. Therefore, the designer of an estimation procedure can choose the particular functional model which appears most adequate for the application.

4.8.4 Filtering, Prediction and Collocation

We now give an important example of the estimation with the Gauss–Helmert model: an extension of the Wiener filter, discussed in Sect. 4.2.5, p. 93. We generalize it to the case where the mean value of the signal is not zero and include the prediction of new signal values, a procedure called collocation.

The General Case for Collocation with Prediction. Wiener filtering aims at finding best estimates $\widehat{\boldsymbol{s}}$ and $\widehat{\boldsymbol{n}}$ for the K-dimensional signal $\boldsymbol{s}$ and the noise $\boldsymbol{n}$ both having zero mean for a given observed K-vector $\boldsymbol{y} = [y_k], k = 1, ..., K$, the sum of both, and knowing the regular $K \times K$ covariance matrices Σ_{ss} and Σ_{nn} of the observational noise and of the signal. We generalize this model in two ways: (1) We assume the mean not to be zero but to be some linear function $\mathsf{A}\boldsymbol{x}$ of some unknown parameters $\boldsymbol{x}$ often called the *trend* of the signal. (2) We want to *predict* some L signal values $\boldsymbol{z} = [z_l], l = 1, ..., L$, not identical to those observed but related to the signal $\boldsymbol{s}$ by their $L \times K$ covariance matrix Σ_{zs}. This generalized model is also called *collocation model*.

trend $\mathsf{A}\boldsymbol{x}$

collocation model

The functional model we want to realize is

$$\boldsymbol{y} = \mathsf{A}\boldsymbol{x} + \boldsymbol{s} + \boldsymbol{n}\,. \tag{4.476}$$

The *stochastical model* collects the prior information for the noise $\underline{\boldsymbol{n}}$, the signal $\underline{\boldsymbol{s}}$ and the new signal values $\underline{\boldsymbol{z}}$ in the vector

$$\underline{\boldsymbol{l}} \sim \mathcal{M}(\tilde{\boldsymbol{l}}, \Sigma_{ll}) \quad \text{or} \quad \begin{bmatrix} \underline{\boldsymbol{n}} \\ \underline{\boldsymbol{s}} \\ \underline{\boldsymbol{z}} \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} \tilde{\boldsymbol{n}} \\ \tilde{\boldsymbol{s}} \\ \tilde{\boldsymbol{z}} \end{bmatrix}, \begin{bmatrix} \Sigma_{nn} & 0 & 0 \\ 0 & \Sigma_{ss} & \Sigma_{sz} \\ 0 & \Sigma_{zs} & \Sigma_{zz} \end{bmatrix} \right) . \tag{4.477}$$

The mean values $\tilde{\boldsymbol{n}}$ and $\tilde{\boldsymbol{s}}$ are fixed values with value zero. We will take $\boldsymbol{n}_{\text{pr}} = \mathbf{0}$ and $\boldsymbol{s}_{\text{pr}} = \mathbf{0}$ as observational values for the prior. We assume the 3×3 block matrix to be regular. With the matrix

$$B = [I_K, I_K, 0]^{\mathsf{T}} \tag{4.478}$$

we have the constraint, i.e., the *functional model*

$$\boldsymbol{g}(\widehat{\boldsymbol{l}}, \widehat{\boldsymbol{x}}) = A\widehat{\boldsymbol{x}} + B^{\mathsf{T}}\widehat{\boldsymbol{l}} - \boldsymbol{y} = A\widehat{\boldsymbol{x}} + \widehat{\boldsymbol{s}} + \widehat{\boldsymbol{n}} - \boldsymbol{y} = \mathbf{0} , \tag{4.479}$$

consistent with the functional model (4.476). Observe, we treat the given vector $\boldsymbol{y}$ as fixed and want to have best estimates for the signal vector $\boldsymbol{s}$ and the noise vector $\boldsymbol{n}$. There is no constraint on the unknown signal vector $\boldsymbol{z}$; however, it is linked to the unknown signal $\boldsymbol{s}$ by their covariance matrix. Equation (4.479) has the form of a Gauss–Helmert model, i.e., model D.

For using the equations for model D in Table 4.10, p. 172, we need to specify starting values for the estimate $\widehat{\boldsymbol{l}}$, though our problem is linear. We choose the initial values $\widehat{\boldsymbol{l}}^a = \boldsymbol{l}$ for the estimated observations and $\widehat{\boldsymbol{x}}^a = \mathbf{0}$ for the estimated parameters. We first obtain best estimates for the parameters $\boldsymbol{x}$ from the normal equations (4.447), p. 165 with $\boldsymbol{c}_g = \boldsymbol{y}$, cf. (4.444), p. 165,

$$A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}A\,\widehat{\boldsymbol{x}} = A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}\boldsymbol{y} \tag{4.480}$$

or

$$A^{\mathsf{T}}(\Sigma_{nn} + \Sigma_{ss})^{-1}A\,\widehat{\boldsymbol{x}} = A^{\mathsf{T}}(\Sigma_{nn} + \Sigma_{ss})^{-1}\boldsymbol{y} . \tag{4.481}$$

Then the best estimate for the fitted observations therefore is

$$\widehat{\boldsymbol{l}} = \boldsymbol{l} + \Sigma_{ll}B(\Sigma_{nn} + \Sigma_{ss})^{-1}(\boldsymbol{y} - A\widehat{\boldsymbol{x}}) . \tag{4.482}$$

We therefore obtain best estimates for the noise, the observed signal and the new signal,

$$\begin{bmatrix} \widehat{\boldsymbol{n}} \\ \widehat{\boldsymbol{s}} \\ \widehat{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} \Sigma_{nn} \\ \Sigma_{ss} \\ \Sigma_{zs} \end{bmatrix} (\Sigma_{nn} + \Sigma_{ss})^{-1}(\boldsymbol{y} - A\widehat{\boldsymbol{x}}) . \tag{4.483}$$

Generally, the predicted L-vector $\widehat{\boldsymbol{z}}$ will not be zero due to its correlation with the estimated signal $\widehat{\boldsymbol{s}}$.

We discuss two specializations.

Wiener Filter and Prediction. The *Wiener filter* can be derived by assuming the trend $A\boldsymbol{x}$ to be zero. Then we obtain the estimated fitted observations from (4.444), p. 165, (4.448), and (4.427), p. 163,

$$\widehat{\boldsymbol{l}} = \widehat{\boldsymbol{l}}^a + \widehat{\Delta\boldsymbol{l}} = \boldsymbol{l} + \Sigma_{ll}B(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}(\boldsymbol{y} + B^{\mathsf{T}}(\widehat{\boldsymbol{l}}^a - \boldsymbol{l})) , \tag{4.484}$$

or, explicitly again,

$$\begin{bmatrix} \widehat{\boldsymbol{n}} \\ \widehat{\boldsymbol{s}} \\ \widehat{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} \Sigma_{nn} \\ \Sigma_{ss} \\ \Sigma_{zs} \end{bmatrix} (\Sigma_{nn} + \Sigma_{ss})^{-1}\boldsymbol{y} . \tag{4.485}$$

The expression for $\widehat{\boldsymbol{s}}$ is consistent with (4.107), p. 93, derived as Bayes estimate for $\boldsymbol{s}$.

The covariance matrix of the estimated observations can be derived from (4.482) observing that $\widehat{\boldsymbol{l}}$ is linearly dependent on $\boldsymbol{l}$ and all other entities are fixed. By variance propagation we obtain

$$\Sigma_{\widehat{l}\widehat{l}} = \Sigma_{ll} - \Sigma_{ll}B(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}B^{\mathsf{T}}\Sigma_{ll} . \tag{4.486}$$

We obtain

$$\mathbb{D}\left(\begin{bmatrix}\widehat{\boldsymbol{n}}\\ \widehat{\boldsymbol{s}}\\ \widehat{\boldsymbol{z}}\end{bmatrix}\right) = \begin{bmatrix}\Sigma_{nn} & 0 & 0\\ 0 & \Sigma_{ss} & \Sigma_{sz}\\ 0 & \Sigma_{zs} & \Sigma_{zz}\end{bmatrix} - \begin{bmatrix}\Sigma_{nn}\\ \Sigma_{ss}\\ \Sigma_{zs}\end{bmatrix}(\Sigma_{nn}+\Sigma_{ss})^{-1}[\Sigma_{nn} \;\; \Sigma_{ss} \;\; \Sigma_{zs}^{\mathsf{T}}]\,. \quad (4.487)$$

Using

$$W_{yy} = (\Sigma_{nn}+\Sigma_{ss})^{-1}\,, \quad (4.488)$$

this is

$$\mathbb{D}\left(\begin{bmatrix}\widehat{\boldsymbol{n}}\\ \widehat{\boldsymbol{s}}\\ \widehat{\boldsymbol{z}}\end{bmatrix}\right) = \begin{bmatrix}\Sigma_{nn}-\Sigma_{nn}W_{yy}\Sigma_{nn} & -\Sigma_{nn}W_{yy}\Sigma_{ss} & -\Sigma_{nn}W_{yy}\Sigma_{zs}^{\mathsf{T}}\\ -\Sigma_{ss}W_{yy}\Sigma_{nn} & \Sigma_{ss}-\Sigma_{ss}W_{yy}\Sigma_{ss} & -\Sigma_{ss}W_{yy}\Sigma_{zs}^{\mathsf{T}}\\ -\Sigma_{zs}W_{yy}\Sigma_{nn} & -\Sigma_{zs}W_{yy}\Sigma_{ss} & \Sigma_{zz}-\Sigma_{zs}W_{yy}\Sigma_{zs}^{\mathsf{T}}\end{bmatrix}\,. \quad (4.489)$$

Now all submatrices in the upper left 2×2-block matrix are equal, except for their sign, therefore we obtain

$$\mathbb{D}\left(\begin{bmatrix}\widehat{\boldsymbol{n}}\\ \widehat{\boldsymbol{s}}\end{bmatrix}\right) = \begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \otimes \Sigma_{ss}(\Sigma_{nn}+\Sigma_{ss})^{-1}\Sigma_{nn} \quad (4.490)$$

This result confirms the covariance matrix of the estimated signal in (4.109), p. 94. Moreover, estimated noise and signal are negatively correlated to 100%. This is reasonable, since if the estimated signal is larger, the estimated noise needs to be smaller, since they sum to the fixed observed values $\boldsymbol{y}$. The uncertainty of both estimates directly depends on uncertainty of their prior values.

The covariance matrix of the predicted signal is

$$\Sigma_{\widehat{z}\widehat{z}} = \Sigma_{zz} - \Sigma_{zs}(\Sigma_{nn}+\Sigma_{ss})^{-1}\Sigma_{zs}^{\mathsf{T}}\,. \quad (4.491)$$

Pure prediction in addition assumes the observed value $\boldsymbol{y}$ is identical to the signal $\boldsymbol{s}$. This can be realized by specializing the covariance for the noise to be $\Sigma_{nn} = 0$. This leads to

$$\widehat{\boldsymbol{z}} = \Sigma_{zs}\Sigma_{ss}^{-1}\boldsymbol{s}\,, \quad \Sigma_{\widehat{z}\widehat{z}} = \Sigma_{zz} - \Sigma_{zs}\Sigma_{ss}^{-1}\Sigma_{zs}^{\mathsf{T}}\,. \quad (4.492)$$

4.9 Methods for Closed Form Estimations

4.9.1 Linearizing by Parameter Substitution 177
4.9.2 Problems Linear in the Parameters 177

Solutions for parameter estimation are called *direct* or *closed form* if no approximate values for the parameters are required and no iteration scheme is necessary. They are useful in real-time applications, where constant speed is of advantage. There are no closed form solutions which are both statistically optimal and useful for providing approximate values for the statistically rigorous solution discussed before, except for special cases, cf. Sect. 10.5, p. 395.

This last section on estimation addresses functional models which allow a direct or closed form solution for the estimated parameters without requiring any approximate values. Though not optimal, the uncertainty of the estimates can be given explicitly. We give the general estimation schemes and demonstrate them using the estimation of circles, conics, and quadrics. We will use these techniques throughout the following chapters, and especially address direct solutions for geometric entities and transformations in Sect. 10.5, p. 395 and in Part III on *Orientation and Reconstruction*. On the page `http://cmp.felk.cvut.cz/minimal` the reader finds a large number of direct solutions for geometric problems with a minimum number of observations.

Closed form solutions can be achieved when the parameters are linear in the functional model, which may occur in two cases:

1. The functional model has a linear form in the parameters in the first place.
2. There is a one-to-one parameter transformation such that in the transformed model, the new parameters appear linearly.

We start with the second situation due to its simplicity.

4.9.1 Linearizing by Parameter Substitution

Algebraically transforming the parameters often allows us to linearize the problem. Thus we may develop an invertible algebraic transformation,

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \qquad \boldsymbol{x} = \boldsymbol{f}^{-1}(\boldsymbol{y})\,, \tag{4.493}$$

such that the new parameters $\boldsymbol{y}$ can be determined linearly with one of the methods mentioned above.

Example 4.9.13: ***Circle fitting.*** Fitting a circle

$$(p_i - p_0)^2 + (q_i - q_0)^2 - r^2 = 0 \qquad i = 1, ..., I \tag{4.494}$$

to observed points (p_i, q_i) can be done by substituting

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} p_0 \\ q_0 \\ p_0^2 + q_0^2 - r^2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_0 \\ q_0 \\ r \end{bmatrix} = \begin{bmatrix} u \\ v \\ \sqrt{u^2 + v^2 - w} \end{bmatrix}. \tag{4.495}$$

Then we can solve the linear equation system

$$p_i^2 + q_i^2 = 2p_i u + 2q_i v - w \qquad i = 1, ..., I \tag{4.496}$$

for the substitute parameters $[u, v, w]$, and derive the centre $[p_0, q_0]$ and radius r from (4.495) (Bookstein, 1979). The solution given here shows severe bias if the given points only cover a small part of the circumference. We give an unbiased solution in Sect. 4.9.2.5, p. 181. ◇

4.9.2 Problems Linear in the Parameters

We start with a Gauss–Helmert model (cf. Sect. 4.8.1, p. 162) with a special structure. Let the G nonlinear constraints $\boldsymbol{g}(\tilde{\boldsymbol{l}}, \tilde{\mathbf{x}})$ between the true values of the N observations $\tilde{\boldsymbol{l}}$, and the U true parameters $\tilde{\mathbf{x}}$ be *linear in the unknown parameters*, and thus of the form

constraints linear in the parameters

$$\underset{G\times U}{\boldsymbol{g}(\tilde{\boldsymbol{l}}, \tilde{\mathbf{x}})} = \underset{G\times U}{\boldsymbol{A}(\tilde{\boldsymbol{l}})}\ \underset{U\times 1}{\tilde{\mathbf{x}}} = \underset{G\times 1}{\mathbf{0}}\,. \tag{4.497}$$

As the constraint is homogeneous, and thus has no additive constant vector, the vector $\tilde{\mathbf{x}}$ can be scaled arbitrarily; thus, it is homogeneous as defined in Sect. 5.1.1, p. 195. This is indicated by the upright notation.

As for the algorithm of the Gauss–Helmert model above, we now assume we have mutually stochastically independent groups $\boldsymbol{l}_i$ of observations, each of which takes part in a small set of constraints $\boldsymbol{g}_i(\boldsymbol{l}_i) = \mathbf{0}$. Thus the G constraints consist of I groups $\boldsymbol{g}_i$ of constraints, each depending on the observational group $\boldsymbol{l}_i$. Therefore, we can also write the model as

$$\boldsymbol{g}_i(\tilde{\boldsymbol{l}}, \tilde{\mathbf{x}}) = \boldsymbol{a}_i^{\mathsf{T}}(\tilde{\boldsymbol{l}})\,\tilde{\mathbf{x}} = \mathbf{0}\,, \quad i = 1, \ldots, I\,. \tag{4.498}$$

The number G of constraints needs to be at least $U - 1$. If the number of constraints is minimal, thus $G = U - 1$, and the constraints are linearly independent, and thus the

direct minimal solution for a nonlinear problem

matrix $\mathsf{A}(\boldsymbol{l})$ has rank $U-1$, the homogeneous equation system can be solved directly. This is of high importance, since we have a *direct minimal solution* of a *nonlinear model*.

Though this model structure is very special, as the unknown parameters appear only linear in the constraints, it will occur frequently when estimating geometric entities in Parts II and III. Observe, a problem nonlinear in the parameters may be linearized by parameter substitution, see the previous section.

In the noise-free case, the solution for given true observational values will be the right eigenvector of the matrix $\mathsf{A}(\tilde{\boldsymbol{l}})$.

In contrast to the Gauss–Helmert model we assume that *the observed* $\boldsymbol{l}$ *values are not to be corrected and only the parameters* $\boldsymbol{x}$ *are to be estimated.* This leads to solutions which are called algebraically optimal.

A solution of an optimization problem is called *algebraically optimal* if it does not refer to or if it approximates the statistical properties of the observations, but instead optimizes some heuristically chosen algebraic expression.

Since due to observational deviations the used constraints will not be satisfied, the optimization will be based on the residual vector of these constraints. In our problems, we use a possibly weighted sum of squares of these residuals together with a single quadratic constraint. As a favourable consequence, the optimization leads to a simple or generalized eigenvalue or to a singular value problem.

4.9.2.1 Estimation Scheme

Due to deviations of the given observational values from the true ones, we obtain a residual vector,

$$\boldsymbol{g}(\boldsymbol{l}, \mathbf{x}) = \mathsf{A}(\boldsymbol{l})\,\mathbf{x} \tag{4.499}$$

which is nonzero for practically any choice of $\mathbf{x}$. The model does not require the constraints to depend linearly on the observations. This is useful when estimating implicitly given curves or surfaces.

Example 4.9.14: ***Implicitly given curves.*** Planar curves in the (p,q)-plane can be implicitly specified by $g(p,q)=0$. When the function $g(p,q)$ is a weighted sum of basis functions $f_u(p,q)$, namely $g(p,q) = \sum_{u=1}^{U} x_u f_u(p,q)$, then from a set of I given points $\boldsymbol{p}_i$ with coordinates $\boldsymbol{p}_i = [p_i, q_i]^\mathsf{T}$, we can write all constraints as

$$\mathsf{A}(\widehat{\boldsymbol{p}})\widehat{\mathbf{x}} = \mathbf{0} \tag{4.500}$$

with

$$\mathsf{A}(\widehat{\boldsymbol{p}}) = \begin{bmatrix} f_1(\widehat{\boldsymbol{p}}_1) & \dots & f_u(\widehat{\boldsymbol{p}}_1) & \dots & f_U(\widehat{\boldsymbol{p}}_1) \\ \dots & \dots & \dots & \dots & \dots \\ f_1(\widehat{\boldsymbol{p}}_i) & \dots & f_u(\widehat{\boldsymbol{p}}_i) & \dots & f_U(\widehat{\boldsymbol{p}}_i) \\ \dots & \dots & \dots & \dots & \dots \\ f_1(\widehat{\boldsymbol{p}}_I) & \dots & f_u(\widehat{\boldsymbol{p}}_I) & \dots & f_U(\widehat{\boldsymbol{p}}_I) \end{bmatrix} \qquad \widehat{\mathbf{x}} = \begin{bmatrix} \widehat{x}_1 \\ \dots \\ \widehat{x}_u \\ \dots \\ \widehat{x}_U \end{bmatrix} . \tag{4.501}$$

This especially holds for planes, straight lines, circles, and conics, where the basis functions are polynomials in p and q. For example, a general circle has the form $g(p,q) = x_1(p^2+q^2) + x_2 p + x_3 q + x_4 = 0$, see below. The model can be generalized to surfaces in 3D implicitly given by $g(p,q,r)=0$, or to curves in 3D given by the intersection of two surfaces. ◇

It is a plausible strategy to minimize some, possibly weighted, norm of $\boldsymbol{g}$. The estimation within the Gauss–Helmert model would suggest to minimize, cf. (4.452), p. 166

$$\Omega = \boldsymbol{g}^\mathsf{T} \Sigma_{gg}^{-1} \boldsymbol{g} \tag{4.502}$$

with respect to the parameters $\mathbf{x}$, where the covariance matrix of the residual constraints is

$$\Sigma_{gg} = \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{l}}\right) \Sigma_{ll} \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{l}}\right)^\mathsf{T} , \tag{4.503}$$

which cannot generally be optimized in closed form. We discuss two approximations which lead to a direct solution.

4.9.2.2 Minimum Norm Solution

The simplest solution is to choose the vector $\mathbf{x}$ for which the norm $|\boldsymbol{g}(\mathbf{x})|^2 = \boldsymbol{g}(\mathbf{x})^\mathsf{T}\,\boldsymbol{g}(\mathbf{x})$ of the residual vector is minimal while excluding the trivial solution $\mathbf{x} = \mathbf{0}$.

This can be achieved by enforcing the constraint $|\mathbf{x}| = 1$ and leads to the optimal solution

$$\widehat{\mathbf{x}} = \text{argmin}_{x,|x|=1}\, |A(\boldsymbol{l})\,\mathbf{x}|^2\,. \tag{4.504}$$

This optimization criterion can be derived from (4.502) by setting $W_{gg} \propto I_G$, which is a crude approximation. It often is referred to as *algebraic minimization*, as the minimization principle just follows algebraic simplicity, namely by minimizing the Euclidean distance of the residual constraint from the origin.

Generally, the solution can be found by a singular value decomposition (SVD, cf. Appendix A.10, p. 777) of the matrix $A(\boldsymbol{l})$ which partitions A into the product of two orthogonal matrices U and V and a rectangular diagonal matrix S:

algebraic solution with SVD

$$\underset{G\times U}{A} = \underset{G\times G}{U}\;\underset{G\times U}{S}\;\underset{U\times U}{V}^\mathsf{T}, \tag{4.505}$$

with the singular values s_g on the diagonal of the rectangular $G \times U$ matrix S, the G left singular vectors $\boldsymbol{u}_g$ in $U = [\boldsymbol{u}_g], g = 1, ..., G$, and the U right singular vectors $\boldsymbol{v}_u$ in $V = [\boldsymbol{v}_u], u = 1, ..., U$. The solution is given by the column of V belonging to the smallest singular value s_j,

$$\widehat{\mathbf{x}} = \boldsymbol{v}_j \quad \text{with} \quad j = \text{argmin}_g(s_g)\,. \tag{4.506}$$

Instead of determining $\mathbf{x}$ from the SVD of A, it also could be determined from the eigenvector of $A^\mathsf{T}A$ belonging to the smallest eigenvalue. This may be computationally more efficient if the number of constraints is not very high.

algebraic solution with eigenvalue decomposition

If the parameters $\mathbf{x}$ are to be determined from a *minimal number of constraints*, the number G of constraints is less than the number U of unknown parameters. Then the parameter vector $\mathbf{x}$ is element of the $(U-G)$-dimensional null space of A. Instead of using the SVD, a basis $\mathsf{null}(A)$ of the null space can be determined by solving the homogeneous equation system $A\mathbf{x} = \mathbf{0}$ by using a QR decomposition (Coleman and Sorensen, 1984). The null space of the $G \times U$ matrix A, when having a full rank G, can be determined from

minimal solution with QR decomposition

$$\mathsf{null}(A) = \underset{G\times U-G}{Q_2} \quad \text{with} \quad [Q, R] = \mathtt{qr}(A^\mathsf{T}) \quad \text{and} \quad \underset{U\times U}{Q} = [\,\underset{U\times G}{Q_1}\,,\ \underset{U\times(U-G)}{Q_2}\,] \tag{4.507}$$

with the QR-decomposition $\mathtt{qr}(\cdot)$, cf. Sect. A.9, p. 776.

The computing times for determining the null space, when using the QR decomposition instead of the SVD, are smaller in the order of a factor of 20. This is of advantage when using a direct algebraic solution within a RANSAC procedure for outlier detection, cf. Sect. 4.7.7, p. 153.

The approximation of the estimate for redundant constraints is crude. Moreover, as the functions $\boldsymbol{a}_i(\boldsymbol{l})$ may be scaled arbitrarily, the solution cannot be unique. Even in the case of proper conditioning of the functions $\boldsymbol{a}_i$, e.g., by requiring that they have norm 1, the estimated parameters show significant bias.

The following solution, proposed by Taubin (1991), reduces the bias.

4.9.2.3 Taubin's Algebraic Minimization

In many applications, each constraint is a scalar g_i and only depends on *one* observational group $\boldsymbol{l}_i$, as also assumed in the example on implicitly given curves. Thus here we have $G = I$ and we will use the index i for the constraint for each observational group. Moreover, often all the given observational groups are of the same type and can be assumed to have the same accuracy $\sigma^2 I$. Based on this assumption, Taubin (1993) proposed starting the geometrically and also statistically motivated optimization function (4.502), expressed as

$$\Omega(\boldsymbol{x}) = \sum_{i=1}^{I} \frac{g_i^2(\boldsymbol{l}, \boldsymbol{x})}{\sigma_{g_i}^2} \,. \tag{4.508}$$

Taubin suggested replacing the individual variances $\sigma_{g_i}^2$ of the residual constraints with the mean variance,

$$\overline{\sigma_{g_i}^2} = \frac{1}{I} \sum_{i=1}^{I} \sigma_{g_i}^2 \,. \tag{4.509}$$

For determining the variances $\sigma_{g_i}^2$, we need the Jacobians (cf. (4.497), p. 177),

$$\frac{\partial g_i}{\partial \boldsymbol{l}} = \frac{\partial \mathbf{x}^\mathsf{T} \boldsymbol{a}_i(\boldsymbol{l})}{\partial \boldsymbol{l}} = \mathbf{x}^\mathsf{T} C_i \,, \quad \text{with} \quad C_i = \frac{\partial \boldsymbol{a}_i}{\partial \boldsymbol{l}} \,, \tag{4.510}$$

and obtain the individual variances

$$\sigma_{g_i}^2 = \sigma^2 \, \mathbf{x}^\mathsf{T} C_i C_i^\mathsf{T} \mathbf{x} \,. \tag{4.511}$$

Omitting the constant factor σ^2, we therefore minimize

$$\Omega_T(\boldsymbol{x}) = \frac{\sum_i g_i^2}{\sum_{i=1}^{I} \sigma_{g_i}^2} \,, \tag{4.512}$$

which can be written as

$$\Omega_T(\boldsymbol{x}) = \frac{\mathbf{x}^\mathsf{T} M \mathbf{x}}{\mathbf{x}^\mathsf{T} N \mathbf{x}} \tag{4.513}$$

with the two matrices

$$M = \frac{1}{I} \sum_{i=1}^{I} \boldsymbol{a}_i(\boldsymbol{l}) \boldsymbol{a}_i^\mathsf{T}(\boldsymbol{l}) \,, \qquad N = \frac{1}{I} \sum_{i=1}^{I} C_i(\boldsymbol{l}) C_i^\mathsf{T}(\boldsymbol{l}) \,. \tag{4.514}$$

Thus the optimal estimate is the eigenvector of the generalized eigenvalue problem,

$$(M - \lambda N)\boldsymbol{x} = \mathbf{0} \,, \tag{4.515}$$

belonging to the smallest generalized eigenvalue.

4.9.2.4 Covariance Matrix of the Estimated Parameters

The covariance matrix of the estimated parameters can be determined using variance propagation of implicit functions (cf. Sect. 2.7.5, p. 43) if no special expressions exist, as in the case of a fitting circle.

We start with the total differential

$$\Delta \boldsymbol{g}(\boldsymbol{l}, \mathbf{x}) = A(\boldsymbol{l}) \, \Delta \mathbf{x} + B^\mathsf{T}(\boldsymbol{l}, \mathbf{x}) \, \Delta \boldsymbol{l} = \mathbf{0} \tag{4.516}$$

of the constraint equation $\boldsymbol{g}(\tilde{\boldsymbol{l}}, \tilde{\mathbf{x}}) = \mathbf{0}$ with the Jacobian

$$B(\mathbf{x}, \boldsymbol{l}) = \frac{\partial \boldsymbol{g}(\mathbf{x}, \boldsymbol{l})}{\partial \boldsymbol{l}^\mathsf{T}} \,, \tag{4.517}$$

which generally depends not only on the parameters, but also on the observed values. From (4.516), we derive an explicit expression for random perturbations of $\mathbf{x}$,

$$\underline{\Delta \mathbf{x}} = -A^+ B^\mathsf{T} \underline{\Delta \boldsymbol{l}} \,, \tag{4.518}$$

induced by random perturbations $\underline{\Delta \boldsymbol{l}}$ in the observed values, with the rank-constrained pseudo-inverse of A,

$$A^+ = V D^+ U^\mathsf{T} , \tag{4.519}$$

where only the largest $U-1$ singular values of A are kept,

$$D^+ = \begin{cases} 1/D_{uu} \text{ if } u \le U-1 \\ 0 \text{ else} \end{cases} . \tag{4.520}$$

This is necessary to enforce the correct rank of the covariance matrix, which is 1 less the dimension U of the homogeneous vector due to the normalization constraint.

Remark: The result (4.518) is also obtained by linearizing the constraints at the true parameters and observations, which yields $A(\underline{\boldsymbol{l}})\mathbf{x} = A(\tilde{\boldsymbol{l}})\tilde{\mathbf{x}} + A(\tilde{\boldsymbol{l}})\Delta\mathbf{x} + B^\mathsf{T}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{l}})\underline{\Delta \boldsymbol{l}} = \underline{\boldsymbol{w}}$. With $\underline{\boldsymbol{y}} = -B^\mathsf{T}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{l}})\underline{\Delta \boldsymbol{l}}$, this is equivalent to $\underline{\boldsymbol{y}} + \underline{\boldsymbol{w}} = A(\tilde{\boldsymbol{l}})\Delta\mathbf{x}$, due to $A(\tilde{\boldsymbol{l}})\tilde{\mathbf{x}} = \mathbf{0}$. Now, solving for the corrections $\Delta\mathbf{x}$ using the Gauss–Markov model $\underline{\boldsymbol{y}} \sim \mathcal{N}(A(\tilde{\boldsymbol{l}})\Delta\mathbf{x}, I_G)$ with constraint $\hat{\mathbf{x}}^\mathsf{T}\hat{\mathbf{x}} = 1$ or linearized $\tilde{\mathbf{x}}^\mathsf{T}\Delta\mathbf{x} = 0$ leads to (4.518) (cf. Sect. (4.3.2), p. 101 and App. (A.12.2), p. 779, and also Koch (1999, theorem on p. 187)) Since $\underline{\boldsymbol{y}} = -B^\mathsf{T}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{l}})\underline{\Delta \boldsymbol{l}}$ actually has covariance matrix $B^\mathsf{T}\Sigma_{ll}B$, not I_G, the algebraic solution is statistically suboptimal if we have redundant observations. ◇ Exercise 4.12

Therefore, the covariance matrix of the parameters $\widehat{\underline{\mathbf{x}}} = \mathbf{x} + \widehat{\underline{\Delta\mathbf{x}}}$ is given by

$$\Sigma_{\widehat{\mathbf{x}}\widehat{\mathbf{x}}} = A^+ B^\mathsf{T} \Sigma_{ll} B A^{+\mathsf{T}} . \tag{4.521}$$

In principle, both Jacobians A and B should be evaluated at the fitted values. In the above-mentioned scheme, we do not determine the fitted observations. Hence, we again assume that the relative accuracy of the observations is high enough, so that the Jacobians, evaluated at the given values $\boldsymbol{l}$ and the estimated parameters $\hat{\mathbf{x}}$, are sufficiently good approximations.

Since the algebraic optimization scheme only depends on the matrix A, we only show that the problem can be brought into the form (4.499) and do not explicitly discuss the estimation steps or the determination of the covariance matrix when we refer to an algebraically optimal solution in the following.

4.9.2.5 Examples: Circle, Ellipse, and Quadric Fitting

Circle fitting. A 2D point $\boldsymbol{p}_i$ with coordinates $\boldsymbol{p}_i = [p_i, q_i]^\mathsf{T}$ and weights w_i is on a general circle if

$$g_i(\boldsymbol{p}_i, \mathbf{x}) = x_1(p_i^2 + q_i^2) + x_2 p_i + x_3 q_i + x_4 = \boldsymbol{a}_i^\mathsf{T}(\boldsymbol{p})\, \mathbf{x} = 0 , \tag{4.522}$$

with

$$\boldsymbol{a}_i(\boldsymbol{p}) = [p_i^2 + q_i^2 \mid p_i \mid q_i \mid 1]^\mathsf{T} , \qquad \mathbf{x} = [x_1 \mid x_2 \mid x_3 \mid x_4]^\mathsf{T} . \tag{4.523}$$

Observe, this allows us to represent a circle with infinite radius if we choose $x_1 = 0$. The Jacobian C_i is given by

$$C_i = \begin{bmatrix} 2p_i & 2q_i \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} . \tag{4.524}$$

With $z_i = p_i^2 + q_i^2$ and the mean values $\bar{p} = \sum_i w_i p_i / I$, etc., we therefore have

$$M = \begin{bmatrix} \overline{z^2} & \overline{zp} & \overline{zq} & \overline{z} \\ \overline{pz} & \overline{p^2} & \overline{pq} & \overline{p} \\ \overline{qz} & \overline{qp} & \overline{q^2} & \overline{q} \\ \overline{z} & \overline{p} & \overline{q} & 1 \end{bmatrix} , \qquad N = \begin{bmatrix} 4\overline{z} & 2\overline{p} & 2\overline{q} & 0 \\ 2\overline{p} & 1 & 0 & 0 \\ 2\overline{q} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4.525}$$

The optimal parameters result from the generalized eigenvalue problem, $M\mathbf{x} = \lambda N\mathbf{x}$. The vector $\boldsymbol{x}$ the eigenvector belonging to the smallest eigenvalue. The solution is still biased (Al-Sharadqah and Chernov, 2009). The authors show that using a slightly modified matrix N leads to a direct unbiased solution, namely by using

$$N = 2N_T - N_P\,, \tag{4.526}$$

where N_T is Taubin's matrix from (4.525) and the matrix N_P, proposed by Pratt (1987), is

$$N_P = \begin{bmatrix} 0 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 \end{bmatrix}. \tag{4.527}$$

The unknown 4-vector $\widehat{\mathbf{x}}$ again is obtained as the eigenvector belonging to the smallest eigenvalue of $M - \lambda N$.

The centre $\boldsymbol{p}_0 = [p_0, q_0]^\mathsf{T}$ and the radius r of the circle $(p_i - p_0)^2 + (q_i - q_0)^2 - r^2 = 0$ can be derived from

$$\widehat{p}_0 = -\frac{\widehat{x}_2}{2\widehat{x}_1}\,, \quad \widehat{q}_0 = -\frac{\widehat{x}_3}{2\widehat{x}_1} \tag{4.528}$$

and

$$\widehat{r} = \sqrt{\frac{\widehat{x}_2^2 + \widehat{x}_3^2 - 4\widehat{x}_1\widehat{x}_4}{4\widehat{x}_1^2}}\,. \tag{4.529}$$

For determining the covariance matrix of the circle parameters, we use the cosines c_i and sines s_i of the directions of the given points to the circle centre,

$$\widehat{c}_i = \frac{p_i - \widehat{p}_0}{\widehat{r}}\,, \quad \widehat{s}_i = \frac{q_i - \widehat{q}_0}{\widehat{r}}\,. \tag{4.530}$$

The covariance matrix of the circle parameters is then given by

$$\mathbb{D}\left(\begin{bmatrix} \widehat{p}_0 \\ \widehat{q}_0 \\ \widehat{r} \end{bmatrix}\right) = \frac{\widehat{\sigma}_0^2}{\sum_i w_i} \begin{bmatrix} \overline{c^2} & \overline{cs} & \overline{c} \\ \overline{cs} & \overline{s^2} & \overline{s} \\ \overline{c} & \overline{s} & \overline{w} \end{bmatrix}^{-1}, \tag{4.531}$$

again using the over-bar for denoting the weighted mean. The estimated variance factor results from

$$\widehat{\sigma_0}^2 = \frac{\sum_i w_i \widehat{v}_i^2}{I - 3}\,, \tag{4.532}$$

with the estimated residuals $\widehat{v}_i = \widehat{r} - |\boldsymbol{p}_i - \widehat{\boldsymbol{p}}_0|$.

General Conic and Ellipse.[11] A general conic can be represented by

$$g_i(\boldsymbol{p}_i, \mathbf{x}) = x_1 p_i^2 + x_2 p_i q_i + x_3 q_i^2 + x_4 p_i + x_5 q_i + x_6 = \boldsymbol{a}_i^\mathsf{T}(\boldsymbol{p})\,\mathbf{x} = 0 \tag{4.533}$$

with

$$\boldsymbol{a}(\boldsymbol{p}_i) = [p_i^2 \mid p_i q_i \mid q_i^2 \mid p_i \mid q_i \mid 1]^\mathsf{T} \tag{4.534}$$

and the 6-vector $\mathbf{x}$. There are several ways to constrain the vector $\mathbf{x}$.

- Simple algebraic minimization enforces the norm $|\mathbf{x}|^2 = \mathbf{x}^\mathsf{T}\mathbf{x} = 1$ to be 1, thus $N = I_6$. As the linear terms in the conic determine the position of the conic, the resultant estimate will not be translation-independent.
- If we know the conic is nonsingular, then we can use the constraint (Bookstein, 1979) $x_1^2 + 2x_2^2 + x_3^2 = 1$, which can be expressed as $\mathbf{x}^\mathsf{T} N\mathbf{x} = 1$, with

[11] We will discuss conics in more detail in Sect. 5.7, p. 236.

$$N = \begin{bmatrix} \text{Diag}([1,2,1]) & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} . \tag{4.535}$$

- If we know that the conic is an ellipse, we can enforce the constraint $4x_1x_3 - x_2^2 = 1$, as proposed by Fitzgibbon et al. (1999), using

$$N = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4.536}$$

- Taubin's solution uses the Jacobian

$$C_i = \begin{bmatrix} 2p_i & 0 \\ q_i & p_i \\ 0 & 2q_i \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{4.537}$$

  and the matrix N from (4.514).
- Finally, Kanatani et al. (2012) provide an iterative scheme for the unbiased estimation of the conic parameters.

In all cases, it is necessary to transform the given data such that they lie in the square $[-1,+1]^2$ in order to arrive at well-conditioned matrices.

If the data are well-distributed across the ellipse, then the simple algebraic solution will do; otherwise, the solution by Taubin is preferable. Fitzgibbon's solution is the only one which forces the conic to be an ellipse, which is a positive feature, if the goal is to fit an ellipse. However, the ellipse is enforced also if the data perfectly lie on a hyperbola.

If the data cover only a small part, say $< 20\%$, of the circumfence, the result will be always highly unstable.

All the mentioned methods except the one by Kanatani et al. (2012) are suboptimal. A thorough comparison of the quality of these methods with the solution based on the Gauss–Helmert model still is lacking.

Quadric.[12] The method can be directly generalized to surfaces in 3D using basis functions $f_i(p,q,r)$. Especially quadrics can be determined with monomials up to order 2

$$g_i([p_i;q_i;r_i],\mathbf{x}) = \sum_{(k,l,m),0\le k+l+m\le 2} x_{klm} p_i^k q_i^l r_i^m , \qquad \sum_{(k,l,m),0\le k+l+m\le 2} x_{klm}^2 = 1 . \tag{4.538}$$

Unfortunately, the closed form solution for the ellipse cannot be transferred to determine the parameters of a constrained quadric, e.g., to an ellipsoid, as the necessary constraints are not quadratic.

4.10 Estimation in Autoregressive Models

We now discuss estimation tasks for models with autoregressive processes, introduced in Sect. 2.8.3, p. 52. Recall, an autoregressive process is a sequence $\{\underline{x}_k\}$ of random variables which are related by a recursive generation procedure $\underline{x}_k = \sum_{p=1}^{P} a_p \underline{x}_{k-p} + \underline{e}_k$.

[12] We will discuss quadrics in more detail in Sect. 5.7, p. 236.

We therefore have the following estimation problem: Given a sequence $\{x_k\}$, which is assumed to follow an AR model, determine its order P, its coefficients a_p and the variance σ_e^2 of its prediction errors $\underline{e}_k$

We only give a few estimation procedures here, which will be useful for analysing surface profiles. More details can be found in Box and Jenkins (1976).

Estimating Parameters of AR(P)-Processes. There are several methods to identify the parameters of an autoregressive process introduced in Sect. 2.8.3, p. 52, see, e.g., Box and Jenkins (1976). We give one method for estimating the parameters for an AR model of given order P and then discuss the selection of the appropriate order.

One method to identify the parameters of an autoregressive process of a given order P is to multiply the generating equation (2.197),

$$\underline{x}_n = \sum_{p=1}^{P} a_p \underline{x}_{n-p} + \underline{e}_n \,, \quad \underline{e}_n \sim \mathcal{M}\left(0, \sigma_e^2\right) \,, \tag{4.539}$$

with $\underline{x}_m$ and take the expectation. This yields what is called the *Yule–Walker equations*. If the process is initiated with $\underline{x}_n = 0, n = 1, ..., P$, it has mean zero and we obtain

$$\mathbb{C}\mathrm{ov}(\underline{x}_m, \underline{x}_n) = \sum_{p=1}^{P} a_p \mathbb{C}\mathrm{ov}(\underline{x}_m, \underline{x}_{n-p}) \qquad m < n \,. \tag{4.540}$$

We now divide (4.540) by the variance $\sigma_x^2 = \mathbb{C}\mathrm{ov}(\underline{x}_n, \underline{x}_n)$, and with the lag $l = |n - m|$ use the correlation coefficients

$$\rho_l = \frac{C_{xx}(l)}{\sigma_x^2} = \rho_{-l} \,, \qquad \rho_0 = 1 \tag{4.541}$$

to obtain the following equations:

$$\rho_l = \sum_{p=1}^{P} a_p \rho_{l-p} \,, \quad l > 0 \,. \tag{4.542}$$

The first P equations can be written as the set of the *Yule–Walker equations*,

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \dots \\ \rho_{P-1} \\ \rho_P \end{bmatrix} = \begin{bmatrix} 1 & \rho_1 & \dots & \rho_{P-2} & \rho_{P-1} \\ \rho_1 & 1 & \dots & \rho_{P-1} & \rho_{P-2} \\ \dots & \dots & \dots & \dots & \dots \\ \rho_{P-2} & \rho_{P-3} & \dots & 1 & \rho_1 \\ \rho_{P-1} & \rho_{P-2} & \dots & \rho_1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_{P-1} \\ a_P \end{bmatrix} \,, \tag{4.543}$$

for determining the coefficients of the autoregressive process (see Yule, 1927; Walker, 1931).

Practically we replace the correlation coefficients by estimates from a single sequence $\widehat{\rho}_l = \widehat{C}_{xx}(l)/\widehat{\sigma}_x^2$ by their estimates using the empirical covariances

$$\widehat{C}_{xx}(l) = \frac{1}{K-1} \sum_{n=1}^{K-l} (x_n - \widehat{\mu}_x)(x_{n+l} - \widehat{\mu}_x) \quad \text{with} \quad \widehat{\mu}_x = \frac{1}{K} \sum_{n=1}^{K} x_n \tag{4.544}$$

and observing $\widehat{\sigma}_x^2 = \widehat{C}_{xx}(0)$ (see Akaike, 1969).

Estimating the Variance and the Order of the AR-Model. As $\mathbb{E}(\underline{e}_n) = 0$, the variance σ_e^2 can easily be derived from

$$\widehat{\sigma}_e^2 = \frac{1}{N-P-1} \sum_{n=P}^{N} \widehat{e}_n^2 \,, \qquad \widehat{e}_n = x_n - \sum_{p=1}^{P} \widehat{a}_p x_{n-p} \tag{4.545}$$

The best order $\widehat{P}$ can be determined by choosing P such that it minimizes a model selection criterion, e.g., the AIC criterion, see (4.350), p. 138,

$$\mathrm{AIC}_P = -N[\log(\widehat{\sigma}_e^2)] + P\,. \tag{4.546}$$

4.11 Exercises

Basics

1. (1) Refer to Sect. 4.1, p. 75. Give examples of situations where you would prefer one of the four principles for estimation.
2. (1) Given are repeated measurements $l_n, n = 1, ..., N$, of the length x of a table using a measuring tape. Specify all elements of the Gauss–Markov model for estimating x. Assume the measurements are statistically independent. How does the mathematical model change if you exchange the measuring tape for a measuring rod, having a different accuracy, after $M \approx N/2$ observations.
3. (1) You measure a planar roof of a building using a laser range finder and obtain four 3D points $(X, Y, Z)_i, i = 1, ..., 4$. Specify a mathematical model for estimating the roof plane whether the variables are observed, given (i.e., fix), or unknown? Is the same model useful for estimating a facade plane?
4. (3) Given is a rectangular room. You measure all six distances $s_i, i = 1, ..., 6$, between the four corner points.
   a. (1) Specify a mathematical model for estimating the length a and the width b of the room. Be specific on which variables are observed, given (i.e., fixed) and unknown.
   b. (1) Assume the uncertainty of the measuring device is not perfectly known. How does the mathematical model change?
   c. (2) Assume the room is not perfectly rectangular (no room actually is!). How can you express this in the mathematical model? When is such a refinement of the mathematical model adequate?
   d. (1) Assume the measurements may contain outliers. How does the mathematical model change?
5. (2) The update equations for sequential estimation are said to allow for imposing the constraint $\Sigma_{22} = \mathbf{0}$ in the second step (see the discussion after (4.147), p. 97). Which conditions need to be fulfilled for such a step?
6. (2) Given is the total least squares model $\boldsymbol{y} + \boldsymbol{v}_y = (\mathbf{M} + \mathbf{V}_M)\boldsymbol{x}$ with $\mathbb{D}(\underline{\boldsymbol{v}}_y) = \Sigma_{yy}$, $\mathbb{D}(\mathrm{vec}\underline{\mathbf{V}}_M) = \Sigma_{mm}$, and $\mathbb{C}\mathrm{ov}(\underline{\boldsymbol{y}}, \mathrm{vec}\underline{\mathbf{V}}_M) = \mathbf{0}$.
   a. Give the linearized Gauss–Helmert model with observation vector $\boldsymbol{l} = [\boldsymbol{y}^\mathsf{T}, (\mathrm{vec}\mathbf{M})^\mathsf{T}]^\mathsf{T}$. Specifically give the Jacobians $\mathbf{A}$ and $\mathbf{B}$, see Sect. 4.8.2, p. 163, (Schaffrin and Snow, 2010).
   b. Which structure for Σ_{mm} is assumed in Mühlich and Mester (1999) as a function of $\mathbf{W}_L$ and $\mathbf{W}_R$?
7. (2) Systematic and gross errors may refer to errors in the functional or the stochastical model. Give examples for
   - systematic errors which are modelled as errors in the functional model,
   - gross errors which are modelled as errors in the functional model,
   - systematic errors which are modelled as errors in the stochastical model, and
   - gross errors which are modelled as errors in the stochastical model.

   In each case refer to sections in this book.

Proofs

8. (1) Prove (4.81), p. 89. *Hint:* Use (4.60), p. 87 and the idempotence of R.
9. (2) Prove (4.73), p. 88. *Hint:* Perform variance propagation of the joint vector $[\widehat{\underline{\boldsymbol{x}}}^\mathsf{T}, \widehat{\underline{\boldsymbol{v}}}^\mathsf{T}]$ that is a function of the observations $\underline{\boldsymbol{l}}$.
10. (2) Prove (4.144), p. 97 using the Woodbury identity (A.14), p. 769.
11. (2) Prove (4.171), p. 101 using the first equation of (4.163), p. 100.
12. (2) Prove that (4.518), p. 180 can be derived using the Gauss–Markov model, $\boldsymbol{y} + \widehat{\boldsymbol{w}} \sim \mathcal{N}(A(\tilde{\boldsymbol{l}})\widehat{\Delta\mathbf{x}}, I_G)$, with constraints $\widehat{\mathbf{x}}^\mathsf{T}\widehat{\mathbf{x}} = 1$ and $\boldsymbol{y} = -B^\mathsf{T}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{l}})\Delta\boldsymbol{l}$, see the remark on p. 181.
13. (1) Prove Steiner's rule (4.243), p. 116.
14. (2) Prove that the redundancy number r_n is in the range [0,1], if the observation l_n is uncorrelated with all others. Give an example with correlated observations where at least one redundancy number r_n does not lie in the range [0,1]. *Hint:* Generate a random estimation problem: Choose a random design matrix A, a random covariance matrix Σ_{ll} and determine the redundancy numbers.
15. (1) Prove that the eigenvalues of the subblock R_{ii} of the redundancy matrix R are in the range [0,1] if the observations in the group $\boldsymbol{l}_i$ are uncorrelated with all others, cf. (4.299), p. 128.
16. (1) Prove (4.401), p. 155.
17. (2) Prove ((4.301), p. 129) using sequential estimation in Sect. 4.2.7.2, p. 96.
18. (2) Show that the estimated mean of a sample $\{x_n\}, n = 1, ..., N$, does not change if you assume that all observations have the same variance and that they are correlated with the same correlation coefficient ρ. Give the variance $\sigma^2_{\widehat{\mu}}$ of the estimated mean $\widehat{\mu}$ as a function of the number N and the correlation coefficient ρ. Under what condition is $\sigma^2_{\widehat{\mu}}$ not negative? What is $\sigma_{\widehat{\mu}}$ for $\rho = -1$, for $\rho = +1$ and for $\rho = 0$? *Hint:* The inverse of a matrix $aI_N + b\mathbf{1}_N\mathbf{1}_N^\mathsf{T}$, with the vector $\mathbf{1}_N$ consisting of N 1s, is $cI_N + d\mathbf{1}_N\mathbf{1}_N^\mathsf{T}$, where c and d depend on a, b and N.
19. (2) Prove (4.456), p. 166. *Hint:* First assume no constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ and show that the estimated residuals can be written as $\widehat{\boldsymbol{v}} = \boldsymbol{k}_1 - \Sigma_{ll}BW_{gg}P_AB^\mathsf{T}\boldsymbol{l}$ with the projection matrix $P_A = (I - A(A^\mathsf{T}W_{gg}A)^{-1}A^\mathsf{T}W_{gg})$ and some constant vector $\boldsymbol{k}_1$ depending on the approximate values. Derive the covariance matrix for $\widehat{\boldsymbol{v}}$. Then, assume additional constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ are present and use the proof of (4.165), p. 100 to show that the correction to estimated parameters can be written as $\widehat{\Delta\boldsymbol{l}} = \boldsymbol{k}_2 + \Sigma_{\widehat{x}\widehat{x}}A^\mathsf{T}W_{gg}B^\mathsf{T}\boldsymbol{l}$ with some constant vector $\boldsymbol{k}_2$. Why does this not change the result?
20. (2) Assume the unknown parameters are partitioned as in Sect. 4.2.6, p. 94: hence, $\boldsymbol{x} = [\boldsymbol{k}^\mathsf{T}, \boldsymbol{p}^\mathsf{T}]^\mathsf{T}$, $U = U_k + U_p$, $A = [C, D]$ and the coefficient matrix C reduced to the coordinates $\boldsymbol{k}$ is $\overline{C} = C - D(D^\mathsf{T}W_{gg}D)^{-1}D^\mathsf{T}W_{gg}C$, cf. (4.122), p. 95. Show that for the Gauss–Helmert model without constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ the three matrices, relevant for the evaluation w.r.t. outliers in the observations,

$$R = \Sigma_{ll}BW_{gg}(I - A\Sigma_{\widehat{x}\widehat{x}}A^\mathsf{T}W_{gg})B^\mathsf{T} \tag{4.547}$$

$$\overline{U}_k = \Sigma_{ll}BW_{gg}\ \overline{C}(\overline{C}^\mathsf{T}W_{gg}\overline{C})^{-1}\overline{C}^\mathsf{T}\ B^\mathsf{T} \tag{4.548}$$

$$U_p = \Sigma_{ll}BW_{gg}\ D(D^\mathsf{T}W_{gg}D)^{-1}D^\mathsf{T}W_{gg}\ B^\mathsf{T} \tag{4.549}$$

are idempotent, thus

$$\mathrm{tr}(R) = \mathrm{rk}(R) = R = G - U\,, \quad \mathrm{tr}(\overline{U}_k) = \mathrm{rk}(\overline{U}_k) = U_k\,, \quad \mathrm{tr}(U_p) = \mathrm{rk}(U_p) = U_p \tag{4.550}$$

and $\mathrm{rk}(R + \overline{U}_k + U_p) = \mathrm{tr}(R + \overline{U}_k + U_p) = G$.

Computer Experiments

21. (3) The following simple camera experiment illustrates the concepts discussed in Sects. 4.2 to 4.6. Later we will assume the pose or the lens of the camera not to be ideal, and the observed data to be perturbed by blur or to contain outliers.
The lengths of flat objects lying on a conveyor belt are to be determined using a digital camera. For simplicity the camera is mounted such that the sensor plane is parallel to the plane of the conveyor belt. We only deal with the length along the conveyor belt. Then we can expect a linear relation (offset x_1 and scale x_2) between the positions s on the conveyor belt and the observed positions l' in the camera. For calibrating the configuration we observe N positions $s_n, n = 1, ..., N$, on a reference table and their image positions l_n.
Perform the following exercizes numerically with the data:

$$N = 8\,, \quad x_1 = 30\ [\text{pixel}]\,, \quad x_2 = 0.7\ [\text{pixel/mm}]\,, \quad \sigma_l = 0.3\ [\text{pixel}]\,.$$

n	1	2	3	4	5	6	7	8
s_n [mm]	78.18	152.38	228.98	442.68	538.34	825.82	913.34	996.13
l'_n [pixel]	85.02	136.51	190.38	339.95	406.85	607.77	669.05	727.18

(see HOME/4_CE.txt[13].).

a. (2) Assume the regression model has the following form

$$\mathbb{E}(\underline{l}_n) = x_1 + x_2 s_n\,, \quad \mathbb{D}(\underline{l}_n) = \sigma_l^2\,, \qquad n = 1, ..., N\,. \tag{4.551}$$

Give an explicit expression for the covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ of the estimated parameters $\boldsymbol{x}$.

b. (1) Now assume the data are centred, i.e., $\sum_n s_n = 0$ and $\sum_n l_n = 0$. Give explicit expressions for the standard deviations of the offset $\widehat{x}_1$ and the scale $\widehat{x}_2$. Interpret the expressions.

c. (1) Give an explicit expression for the estimated variance factor $\widehat{\sigma}_0^2$. *Hint:* Use $\Omega = \widehat{\boldsymbol{v}}^\mathsf{T} \mathsf{W}_{ll}^a \widehat{\boldsymbol{v}} = (\boldsymbol{l} - \boldsymbol{a})^\mathsf{T} \mathsf{W}_{ll}^a (\boldsymbol{l} - \boldsymbol{a}) - \boldsymbol{n}^\mathsf{T} \widehat{\boldsymbol{x}}$, see (4.81), p. 89

d. (2) Give explicit expressions for the elements of the hat matrix. Show that its trace is U.

e. (2) Give an explicit expression for the redundancy numbers r_n. Give an example for $s_i, i = 1, ..., 5$, where the fifth observation is at a leverage point, e.g., with $r_5 = 0.1$. Explain the situation. Give the standard deviation $\sigma_{\widehat{v}_5}$. Is a residual $\widehat{v}_5 = 1$ pixel significantly deviating from 0, when using a significance level of 95%?

f. (1) Extend the model by an additional quadratic term, $x_3 s^2$. This may be used to model lens distortion effects or the improper positioning of the camera over the conveyor belt. Assume you know the quadratic effect is small; specifically you assume $\underline{x}_3$ is a random variable with $\underline{x}_3 \sim \mathcal{M}(x_{3,0}, \sigma^2_{x_{3,0}})$, with $x_{3,0} = 0$. Specify a Gauss–Markov model which models this Bayes estimation problem with prior information on $\underline{x}_3$. Observe: You implicitly also treat the two other unknowns, x_1 and x_2, as stochastic variables, without having prior information for them. What weight matrix W_{xx} do you choose for the prior information?

g. (2) Use the regression model, however, without using prior information for x_3. Partition the parameters of the extended model into two groups, where the first is $\boldsymbol{k} := [x_1, x_2]^\mathsf{T}$, the second is $\boldsymbol{t} := x_3$. Build up the normal equation system reduced to the first group. Show that the solution for the parameters $[x_1, x_2]^\mathsf{T}$ numerically

[13] see Sect. 1.3.2.4, p. 16

is the same as with the original system with all three unknowns (without Bayes). Build the design matrix $\overline{C}$ reduced to the first two parameters and verify $\overline{N} = \overline{C}^{\mathsf{T}}\overline{C}$.

h. (2) Use the regression model, however, without using prior information for x_3. Give the redundancy numbers r_n and the contributions u_{t_n} and $\overline{u}_{k_n}$ to the parameters $t := x_3$ and $\boldsymbol{k} := [x_1, x_2]^{\mathsf{T}}$. Check their sums to be R, $U_t = 1$, and $U_k = 2$.

i. (1) Now, use the given observations. Use the regression model, however, without using prior information for x_3, as first step of a sequential estimation procedure. Add the fictitious observation $x_{3,0} = 0$ with its standard deviation $\sigma_{x_{3,0}}$ in a second step. Give the test statistic for the second estimation step. Describe the two hypotheses H_0 and H_a of this test on the prior information.

j. (2) Use the regression model, however, without using prior information for x_3. Introduce the prior information as a weak constraint with its standard deviation $\sigma_{x_{3,0}}$. Realize both versions of the model, and change $\sigma_{x_{3,0}}$ to 0 and $w_{x_{3,0}}$ to 0. Interpret both extreme cases.

22. (3) Analyse the radius of convergence of iterative estimation algorithms as a function of the chosen mathematical model and the stability of the configuration with a simulation study (refer to Sect. 4.8.3.2, p. 172). Choose two functional models for a fitting line. Implement the two iterative estimation algorithms. Use simulated data for the following investigations. Generate the parameters and observations such that they correspond to some real problem of your choice.

    a. Show that from approximate values close enough to the true data the two models for each case converge to the same values, and the resulting $\widehat{\sigma}_0$ and the theoretical covariance matrix $\Sigma_{\widehat{l}\widehat{l}}$ for the fitted observations coincide.
    b. Investigate the radius of convergence by changing the distance of the approximate values from the true values of the parameters. Draw the number of iterations as a function of the distance. What distance do you use? Is this distance independent of the unit in which the data are given?
    c. Choose observations which yield instable results in the sense that some redundancy numbers are below 0.01. Repeat the investigation on convergence. Do the convergence properties change?

23. (3) This exercise analyses the effect of the gauge of a set of 2D points on geometric entities derived from their coordinates. Refer to Sect. 4.5, p. 108 and Sect. 4.6.2.3, p. 120. Generate a set of five randomly positioned 2D points. Assume they are observed with a covariance matrix defined by an exponential covariance function, see Sect. 2.8.2, p. 50 and Sect. (4.266), p. 122.

    a. Generate the 10×10 covariance matrix $\Sigma^{(a)}$ based on the chosen covariance function. Show that its rank is 10.
    b. Transform the covariance matrix such that the gauge is in points 1 and 2, leading to the covariance matrix $\Sigma^{(12)}$. What rank does the resulting covariance matrix have?
    c. Transform the covariance matrix such that the gauge is in points 4 and 5, leading to the covariance matrix $\Sigma^{(45)}$.
    d. Determine the standard deviation of the distances s_{12} and s_{25} between the points 1, 3, and 5 using the three covariance matrices. Do the standard deviations differ or not?
    e. Determine the standard deviation of the direction $\phi_{3,5}$ with the three covariance matrices. Do the standard deviations differ or not?
    f. Determine the standard deviation of the angle $a_{123} = \phi_{23} - \phi_{21}$ with the three covariance matrices. Do the standard deviations differ or not?

    Discuss and explain the results.

24. (3) Analyse the precision and detectability w.r.t. outliers for a planar triangulation by a simulation study, see Fig. 4.17. Assume an unknown 2D point $\boldsymbol{x}$ close to the origin

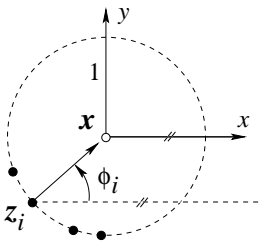


Fig. 4.17 Planar triangulation. Point $\boldsymbol{x}$ is observed from I points $\boldsymbol{z}_i$, leading to directions ϕ_i.

($\tilde{\boldsymbol{x}} = \boldsymbol{0}$) is observed from I points $\boldsymbol{z}_i$ on a circle of radius 1 m leading to directions ϕ_i. Assume the directions have the standard deviation σ. Write a program for simulating data for a given set of $\phi_i, i = 1, ..., I$, and a program for determining the best estimate for the 2D point $\boldsymbol{x}$, which also provides the covariance matrix of the estimated points, the covariance matrix of the residuals, the redundancy numbers of the observations, and the test statistic for checking the observations for outliers.

a. Generate configurations with two, three, and ten points not covering the complete perimeter such that you obtain a covariance matrix $\Sigma_{\hat{x}\hat{x}}$ where the semi-axes of the standard ellipse have a ratio of (1) approximately 1 : 1 and (2) approximately 1 : 10.
b. Generate configuration with three, four, and ten points such that all redundancy numbers (1) are equal and (2) differ by at least a factor of 10 while the standard ellipse has semi-axes with a ratio of approximately 1 : 10.
c. Generate configuration with three, four, and ten points where the correlations $\rho_{v_i v_j}$ between the residuals are below 0.4 except for one pair where the correlation is 99%, while the standard ellipse has semi-axes with a ratio (1) of approximately 1 : 2 and (2) of approximately 1 : 10.
d. Take the last case with ten points and a round standard ellipse for the estimated point. Simulate observations and perform a rigorous outlier test. Repeat the experiment 10 000 times. How large is the percentage of erroneously finding an outlier? Does it correspond to the specified significance level?
e. Introduce a just detectable outlier in observation ϕ_1. Simulate observations and perform a rigorous outlier test. Repeat the experiment 10 000 times. How large is the percentage of finding the outlier? Does it correspond to the assumed power of the test (see Sect. 3.1.1, p. 62)?

25. (3) Analyse the probability of finding outliers with RANSAC. Use or write a RANSAC procedure for fitting a straight 2D line through a set of points and a program for generating simulated data. Incorporate an internal indicator, which predicts whether RANSAC was successful. Write a routine which externally determines whether RANSAC actually was successful. For the following experiment assume the true line is $y = 0.1 + 0.4\,x$ and the generated points lie in the square $[-1, +1]^2$. Assume a success rate for the algorithm of $P = 0.99$.
Vary the following parameters, each leading to five alternatives.

- The standard deviation σ of the points varies from 0.001 to 0.1 in steps of a factor of $\sqrt{10}$.
- The percentage ε of outliers varies between 2% and 50% in steps of a factor of $\sqrt{5}$.
- The size of the outliers lies uniformly in the range of $k[5, 50]\sigma$. The range varies with $k = 1$ to $k = 100$ with a factor of $\sqrt{10}$.

- The number of generated points varies from $I = 20$ to $I = 500$ in steps of a factor of $\sqrt{5}$.

These are 625 experiments.

a. Before you analyse these experiments, document your expectations concerning the internal and external success rate and the coincidence of both as a function of the four parameters σ, ε, k, and I.
b. Repeat each experiment $N = 1000$ times and determine the probabilities that the internal and the external checks are successful and not successful.
c. Interpret the result by comparing it to your expectations.
d. Where does the outcome of the experiment differ most from the expectations?
e. Under what conditions is the expected success rate closest to the achieved success rate?
f. How would you come to a more realistic prediction of the real success rate?

Part II
Geometry

Projective geometry is an adequate tool for representation, transformation and reasoning in 2D and 3D when using perspective images (see the motivating paper by Faugeras and Papadopoulo, 1998).

In this volume objects are represented geometrically, mainly by a set of geometric features, such as points, line segments or surface regions. This includes virtual geometric objects such as vanishing points, which are either the intersection point at infinity of parallel lines or simply their direction. Though we use the calculus of algebraic projective geometry, we always think of the geometry of objects (not at infinity) or their images, which are elements in a Euclidean space.

Homogeneous coordinates are used to represent geometric entities including elements at infinity. This is the topic of Sect. 5.1, p. 195, addressing elements in 2D and 3D, namely points, lines, planes, conics and quadrics. We will use covariance matrices of homogeneous coordinate vectors to represent the uncertainty of geometric elements in Chap. 10.

Transformations of these entities, including motions, similarities or straight line-preserving mappings, called homographies are the topic of Chap. 6, p. 247. Due to the specific representation with homogeneous coordinates, they in essence are matrix-vector multiplications, why concatenation and inversion of transformation is simplified, namely reducing to matrix multiplication and inversion. *Geometric reasoning* with these entities addresses the construction of new entities (e.g., intersection of lines) and expressing specific geometric relations between them (e.g., the incidence of a point and a plane) is dealt with in Chap. 7, p. 291. Most of these constructions and constraints lead to multilinear forms in the coordinates, which will be used in Chap. 10 to simplify uncertainty propagation. The construction of new entities covers the determination of conics and transformations from a minimal set of geometric entities, which are useful within random sample consensus type robust estimators. The matrix representation of joins and intersections presented here follows Förstner et al. (2000), is motivated by early work of Brand (1947), and eases statistical reasoning.

Rotations in 3D, the topic of Chap. 8, p. 325, are the nonlinear part in spatial motions and similarities. They deserve special attention, as there is a variety of representations, each of which is useful for certain tasks. Representing rotations using the exponential map of skew matrices is the basis for representing uncertain rotations in Chap. 10 and generalizes to the other transformations.

Oriented projective geometry, the topic of Chap. 9, p. 343, allows us to distinguish lines of opposite directions, slightly modifying the concept of homogeneous coordinates. This is useful for spatial reasoning about left-to-right or in-front-of relations, either in the plane or in space. All geometric construction developed in Chap. 7 can be used for oriented geometric entities.

Chapter 10, p. 359 on *uncertain projective geometry* addresses the representation, the testing, and the estimation of statistically uncertain geometric entities and transformations. We develop adequate representations for uncertain geometric entities and transformations and provide algebraically and statistically optimal solutions including the uncertainty of the estimated parameters for a number of relevant tasks, partly including closed form solutions. We describe the concept of what is called reduced homogeneous coordinates (Förstner, 2010a), which allows us to efficiently estimate the coordinates of a large number of geometric entities, and in this way it extends the work of Kanatani (1993) and Heuel (2004).

Chapter 5
Homogeneous Representations of Points, Lines and Planes

5.1 Homogeneous Vectors and Matrices 195
5.2 Homogeneous Representations of Points and Lines in 2D 205
5.3 Homogeneous Representations in $\mathbb{P}^n$ 209
5.4 Homogeneous Representations of 3D Lines 216
5.5 On Plücker Coordinates for Points, Lines and Planes 221
5.6 The Principle of Duality 229
5.7 Conics and Quadrics 236
5.8 Normalizations of Homogeneous Vectors 241
5.9 Canonical Elements of Coordinate Systems 242
5.10 Exercises 245

This chapter motivates and introduces homogeneous coordinates for representing geometric entities. Their name is derived from the homogeneity of the equations they induce. Homogeneous coordinates represent geometric elements in a projective space, as inhomogeneous coordinates represent geometric entities in Euclidean space. Throughout this book, we will use Cartesian coordinates: inhomogeneous in Euclidean spaces and homogeneous in projective spaces. A short course in the plane demonstrates the usefulness of homogeneous coordinates for constructions, transformations, estimation, and variance propagation. A characteristic feature of projective geometry is the symmetry of relationships between points and lines, called duality. In this chapter we aim at exploiting the algebraic properties of the representations of geometric entities and at giving geometrically intuitive interpretations.

5.1 Homogeneous Vectors and Matrices

5.1.1 Definition and Notation 195
5.1.2 A Short Course 197

5.1.1 Definition and Notation

Definition 5.1.2: ***Homogeneous coordinates (J. Plücker 1829).*** Homogeneous coordinates $\mathbf{x}$ of a geometric entity $\mathcal{X}$ are invariant with respect to multiplication by a scalar $\lambda \neq 0$: thus $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same entity $\mathcal{X}$. ◇

We will find homogeneous representations for geometric entities, such as points, lines and planes, but also for transformations. The homogeneous representation is not unique, as $\lambda \neq 0$ can be chosen arbitrarily; $\mathbf{x}$ and $-\mathbf{x}$ represent the same entity. Uniqueness of the entity is guaranteed as long as not all coordinates vanish, thus $|\mathbf{x}| \neq 0$.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_5

In certain applications it will be useful to restrict the freedom of scaling and to distinguish between opposite directions, e.g., when reasoning about the left or the right side of an entity or when modelling a real camera: points always have to be located in front of a camera, and this needs to be reflected in the modelling. This leads to *oriented* entities whose representation is only invariant to the multiplication with a positive scalar.

It will occasionally be useful to reduce the ambiguity of the scaling and normalize homogeneous entities. We will distinguish between spherical normalization, which we denote by the index s, e.g. $\mathbf{x}^s$ where $|\mathbf{x}^s| = 1$, and similarly the Euclidean normalization $\mathbf{x}^e$ where some of the elements of $\mathbf{x}^e$ can be interpreted as elements in Euclidean space.

Due to these representational properties, we need to clarify the usage of the equal sign "=" in the context of homogeneous entities. It has three uses:

1. The equal sign is used to indicate equality, following the convention in mathematics.
2. The equal sign is used to indicate a value assignment as in some computer languages. For example, $\mathbf{l} = \mathbf{x} \times \mathbf{y}$ is read as the vector $\mathbf{l}$ is to be determined as the cross product of $\mathbf{x}$ and $\mathbf{y}$. This is sometimes written as $\mathbf{l} := \mathbf{x} \times \mathbf{y}$.
3. The equal sign is used to state that the representations on the left and the right-hand sides refer to the same object. Thus the two representations are equal up to scaling. The equation above $\mathbf{l} = \mathbf{x} \times \mathbf{y}$ (a homogeneous relation), thus can be read as a condition for the line parameters $\mathbf{l}$ to be equal to the parameters of the line connecting the points $\chi(\mathbf{x})$ and $y(\mathbf{y})$. This sometimes is written as $\mathbf{l} \cong \mathbf{x} \times \mathbf{y}$, or as $\lambda\mathbf{l} = \mathbf{x} \times \mathbf{y}$ with some $\lambda \neq 0$, making the scale factor explicit.

We will use the simple equal sign and only specify the relations 2 and 3 if the context requires.

In contrast to inhomogeneous entities such as $\boldsymbol{l}$, $\boldsymbol{X}$, and R, homogeneous entities are designated with upright letters, such as $\mathbf{l}$, $\mathbf{X}$, and P. Planes are designated with letters from the beginning of the alphabet, lines with letters from the middle of the alphabet and points with letters from the end of the alphabet. Points and lines in the plane will be called 2D points and 2D lines, in contrast to 3D points and 3D lines in space.

We distinguish between *names* and *representations* of geometric entities. The symbol $\mathcal{X}$ denotes the name of the point whereas its coordinates are denoted by $\boldsymbol{X}$ or $\mathbf{X}$; thus, we can write $\mathcal{X}(\boldsymbol{X})$ or $\mathcal{X}(\mathbf{X})$ depending on our assumptions about the representation of the point $\mathcal{X}$. The notations used are collected in Tables 5.1 and 5.2.

Table 5.1 Names of basic geometric entities in 2D and 3D

element	2D	3D
planes		$\mathcal{A}$, $\mathcal{B}$, ...
lines	$\mathcal{l}$, $\mathcal{m}$, ...	$\mathcal{L}$, $\mathcal{M}$, ...
points	$\mathcal{x}$, $\mathcal{y}$, ...	$\mathcal{X}$, $\mathcal{Y}$, ...

Table 5.2 Notation for inhomogeneous and homogeneous vectors and matrices

	2D	3D	transformations
inhomogeneous	$\boldsymbol{x}$	$\boldsymbol{X}$	R
homogeneous	$\mathbf{l}$, $\mathbf{x}$	$\mathbf{A}$, $\mathbf{L}$, $\mathbf{X}$	H

Homogeneous coordinates have a number of advantages which make them indispensable in our context:

- They allow us to represent entities at infinity, which occurs frequently, e.g., when dealing with vanishing points. Conceptually, homogeneous coordinates are the natural

representation of elements of a projective space, by which we mean the corresponding Euclidean space together with the elements at infinity of all lines in that plane.
- Homogeneous coordinates allow us to easily represent straight line-preserving transformations, thus not only translations, rotations or affine transformations but also projective transformations, e.g., when representing the mapping from 3D object space to 2D image space in a pinhole camera.
- They simplify concatenation and inversion of straight line-preserving transformations, since all transformations are represented as a matrix vector product.
- They simplify the construction of geometric elements from given ones as well as the expression of geometric constraints as sets of homogeneous equations.
- All geometric operations, constructions, and transformations are bilinear forms. As a consequence, the uncertainty of vectors and matrices using covariance matrices can easily be propagated, as the necessary Jacobians are derived without effort.

We will first introduce the basic ideas in a short course, motivating the content of the chapter, and then discuss the individual concepts in detail.

5.1.2 A Short Course

This subsection is meant to give an intuitive introduction to the use of homogeneous coordinates in 2D space and to exemplify their advantages for points and lines and their relations and transformations.

5.1.2.1 Representation with Homogeneous Coordinates

The Hessian normal form of a straight line ℓ in the xy-plane is given by *Hessian normal form*

$$x\cos\phi + y\sin\phi - d = 0\,, \tag{5.1}$$

see Fig. 5.1. Whereas the point $\mathcal{X}$ is represented with its inhomogeneous coordinates $\boldsymbol{x} = [x, y]^\mathsf{T}$, thus $\mathcal{X}(x, y)$, the line ℓ is represented with the Hessian coordinates $\boldsymbol{h} = [\phi, d]^\mathsf{T}$, namely the direction ϕ of its normal in mathematically positive, i.e., counterclockwise, direction counted from the x-axis, and its distance d from the origin O,[1] thus $\ell(\phi, d)$.

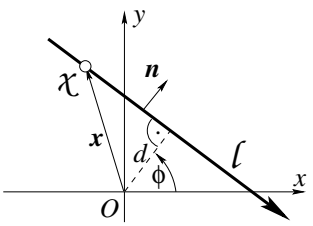


Fig. 5.1 Straight line with parameters of Hessian normal form. The normal direction $\boldsymbol{n}$ of the line points to the left w.r.t. the direction (or orientation, cf. Sect. 9.1.1.3, p. 346) of the line

Equation (5.1) may be written in different forms and allows different interpretations:

- The equation represents the incidence of the point $\mathcal{X}(\boldsymbol{x})$ with the line $\ell(\boldsymbol{h})$. This symmetric incidence relation $\iota(\mathcal{X}, \ell)$ is equivalent to the dual relations: "The point $\mathcal{X}$ lies on the line ℓ" and "The line ℓ passes through the point $\mathcal{X}$".
- The equation may be written as

[1] We assume the distance is measured in the direction of the normal.

$$\boldsymbol{n}^\mathsf{T}\boldsymbol{x} = d \qquad \text{with} \qquad \boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \tag{5.2}$$

if we use the normal vector $\boldsymbol{n}$. It suggests the line is to be represented by three parameters, $[n_x, n_y, d]^\mathsf{T}$. However, they satisfy one constraint, namely $|\boldsymbol{n}| = 1$. The representation with $\boldsymbol{n}$ does not have a singularity when estimating the direction $\boldsymbol{n}$, unlike the angle representation of the direction with ϕ (see the discussion on using quaternions for representing rotations, Sect. 8.1.5.2, p. 335). This has significant advantages.

- The equation may be written as

$$\mathbf{x}^{e\mathsf{T}}\mathbf{l}^e = 0 \qquad \text{with} \qquad \mathbf{x}^e = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{l}^e = \begin{bmatrix} \cos\phi \\ \sin\phi \\ -d \end{bmatrix}. \tag{5.3}$$

This suggests that both the point and the line are to be represented with 3-vectors, thus $\mathcal{X}(\mathbf{x}^e)$ and $\mathcal{L}(\mathbf{l}^e)$. They are homogeneous vectors, as multiplying them with an arbitrary scalar $\neq 0$ does not change the incidence relation. But they are normalized in a well-defined way, namely such that the inhomogeneous parameters (x, y) and d can be directly inferred. We will discuss normalization below.
Moreover, the equation is symmetric in $\mathbf{x}$ and $\mathbf{l}$ as $\mathbf{x}^\mathsf{T}\mathbf{l} = \mathbf{l}^\mathsf{T}\mathbf{x} = 0$, which algebraically reflects the symmetry of the incidence property $\iota(\mathcal{X}, \mathcal{L})$.

- The equation may more generally be written as

$$\mathbf{x}^\mathsf{T}\mathbf{l} = 0 \tag{5.4}$$

with the vectors

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{5.5}$$

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \pm\sqrt{a^2+b^2} \begin{bmatrix} \cos\phi \\ \sin\phi \\ -d \end{bmatrix}. \tag{5.6}$$

The factors $w \neq 0$ and $|[l_1, l_2]| = \sqrt{a^2+b^2} \neq 0$ can be chosen arbitrarily. Therefore, points and lines can be represented by nearly arbitrary 3-vectors, namely by restricting the absolute value of $w = x_3$ and the absolute value of $[a, b]^\mathsf{T} = [l_1, l_2]^\mathsf{T}$ not to be zero.
As the relation (5.4) is a homogeneous equation, the corresponding representations of the points are homogeneous, and the 3-vectors $\mathbf{x}$ and $\mathbf{l}$ are called the homogeneous coordinates of the point $\mathcal{X}$ and the line $\mathcal{L}$ respectively.

homogeneous coordinates

We can easily determine the Euclidean representation of the point and the line from

$$x = \frac{u}{w} \quad y = \frac{v}{w}, \qquad \phi = \operatorname{atan2}(b, a) \quad d = -\frac{c}{\sqrt{a^2+b^2}} \tag{5.7}$$

or

$$\boldsymbol{x} = \frac{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}{x_3}, \qquad \phi = \operatorname{atan2}(l_2, l_1) \quad d = -\frac{l_3}{\left|\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right|}. \tag{5.8}$$

5.1.2.2 Normalizations

Homogeneous coordinates of a point or a line are not unique. Uniqueness may be achieved by normalization, i.e., by fixing the scale factor. Two types of normalizations are common, Euclidean and spherical.

Euclidean Normalization. By *Euclidean normalization* the vector is transformed such that the Euclidean properties become visible (Fig. 5.2). We obtain

$$\mathbf{x}^e = \mathsf{N}^e(\mathbf{x}) = \frac{1}{w}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}, \qquad \mathbf{l}^e = \mathsf{N}^e(\mathbf{l}) = \frac{1}{\sqrt{a^2+b^2}}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} \\ -d \end{bmatrix}. \tag{5.9}$$

Therefore, following Brand (1966), we introduce the following notation for points and lines to specify the Euclidean part and the homogeneous part of a homogeneous vector. The Euclidean part, indexed by 0, implicitly contains the Euclidean properties: for points the two coordinates, and for lines the distance to the origin,

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix}, \qquad \mathbf{l} = \begin{bmatrix} \boldsymbol{l}_h \\ l_0 \end{bmatrix}. \tag{5.10}$$

Euclidean normalization then reads as

$$\mathbf{x}^e = \frac{\mathbf{x}}{x_h}, \qquad \mathbf{l}^e = \frac{\mathbf{l}}{|\boldsymbol{l}_h|}. \tag{5.11}$$

Spherical Normalization. By *spherical normalization* all coordinates of a homogeneous vector are processed the same way and the complete vector is normalized to 1 (Fig. 5.3). The spherically normalized homogeneous coordinates of a 2D point $\mathcal{X}$ and of a 2D line $\mathcal{l}$ are

$$\mathbf{x}^s = \mathsf{N}(\mathbf{x}) = \frac{1}{\sqrt{u^2+v^2+w^2}}\begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \mathbf{l}^s = \mathsf{N}(\mathbf{l}) = \frac{1}{\sqrt{a^2+b^2+c^2}}\begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{5.12}$$

Thus the spherically normalized homogeneous coordinates of all 2D points and 2D lines build the unit sphere S^2 in $\mathbb{R}^3$.

We will frequently use spherically normalized homogeneous vectors. They have several advantages:

1. They lie on a sphere, which is a closed manifold without any borders. Thus geometrically, i.e., if we do not refer to a special coordinate system, there are no special points in the projective plane.
2. The redundancy in the representation – we use three coordinates for a 3D entity – requires care in iterative estimation procedures, as the length constraint needs to be introduced. Iteratively correcting spherically normalized vectors can be realized in the tangent space which for 2D points is a tangent plane at the spherically normalized vector.
3. As the points $\mathbf{x}^s$ and the point $-\mathbf{x}^s$ represent the same 2D point, the representation is not unique. Taking these two points as two different ones leads to the concept of oriented projective geometry, which among other things can distinguish between lines with different orientation (Chap. 9, p. 343).

5.1.2.3 Geometric Interpretation of Homogeneous Coordinates and the Projective Plane

The last two paragraphs suggest an intuitive and important geometric interpretation of homogeneous coordinates as embedding the real plane $\mathbb{R}^2$ with origin O_2 and axes x and y into the 3D Euclidean space $\mathbb{R}^3$ with origin O_3 and axes u, v and w, cf. Fig. 5.2, left. The Euclidean normalized coordinate vector $\mathbf{x}^e = [u, v, w]^\mathsf{T} = [x, y, 1]^\mathsf{T}$ lies in the plane $w = 1$. The origin O_2 has coordinates $\mathbf{x}_{O_2} = [0, 0, 1]^\mathsf{T}$. The u- and the v-axes are parallel to the x- and the y-axes respectively. Thus, adding the third coordinate, 1, to an inhomogeneous coordinate vector $\boldsymbol{x}$ to obtain $\mathbf{x}^e$ can be interpreted as embedding the real Euclidean plane

Euclideanly normalized vector

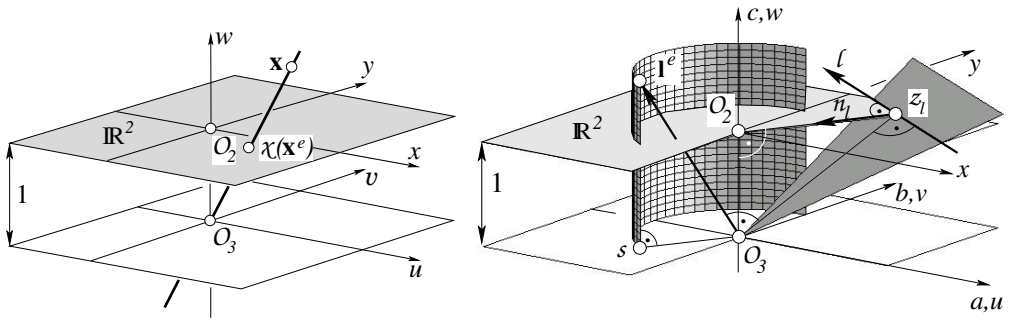


Fig. 5.2 Representation with homogeneous coordinates, Euclidean normalization. **Left:** 2D point. The real plane $\mathbb{R}^2$ is embedded into the 3D space $\mathbb{R}^3$ with coordinates u, v and w. Any vector $\mathbf{x}$ on the line joining the origin O_3 of the (uvw)-coordinate system and the point χ, except the origin itself, can represent the point χ on the Euclidean plane $\mathbb{R}^2$. The intersection of the line χO_3 with the plane $w = 1$ yields the Euclideanly normalized homogeneous coordinate vector $\mathbf{x}^e$ of χ. **Right:** 2D line. The real plane $\mathbb{R}^2$ is embedded into the 3D space $\mathbb{R}^3$. Coordinates a, b and c are used to represent 2D lines. The 2D line ℓ is represented by the normal $\mathbf{l}$ of the plane passing through the origin O_3 and the line. When the Euclideanly homogeneous coordinates $\mathbf{l}^e$ (5.9), p. 199 are used, their first two elements are normalized to 1, and the vector $\mathbf{l}^e$ lies on the unit cylinder (distance $sO_3 = 1$) parallel to the w-axis. The distance of the line ℓ from the origin O_2, which is in the direction of the normal n_l, is identical to the c-component of $\mathbf{l}^e$

$\mathbb{R}^2$ into $\mathbb{R}^3$. Points with coordinates $\mathbf{x} = \lambda\mathbf{x}^e$ are on a straight line through the origin O_3. They represent the same point, namely χ. You can also say: the straight line χO_3, taking χ, which is embedded in the 3D (uvw)-space, represents the homogeneous point χ.

A similar geometric interpretation can be given for lines. Here, we embed the real plane $\mathbb{R}^2$ into $\mathbb{R}^3$, but with an (a, b, c)-coordinate system at O_3.

The vector $\mathbf{l}^e := [a, b, c]^\mathsf{T} = [\cos\phi, \sin\phi, -d]^\mathsf{T}$ lies on the vertical cylinder $a^2 + b^2 = 1$ with unit radius, see Fig. 5.2, right. The vector $\mathbf{l}^e$ is the normal of the plane through O_3 and ℓ, as $\mathbf{x}^\mathsf{T}\mathbf{l}^e = 0$ for all points on ℓ. The coordinate d of this vector is equal to the distance of the line ℓ from the origin O_2, as can be proven geometrically by investigating the coplanar triangles (O_2, z_l, O_3) and $(s, \mathbf{l}^e, O_3)$.

Exercise 5.13

The spherically normalized homogeneous coordinates can be geometrically interpreted in a similar way.

spherically normalized vectors

The point $\mathbf{x}^s$ lies on the unit sphere S^2 in the three-dimensional (uvw)-space $\mathbb{R}^3$, see Fig. 5.3, left. Obviously, the negative vector $-\mathbf{x}^s$, also representing the point χ, lies on the unit sphere. All points on the unit sphere S^2, except those on the equator $u^2 + v^2 = 1$, represent points of $\mathbb{R}^2$.

The points on the equator have a well-defined meaning: when a point χ moves away from the origin O_2 towards infinity, its spherically normalized homogeneous vector moves towards the equator. Thus, points on the equator of S^2 represent points χ_∞ at infinity. They are represented by homogeneous coordinate vectors with $w = 0$, independently of their normalization.

point at infinity

If we take the union of all points in the Euclidean plane $\mathbb{R}^2$ and all points at infinity, we obtain what is called the *projective plane* $\mathbb{P}^2$. Both can be represented by the unit sphere, with opposite points identified.

projective plane

The point $\mathbf{l}^s$ also lies on the unit sphere in the three-dimensional (abc)-space $\mathbb{R}^3$, see Fig. 5.3, right. It is the unit normal of the plane through O_3 and the line ℓ. This plane intersects the unit sphere in a great circle.

The relation between this circle and the normal $\mathbf{l}^s$ is called *polarity on the sphere*: the point $\mathbf{l}^s$ is what is called the *pole* of the circle; the circle is the *polar* of the point $\mathbf{l}^s$.

polarity on the sphere

If a line moves towards infinity, its homogeneous vector moves towards the c-axis. Therefore, the origin O_2 or its antipode represent the line ℓ_∞ at infinity. Since lines are dual to points, cf. below, this unit sphere S^2 represents the dual projective plane.

This visualization of the projective plane is helpful for understanding certain constructions and will be used throughout.

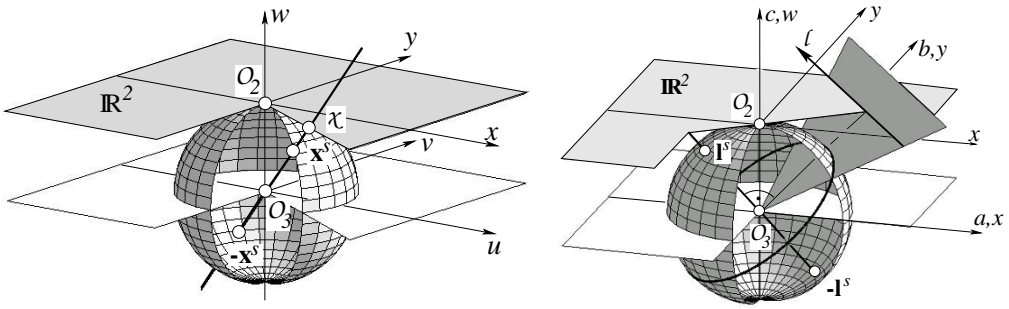


Fig. 5.3 Spherical normalization. **Left:** 2D point. The spherically normalized point $\mathbf{x}^s$ lies on the upper hemisphere of the unit sphere S^2. The antipode $-\mathbf{x}^s$ also represents the point $\mathcal{X}$. Points on the equator $u^2 + v^2 = 1$ represent points at infinity. **Right:** 2D line ℓ lying in $\mathbb{R}^2$. The spherically normalized homogeneous vector $\mathbf{l}^s$ is the unit normal of the plane through O_3 and ℓ. When seeing the plane $(O_3\ell)$ from $\mathbf{l}^s$, the origin O_3 is on the left side of the line ℓ. Therefore, the antipode point $-\mathbf{l}^s$ represents the same line, however, with opposite direction

5.1.2.4 Line Joining Two Points, Intersection of Two Lines, and Elements at Infinity

Now, let us determine the line

$$\ell = \mathcal{X} \wedge y \tag{5.13}$$

joining two points $\mathcal{X}$ and y. The symbol $\wedge$ (read: wedge) indicates the join. If the two points are given with their homogeneous coordinates, thus $\mathcal{X}(\mathbf{x})$ and $y(\mathbf{y})$, the joining line is given by

$$\ell = \mathcal{X} \wedge y: \quad \mathbf{l} = \mathbf{x} \times \mathbf{y} = \mathsf{S}(\mathbf{x})\mathbf{y}\,, \tag{5.14}$$

as then the vector $\mathbf{l}$ is perpendicular to $\mathbf{x}$ and $\mathbf{y}$, thus $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$ and $\mathbf{y}^\mathsf{T}\mathbf{l} = 0$; thus, the line passes through both points. Matrix $\mathsf{S}(\mathbf{x})$ is the skew symmetric matrix induced by the 3-vector $\mathbf{x}$, Exercise 5.1

$$\mathsf{S}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{5.15}$$

A first remark on notation: The symbol for the join of two geometric entities is not unique in the literature. The wedge sign "$\wedge$" often is used for the cross product in physics. This is the reason for using it here for the join of two points, as the homogeneous coordinates of the resulting line is obtained by the cross product. Observe: some authors use the sign $\vee$ for the join of two points.

We will overload the symbol in two ways: (1) We will use it also for the join of geometric entities in 3D, namely 3D points and 3D lines. (2) We will also use it for the corresponding algebraic entities. Thus we could have written in (5.14) the expression $\mathbf{l} = \mathbf{x} \wedge \mathbf{y}$, keeping in mind how the operation is performed algebraically. Applying the wedge to two 3-vectors therefore is identical to determining their cross product, independently of what the two vectors represent. *overloading of $\wedge$ and $\cap$*

A similar reasoning leads to the homogeneous coordinates of the intersection point,

$$\mathcal{X} = \ell \cap m: \quad \mathbf{x} = \mathbf{l} \times \mathbf{m} = \mathsf{S}(\mathbf{l})\mathbf{m}\,, \tag{5.16}$$

of two lines $\ell(\mathbf{l})$ and $m(\mathbf{m})$ given with homogeneous coordinates, where the sign $\cap$ (read: cap) indicates the intersection.

A second remark on notation: It would be more consistent to use the sign $\vee$ for the intersection. We found that in longer expressions it is difficult to distinguish visually between the $\wedge$-symbol and the $\vee$-symbol. Therefore, we intentionally use the sign $\cap$ for the intersection, which corresponds to the sign for the intersection of sets. This appears intuitive, as the intersection point is the set-intersection of the two lines, taken as the set

of infinitely many points. Again we will overload the symbol both for 3D entities, namely 3D lines and planes, and for algebraic entities.

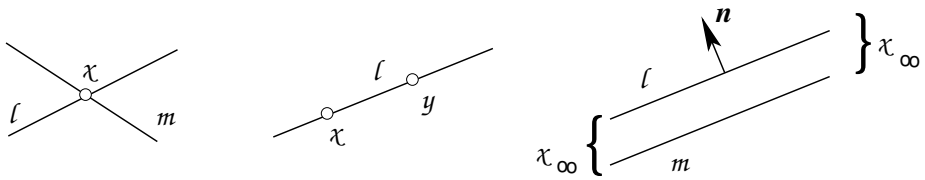


Fig. 5.4 Intersection $\mathcal{X} = \mathcal{l} \cap \mathcal{m}$ **(left)** and join $\mathcal{l} = \mathcal{X} \wedge \mathcal{y}$ **(centre)** of two geometric entities. The intersection of two parallel lines **(right)** leads to the point at infinity $\mathcal{X}_\infty$. The figure indicates this point to be the result of two different limiting processes. Given the direction $[u, v]^\mathsf{T}$ of the line, we may end up with $\mathbf{x}_\infty = \lim_{w \downarrow 0} [u, v, w]^\mathsf{T} = [u, v, 0]^\mathsf{T}$. But due to homogeneity, we also have $-\mathbf{x}_\infty = \lim_{w \uparrow 0} [u, v, w]^\mathsf{T} = \lim_{-w \downarrow 0} [-u, -v, -w]^\mathsf{T} = [-u, -v, 0]^\mathsf{T}$, the vector pointing in the opposite direction. In Sect. 9, p. 343, oriented projective geometry, we will distinguish between these two directions

If all vectors are spherically normalized, we arrive at a very intuitive interpretation of the construction equations, see Fig. 5.5. 2D points correspond to and are represented as points on the unit sphere, whereas 2D lines correspond to great circles on the unit sphere and are represented as unit normals, thus also as points on the unit sphere. The two constructions read as

$$\mathbf{l}^s = \mathsf{N}(\mathbf{x}^s \times \mathbf{y}^s) \qquad \text{and} \qquad \mathbf{x}^s = \mathsf{N}(\mathbf{l}^s \times \mathbf{m}^s), \tag{5.17}$$

which can be derived geometrically from the two graphs in Fig. 5.5.

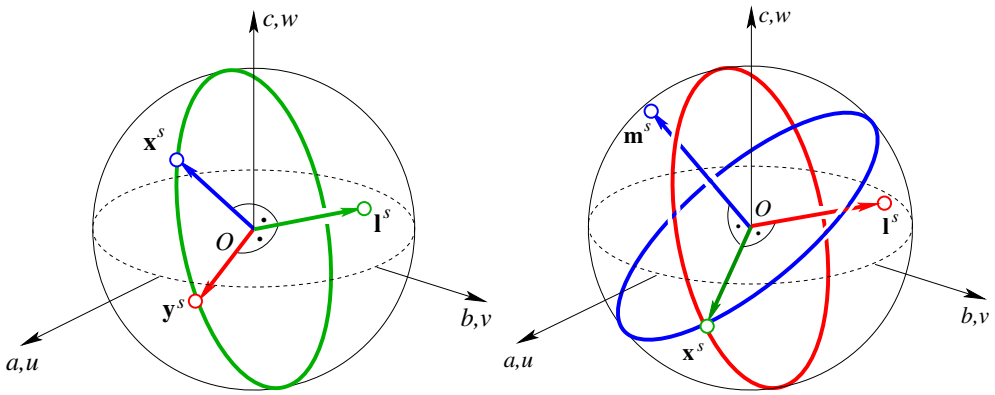


Fig. 5.5 Join of two points and intersection of two lines on the projective planes $\mathbb{P}^2$ and its dual plane $\mathbb{P}^{*2}$ superimposed on the same unit sphere (for the definition of $\mathbb{P}^{*2}$ cf. (5.38), p. 209). **Left:** The 2D line $\mathcal{l}$ joining any two 2D points $\mathbf{x}^s, \mathbf{y}^s$ on the projective plane is the great circle through these points. The normal $\mathbf{l}^s$ of the plane containing the great circle is determined by the normalized cross product of the two homogeneous coordinate vectors. **Right:** The 2D intersection point of any two 2D lines on the projective plane is the intersection of the two great circles defined by their normals $\mathbf{l}^s$ and $\mathbf{m}^s$. The direction of the intersection point $\mathbf{x}^s$ is the normalized cross product of the two normals of the planes containing the two great circles. If the intersection $\mathbf{x}^s$ lies on the equator, its last coordinate is zero, indicating the point is at infinity; thus, the two lines are parallel. Observe, the cross products are unique, a property which we will exploit when discussing oriented elements

Two parallel lines do not intersect in a point in the real plane but at infinity, which cannot be represented with inhomogeneous coordinates. However, the cross product of their homogeneous coordinates exists. This allows us to explicitly represent points at infinity.

Let the two lines, see Fig. 5.4, right, have the common normal $\boldsymbol{n}$ and two different distances d_1 and d_2 from the origin, with

$$\boldsymbol{n}^{\perp} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \boldsymbol{n} \tag{5.18}$$

perpendicular to the normal $\boldsymbol{n}$ of the lines. Then the homogeneous coordinates of their intersection point $\mathcal{X}_\infty$ are obtained from

Exercise 5.14

$$\mathbf{x}_\infty = \begin{bmatrix} \boldsymbol{n} \\ -d_1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{n} \\ -d_2 \end{bmatrix} = \begin{bmatrix} (d_2 - d_1)\boldsymbol{n}^{\perp} \\ 0 \end{bmatrix} \cong \begin{bmatrix} \boldsymbol{n}^{\perp} \\ 0 \end{bmatrix}. \tag{5.19}$$

Thus, the first two components $[u, v]^\mathsf{T}$ of the 3-vector of a point at infinity,

point at infinity

$$\mathcal{X}_\infty : \quad \mathbf{x}_\infty = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix}, \tag{5.20}$$

represent the direction towards the point at infinity, whereas the third component is zero.

Two points, $\mathcal{X}_\infty([u_x, v_x, 0]^\mathsf{T})$ and $\mathcal{Y}_\infty([u_y, v_y, 0]^\mathsf{T})$, at infinity span what is called the *line at infinity*,

Exercise 5.2

line at infinity

$$\mathcal{L}_\infty : \quad \mathbf{l}_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{5.21}$$

as the cross product yields $[0, 0, u_x v_y - u_y v_x]^\mathsf{T}$, which is proportional to $[0, 0, 1]^\mathsf{T}$. Any other point at infinity lies on the line at infinity.

All points with $\mathbf{x} \in \mathbb{R}^3 \setminus \mathbf{0}$, assuming proportional vectors represent the same point, are elements of the projective plane $\mathbb{P}^2$.[2] Reasoning with such projective elements is at the heart of projective geometry.

projective plane

All lines with $\mathbf{l} \in \mathbb{R}^3 \setminus \mathbf{0}$ are elements of the corresponding dual projective plane. This corresponds to the notion of a vector space for points and its dual for its linear forms. We will exploit the concept of duality between points and lines and generalize it to both 3D points and the corresponding transformations.

dual projective plane

Observe, the coordinates of the line $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ are not the same as the coordinates of the line $\mathcal{L} = \mathcal{Y} \wedge \mathcal{X}$, but are their negatives, as the cross product is anti-symmetric. This allows us to distinguish between lines with different directions if we follow certain sign conventions. For example, if we assume points to be represented with positive third component, we can distinguish between the sign of the lines $\mathcal{X} \wedge \mathcal{Y}$ and $\mathcal{Y} \wedge \mathcal{X}$, as their normals differ by 180°. If we consistently consider the sign conventions, we arrive at the *oriented projective geometry*, which is the topic of Chap. 9.

oriented projective geometry

The 2D coordinate system can be described by its origin $\mathcal{X}_0$ and its axes $\mathcal{L}_x$ and $\mathcal{L}_y$, with coordinates identical to unit 3-vectors $\boldsymbol{e}_i^{[3]}$,

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{e}_3^{[3]}, \qquad \mathbf{l}_x = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \boldsymbol{e}_2^{[3]}, \qquad \mathbf{l}_y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{e}_1^{[3]}. \tag{5.22}$$

Note that the x-axis seen as a line $\mathcal{L}_x$ has the Euclidean normal $[0, 1]^\mathsf{T}$ and passes through the origin, therefore $\mathbf{l}_x = \boldsymbol{e}_2$, not $\mathbf{l}_x = \boldsymbol{e}_1$. We will discuss the elements of coordinate systems in detail in Sect. 5.9.

5.1.2.5 Duality of Points and Lines

Each geometric element, operation, and relation has what is called a *dual*, indicated by $\overline{(\cdot)}$. The concept of duality results from the underlying three-dimensional vector space $\mathbb{R}^3$ for

[2] Mathematically, this is the quotient space $\mathbb{P}^2 = (\mathbb{R}^3 \setminus \mathbf{0})/(\mathbb{R} \setminus \mathbf{0})$, indicating that all vectors $\mathbf{x} \in \mathbb{R}^3 \setminus \mathbf{0}$ are taken as equivalent if they are multiplied with some $\lambda \in \mathbb{R} \setminus \mathbf{0}$.

representing points, with the dual vector space $\mathbb{R}^{*3}$, which contains all linear forms $\mathbf{l}^\mathsf{T}\mathbf{x}$, represented by the vector $\mathbf{l}$. As the two spaces are isomorphic, there is a natural mapping $D : \mathbb{R}^3 \mapsto \mathbb{R}^{*3}$, namely the identity mapping $\mathbf{x} \mapsto \mathbf{l}$.

Given the point $\mathbf{x} = [u, v, w]^\mathsf{T}$, the line $\mathbf{l}$ which is dual to this point has the same coordinates as the point

$$\mathbf{l} = \overline{\mathbf{x}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{5.23}$$

and vice versa.

Therefore, a given 3-vector $[r, s, t]^\mathsf{T}$ can be interpreted as either a 2D point $\mathbf{x} = [r, s, t]^\mathsf{T}$ or a 2D line $\mathbf{l} = [r, s, t]^\mathsf{T}$; they are dual to each other.

The point $\mathcal{X}$ and the dual line $\mathcal{l}$ are positioned on opposite sides of the origin O, with distances d_{xO} and d_{lO} to the origin multiplying to 1, thus $d_{xO}\, d_{lO} = 1$ (Table 7.3, p. 298). The line through $\mathcal{X}$ and perpendicular to the line $\mathcal{l}$ passes through the origin, see Fig. 5.6. We will see that this property transfers to 3D.

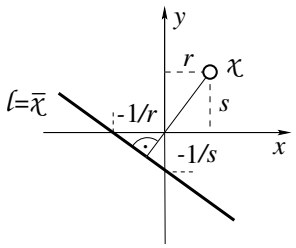


Fig. 5.6 Duality of points and lines in the plane. Point $\mathcal{x}$ and line $\mathcal{l}$ are dual w.r.t. each other. They have the same homogeneous coordinates $[r, s, t = 1]^\mathsf{T}$: $\mathcal{x}(\boldsymbol{x} = [r, s]^\mathsf{T})$ and $\mathcal{l}(rx + sy + 1 = 0)$, from which the intersection points $[-1/r, 0]$ and $[0, -1/s]$ with the axis can be derived

For spherically normalized homogeneous coordinates, we see from Fig. 5.3 that a point $\mathbf{x}^s$ and its dual line $\mathbf{l}^s = \overline{\mathbf{x}}^s$ are related by polarity on the sphere, cf. Sect. 5.1.2.3, p. 200. For more on duality cf. Sect. 5.6, p. 229.

5.1.2.6 Transformation of Points

Linear mappings of homogeneous coordinates can be used to represent classical transformations. For example, we have the translation $\mathcal{T}$ and the rotation $\mathcal{R}$,

$$\mathcal{x}' = \mathcal{T}(\mathcal{x}) : \quad \mathbf{x}' = \mathsf{T}\mathbf{x} \quad \text{with} \quad \mathsf{T}([t_x, t_y]^\mathsf{T}) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

and

$$\mathcal{x}' = \mathcal{R}(\mathcal{x}) : \quad \mathbf{x}' = \mathsf{R}\mathbf{x} \quad \text{with} \quad \mathsf{R}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.25}$$

homogeneous matrix

which can easily be verified. Observe, the two 3×3 matrices are homogeneous entities: their multiplication with a scalar $\mu \neq 0$ does not change the transformation, as the resulting vector is multiplied with $\mu \neq 0$, leaving the resulting point unchanged.

Concatenation and inversion are obviously easy, since the geometric transformations are represented as matrix vector products.

matrix representation for point

Observe, the join of two points in (5.14) is also a matrix vector multiplication, suggesting that the skew matrix $\mathsf{S}(\mathbf{x})$ is a matrix representation of the point. We will generalize this

property for all basic geometric entities and transformations and derive a representation with homogeneous vectors and matrices.

We will see that a general linear mapping of homogeneous coordinates is straight line-preserving.

5.1.2.7 Variance Propagation and Estimation

All relations discussed so far are bilinear in the elements of the coordinates involved. Therefore, we may easily derive the Jacobians needed for variance propagation.

For example, from

$$\mathbf{l} = \mathbf{x} \times \mathbf{y} = -\mathbf{y} \times \mathbf{x} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x} \tag{5.26}$$

we immediately obtain the two Jacobians

$$\frac{\partial(\mathbf{x} \times \mathbf{y})}{\partial \mathbf{y}} = \mathsf{S}(\mathbf{x}) \qquad \text{and} \qquad \frac{\partial(\mathbf{x} \times \mathbf{y})}{\partial \mathbf{x}} = -\mathsf{S}(\mathbf{y}). \tag{5.27}$$

The line coordinates are nonlinear functions of the point coordinates, namely sums of products. Following Sect. 2.7.6, p. 44, in a first approximation if the two points are stochastically independent with covariance matrices Σ_{xx} and Σ_{yy}, we obtain the covariance matrix of the joining line,

$$\Sigma_{ll} = \mathsf{S}(\boldsymbol{\mu}_x)\Sigma_{yy}\mathsf{S}^\mathsf{T}(\boldsymbol{\mu}_x) + \mathsf{S}(\boldsymbol{\mu}_y)\Sigma_{xx}\mathsf{S}^\mathsf{T}(\boldsymbol{\mu}_y). \tag{5.28}$$

Of course, we will need to discuss the meaning of the covariance matrix of a homogeneous entity and the degree of approximation resulting from the homogeneity of the representation.

Finally, we will discuss estimation techniques for homogeneous entities: we may use the homogeneity of the constraints to advantage to obtain approximate values. For example, let us assume N points $\mathbf{x}_n$, $n = 1, ..., N$, are given and we want to determine a best fitting straight line. Due to measurement deviations, the points and the unknown line will not satisfy the constraints $\mathbf{x}_n^\mathsf{T}\mathbf{l} = 0$ but will result in some residual $\mathbf{x}_n^\mathsf{T}\mathbf{l} = w_n$. Minimizing the length of the vector $\boldsymbol{w} = [w_1, ..., w_n, ..., w_N]^\mathsf{T}$, i.e., minimizing the sum of squared residuals $\boldsymbol{w}^\mathsf{T}\boldsymbol{w} = \sum_{n=1}^{N} w_n^2$ w.r.t. the line parameters under the constraint $|\mathbf{l}| = 1$, leads to minimizing the Rayleigh ratio

$$r = \frac{\mathbf{l}^\mathsf{T}\left(\sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^\mathsf{T}\right)\mathbf{l}}{\mathbf{l}^\mathsf{T}\mathbf{l}} \to \min, \tag{5.29}$$

which is known to be equivalent to solving an eigenvalue problem. As this method does not take the possibly different uncertainties of the points into account, we also need to discuss statistically optimal estimates. The special structure of the constraints will simplify the setup of the corresponding estimation problem.

5.2 Homogeneous Representations of Points and Lines in 2D

5.2.1 2D Point 206
5.2.2 2D Line 207

This section gives formal definitions of homogeneous coordinates of 2D points and 2D lines. It completes the descriptions of the concepts given so far for 2D space: the 3-vectors establishing the projective plane for points and the dual projective plane for lines. Both contain points and lines at infinity.

5.2.1 2D Point

A 2D point χ with inhomogeneous coordinates $\boldsymbol{x} = [x, y]^\mathsf{T}$ has homogeneous coordinates[3]

$$\chi(\mathbf{x}): \qquad \mathbf{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix} = \begin{bmatrix} u \\ v \\ \hline w \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ \hline w \end{bmatrix} . \tag{5.30}$$

The factor $w \neq 0$ can be chosen arbitrarily. If the inhomogeneous coordinates $\boldsymbol{x}$ are given, we use $w = 1$.

The coordinate $x_h = w$ is called the *homogeneous part*, specifically the homogeneous coordinate of the coordinate vector $\mathbf{x}$, while the other part, $\boldsymbol{x}_0 = [u, v]^\mathsf{T}$, is the inhomogeneous or *Euclidean part*. The homogeneous part determines the scale factor when going from inhomogeneous coordinates to homogeneous ones. This is a general feature of homogeneous representations (Brand, 1966).

We use the notation $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T}$ when we have to handle multiple points. The notation $\mathbf{x} = [u, v, w]^\mathsf{T}$ is the most common in computer graphics, and the notation $\mathbf{x} = [\boldsymbol{x}_0^\mathsf{T}, x_h]^\mathsf{T}$ is preferred when we explicitly want to point out the Euclidean interpretation of the homogeneous entity.

If $w \neq 0$, any 3-vector $\mathbf{x}$ can represent a point in the Euclidean plane $\mathbb{R}^2$ with coordinates

$$\boldsymbol{x} = \boldsymbol{c}(\mathbf{x}) \quad \text{with} \quad \boldsymbol{c}(\mathbf{x}) = \frac{\boldsymbol{x}_0}{x_h} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \end{bmatrix} . \tag{5.31}$$

Obviously, points in $\mathbb{R}^2$ can be described by both representations, as long as $w \neq 0$; thus, we may specify $\chi(\boldsymbol{x})$ or $\chi(\mathbf{x})$.

Corollary 5.2.1: *Point at infinity.* A point $[u, v, 0]^\mathsf{T}$ with homogeneous component $x_h = w = 0$ is called *ideal point* or *point at infinity*. ◇

Points at infinity may be used to represent directions in the plane with Euclidean direction vector $[u, v]^\mathsf{T}$. This can be seen from the limiting process,

$$\lim_{w \to 0} \begin{bmatrix} u/w \\ v/w \end{bmatrix} ,$$

which moves the point $[u/w, v/w]^\mathsf{T}$ towards infinity in the direction $[u, v]^\mathsf{T}$ as seen from the origin. However, opposite directions are unified, i.e., treated as identical, since $[u, v, 0]^\mathsf{T} \cong [-u, -v, 0]^\mathsf{T}$. A direction $\measuredangle$ in the plane thus has homogeneous coordinates,

$$\mathbf{d} = \begin{bmatrix} \cos \alpha_x \\ \cos \alpha_y \\ 0 \end{bmatrix} , \tag{5.32}$$

where the angles α_x and α_y are the angles between the direction and the x- and y-axes, respectively.

As opposite directions are unified, the points at infinity may lie in either direction of the line. This definition transfers to all other dimensions. If we do not want to exploit the difference of opposite directions, we need to use the concepts of oriented projective geometry, discussed in Sect. 9, p. 343.

Similarly to collecting all points $\boldsymbol{x} \in \mathbb{R}^2$ in the Euclidean plane, we collect all points with homogeneous coordinates in the projective plane.

[3] This is a convention also found in computer graphics. However, we also could put the homogeneous component w as the first coordinate and start counting the vector elements at 0, thus $\mathbf{x} = [x_i] = [x_0, x_1, x_2]^\mathsf{T}$. This results in a mathematically more convincing representation, as it directly generalizes to higher dimensions; since the first element is the homogeneous one, the last index is identical to the dimension of the space. For 3D lines, however, this type of argumentation cannot be used. We therefore adopt the most commonly used convention here.

Definition 5.2.3: *Projective plane.* The projective plane, $\mathbb{P}^2(\mathbb{R})$, contains all points $\mathcal{X}$ with real-valued 3-vectors $\boldsymbol{x} = [u, v, w]^\mathsf{T} \in \mathbb{R}^3 \setminus \mathbf{0}$,

$$\mathcal{X} \in \mathbb{P}^2(\mathbb{R}) : \quad \mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3 \setminus \mathbf{0}\,, \tag{5.33}$$

with

$$\mathcal{X}(\mathbf{x}) \equiv \mathcal{Y}(\mathbf{y}) \;\Leftrightarrow\; \mathbf{x} = \lambda \mathbf{y}, \quad \text{for some } \lambda \neq 0. \tag{5.34}$$

Two points $\mathcal{X}(\mathbf{x})$ and $\mathcal{Y}(\mathbf{y})$ are equal if their homogeneous coordinates are identical up to a scale factor $\lambda \neq 0$. ◇

The projective plane $\mathbb{P}^2$ consists of all points $[x, y]^\mathsf{T}$ of the Euclidean plane $\mathbb{R}^2$ and the points $[u, v, 0]^\mathsf{T}$ at infinity, which itself is identical to a projective line $\mathbb{P} = \mathbb{P}^1$, cf. below. Similarly, we denote the projective plane by $\mathbb{P}^2$.

As $\mathbf{x} \in \mathbb{R}^3 \setminus \mathbf{0}$ and all $\lambda\mathbf{x}$ with $\lambda \neq 0$ represent the same point, and since $\lambda\mathbf{x}$ is a 3D line passing through the origin of the $\mathbb{R}^3$, we may identify points in the projective plane $\mathbb{P}^2$ with lines through the origin of $\mathbb{R}^3$. We can use this equivalence relation to visualize $\mathbb{P}^2$ as the set of points on the sphere S^2 with opposite points identified, cf. Sect. 5.1.2.3, p. 199.

The three coordinates of the homogeneous vector actually have only two degrees of freedom, as the scale of the vector is arbitrary. Thus, a 2D point is still specified by a minimum of two independent parameters.

degrees of freedom of a 2D point

Oriented projective geometry distinguishes between the point with homogeneous coordinates $\mathbf{x}$ and the point with homogeneous coordinate $-\mathbf{x}$. We will discuss this in Chap. 9, p. 343.

5.2.2 2D Line

A 2D line $\mathcal{l}$ with implicit representation

$$ax + by + c = 0 \tag{5.35}$$

or representation in Hessian normal form (cf. Fig. 5.7),

$$x \cos\phi + y \sin\phi - d = 0\,, \tag{5.36}$$

has homogeneous coordinates

$$\mathcal{l} : \quad \mathbf{l} = \begin{bmatrix} \boldsymbol{l}_h \\ l_0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ \hline c \end{bmatrix} = \pm\sqrt{a^2 + b^2} \begin{bmatrix} \cos\phi \\ \sin\phi \\ \hline -d \end{bmatrix}. \tag{5.37}$$

The sub-vector $\boldsymbol{l}_h = [a, b]^\mathsf{T}$ is called the homogeneous part of the line $\mathbf{l}$, while $l_0 = c$ is the inhomogeneous or Euclidean part of the line coordinates. Observe, the homogeneous part has length 1 when the Hessian form, the 3-vector in the last term in (5.37), is used. The homogeneous part $\boldsymbol{l}_h$ is the normal of the line proportional to $\boldsymbol{n}$, while l_0 is proportional to the distance d of the line from the origin and has opposite sign, thus $\text{sign}(l_0) = -\text{sign}(d)$.

Therefore, the partitioning of the line vector into homogeneous and Euclidean parts is different from that for points but facilitates the following geometric expressions. 2D lines with finite distance to the origin can be described in all three representations, as long as $\sqrt{a^2 + b^2} \neq 0$.

Exercise 5.3

Definition 5.2.4: *Line at infinity.* The line $\mathcal{l}_\infty$ with homogeneous coordinates $[0, 0, 1]^\mathsf{T}$ with homogeneous part $\boldsymbol{l}_h = \mathbf{0}$ is called the *line at infinity.* ◇

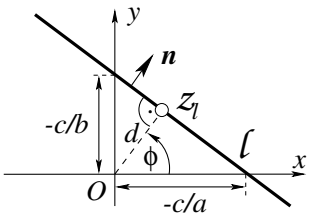


Fig. 5.7 2D line example. Line l with coordinates $\mathbf{l} = [a, b, c]^\mathsf{T} = [3, 4, -10]^\mathsf{T}$ in 2D. Parameters ϕ and d of the Hessian normal form: $0.6x + 0.8y - 2 = 0$; thus, the normal is $\boldsymbol{n} = [0.6, 0.8]^\mathsf{T}$, the direction angle of the normal is $\phi = \arctan(0.8/0.6) \approx 53.13°$ and the distance of the line to the origin is $d = +2$. Alternative equivalent representations of the line are: $y = 2.5 - 0.75x$, $3x + 4y - 10 = 0$ or $x/3.333 + y/2.5 = 1$. The coordinates of the footpoint z are $[1.2, 1.6]$ and can be used to represent the line

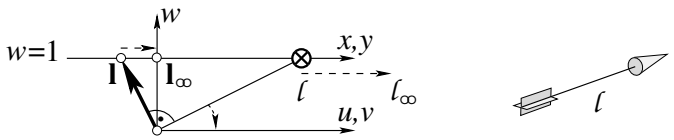


Fig. 5.8 Line at infinity. If the homogeneous line vector $\mathbf{l}$ of a finite line l is shifted to $\mathbf{l}_\infty = [0, 0, 1]^\mathsf{T}$, the line is shifted to infinity. We assumes the line l is directed, with its direction into the drawing plane. The cross symbolizes the feathers (cf. right; if the direction of the line shows in the opposite direction, a dot would symbolize the tip of the arrow)

The direction of the line at infinity is not specified, as can be seen when calculating the angle ϕ using (5.7), see Fig. 5.8.

Corollary 5.2.2: ***Line at infinity of a plane.*** The line at infinity of a plane contains the points at infinity of all lines in that plane. ◇

For the projective plane $\mathbb{P}^2$ this holds as all points at infinity satisfy the constraint $\mathbf{x}_\infty^\mathsf{T}\mathbf{l}_\infty = 0$. The line at infinity can be visualized as the horizon of the plane. The concept transfers to arbitrary planes in 3D, cf. Sect. 5.4.4, p. 219.

Example 5.2.15: ***The horizon as the image of the line at infinity of a horizontal plane.*** Figure 5.9 shows a perspective image of a scene with a horizontal plane covered by a regular hexagonal pattern. Here, the 2D elements in the image are denoted by a prime, $'$, in order to distinguish them from the 3D elements, a convention which we will use in Part III.

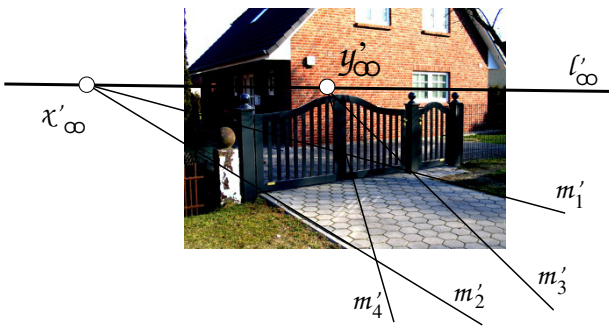


Fig. 5.9 Perspective view of a scene with the horizon l'_∞, which is the vanishing line of the horizontal plane. It may be constructed by the two vanishing points x'_∞ and y'_∞ which themselves are the intersections of the images of two pairs of lines: $l'_\infty = (m'_1 \cap m'_2) \wedge (m'_3 \cap m'_4)$ which are parallel on the ground plane, here consisting of hexagonal plates

The image line l'_∞ is the image of the line at infinity of the ground plane, i.e., the image of the horizon. It can be constructed from the join of the images x'_∞ and y'_∞ of two vanishing points, which themselves are determined as the intersections of the images of two pairs of parallel lines in the ground plane derived from the regular pattern. From the image of the horizon we can infer parts of the rotation of the camera w.r.t.

a coordinate system of the building. We will discuss perspective mappings and how to infer information about the camera and the 3D scene from a single image, e.g., how to derive rectified images of the facade or the ground plane in later sections. ◇

Corollary 5.2.3: *Point at infinity of a set of parallel lines.* Parallel lines have a common point at infinity. ◇

Proof: We show the proof for 2D lines. Parallel lines have the same normal, thus can be represented as $\mathbf{l}_i = [a, b, c_i]^\mathsf{T}$ with fixed normal $[a, b]^\mathsf{T}$ and arbitrary c_i. The point at infinity of these lines has the coordinates $\mathbf{x}_{l\infty} = [-b, a, 0]$, as it points in the direction $[-b, a]^\mathsf{T}$ and is at infinity. ◇

The notion of a point at infinity of sets of parallel lines in the plane transfers to sets of parallel lines in 3D space. All lines ℓ with parameters $[a, b, c]^\mathsf{T} \neq \mathbf{0}$ build what is called the *dual projective plane* $\mathbb{P}^{*2}$.

Definition 5.2.5: *Dual projective plane.* The dual projective plane $\mathbb{P}^{*2}$ contains all lines ℓ with 3-vectors, $\mathbf{l} = [a, b, c]^\mathsf{T} \in \mathbb{R}^3 \setminus \mathbf{0}$,

$$\ell \in \mathbb{P}^{*2}: \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \in \mathbb{R}^3 \setminus \mathbf{0}\,, \tag{5.38}$$

with

$$\ell(\mathbf{l}) \equiv m(\mathbf{m}) \;\Leftrightarrow\; \mathbf{l} = \lambda \mathbf{m}, \quad \text{for some } \lambda \neq 0. \tag{5.39}$$

◇

Since the dual projective plane $\mathbb{P}^{*2}$ is isomorphic to $\mathbb{P}^2$, we often will not distinguish between the two, as when discussing the join of two points and the intersection of two lines in Sect. 5.1.2.4, and where we superimposed the projective plane of points and the dual projective plane of lines, see Fig. 5.5, p. 202.

If the orientations of two lines with coordinates $\mathbf{l}$ and $-\mathbf{l}$ are treated as distinct, we need to employ the concept of oriented projective geometry, cf. Chap. 9, p. 343.

2D Line in Point-Direction Form. The classic explicit representation with reference point $\mathbf{x}_0$, given with inhomogeneous coordinates $\boldsymbol{x}_0 \neq \mathbf{0}$ and direction vector $\boldsymbol{d}$, reads as

$$\boldsymbol{x}(\alpha) = \boldsymbol{x}_0 + \alpha \boldsymbol{d} \qquad \alpha \in \mathbb{R}\,. \tag{5.40}$$

The point at infinity of the line cannot be represented this way.

A similar expression can be obtained with homogeneous coordinates if the line is given by two points $\mathbf{x}_0$ and $\mathbf{x}_1$, namely

$$\mathbf{x}(\alpha) = (1 - \alpha)\mathbf{x}_0 + \alpha(\mathbf{x}_1 - \mathbf{x}_0)\,, \tag{5.41}$$

as the two points need to lie on ℓ, therefore $\mathbf{x}_i^\mathsf{T}\mathbf{l} = 0$ and hence also $\mathbf{x}^\mathsf{T}(\alpha)\ \mathbf{l} = 0$. This representation allows us to choose α such that $\mathbf{x}(\alpha)$ is the point at infinity.

5.3 Homogeneous Representations in $\mathbb{P}^n$: 3D Points, the Plane and 1D Points

5.3.1 3D Point 210
5.3.2 Plane 211
5.3.3 1D Point 214
5.3.4 The Projective Space $\mathbb{P}^n$ 214

This section generalizes the concept of the projective plane to other dimensions. Most important is the three-dimensional projective space for representing points and planes in

3D. It includes points at infinity sitting on the plane of infinity, which can be visualized as the celestial sphere. The one-dimensional projective line, generalizing the real line by including the point at infinity, is the prototype of a line in higher dimensions and later serves to define the cross ratio. The generalization to dimensions higher than three is required for the discussion of 3D lines and transformations. We will finally arrive at a recursive definition of projective space as the union of the real space of the same dimension and the elements at infinity.

5.3.1 3D Point

A 3D point $\mathcal{X}$ with inhomogeneous coordinates $\boldsymbol{X} = [X, Y, Z]^\mathsf{T}$ analogously has coordinates

$$\mathcal{X}: \quad \mathbf{X} = \begin{bmatrix} \boldsymbol{X}_0 \\ X_h \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ \hline T \end{bmatrix} = \begin{bmatrix} TX \\ TY \\ TZ \\ \hline T \end{bmatrix}. \tag{5.42}$$

The factor $T \neq 0$ can again be chosen arbitrarily. If the inhomogeneous coordinates are given, we use $T = 1$.

The coordinate $X_h = T$ is called the *homogeneous part*, specifically the homogeneous coordinate of the coordinate vector $\mathbf{X}$, while the other part $\boldsymbol{X}_0 = [U, V, W]^\mathsf{T}$ is the inhomogeneous or *Euclidean part*.

Inversely, if $T \neq 0$, any 4-vector $\mathbf{X}$ can be interpreted as a point in $\mathbb{R}^3$ with coordinates

$$\boldsymbol{X} = \frac{\boldsymbol{X}_0}{X_h} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} U/T \\ V/T \\ W/T \end{bmatrix}. \tag{5.43}$$

3D point at infinity

Points $[U, V, W, 0]^\mathsf{T}$ with homogeneous component $X_h = T = 0$ are *points at infinity*. They occur as points at infinity of 3D lines and therefore as pre-images of vanishing points in images, cf. Sect. (12.3), p. 523. They may be used to represent directions in $\mathbb{R}^3$ with Euclidean direction vector $[U, V, W]^\mathsf{T}$ and can be visualized as infinitely remote stars on the celestial sphere, see Fig. 5.10.

A direction $\mathcal{D}$ in 3D space thus has homogeneous coordinates

$$\mathbf{D} = \begin{bmatrix} \cos\alpha_X \\ \cos\alpha_Y \\ \cos\alpha_Z \\ 0 \end{bmatrix}, \tag{5.44}$$

where the angles α_X, α_Y, and α_Z are the angles between the direction and the X-, the Y-, and the Z-axes, respectively. Especially when referring to a point on the earth's surface, with the Z-axis representing the direction to the zenith, the angle α_Z is the zenith angle of the direction $\mathcal{D}$.

All points $\mathcal{X}$ with homogeneous coordinates $\mathbf{X} = [U, V, W, T]^\mathsf{T} \in \mathbb{R}^4 \setminus \mathbf{0}$ form what is called the *projective space* $\mathbb{P}^3$.

Definition 5.3.6: *Projective space.* The projective space $\mathbb{P}^3$ contains all 3D points $\mathcal{X}$ with homogeneous coordinates $\mathbf{X} = [U, V, W, T]^\mathsf{T} \in \mathbb{R}^4 \setminus \mathbf{0}$,

$$\mathcal{X} \in \mathbb{P}^3: \quad \mathbf{X} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} \in \mathbb{R}^4 \setminus \mathbf{0}, \tag{5.45}$$

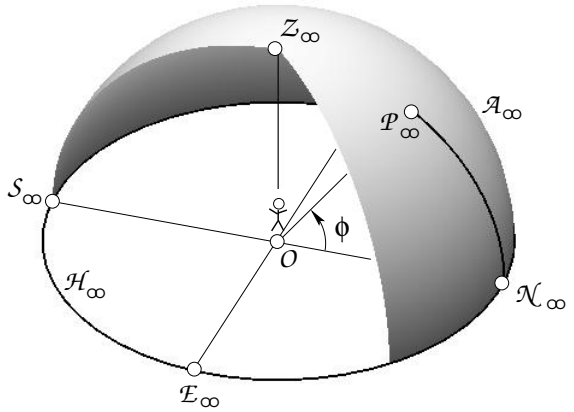


Fig. 5.10 Celestial sphere at a point O on the earth's surface. It shows the projective plane $\mathcal{A}_\infty$ of all points at infinity in a three-dimensional projective space $\mathbb{P}^3$, specifically the north pole $\mathcal{P}_\infty$, here shown for a latitude of $\phi = 52°$, and the zenith $\mathcal{Z}_\infty$. It also contains the horizon $\mathcal{H}_\infty$ with the south point $\mathcal{S}_\infty$, the east point $\mathcal{E}_\infty$ and the north point $\mathcal{N}_\infty$ used for local navigation. If the plane at infinity is the 2-sphere S^2 in $\mathbb{R}^3$, this visualization contains all finite points somewhere inside the unit sphere. The unit ball B^2 is therefore a way to visualize all points in $\mathbb{P}^3$. We will formalize this visualization in Sect. 5.9, p. 242

with

$$\mathcal{X}(\mathbf{X}) \equiv \mathcal{Y}(\mathbf{Y}) \;\Leftrightarrow\; \mathbf{X} = \lambda \mathbf{Y}, \quad \text{for some } \lambda \neq 0. \tag{5.46}$$

The coordinates of identical points may differ by a factor $\lambda \neq 0$. ◇

The projective space $\mathbb{P}^3$ consists of all points $[X, Y, Z]^\mathsf{T}$ of the Euclidean space $\mathbb{R}^3$ and the points $[U, V, W, 0]^\mathsf{T}$ at infinity, which build a projective plane $\mathbb{P}^2$. The four coordinates of the homogeneous vector actually have only three degrees of freedom, as the scale of the vector is arbitrary. Thus, a 3D point is still specified by the minimum of three independent parameters.

5.3.2 Plane

The plane $\mathcal{A}$ with implicit representation

$$AX + BY + CZ + D = 0\,, \tag{5.47}$$

or, when represented with normal vector $\boldsymbol{N}$ and distance S to the origin, which again is measured in the direction of the normal,

$$\boldsymbol{X}^\mathsf{T}\boldsymbol{N} - S = 0, \qquad |\boldsymbol{N}| = 1\,, \tag{5.48}$$

has homogeneous coordinates

$$\mathcal{A}: \quad \mathbf{A} = \begin{bmatrix} \boldsymbol{A}_h \\ A_0 \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ \hline D \end{bmatrix} = \pm\sqrt{A^2 + B^2 + C^2} \begin{bmatrix} N_X \\ N_Y \\ N_Z \\ \hline -S \end{bmatrix} \tag{5.49}$$

with arbitrary factor $\sqrt{A^2 + B^2 + C^2}$, see Fig. 5.11, p. 212. It is obvious that a plane has three degrees of freedom. The sub-vector $\boldsymbol{A}_h = [A, B, C]^\mathsf{T}$ is called the homogeneous part of the plane vector $\mathbf{A}$, while $A_0 = D$ is the inhomogeneous or Euclidean part of the plane coordinates.

The homogeneous part $\boldsymbol{A}_h$ is the nonnormalized normal vector of the plane, while A_0 is proportional to the distance of the plane to the origin, and again has the opposite sign. If $\boldsymbol{A}_h$ is normalized to 1, the fourth element is the signed distance of the plane to the origin.

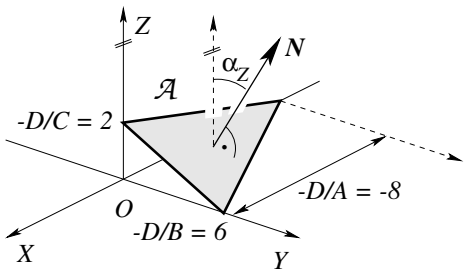


Fig. 5.11 Plane $\mathcal{A}$ in 3D with normal vector $\boldsymbol{N}$. Here, we have the special case $\mathbf{A} = [-3, 4, 12, -24]^\mathsf{T}$ with the normal $\boldsymbol{N} = [-3, 4, 12]^\mathsf{T}/13$ and distance $S = 24/13 \approx 1.846$ to the origin. The zenith angle of the normal or the tilt angle of the plane is $\alpha_Z = \arccos(12/13) \approx 22.6°$; the slope is $\tan\alpha_Z \approx 0.417 = 41.7\%$

The normal vector $\boldsymbol{N} = [\cos\alpha_X, \cos\alpha_Y, \cos\alpha_Z]^\mathsf{T} = \boldsymbol{A}_h/|\boldsymbol{A}_h|$ contains the cosines of the angles of the normal direction $\mathcal{N}$ with the three coordinate axes. The slope of the plane can either be represented by its tilt angle, i.e., the angle with the XY-plane, which is identical to α_Z, or by the tangent $\tan\alpha_Z$ of this angle, often given as a percentage, thus $100\tan\alpha_Z\%$.

Planes in $\mathbb{R}^3$ can be described in all three representations of (5.49) as long as $|\boldsymbol{A}_h| \neq 0$. The plane $\mathcal{A}_\infty$ with coordinates $[0, 0, 0, 1]^\mathsf{T}$, i.e., with homogeneous part $\boldsymbol{A}_h = \mathbf{0}$, represents the *plane at infinity*.

plane at infinity

Corollary 5.3.4: *Plane at infinity of the 3D space* $\mathbb{P}^3$. The plane at infinity, $\mathcal{A}_\infty(\mathbf{A}_\infty)$, of the 3D space $\mathbb{P}^3$ contains the points at infinity, $\mathcal{X}_\infty(\mathbf{X}_\infty)$, of all 3D lines. ⋄

This holds as for all points at infinity we have $\mathbf{X}_\infty^\mathsf{T}\mathbf{A}_\infty = 0$. The plane at infinity can be visualized as the celestial sphere of our 3-space (Fig. 5.10, p. 211).

5.3.2.1 The Dual Projective Space $\mathbb{P}^{*3}$

All planes $\mathcal{A}$ with coordinates $[A, B, C, D]^\mathsf{T} \neq \mathbf{0}$ form what is called the *dual projective space* $\mathbb{P}^{*3}$.

Definition 5.3.7: *Dual projective space.* The dual projective space $\mathbb{P}^{*3}$ contains all planes $\mathcal{A}$ with homogeneous coordinates $\mathbf{A} = [A, B, C, D]^\mathsf{T} \in \mathbb{R}^4 \setminus \mathbf{0}$,

$$\mathcal{A} \in \mathbb{P}^{*3}: \quad \mathbf{A} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \in \mathbb{R}^4 \setminus \mathbf{0} \tag{5.50}$$

with

$$\mathcal{A}(\mathbf{A}) \equiv \mathcal{B}(\mathbf{B}) \;\Leftrightarrow\; \mathbf{A} = \lambda\mathbf{B}, \quad \text{for some } \lambda \neq 0, \tag{5.51}$$

where planes are identical if their homogeneous coordinates differ by a factor $\lambda \neq 0$. ⋄

Again, as with $\mathbb{P}^{*2}$ in 2D, the dual projective space $\mathbb{P}^{*3}$ is isomorphic to the projective space $\mathbb{P}^3$, and we will generally not between them. A plane has three degrees of freedom.

If the plane is not at infinity, i.e., $\boldsymbol{A}_h \neq \mathbf{0}$, we may determine the normalized normal vector $\boldsymbol{N}$ and the distance S to the origin from

$$\boldsymbol{N} = \frac{\boldsymbol{A}_h}{|\boldsymbol{A}_h|} \qquad S = -\frac{A_0}{|\boldsymbol{A}_h|}. \tag{5.52}$$

The orientation of a plane is obviously defined by its normal vector; thus, two planes with coordinates $\mathbf{A}$ and $-\mathbf{A}$ are treated as different in oriented projective geometry, since their normals differ, cf. Chap. 9, p. 343. With each plane in 3D we may associate its line at infinity, which only depends on its normal direction, cf. Sect. 5.4.4, p. 219.

5.3.2.2 Plane in Point-Direction Form

The classical explicit representation for all points $\mathcal{X}$ on a plane $\mathcal{A}$ is given by a reference point $\mathcal{X}_0$ and two nonparallel directions $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ which will usually be chosen to be mutually perpendicular, see Fig. 5.12:

$$\boldsymbol{X} = \boldsymbol{X}_0 + t_1\boldsymbol{D}_1 + t_2\boldsymbol{D}_2, \tag{5.53}$$

where t_1, t_2 are the coordinates in the plane. Its normal is given by $\boldsymbol{A}_h = \boldsymbol{D}_1 \times \boldsymbol{D}_2/|\boldsymbol{D}_1 \times \boldsymbol{D}_2|$, and its distance S to the origin is $A_0 = \boldsymbol{A}_h^\mathsf{T}\boldsymbol{X}_0$.

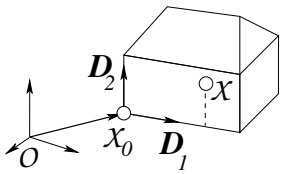


Fig. 5.12 Plane in point-direction form

Extending the vectors by a homogeneous part yields the homogeneous coordinates $\mathbf{X}$ in 3D of a point $\mathcal{X} \in \mathcal{A}$ as an explicit function of the plane homogeneous coordinates $\mathbf{t}$. The coordinate system on the plane is given by the origin $\mathcal{X}_0$ and the two directions $\mathbf{D}_i^\mathsf{T} = [\boldsymbol{D}_i^\mathsf{T}, 0]^\mathsf{T}$ (Heikkila, 2000):

$$\mathbf{X} = \mathbf{X}_0 + t_1\mathbf{D}_1 + t_2\mathbf{D}_2 = \underbrace{\begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{D}_2 & \boldsymbol{X}_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathsf{H}} \begin{bmatrix} t_1 \\ t_2 \\ 1 \end{bmatrix} = \mathsf{H}\mathbf{t}\,. \tag{5.54}$$

The direction vectors $\mathbf{D}_i$ are points at infinity and also define the units $|\boldsymbol{D}_i|, i = 1, 2$ in which the coordinates on the plane are measured. The points on the line at infinity of the plane cannot be represented this way.

5.3.2.3 Plane in Three-Point Representation

The following explicit representation for a point lying on a plane passing through three given points includes the points on the line at infinity of the plane,

$$\mathbf{X} = t_0\mathbf{X}_0 + t_1\mathbf{X}_1 + t_2\mathbf{X}_2 = [\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2] \begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix} = \mathsf{X}\mathbf{t}\,, \tag{5.55}$$

indicating that the vector $\mathbf{X}$ is an element of the column space of the matrix X containing the homogeneous points as columns. Again, this is a mapping from the plane coordinates $\mathbf{t}$ to the space coordinates $\mathbf{X}$, see the example in next figure.

If the vector $\boldsymbol{t} = [t_0, t_1, t_2]^\mathsf{T}$ is normalized such that $t_0 + t_1 + t_2 = 1$, its elements $t_i, i = 0, 1, 2$, are called the barycentric coordinates of $\mathcal{X}$ with respect to the triangle $(\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2)$. We will discuss barycentric coordinates in more detail in Sect. 9.1.2.1, p. 349.

5.3.3 1D Point

A 1D point $\mathcal{X}$ with inhomogeneous coordinate x has homogeneous coordinates,

$$\mathcal{X} : \mathbf{x} = \begin{bmatrix} x_0 \\ x_h \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} vx \\ v \end{bmatrix}. \tag{5.56}$$

The factor $v \neq 0$ can be chosen arbitrarily.

Analogously, the coordinate $x_h = v$ is called the *homogeneous part*, specifically the homogeneous coordinate of the coordinate vector $\mathbf{x}$, while the other part, $x_0 = v$, is the inhomogeneous or *Euclidean part*. If $v \neq 0$, any 2-vector $\mathbf{x}$ can be interpreted as a point in $\mathbb{R}$ with coordinates $x = u/v$.

Points $[u, 0]^\mathsf{T}$ with homogeneous component $x_h = v = 0$ are *ideal points* or *points at infinity*, as their inhomogeneous coordinates are not finite. They represent the two points at infinity, where opposite directions are unified, as $[u, 0]^\mathsf{T} \cong [-u, 0]^\mathsf{T}$.

Definition 5.3.8: *Projective line.* The projective line $\mathbb{P} = \mathbb{P}^1$ contains all 1D points $\mathcal{X}$ with homogeneous coordinates $\mathbf{x} = [u, v]^\mathsf{T} \in \mathbb{R}^2 \setminus \mathbf{0}$,

$$\mathcal{X} \in \mathbb{P} : \quad \mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix} \in \mathbb{R}^2 \setminus \mathbf{0}, \tag{5.57}$$

with

$$\mathcal{X}(\mathbf{x}) \equiv \mathcal{Y}(\mathbf{y}) \;\Leftrightarrow\; \mathbf{x} = \lambda \mathbf{y}, \quad \text{for some } \lambda \neq 0. \tag{5.58}$$

The coordinates of identical points may differ by a factor $\lambda \neq 0$. ◇

The projective line $\mathbb{P}$ consists of all points $\mathcal{X}$ of the Euclidean line $\mathbb{R}$ and the points $\mathcal{X}_\infty([u, 0]^\mathsf{T}), u \neq 0$ at infinity.

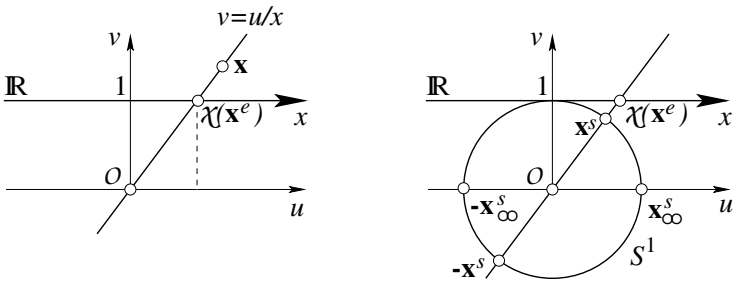


Fig. 5.13 The projective line $\mathbb{P} = \mathbb{P}^1$. **Left:** The point $\mathcal{X}$ on the real line $\mathbb{R}$ with inhomogeneous coordinate x has homogeneous coordinates $\mathbf{x} = [vx, v]^\mathsf{T}$, and if of Euclidean normalization $\mathbf{x}^e = [x, 1]^\mathsf{T}$. Due to homogeneity it represents the line $v = u/x$ through the origin O of the uv-plane. If $v = 0$, the line is horizontal, representing the point $\mathcal{X}_\infty$ at infinity. **Right:** Spherical normalization of the homogeneous coordinates of a point on the projective line leads to a point $\mathbf{x}^s$ sitting on the unit circle S^1, $-\mathbf{x}^s$ representing the same point. The point $\mathcal{X}_\infty$ at infinity has spherically normalized coordinates $[\pm 1, 0]^\mathsf{T}$

Exercise 5.4

As a 1D point $\mathcal{X}$ with homogeneous coordinates $\mathbf{x} \in \mathbb{R}^2 \setminus \mathbf{0}$ and $\lambda\mathbf{x}$ with $\lambda \neq 0$ represent the same point (see Fig. 5.13), and $\lambda\mathbf{x}$ is a 2D line passing through the origin of the uv-plane $\mathbb{R}^2$ with equation $v = u/x$, points in the projective line $\mathbb{P}$ can be identified with all lines through the origin of $\mathbb{R}^2$.

5.3.4 The Projective Space $\mathbb{P}^n$

Eventually, we will need homogeneous entities in higher-dimensional spaces. For example, 3D lines will be represented by a homogeneous 6-vector. Another example is the vector

$$\mathbf{h} = \text{vec}\mathsf{H} = \begin{bmatrix} H_{11} \\ H_{21} \\ H_{31} \\ H_{12} \\ \dots \\ H_{33} \end{bmatrix}, \tag{5.59}$$

containing the elements of 3×3 matrix H for transforming points $\mathbf{x} \in \mathbb{P}^2$ with $\mathbf{x}' = \mathsf{H}\mathbf{x}$. Since the transformation is invariant w.r.t. a scaling of the matrix H, it is a homogeneous matrix. Therefore also the vector $\mathbf{h}$ is homogeneous, namely $\mathbf{h} \in \mathbb{P}^8$.

We therefore define:

Definition 5.3.9: ***Projective space.*** The projective space $\mathbb{P}^n(\mathbb{R})$ contains all $(n+1)$-dimensional points $\mathcal{X}$ with homogeneous real-valued coordinates $\mathbf{x} \in \mathbb{R}^{n+1} \setminus \mathbf{0}$,

$$\mathcal{X}(\mathbf{x}) \in \mathbb{P}^n(\mathbb{R}) : \quad \mathbf{x} \in \mathbb{R}^{n+1} \setminus \mathbf{0}\,, \tag{5.60}$$

with

$$\mathcal{X}(\mathbf{x}) \equiv y(\mathbf{y}) \;\Leftrightarrow\; \mathbf{x} = \lambda\mathbf{y}, \quad \text{for some } \lambda \neq 0. \tag{5.61}$$

◇

We recursively define a projective space $\mathbb{P}^n$ as the union of the real space $\mathbb{R}^n$ and the elements at infinity. These elements at infinity themselves build a projective space $\mathbb{P}^{n-1}$. Hence we have

$$\mathbb{P}^n = \mathbb{R}^n \cup \mathbb{P}^{n-1} \qquad \mathbb{P}^0 : [x_1] \in \mathbb{R} \setminus 0. \tag{5.62}$$

If we want to represent all points $\mathcal{X} \in \mathbb{P}^n$ using homogeneous coordinates $\mathbf{x} \in \mathbb{R}^{n+1} \setminus \mathbf{0}$, the real points and the points at infinity need to be represented by $n+1$-vectors. Thus, we need to specify the hyperplane at infinity, e.g., by fixing the last coordinate $x_{n+1} = 0$. This embeds the $n-1$-dimensional plane at infinity whose points are directions represented by the (non-normalized) n-vectors $[x_1, ..., x_n]^\mathsf{T}$, into $\mathbb{P}^n$ by extending these vectors by one element, leading to $[x_1, ..., x_n, 0]^\mathsf{T}$. Therefore, in (5.62) the elements of $\mathbb{P}^{n-1}$ are represented with $n+1$-vectors, $[x_1, ..., x_n, 0]^\mathsf{T}$. In the next step of the recursion these elements are identified with the n-vectors $[x_1, ..., x_n]^\mathsf{T}$.

projective point $\mathbb{P}^0$

The recursion starts with the zero-dimensional projective space $\mathbb{P}^0$, which is algebraically represented by the homogeneous 1-vector $[x_1] \in \mathbb{R}\setminus 0$. In analogy to the projective plane $\mathbb{P}^2$ and the projective line $\mathbb{P} = \mathbb{P}^1$, we call it *projective point* $o \in \mathbb{P}^0$. It only contains one element, namely the origin $o([1])$. The situation can be visualized; for $n = 2$ see Fig. 5.14.

The projective space $\mathbb{P}^2$ can be represented as the unit sphere S^2, i.e., by points $\mathbf{x}^s = [x_1^s, x_2^s, x_3^s]^\mathsf{T}$ with $|\mathbf{x}^s| = 1$, cf. Sect. 5.1.2.3, p. 199. The points on the upper and the lower hemispheres represent the real plane $\mathbb{R}^2$. The points at infinity $\mathcal{X}_\infty$ with coordinates $[x_1, x_2, 0]^\mathsf{T}$ are represented by the equator, a circle S^1. Referring to Fig. 5.13, p. 214, right, we can take the unit circle as a representation of the projective line $\mathbb{P} = \mathbb{P}^1$, i.e., by points $\mathbf{x}^s = [x_1^s, x_2^s]^\mathsf{T}$ with $|\mathbf{x}^s| = 1$. Again, the left hemicircle, and the right hemicircle represent the real line $\mathbb{R} = \mathbb{R}^1$ and its point at infinity x_∞ with coordinates $[x_1, 0]^\mathsf{T}$. Going one step further, the projective point $\mathbb{P}^0$ with coordinate $[x_1] = [\pm 1]$ cannot be subdivided.

Thus we have

$$\mathbb{P}^2 = \mathbb{R}^2 \cup \mathbb{R}^1 \cup \mathbb{P}^0. \tag{5.63}$$

complex projective space

Occasionally, we will need the notion of a complex projective space $\mathbb{C}$. Then the elements of the homogeneous coordinates are complex numbers, and we might write $\mathbb{P}^n(\mathbb{C})$, making the character of the homogeneous vector $\mathbf{x} \in \mathbb{C}^{n+1}$ explicit.

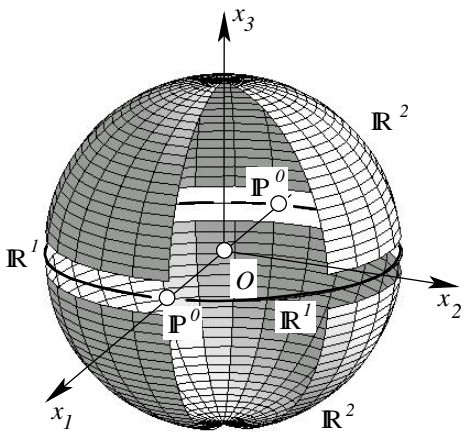


Fig. 5.14 Partitioning of the projective plane $\mathbb{P}^2 = \mathbb{R}^2 \cup \mathbb{R}^1 \cup \mathbb{P}^0$, cf. (5.63). First part: the real plane $\mathbb{R}^2$, the set of all spherically normalized point vectors $\mathbf{x}^s$ on S^2. Second part: the projective line $\mathbb{P}^1$ on the equator of S^2, representing the points at infinity, thus $\mathbb{P}^2 = \mathbb{R}^2 \cup \mathbb{P}^1$. Third part: the equator is a projective line decomposed into the real line $\mathbb{R}$ and the point at infinity, which is the projective point $\mathbb{P}^0$, hence $\mathbb{P}^1 = \mathbb{R}^1 \cup \mathbb{P}^0$. The projective point $\mathbb{P}^0$ is represented by two opposite points on the unit sphere

5.4 Homogeneous Representations of 3D Lines

5.4.1 Parametrization of the 3D Line 216
5.4.2 Homogeneous Coordinates of the 3D Line 217
5.4.3 Homogeneous Matrix for the 3D Line 218
5.4.4 The 3D Line at Infinity 219
5.4.5 3D Line by Intersection of Planes 220

5.4.1 Parametrization of the 3D Line

degrees of freedom of a 3D line

The representation of straight 3D lines is more complex than that of 2D lines. A 3D line can be represented by four independent parameters:

1. A 3D line $\mathcal{L}$ may be the intersection of two planes; e.g., the two planes $\mathcal{A}$ and $\mathcal{B}$ in Fig. 5.15, which are orthogonal to two of the coordinate planes, and $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$. Then we need only two parameters for each plane, namely the lines $\mathcal{F}$ and $\mathcal{G}$ in the XZ and the XY planes.
2. The connection or join of two points, say $\mathcal{X}$ and $\mathcal{Y}$ in Fig. 5.15, which are the intersections of the line with two coordinate planes, thus $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$. In this case also, only the two coordinates $[X, Y]$ of $\mathcal{X}$ and $[Y, Z]$ of $\mathcal{Y}$ are needed.

The four parameters obviously depend on the chosen reference points or planes.

Another common representation is given by a reference point and a normalized direction with five parameters: three for the reference point and two for the direction. Obviously, one parameter is redundant, as the reference point can be chosen arbitrarily on the line. Choosing the point closest to the origin as reference point imposes a constraint (cf., e.g., Mulawa, 1989), thus reducing the number of free parameters to four.

This representation is not homogeneous but closely related to the one we will use.

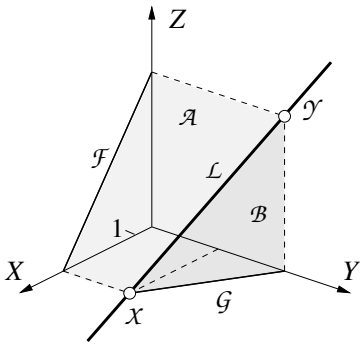


Fig. 5.15 Degrees of freedom of a 3D line. Only 4 parameters are necessary to represent a 3D line $\mathcal{L}$: for example, either the coordinates of the intersection points $\mathcal{X}$ and $\mathcal{Y}$ with two reference planes, here the XY and the YZ planes, or the parameters of the planes $\mathcal{A}$ and $\mathcal{B}$ perpendicular to two reference planes, here perpendicular to the XZ-plane and the XY plane, induced by the two lines $\mathcal{F}$ and $\mathcal{G}$. The figure shows the special case $\mathbf{X} = [4, 3, 0, 1]^\mathsf{T}$, $\mathbf{Y} = [0, 6, 7, 1]^\mathsf{T}$, $\mathbf{A} = [7, 0, 4, -28]^\mathsf{T}$, $\mathbf{B} = [6, 8, 0, -48]^\mathsf{T}$, and $\mathbf{L} = [-4, 3, 7, 21, -28, 24]^\mathsf{T}$. For the representation of the 3D line, see below

5.4.2 Homogeneous Coordinates of the 3D Line

Let a 3D line $\mathcal{L}$ be defined as the join of two 3D points $\mathcal{X}$ and $\mathcal{Y}$, namely $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ as shown in Fig. (5.16), p. 218. The point $\mathcal{Z}$ on the line is closest to the origin, whereas the coordinates of vector $\boldsymbol{L}_0$ are given by the cross product $\boldsymbol{L}_0 = \boldsymbol{X} \times \boldsymbol{Y}$. The three vectors $\boldsymbol{Y} - \boldsymbol{X}$, $\boldsymbol{L}_0$ and $\boldsymbol{Z}$ form a right-handed tripod. Two of them are sufficient to specify the 3D line.

We define the coordinates $\mathbf{L}$ of the 3D line $\mathcal{L}$ as a function of the inhomogeneous coordinates $\boldsymbol{X} = [X_1, X_2, X_3]^\mathsf{T}$ and $\boldsymbol{Y} = [Y_1, Y_2, Y_3]^\mathsf{T}$ of two points as

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{Y} - \boldsymbol{X} \\ \boldsymbol{X} \times \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} Y_1 - X_1 \\ Y_2 - X_2 \\ Y_3 - X_3 \\ X_2Y_3 - Y_2X_3 \\ X_3Y_1 - Y_3X_1 \\ X_1Y_2 - Y_1X_2 \end{bmatrix} . \tag{5.64}$$

Obviously, the first and the second subvector, $\boldsymbol{Y} - \boldsymbol{X}$ and $\boldsymbol{X} \times \boldsymbol{Y}$, respectively, are perpendicular.

The representation of a line using six elements goes back to Julius Plücker (1801-1868). Therefore, the elements of the 6-vector are called Plücker coordinates of the 3D line, and the orthogonality constraint between the two subvectors is called Plücker constraint. We define:

Definition 5.4.10: ***Plücker coordinates of a 3D line.*** A 3D line $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ joining two 3D points with homogeneous coordinates[4] $\mathbf{X} = [X_1, X_2, X_3, X_4]^\mathsf{T}$ and $\mathbf{Y} = [Y_1, Y_2, Y_3, Y_4]^\mathsf{T}$ has Plücker coordinates

$$\mathbf{L} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix} = \begin{bmatrix} X_h \boldsymbol{Y}_0 - Y_h \boldsymbol{X}_0 \\ \boldsymbol{X}_0 \times \boldsymbol{Y}_0 \end{bmatrix} = \begin{bmatrix} X_4Y_1 - Y_4X_1 \\ X_4Y_2 - Y_4X_2 \\ X_4Y_3 - Y_4X_3 \\ X_2Y_3 - Y_2X_3 \\ X_3Y_1 - Y_3X_1 \\ X_1Y_2 - Y_1X_2 \end{bmatrix} . \tag{5.65}$$

◇

[4] Not to be confused with the inhomogeneous coordinates in (5.64).

The 3-vectors $\boldsymbol{L}_h$ and $\boldsymbol{L}_0$ are the homogeneous and the Euclidean parthomogeneous vector $\mathbf{L}$. The vector $\boldsymbol{L}_0$ also is called the *moment of the 3D line*.

Plücker constraint

The two vectors fulfil the Plücker constraint,

$$\boldsymbol{L}_h^\mathsf{T}\boldsymbol{L}_0 = L_1L_4 + L_2L_5 + L_3L_6 = 0. \tag{5.66}$$

A vector $\lambda\mathbf{L}$ represents the same 3D line as the vector $\mathbf{L}$ if $\lambda \neq 0$. We just need to show that for a general $\mathbf{L}$, fulfilling the Plücker constraint, the direction and the point $\mathcal{Z}$ can be uniquely derived from $\mathbf{L}$. First, the direction can be uniquely determined from $\boldsymbol{L}_h$, namely $\mathsf{N}(\boldsymbol{L}_h) = \boldsymbol{L}_h/|\boldsymbol{L}_h|$. Second, the direction of the vector $\boldsymbol{Z}$ results from $\mathsf{N}(\boldsymbol{L}_h \times \boldsymbol{L}_0)$, independent of the scaling.

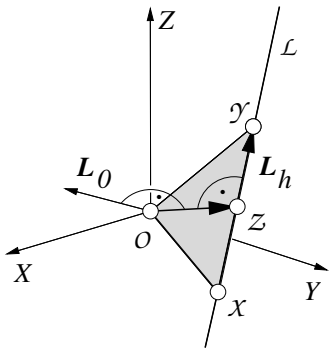


Fig. 5.16 Representation of a 3D line with Plücker coordinates. Line $\mathcal{L}$ through $\mathcal{X}$ and $\mathcal{Y}$ with direction $\boldsymbol{L}_h$, normal $\boldsymbol{L}_0$ and point $\mathcal{Z}$ closest to the origin. The vectors $\boldsymbol{L}_h$, $\boldsymbol{L}_0$ and component $\boldsymbol{Z}_h$ from $\mathbf{Z}$ build a right-handed tripod. The figure shows the special case: $\mathbf{X} = [21, 35, 0, 5]^\mathsf{T}$, $\mathbf{Y} = [3, 12, 12, 2]^\mathsf{T}$, $\mathbf{L} = [-27, -10, 60, 420, -252, 147]^\mathsf{T}$ and $\mathbf{Z}(\mathcal{L}) = [13650, 29169, 11004, 4429]^\mathsf{T}$. For determining $\mathcal{Z}$, cf. (7.144), p. 323

Finally, the distance d_{LO} of the 3D line from $\mathcal{O}$ is given by

$$d_{LO} = \frac{|\boldsymbol{L}_0|}{|\boldsymbol{L}_h|}. \tag{5.67}$$

This can easily be verified by determining 2A, the double area of the triangle $(\mathcal{XOY})$ (cf. Fig. 5.16). We have

$$2A = \left|\frac{\boldsymbol{X}_0}{X_h} \times \frac{\boldsymbol{Y}_0}{Y_h}\right| = \frac{1}{X_hY_h}\underbrace{|\boldsymbol{X}_0 \times \boldsymbol{Y}_0|}_{|\boldsymbol{L}_0|} = \left|\frac{\boldsymbol{Y}_0}{Y_h} - \frac{\boldsymbol{X}_0}{X_h}\right| d_{LO} = \frac{1}{X_hY_h}\underbrace{|X_h\boldsymbol{Y}_0 - Y_h\boldsymbol{X}_0|}_{|\boldsymbol{L}_h|} d_{LO}\,.$$

Observe, if a line passes though the origin the Euclidean part $\boldsymbol{L}_0$ is zero, as any two points and the origin are collinear, thus the cross product $\boldsymbol{L}_0 = \boldsymbol{X}_0 \times \boldsymbol{Y}_0 = \mathbf{0}$, which is consistent with (5.67).

In Sect. 5.5, p. 221 we will give a more formal and more general definition of Plücker coordinates which relies on vector space algebra and covers the homogeneous coordinates of points and planes as special cases.

5.4.3 Homogeneous Matrix for the 3D Line

The coordinates of the 3D line in (5.65) can obviously be identified as determinants built by the homogeneous coordinates of the two points.

They occur as elements in what is called the *Plücker matrix* *Plücker matrix*

$$\begin{aligned}
\Gamma(\mathbf{L}) &= [\gamma_{ij}] \qquad\qquad i,j \in \{1,2,3,4\} && (5.68)\\
&= [X_iY_j - Y_iX_j] && (5.69)\\
&= \mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T} && (5.70)\\
&= \begin{bmatrix}
0 & \begin{vmatrix} X_1 & Y_1 \\ X_2 & Y_2 \end{vmatrix} & \begin{vmatrix} X_1 & Y_1 \\ X_3 & Y_3 \end{vmatrix} & \begin{vmatrix} X_1 & Y_1 \\ X_4 & Y_4 \end{vmatrix} \\
\begin{vmatrix} X_2 & Y_2 \\ X_1 & Y_1 \end{vmatrix} & 0 & \begin{vmatrix} X_2 & Y_2 \\ X_3 & Y_3 \end{vmatrix} & \begin{vmatrix} X_2 & Y_2 \\ X_4 & Y_4 \end{vmatrix} \\
\begin{vmatrix} X_3 & Y_3 \\ X_1 & Y_1 \end{vmatrix} & \begin{vmatrix} X_3 & Y_3 \\ X_2 & Y_2 \end{vmatrix} & 0 & \begin{vmatrix} X_3 & Y_3 \\ X_4 & Y_4 \end{vmatrix} \\
\begin{vmatrix} X_4 & Y_4 \\ X_1 & Y_1 \end{vmatrix} & \begin{vmatrix} X_4 & Y_4 \\ X_2 & Y_2 \end{vmatrix} & \begin{vmatrix} X_4 & Y_4 \\ X_3 & Y_3 \end{vmatrix} & 0
\end{bmatrix} && (5.71)\\
&= \begin{bmatrix}
0 & L_6 & -L_5 & -L_1 \\
-L_6 & 0 & L_4 & -L_2 \\
L_5 & -L_4 & 0 & -L_3 \\
L_1 & L_2 & L_3 & 0
\end{bmatrix} && (5.72)\\
&= \begin{bmatrix} -\mathsf{S}_{L_0} & -\boldsymbol{L}_h \\ \boldsymbol{L}_h^\mathsf{T} & 0 \end{bmatrix}. && (5.73)
\end{aligned}$$

This matrix has a number of remarkable properties:

- It is skew symmetric, due to the skew symmetry of the determinants $X_iY_j - Y_iX_j$.
- It has rank 2, as it is the difference of two dyadic products, $\mathbf{X}\mathbf{Y}^\mathsf{T}$ and $\mathbf{Y}\mathbf{X}^\mathsf{T}$.
- It is linearly dependent on the elements of the Plücker coordinates of the line.
- Replacing the two points $\mathcal{X}$ and $\mathcal{Y}$ by two others sitting on the line, e.g.,

$$[\mathbf{X}', \mathbf{Y}'] = [\mathbf{X}, \mathbf{Y}] \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \tag{5.74}$$

  with $ad - bc \neq 0$, changes all determinants $X_iY_j - X_jY_i$ by the factor $ad - bc$ as

$$\begin{vmatrix} X_i' & Y_i' \\ X_j' & Y_j' \end{vmatrix} = \begin{vmatrix} X_i & Y_i \\ X_j & Y_j \end{vmatrix} \begin{vmatrix} a & b \\ c & d \end{vmatrix}. \tag{5.75}$$

  This proves that the homogeneous representation of a 3D line with Plücker coordinates is independent of the chosen points.

We also partition the Plücker matrix, each submatrix depending on $\boldsymbol{L}_h$ or $\boldsymbol{L}_0$. The Plücker matrix will play a central role in geometric reasoning, e.g., when intersecting a plane with a line or when joining a line and a point.

5.4.4 The 3D Line at Infinity

The distance of a line to the origin is not defined if the first 3-vector $\boldsymbol{L}_h$ vanishes; in this case we have an *ideal line* or a *line at infinity* with the general form

$$\mathcal{L}_\infty : \quad \mathbf{L} = \begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{L}_0 \end{bmatrix}. \tag{5.76}$$

It is defined by the moment vector $\boldsymbol{L}_0$ and thus can be visualized as the intersection of the plane with coordinates $[\boldsymbol{L}_0, 0]$ and the plane at infinity. For example, the line $[0,0,0,0,0,1]^\mathsf{T}$ represents the horizon, i.e., the intersection of the XY-plane having normal $[0,0,1]^\mathsf{T}$, with the plane at infinity. Exercise 5.7

Generally, we have the following relation between a plane and its line at infinity.

Corollary 5.4.5: ***Line at infinity of a plane.*** The line at infinity $\mathcal{L}_{\infty A}$ of plane $\mathcal{A} \neq \mathcal{A}_\infty$ is given by

$$\mathbf{L}_{\infty A} = \begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{A}_h \end{bmatrix}. \tag{5.77}$$

It is identical for all planes parallel to $\mathcal{A}$. ◇

Proof: We only need to analyse the geometric relation of the entities on the celestial sphere representing the plane at infinity. All points $[\boldsymbol{X}_0, 0]^\mathsf{T}$ at infinity on $\mathcal{A}$ point in a direction perpendicular to the normal of the plane, thus fulfil $\boldsymbol{X}_0 \perp \boldsymbol{A}_h$. But these are exactly those points sitting on the 3D line at infinity with moment vector $\boldsymbol{L}_0 = \boldsymbol{A}_h$. ◇

The point at infinity $\mathcal{X}_{\infty L}$ of any line $\mathcal{L}$ is given by

$$\mathbf{X}_{\infty L} = \begin{bmatrix} \boldsymbol{L}_h \\ 0 \end{bmatrix}. \tag{5.78}$$

It is identical for all lines parallel to $\mathcal{L}$.

5.4.5 3D Line by Intersection of Planes

If the line $\mathcal{L}$ is determined as the intersection of two planes $\mathcal{A}$ and $\mathcal{B}$, namely $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$, we obtain the following expression for the Plücker coordinates,

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_h \times \boldsymbol{B}_h \\ A_0 \boldsymbol{B}_h - B_0 \boldsymbol{A}_h \end{bmatrix}. \tag{5.79}$$

This is plausible, as the direction $\boldsymbol{L}_h$ of the line is perpendicular to the normals $\boldsymbol{A}_h$ and $\boldsymbol{B}_h$ of the two planes and the moment of the line $\boldsymbol{L}_0$ must lie in the plane spanned by the two normals. The full proof is left as an exercise.

Exercise 5.15

Observe the similarity of (5.79) to the expression for the join of two points in (5.65): the point coordinates are replaced by plane coordinates, and expressions for the homogeneous and the Euclidean parts of the line coordinates are exchanged. We will discuss this analogy in Sect. 5.6, p. 229 on duality.

3D line in Point-Direction Form. The classical point-direction representation for a point $\mathcal{X}$ on a 3D line $\mathcal{L}$ is given by a reference point $\boldsymbol{X}_0$ and the direction $\boldsymbol{D}$ of the line

$$\boldsymbol{X} = \boldsymbol{X}_0 + t\boldsymbol{D}, \tag{5.80}$$

where t is the line coordinate and again, as in the 2D case, does not allow us to represent the point at infinity of the line. Here, we can also express the homogeneous coordinates $\mathbf{X}$ of the point as a function of its homogeneous coordinates on the 3D line

$$\mathbf{X} = \mathbf{X}_0 + t\mathbf{D} = \underbrace{\begin{bmatrix} \boldsymbol{D} & \boldsymbol{X}_0 \\ 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} t \\ 1 \end{bmatrix} = \mathsf{H}\mathbf{t}\,, \tag{5.81}$$

where the direction is represented as a point at infinity $\mathbf{D}^\mathsf{T} = [\boldsymbol{D}^\mathsf{T}, 0]$.

3D Line in Two-Point Form. When two points $\mathcal{X}_1$ and $\mathcal{X}_2$ are given, then the point $\mathcal{X}$ with homogeneous coordinates

$$\mathbf{X} = t_1\mathbf{X}_1 + t_2\mathbf{X}_2 \tag{5.82}$$

lies on the line joining the two points. The parameters $t_i, i = 1, 2$, can be chosen such that $\mathbf{X}_1$ or $\mathbf{X}_2$ is the point at infinity of the 3D line.

The proof of (5.82) uses the point-line incidence relation; cf. Sect. 7.2.2.2, p. 306.

5.5 On Plücker Coordinates for Points, Lines and Planes

5.5.1 Plücker Coordinates for 2D Entities Derived from Points 221
5.5.2 Plücker Coordinates for 2D Entities Derived from Lines 224
5.5.3 Plücker Coordinates for 3D Points and Planes 224
5.5.4 Plücker Coordinates for 3D Lines 226

We have shown advantages of representing elements at infinity and of some constructions with homogeneous coordinates for geometric entities.

In the following two sections, we want to demonstrate that the already given representations are special cases within the general concept of Plücker coordinates relying on vector space algebra. The concept of Plücker coordinates is related in an elementary way to the minors of matrices whose column spaces are spanned by the generating elements, which are either elements in the vector space $V(\mathbb{R}^n)$, representing points, or elements in its dual vector space $V^*(\mathbb{R}^n)$, representing hyperplanes. The duality of the basic vector spaces also allows a geometric interpretation and the generalization to 3D lines, a topic discussed in the next section.

We will first motivate and formally define the concept of Plücker coordinates for 2D points and 2D lines, and then apply it to 3D geometric entities.

The introduction of line coordinates follows Busemann and Kelley (1953).

5.5.1 Plücker Coordinates for 2D Entities Derived from Points

We represented a 2D line, $\ell(\mathbf{l})$, with its homogeneous coordinates $\mathbf{l}$, closely related to the Hessian normal form. The coordinates could be determined from the join of two given points, $\ell = \chi \wedge y$ by $\mathbf{l} = \mathbf{x} \times \mathbf{y}$.

However, we also can represent all points $z(\mathbf{z})$ sitting on ℓ by an arbitrary linear combination of the two points (cf. (5.41), p. 209)

point on 2D line spanned by two points

$$\mathbf{z} = (1 - \alpha)\mathbf{x} + \alpha\mathbf{y}, \tag{5.83}$$

as obviously $\mathbf{z}^\mathsf{T}\mathbf{l} = 0$ for all α. Thus, the 3-vector $\mathbf{z}$ is in the column space of the 3×2 matrix $[\mathbf{x}, \mathbf{y}]$:

$$\mathbf{z} \in \mathsf{span}([\mathbf{x}, \mathbf{y}]). \tag{5.84}$$

Therefore, we may also represent a 2D line by two arbitrary points on the line, namely by the column space having the generating points as columns.

However, we might also try to find a representation which is independent of the chosen points. This leads to the concept of what is called *Plücker coordinates*.

The concept of Plücker coordinates is to represent a geometric entity by the invariants of its generating base elements. Let us therefore start with three points,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, \tag{5.85}$$

and analyse the determinant,

$$D_x = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 2A\,; \tag{5.86}$$

the index x in D_x indicates the determinant is derived from 2D points. If the coordinates are normalized such that the homogeneous parts x_3, y_3 and z_3 are 1 (cf. Sect. 5.1.2.2, p. 198), the determinant is twice the area of the triangle spanned by the three points. The area is positive if the order of the points is anti-clockwise; otherwise, it is negative.

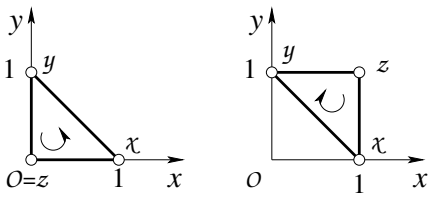


Fig. 5.17 Triangle with positive (left) and negative (right) area

As an example, take the left configuration in Fig. 5.17; here

$$\begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix} = 1. \tag{5.87}$$

We now exploit the fact that the points are collinear only if the determinant is 0. We develop the determinant according to the first column,

$$D_x = |\mathbf{x}, \mathbf{y}, \mathbf{z}| = x_1 \begin{vmatrix} y_2 & z_2 \\ y_3 & z_3 \end{vmatrix} + x_2 \begin{vmatrix} y_3 & z_3 \\ y_1 & z_1 \end{vmatrix} + x_3 \begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix}. \tag{5.88}$$

The 2×2 determinants only depend on the points y and z. We now define the 3-vector,

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} \begin{vmatrix} y_2 & z_2 \\ y_3 & z_3 \end{vmatrix}, \begin{vmatrix} y_3 & z_3 \\ y_1 & z_1 \end{vmatrix}, \begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix} \end{bmatrix}^{\mathsf{T}}. \tag{5.89}$$

Thus the determinant can be written as

$$D_x = \mathbf{x}^{\mathsf{T}}\mathbf{l}. \tag{5.90}$$

If the three points are collinear, the determinant is 0, and we obtain a linear constraint on the point coordinates, saying that the point $\mathcal{X}$ must lie on the line through y and z which can be represented by the line coordinates $\mathbf{l}$. These line coordinates are the minors of the matrix $[\mathbf{y}, \mathbf{z}]$. When collecting the minors in a vector, we are free in choosing a sequence. So for expressing the incidence relation between a point and a line, we choose a sequence which leads to a simple expression, namely the dot product.

This is the motivation to define Plücker coordinates of geometric entities in the following manner:

Definition 5.5.11: ***Plücker coordinates of a geometric entity.*** The Plücker coordinates of a geometric entity

$$\mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_k = \bigwedge_{i=1}^{k} \mathcal{X}_i \tag{5.91}$$

spanned by $k \leq n+1$ points $\mathcal{X}_i$ with linearly independent vectors $\mathbf{X}_i \in \mathbb{R}^{n+1}$ as homogeneous coordinates are given by the $\binom{n+1}{k}$ possible determinants of size $k \times k$ taken from the $(n+1) \times k$ matrix

$$[\mathbf{X}_1, ..., \mathbf{X}_k]. \tag{5.92}$$

◇

Thus we observe:

- With $n = 2$ and $k = 1$, we obtain the Plücker coordinates of a 2D point $\mathcal{X}$. They are the three determinants of the 1×1 minors of the vector $\mathbf{x}$, thus identical to its homogeneous coordinates $\mathbf{x}$.
- With $n = 2$ and $k = 2$, we obtain the Plücker coordinates of a 2D line $\mathcal{l}$ through two points y and z, identically to (5.89)
- As mentioned above, we overload the join operator "$\wedge$" and express the relation between a 2D line and its representation as

  geometric operation join: $\mathcal{X} \wedge y$ *algebraic operation:* $\mathbf{x} \wedge \mathbf{y} = \mathbf{x} \times \mathbf{y}$

  $$\mathcal{l} = \mathcal{X} \wedge y : \quad \mathbf{l} = \mathbf{x} \wedge \mathbf{y}. \tag{5.93}$$

  The expression on the left refers to the geometric entities, the expression on the right to its algebraic representation. This overloading will be of advantage when transferring the relations to 3D. Observe, when using the operator $\wedge$ for two vectors, we refer to the algebraic operation, independently of what the vectors represent.
  From the rules of determinants, we can then directly derive the effect of choosing a different order of the points: Reversing the order of the points changes the sign,

  $$\mathbf{l} = -\mathbf{y} \wedge \mathbf{x}. \tag{5.94}$$

  If we distinguish between the two direction of a line, we can explore the sign rules of the determinants. This will be exploited in Sect. 9, p. 343 (Oriented Projective Geometry). Thus the operator $\wedge$, when used for the join of vectors, is an algebraic operator operating on vectors of the same length and yields the determinants of the minors of the corresponding matrix consisting of the given vectors, taking into account the correct sequence of the vectors and the convention for sorting the minors. *algebraic operator* $\wedge$
- Any point $t(\mathbf{t}) \in \mathbb{P}^2$ can be written as a linear combination $\mathbf{t} = \alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z}$ of three points, if the three points are linearly independent. The complete plane $\mathcal{U} = \mathcal{X} \wedge y \wedge z$, $\mathcal{U}$ standing for universe, is therefore represented by the determinant of three points in general position,

  $$\mathcal{U} = \mathcal{X} \wedge y \wedge z : \quad D_x = \mathbf{x} \wedge \mathbf{y} \wedge \mathbf{z} = |\mathbf{x}, \mathbf{y}, \mathbf{z}|, \tag{5.95}$$

  as only one minor of size 3×3 exists. Since the representation is homogeneous, the geometric entity $\mathcal{U}$ can be represented algebraically by the scalar 1.
- We can collect the determinants l_i, $i = 1, ..., 3$, into $\mathbf{l}$ in any order and with any sign convention. Therefore, several different alternatives for representing geometric entities exist. For example, a common representation is $[l_0, l_1, l_2]^\mathsf{T}$, where l_0 is the Euclidean part and $[l_1, l_2]^\mathsf{T}$ is the homogeneous part, which may be easily generalized to higher dimensions.
  The choice of the sequence and the sign may be taken into account by a weight matrix or metric W when expressing the incidence. We then need to replace $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$ with $\mathbf{x}^\mathsf{T}W\mathbf{l} = 0$, thus taking a weighted inner product $\langle \cdot, \cdot \rangle_W$. This is the reason why we may write

  $$\langle \mathbf{x}, \mathbf{l} \rangle_W = 0 \tag{5.96}$$

for expressing the incidence,[5] and in the case of $W = I$ or in the case of known context, the index W may be omitted, thus $\langle \mathbf{x}, \mathbf{l} \rangle = 0$.

5.5.2 Plücker Coordinates for 2D Entities Derived from Lines

Instead of constructing geometric entities by joining points, we can also construct them by intersecting hyperplanes. The reasoning is totally analogous, and is now given for 2D points, which are defined as the intersection of 2D lines.

First, we find that a 2D line l only passes through the intersection point of two 2D lines m and n if

$$\mathbf{l} = (1 - \alpha)\mathbf{m} + \alpha\mathbf{n}, \tag{5.97}$$

or if

$$\mathbf{l} \in \mathsf{span}([\mathbf{m}, \mathbf{n}]). \tag{5.98}$$

pencil of lines

Equation (5.97) can be taken as the representation of a pencil of lines spanned by the lines m and n. Again, we want to represent the column space of the 3×2 matrix $[\mathbf{m}, \mathbf{n}]$ in a more compact manner, which will lead to the Plücker coordinates of the intersection point.

We again start with three 2D lines, l, m, and n, in general position and develop the determinant of their homogeneous coordinates w.r.t. the first column, obtaining the expansion

$$D_l = |\mathbf{l}, \mathbf{m}, \mathbf{n}| = l_1 \underbrace{\begin{vmatrix} m_2 & n_2 \\ m_3 & n_3 \end{vmatrix}}_{x_1} + l_2 \underbrace{\begin{vmatrix} m_3 & n_3 \\ m_1 & n_1 \end{vmatrix}}_{x_2} + l_3 \underbrace{\begin{vmatrix} m_1 & n_1 \\ m_2 & n_2 \end{vmatrix}}_{x_3} = \mathbf{l}^\mathsf{T}\mathbf{x}. \tag{5.99}$$

The determinant is only 0 if the three lines meet at one point, i.e., if they are concurrent. We can collect the 2×2 determinants which depend only on the two line coordinates $\mathbf{m}$ and $\mathbf{n}$ in the 3-vector $\mathbf{x}$, observing the coordinates to be $\mathbf{m} \times \mathbf{n}$ and requiring the determinant to vanish. Then we obtain a linear constraint $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$ on the line coordinates, i.e., the line has to pass through a point x with homogeneous coordinates $\mathbf{x}$.

Overloading the $\cap$-operator, we can write the intersection and its algebraic representation as

$$\mathit{x} = \mathit{m} \cap \mathit{n} : \qquad \mathbf{x} = \mathbf{m} \cap \mathbf{n}. \tag{5.100}$$

The determinant D_l algebraically represents the single element

$$\mathit{o} = \mathit{l} \cap \mathit{m} \cap \mathit{n} : \quad D_l = \mathbf{l} \cap \mathbf{m} \cap \mathbf{n} = |\mathbf{l}, \mathbf{m}, \mathbf{n}| \tag{5.101}$$

of the projective point $\mathbb{P}^0$.

If we start with 2D lines to generate geometric entities from the intersection of 2D lines, or from hyperplanes generally, we arrive at the same definition of the Plücker coordinates: The Plücker coordinates of a geometric entity, generated as the intersection $\mathcal{A}_1 \cap ... \cap \mathcal{A}_k$ of k hyperplanes $\mathcal{A}$ with homogeneous coordinates $\mathbf{A}_i$, are all $k \times k$ minors of the matrix $[\mathbf{A}_1, ..., \mathbf{A}_k]$.

5.5.3 Plücker Coordinates for 3D Points and Planes

The concept of the previous chapter directly transfers to 3D: 3D lines and planes may similarly be represented as column spaces of two and three 4-vectors of 3D points, and

[5] For example, let $\mathbf{x}^\mathsf{T} = [\boldsymbol{x}_0^\mathsf{T}, x_h]$ and $\mathbf{l}^\mathsf{T} = [l_0, \boldsymbol{l}_h^\mathsf{T}]$; then with $W = \begin{bmatrix} \mathbf{0}_{2\times1} & I_2 \\ 1 & \mathbf{0}_{1\times2} \end{bmatrix}$ we have the incidence constraint $\mathbf{x}^\mathsf{T} W \mathbf{l} = 0$.

in the extreme case of only one vector column, we arrive at the Plücker coordinates of a point.

We start from four points, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, and $\mathcal{T}$, and develop the 4×4 determinant

$$D_X = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}] = \begin{vmatrix} X_1 & Y_1 & Z_1 & T_1 \\ X_2 & Y_2 & Z_2 & T_2 \\ X_3 & Y_3 & Z_3 & T_3 \\ X_4 & Y_4 & Z_4 & T_4 \end{vmatrix} \tag{5.102}$$

w.r.t. the first column. With the Plücker coordinates $\mathbf{A}$ of the plane $\mathcal{A} = \mathcal{Y} \wedge \mathcal{Z} \wedge \mathcal{T}$,

$$\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} \begin{vmatrix} Y_2 & Z_2 & T_2 \\ Y_3 & Z_3 & T_3 \\ Y_4 & Z_4 & T_4 \end{vmatrix}, - \begin{vmatrix} Y_1 & Z_1 & T_1 \\ Y_3 & Z_3 & T_3 \\ Y_4 & Z_4 & T_4 \end{vmatrix}, \begin{vmatrix} Y_1 & Z_1 & T_1 \\ Y_2 & Z_2 & T_2 \\ Y_4 & Z_4 & T_4 \end{vmatrix}, - \begin{vmatrix} Y_1 & Z_1 & T_1 \\ Y_2 & Z_2 & T_2 \\ Y_3 & Z_3 & T_3 \end{vmatrix} \end{bmatrix}^\mathsf{T},$$

which are the minors of the right 4×3 submatrix $[\mathbf{Y}, \mathbf{Z}, \mathbf{T}]$, we can write the determinant D_X as

$$D_X = \mathbf{X}^\mathsf{T} \mathbf{A}. \tag{5.103}$$

If the coordinates are normalized such that the homogeneous parts X_4 to T_4 are 1 (cf. Sect. 5.1.2.2, p. 198), then the determinant is six times the volume of the tetrahedron spanned by the four points. The volume is positive if $\mathcal{X}$ is on the opposite side of the normal of the triangle $(\mathcal{Y}\,\mathcal{Z}\,\mathcal{T})$ when defining the sign of the normal with the right-hand rule following the three points $(\mathcal{Y}\,\mathcal{Z}\,\mathcal{T})$ – cf. Fig. 5.18 – e.g., calculated by $\boldsymbol{N} = (\boldsymbol{Y} - \boldsymbol{T}) \times (\boldsymbol{Z} - \boldsymbol{T})$. For

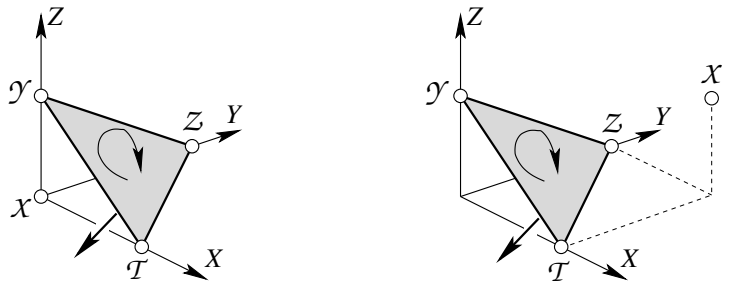


Fig. 5.18 The sign of the volume of a tetrahedron depends onlie on the same side as the normal of the last three points $(\mathcal{Y}, \mathcal{Z}, \mathcal{T})$ using the right-hand rule for defining the sign of the normal. **Left:** volume of $(\mathcal{X}\,\mathcal{Y}\,\mathcal{Z}\,\mathcal{T})$ is positive. **Right:** volume is negative

example, take the left configuration; here

$$\begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 1 > 0$$

and the origin $\mathcal{O} = \mathcal{X}$ is on the same side of the triangle as the normal. If the determinant D_X is 0, the points are coplanar. Then the point $\mathcal{X}$ must lie on the plane $\mathcal{A} = \mathcal{Y} \wedge \mathcal{Z} \wedge \mathcal{T}$ spanned by the other three points with the plane coordinates $\mathbf{A}$ from (5.103), thus $0 = \mathbf{X}^\mathsf{T}\mathbf{A}$.

plane from three points

We therefore have the following results:

- The Plücker coordinates $\mathbf{X}$ of a 3D point $\mathcal{X}$ are identical to its homogeneous coordinates.
- The Plücker coordinates $\mathbf{A}$ of a plane $\mathcal{A}$ through three points are given by (5.103).

- From the rules of determinants. we find: If the points are exchanged cyclically, the 4-vector $\mathbf{A}$ does not change its sign.

$$\mathbf{A} = \mathbf{Y} \wedge \mathbf{Z} \wedge \mathbf{T} = \mathbf{Z} \wedge \mathbf{T} \wedge \mathbf{Y} = \mathbf{T} \wedge \mathbf{Y} \wedge \mathbf{Z}. \tag{5.104}$$

  Following the right-hand rule, the normal of the plane does not change its direction. If we exchange two of the points, the sign of $\mathbf{A}$ changes.

$$\mathbf{A} = -\mathbf{Y} \wedge \mathbf{T} \wedge \mathbf{Z} = -\mathbf{Z} \wedge \mathbf{Y} \wedge \mathbf{T} = -\mathbf{T} \wedge \mathbf{Z} \wedge \mathbf{Y} \tag{5.105}$$

Exercise 5.5

  with the direction of the normal.
- The complete determinant represents the complete 3D space $\mathcal{U}$.

$$\mathcal{U}: \quad D_X = \mathbf{X} \wedge \mathbf{Y} \wedge \mathbf{Z} \wedge \mathbf{T}. \tag{5.106}$$

5.5.4 Plücker Coordinates for 3D Lines

Whereas homogeneous coordinates for points and hyperplanes can easily be given geometrically, homogeneous coordinates for 3D lines can evolve naturally from an explicit reference to the more general concept of Plücker coordinates. We will derive the Plücker coordinates of 3D lines both from the join of point pairs and from the intersection of planes.

5.5.4.1 Plücker Coordinates Derived from Points

We again start with four points in arbitrary positions and develop the determinant D_X, but now w.r.t. the first two columns (cf. Browne, 2009, Sect. 2.8):

$$D_X = |\mathbf{X}, \mathbf{Y}; \mathbf{Z}, \mathbf{T}| \tag{5.107}$$

$$\begin{aligned}
&- \begin{vmatrix} X_1 & Y_1 \\ X_2 & Y_2 \end{vmatrix} \begin{vmatrix} Z_4 & T_4 \\ Z_3 & T_3 \end{vmatrix} \\
&- \begin{vmatrix} X_2 & Y_2 \\ X_3 & Y_3 \end{vmatrix} \begin{vmatrix} Z_4 & T_4 \\ Z_1 & T_1 \end{vmatrix} \\
&- \begin{vmatrix} X_3 & Y_3 \\ X_1 & Y_1 \end{vmatrix} \begin{vmatrix} Z_4 & T_4 \\ Z_2 & T_2 \end{vmatrix} \\
&- \begin{vmatrix} X_4 & Y_4 \\ X_1 & Y_1 \end{vmatrix} \begin{vmatrix} Z_2 & T_2 \\ Z_3 & T_3 \end{vmatrix} \\
&- \begin{vmatrix} X_4 & Y_4 \\ X_2 & Y_2 \end{vmatrix} \begin{vmatrix} Z_3 & T_3 \\ Z_1 & T_1 \end{vmatrix} \\
&- \begin{vmatrix} X_4 & Y_4 \\ X_3 & Y_3 \end{vmatrix} \begin{vmatrix} Z_1 & T_1 \\ Z_2 & T_2 \end{vmatrix}.
\end{aligned} \tag{5.108}$$

This is obviously an expression depending on six determinants for each of the point pairs. The factors are the Plücker coordinates for representing the two 3D lines $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ and $\mathcal{M} = \mathcal{Z} \wedge \mathcal{T}$, respectively, namely the determinants of all six 2×2 minors of the two matrices (see Fig. 5.19)

$$[\mathbf{X}, \mathbf{Y}] \qquad \text{and} \qquad [\mathbf{Z}, \mathbf{T}]. \tag{5.109}$$

Referring to the first line, $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$, the 2×2 determinants in (5.108)

$$g_{ij} = X_i Y_j - Y_i X_j \tag{5.110}$$

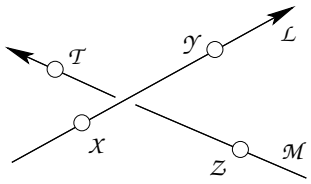


Fig. 5.19 Two 3D lines $\mathcal{L}$ and $\mathcal{M}$ through two pairs of points $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}, \mathcal{T}$. The direction of a line is determined from the sequence of the given points or from the homogeneous part $\boldsymbol{L}_h$ of the line coordinates $\mathbf{L}$, cf. (7.38), p. 301

are elements of the Plücker matrix

$$\Gamma_L = [g_{ij}] = \mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T}\,, \tag{5.111}$$

as already seen above, cf. Sect. 5.4.3, p. 218.

There are numerous ways to collect the six elements in a Plücker vector for the 3D line. We choose the following sequence (Pottmann and Wallner, 2010):

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix} = [g_{41}, g_{42}, g_{43}, g_{23}, g_{31}, g_{12}]^\mathsf{T}. \tag{5.112}$$

The first 3-vector, the homogeneous part $\boldsymbol{L}_h$, gives the direction of the line; the second 3-vector, the Euclidean part $\boldsymbol{L}_0$, the moment of the line. There are several other ways to select and sort the entries of the Plücker matrix $\Gamma_L = \{g_{ij}\}$ which are explored in an exercise. Exercise 5.6

As we have seen, not all 6-vectors represent 3D lines. Only 6-vectors satisfying what is called the *Plücker-constraint*,

$$\boldsymbol{L}_h^\mathsf{T}\boldsymbol{L}_0 = L_1L_4 + L_2L_5 + L_3L_6 = 0\,, \tag{5.113}$$

represent 3D lines, as the direction of the line has to be perpendicular to the normal of the plane through the line and the origin. Therefore, as the 6-vector $\mathbf{L}$ is homogeneous and has to fulfil the Plücker constraint, it has only four degrees of freedom. With the corresponding Plücker coordinates of the line $\mathcal{M} = \mathcal{Z} \wedge \mathcal{T}$,

$$\mathbf{M} = \mathbf{Z} \wedge \mathbf{T} = \begin{bmatrix} Z_4T_1 - Z_1T_4 \\ Z_4T_2 - Z_2T_4 \\ Z_4T_3 - Z_3T_4 \\ \hline Z_2T_3 - Z_3T_2 \\ Z_3T_1 - Z_1T_3 \\ Z_1T_2 - Z_2T_1 \end{bmatrix}, \tag{5.114}$$

we can express the determinant D_X as

$$D_X = -(L_1M_4 + L_2M_5 + L_3M_6 + L_4M_1 + L_5M_2 + L_6M_3) = -\langle \mathbf{L}, \mathbf{M} \rangle_D = -\mathbf{L}^\mathsf{T} D \mathbf{M}\,, \tag{5.115}$$

with the *dualizing matrix* *dualizing matrix*

$$D = \begin{bmatrix} \mathbf{0} & I_3 \\ I_3 & \mathbf{0} \end{bmatrix} \tag{5.116}$$

used as weight matrix in the inner product.

Thus we have the identities in representing the determinant D_X,

$$D_X = |\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}| = \mathbf{X} \wedge \mathbf{Y} \wedge \mathbf{Z} \wedge \mathbf{T} = \langle \mathbf{X}, \mathbf{Y} \wedge \mathbf{Z} \wedge \mathbf{T} \rangle = -\langle \mathbf{X} \wedge \mathbf{Y}, \mathbf{Z} \wedge \mathbf{T} \rangle\,, \tag{5.117}$$

omitting the indices of the inner products, as they are clear from the context. If the determinant is 0, the elements involved are coplanar:

- four coplanar 3D points $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}$,
- a 3D point $\mathcal{X}$ on a plane $\mathcal{A}(\mathcal{Y}, \mathcal{Z}, \mathcal{T})$, or
- two coplanar 3D lines $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ and $\mathcal{M}(\mathcal{Z}, \mathcal{T})$.

5.5.4.2 Plücker Coordinates Derived from Planes

We might also derive the line coordinates starting from planes, instead of from points. This will lead to completely equivalent expressions. We show that the result does not provide new information about the structure of the line coordinates. However, the result will help to motivate the concept of duality.

We therefore expand the determinant based on the coordinates of four planes, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$,

$$D_A = |\mathbf{A}, \mathbf{B}; \mathbf{C}, \mathbf{D}| = \tag{5.118}$$

$$- \begin{vmatrix} A_2 & B_2 \\ A_3 & B_3 \end{vmatrix} \begin{vmatrix} C_4 & D_4 \\ C_1 & D_1 \end{vmatrix} - \begin{vmatrix} A_3 & B_3 \\ A_1 & B_1 \end{vmatrix} \begin{vmatrix} C_4 & D_4 \\ C_2 & D_2 \end{vmatrix} \tag{5.119}$$

$$- \begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix} \begin{vmatrix} C_4 & D_4 \\ C_3 & D_3 \end{vmatrix} - \begin{vmatrix} A_4 & B_4 \\ A_1 & B_1 \end{vmatrix} \begin{vmatrix} C_2 & D_2 \\ C_3 & D_3 \end{vmatrix} \tag{5.120}$$

$$- \begin{vmatrix} A_4 & B_4 \\ A_2 & B_2 \end{vmatrix} \begin{vmatrix} C_3 & D_3 \\ C_1 & D_1 \end{vmatrix} - \begin{vmatrix} A_4 & B_4 \\ A_3 & B_3 \end{vmatrix} \begin{vmatrix} C_1 & D_1 \\ C_2 & D_2 \end{vmatrix} . \tag{5.121}$$

The determinants can be seen to represent the lines $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$ and $\mathcal{M} = \mathcal{C} \cap \mathcal{D}$. The 2×2 determinants are from the matrix

$$\mathsf{G}_L = [G_{ij}] = [A_i B_j - A_j B_i] = \mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T}. \tag{5.122}$$

These are six distinct values, which can be collected in the vector (cf. (5.79), p. 220)

$$\begin{bmatrix} G_{23} \\ G_{31} \\ \hline G_{12} \\ G_{41} \\ G_{42} \\ G_{43} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_h \times \mathbf{B}_h \\ A_h \boldsymbol{B}_0 - B_h \boldsymbol{A}_0 \end{bmatrix} \tag{5.123}$$

analogously to the vector $\mathbf{L}$ derived from points.

We now compare the two approaches, the one with points and the one with planes. We assume the join $\mathcal{l} = \mathcal{X} \wedge \mathcal{Y}$ of two points $\mathcal{X}$ and $\mathcal{Y}$ represents the same 3D line as the intersection $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$. Thus we can choose between these two representations. This implicitly enforces the following four incidence relations:

$$\mathcal{X} \in \mathcal{A} \quad \mathcal{Y} \in \mathcal{A} \quad \mathcal{X} \in \mathcal{B} \quad \mathcal{Y} \in \mathcal{B}\,. \tag{5.124}$$

Interestingly, we find the following relation:

$$\begin{bmatrix} G_{23} \\ G_{31} \\ \hline G_{12} \\ G_{41} \\ G_{42} \\ G_{43} \end{bmatrix} = \lambda \begin{bmatrix} g_{41} \\ g_{42} \\ \hline g_{43} \\ g_{23} \\ g_{31} \\ g_{12} \end{bmatrix} ; \tag{5.125}$$

thus the two vectors containing the determinants G_{ij} and g_{ij} are proportional after exchanging the first and the last three elements.

This can be derived from the four incidence relations (5.124)

$$0 = \mathbf{A}^\mathsf{T}\mathbf{X} = A_1X_1 + A_2X_2 + A_3X_3 + A_4X_4 \tag{5.126}$$
$$0 = \mathbf{A}^\mathsf{T}\mathbf{Y} = A_1Y_1 + A_2Y_2 + A_3Y_3 + A_4Y_4 \tag{5.127}$$
$$0 = \mathbf{B}^\mathsf{T}\mathbf{X} = B_1X_1 + B_2X_2 + B_3X_3 + B_4X_4 \tag{5.128}$$
$$0 = \mathbf{B}^\mathsf{T}\mathbf{Y} = B_1Y_1 + B_2Y_2 + B_3Y_3 + B_4Y_4 \tag{5.129}$$

by collecting any two indices, say $[i, j]$, and the remaining indices $[k, l]$. We obtain relations expressed by determinants, namely the elements g_{ij}, G_{ij}, g_{lk}, and G_{lk}. First, we have

$$\begin{bmatrix} A_i & A_j \\ B_i & B_j \end{bmatrix} \begin{bmatrix} X_i & Y_i \\ X_j & Y_j \end{bmatrix} + \begin{bmatrix} A_k & A_l \\ B_k & B_l \end{bmatrix} \begin{bmatrix} X_k & Y_k \\ X_l & Y_l \end{bmatrix} = \mathbf{0}$$

or

$$\begin{bmatrix} A_i & A_j \\ B_i & B_j \end{bmatrix} \begin{bmatrix} X_i & Y_i \\ X_j & Y_j \end{bmatrix} = - \begin{bmatrix} A_k & A_l \\ B_k & B_l \end{bmatrix} \begin{bmatrix} X_k & Y_k \\ X_l & Y_l \end{bmatrix}.$$

We now take determinants on both sides, use $|\mathsf{A}| = (-1)^d|-\mathsf{A}|$ for any $d \times d$ matrix A, or $|\mathsf{A}| = |-\mathsf{A}|$ for any 2×2 matrix, and obtain $G_{ij}g_{ij} = G_{kl}g_{kl} = G_{lk}g_{lk}$ or $G_{ij}/g_{lk} = G_{lk}/g_{ij}$. Therefore the representations $[g_{ij}]$ and $[G_{ij}]$ for a 3D line are equivalent.

In the following, we choose the representation with g_{ij} based on the points; thus, we refer to the Plücker coordinates in (5.65), p. 217 and the Plücker matrix in (5.68), p. 219.

5.6 The Principle of Duality

5.6.1 Dual Geometric Entities 229
5.6.2 The Dual 3D Line 233
5.6.3 Geometric Relations Between Dual Entities 233
5.6.4 Relation to Grassmann–Cayley and Geometric Algebra 234

The relationships established so far show interesting analogies. For example, the expressions for the intersection of two straight lines, $\mathcal{X} = \mathcal{l} \cap \mathcal{m} : \mathbf{x} = \mathbf{l} \times \mathbf{m}$, and the join of two points, $\mathcal{l} = \mathcal{X} \wedge \mathcal{y} : \mathbf{l} = \mathbf{x} \times \mathbf{y}$, are totally equivalent. The same holds for the incidence relations of a 2D point and a 2D line, $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$, and for a 3D point and a plane, $\mathbf{X}^\mathsf{T}\mathbf{A} = 0$. We also found a similar analogy for the degrees of freedom of a 3D line, once generated as the join of two points and once generated as the intersection of two planes.

5.6.1 Dual Geometric Entities

This analogy is caused by the way we constructed these elements, and it results from the duality of the underlying vector spaces; thus, it is not accidental. The *principle of duality* in the context of geometric reasoning leads to the following statements:

1. For each geometric entity, say $\mathcal{g}$, there exists a dual entity $\overline{\mathcal{g}}$. Vice versa, $\mathcal{g}$ is the dual entity of $\overline{\mathcal{g}}$, thus
$$\overline{\overline{\mathcal{g}}} = \mathcal{g}\,. \tag{5.130}$$
2. Each geometric relation r corresponds to a dual relation $\overline{r}$ as given in Table 5.3.
3. Each true statement corresponds to a dual statement, which again is true if each geometric entity and relation is replaced by its dual.

Table 5.3 Duality relations. $\overline{\mathbf{g}}$ is the dual of $\mathbf{g}$, the symbol $\overline{\wedge}$ is the dual of $\wedge$

entity/operation	dual entity/dual operation	relation
2D point $\mathcal{X}(\mathbf{g})$	2D line $\mathcal{l}(\overline{\mathbf{g}})$	$\overline{\mathbf{g}} = \mathbf{g}$
3D point $\mathcal{X}(\mathbf{G})$	plane $\mathcal{A}(\overline{\mathbf{G}})$	$\overline{\mathbf{G}} = \mathbf{G}$
3D line $\mathcal{L}([\boldsymbol{f}^\mathsf{T}, \boldsymbol{g}^\mathsf{T}]^\mathsf{T})$	3D line $\mathcal{M}([\boldsymbol{g}^\mathsf{T}, \boldsymbol{f}^\mathsf{T}]^\mathsf{T})$	$\overline{[\boldsymbol{f}^\mathsf{T}, \boldsymbol{g}^\mathsf{T}]^\mathsf{T}} = [\boldsymbol{g}^\mathsf{T}, \boldsymbol{f}^\mathsf{T}]^\mathsf{T}$
join $\wedge$	intersection $\cap$	$\overline{\wedge} = \cap$
incidence ι	incidence ι	$\bar{\iota} = \iota$

Example 5.6.16: ***Dualizing a statement.***
Statement "A": Three 2D points $\mathcal{x}$, y and z are collinear, i.e., z is incident to the line $\mathcal{x} \wedge y$, thus $\iota(z, \mathcal{x} \wedge y)$, if the determinant $|\mathbf{x}, \mathbf{y}, \mathbf{z}| = 0$. Statement "A" dualized: Three 2D lines $\mathcal{l}$, $\mathcal{m}$ and $\mathcal{n}$ intersect in a point, i.e., $\mathcal{n}$ is incident with the intersection of $\mathcal{l} \cap \mathcal{m}$, i.e., $\iota(\mathcal{n}, \mathcal{l} \cap \mathcal{m})$, if the determinant $|\mathbf{l}, \mathbf{m}, \mathbf{n}| = 0$. ◇

dualizing or Hodge operator

The dual entity $\overline{\mathcal{g}}(\overline{\mathbf{g}})$ can be determined from $\mathbf{g}$ via the linear dualization operator $\overline{(\bullet)}$, sometimes called the *Hodge operator*,

$$\mathbf{l} = \overline{\mathbf{x}} = I_3\mathbf{x} \qquad \mathbf{A} = \overline{\mathbf{X}} = I_4\mathbf{X} \qquad \mathbf{M} = \overline{\mathbf{L}} = D\mathbf{L} \qquad \text{with} \quad D = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}. \tag{5.131}$$

Obviously, a 2D point and the corresponding dual 2D line have the same homogeneous coordinates. Also, a 3D point and the corresponding dual plane have the same homogeneous coordinates. 3D lines have 3D lines as duals, however, with different homogeneous coordinates, as we will see below. Therefore we have $\overline{\mathcal{l}} \neq \mathcal{l}$, as the dual of a 2D line is a 2D point, but $\overline{\mathbf{l}} = \mathbf{l}$ as the coordinates of a line and its dual are the same.

$\overline{\mathcal{l}} \neq \mathcal{l}$ *but* $\overline{\mathbf{l}} = \mathbf{l}$

We have collected the pairs of dual elements in the plane and in space in Tables 5.4 and 5.7.

The tables need some explanation, which in an intricately way depends on the dimensions of the entities involved. We have to distinguish between three notions of dimension:

1. The dimension d of the manifolds involved. The dimensions of the manifolds point, line and plane are 0, 1, and 2, respectively. For completeness, we also need a representation of the empty space, which can formally be interpreted as a manifold of dimension -1.
2. The dimension of the space the geometric entity is embedded in. Here we only discuss geometric entities in the two-dimensional projective plane $\mathbb{P}^2$ and the three-dimensional projective space $\mathbb{P}^3$ given in separate tables.
3. The dimension n of the projective space of the representing homogeneous vector. Here we have dimensions between 0, for the empty set $\emptyset$ represented by the projective point $o(1) \in \mathbb{P}^0$, and 5, for representing 3D-lines $\mathbf{L} \in \mathbb{P}^5$.

The dimension of the manifold should not be confused with the dimension of the projective space for representing an entity. For example, both 3D points and planes are represented as elements in a three-dimensional projective space $\mathbb{P}^3$, whereas the entity point is a zero-dimensional and the entity plane a two-dimensional manifold.[6]

Now we find: the dimensions d of the manifold $\mathcal{g}$ and the dual $\overline{\mathcal{g}}$ sum up to 1 less than the dimension of the projective space they are embedded in. This is shown in the rows of Tables 5.4 to 5.7. For completeness, we need to include a manifold of dimension -1, as mentioned above.

[6] There exist 2D points $\mathcal{x}$ and 2D lines $\mathcal{l}$ in $\mathbb{P}^2$, which are represented as points $\mathbf{x}$ in $\mathbb{P}^2$ and points $\mathbf{l}$ in $\mathbb{P}^{*2}$. We therefore could also think of 2D points $\mathcal{x}^*$ and 2D lines $\mathcal{l}^*$ in $\mathbb{P}^{*2}$ which then are represented as $\mathbf{x}^* = \mathbf{l}$ in $\mathbb{P}^2$ and as $\mathbf{l}^* = \mathbf{x}$ in $\mathbb{P}^2$. We always work with 2D points $\mathcal{x}$ and lines $\mathcal{l}$ in $\mathbb{P}^2$. Therefore we treat the dual $\mathcal{x}^*$ of a 2D point $\mathcal{x}$ as the line $\mathcal{l} = \mathcal{x}^*$; both $\mathcal{x}$ and $\mathcal{l}$ are geometric elements in $\mathbb{P}^2$. Thus dualizing can be seen as a geometric transformation which maps points to lines or vice versa, namely a projective correlation and its dual, cf. Sect. 6.6, p. 282.

Table 5.4 Dual entities in $\mathbb{P}^2$. Entities in one row are dual pairs. For each we give three values: (1) their dimensions d, (2) the name of the spatial entity g and its dual $\overline{g}$, (3) the projective space of their representations. The empty space is denoted by $\emptyset$, the complete 2D space is denoted by $\mathcal{U}$ (universe). The last column gives the dualizing or Hodge operator

		entity g	dual entity $\overline{g}$	Hodge Operator
1	d	-1	2	
	name	$\emptyset$	$\mathcal{U}$	
	represent. space	$\mathbb{P}^0$	$\mathbb{P}^{*0}$	1
2	d	0	1	
	name	$\mathcal{X}$	$\mathcal{l}$	
	represent. space	$\mathbb{P}^2$	$\mathbb{P}^{*2}$	I_3
3	d	1	0	
	name	$\mathcal{l}$	$\mathcal{X}$	
	represent. space	$\mathbb{P}^{*2}$	$\mathbb{P}^2$	I_3
4	d	2	-1	
	name	$\mathcal{U}$	$\emptyset$	
	represent. space	$\mathbb{P}^{*0}$	$\mathbb{P}^0$	1

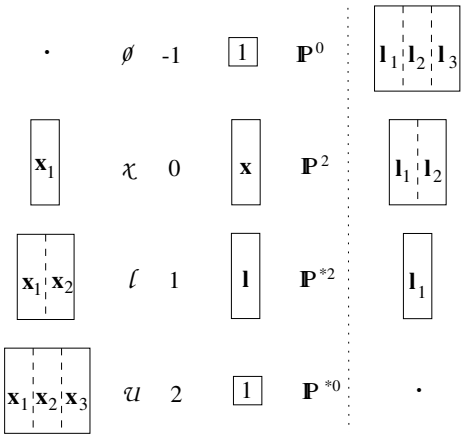


Table 5.5 Visualization of Plücker coordinates with dual representations for 2D elements. Six columns from left to right: (1) The space spanned by the join of 0, 1, 2, and 3 2D points, (2) the resulting entities, the empty space $\emptyset$, the point $\mathcal{X}$, the line $\mathcal{l}$ joining two points and the complete projective plane, the universe $\mathcal{U}$, (3) the dimension of these manifolds, (4) the representation with Plücker coordinates, (5) the dimension of the projective space of the Plücker coordinates, and (6) the space spanned by the intersection of 3, 2, 1, and 0 lines

The dimension n of the representing projective space is 1 less than the number of $k \times k$-minors of the $n \times k$ matrices where $0 \leq k \leq n+1$, cf. Sect. 5.5.2, p. 224; e.g., for $\mathbb{P}^2$ we have: $[\binom{n+1}{k} - 1] = [0, 2, 2, 0]$, formally including the case $k = 0$.

The two extreme cases with $\mathbb{P}^0$ require an explanation. In 2D, starting from a zero-dimensional point $\mathcal{X}$ reduction by one dimension formally leads to the empty space, $\emptyset$, which is a -1-dimensional manifold. The empty space is the intersection of three lines in arbitrary position, which according to (5.101), p. 224 is represented by $o \in \mathbb{P}^0$.

Starting from the line in 2D space and increasing the dimension by 1 yields the complete 2D space $\mathcal{U}$ (universe), which according to (5.95), p. 223 is represented by the join of three points in arbitrary position, which by analogy is the only element in the dual space $\mathbb{P}^{*0}$. Also, this entity is a nonzero homogeneous 1-vector; thus, it can be set to $[x_1] = [1]$.

The discussion reveals the full symmetry of all relations and the clear structure of the construction of homogeneous representations of geometric entities. It is closely related

Table 5.6 Dual entities in $\mathbb{P}^3$. For explanation see the table before. Observe, the dual of a 3D line is a 3D line.

		entity $\mathcal{G}$	dual entity $\overline{\mathcal{G}}$	Hodge-Operator
1	d	-1	3	
	name	$\emptyset$	$\mathcal{U}$	
	represent. space	$\mathbb{P}^0$	$\mathbb{P}^{*0}$	1
2	d	0	2	
	name	$\mathcal{X}$	$\mathcal{A}$	
	represent. space	$\mathbb{P}^3$	$\mathbb{P}^{*3}$	I_4
3	d	1	1	
	name	$\mathcal{L}$	$\mathcal{M} = \overline{\mathcal{L}}$	
	represent. space	$\mathbb{P}^5$	$\mathbb{P}^{*5}$	$D = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}$
4	d	2	0	
	name	$\mathcal{A}$	$\mathcal{X}$	
	represent. space	$\mathbb{P}^{*3}$	$\mathbb{P}^3$	I_4
5	d	3	-1	
	name	$\mathcal{U}$	$\emptyset$	
	represent. space	$\mathbb{P}^{*0}$	$\mathbb{P}^0$	1

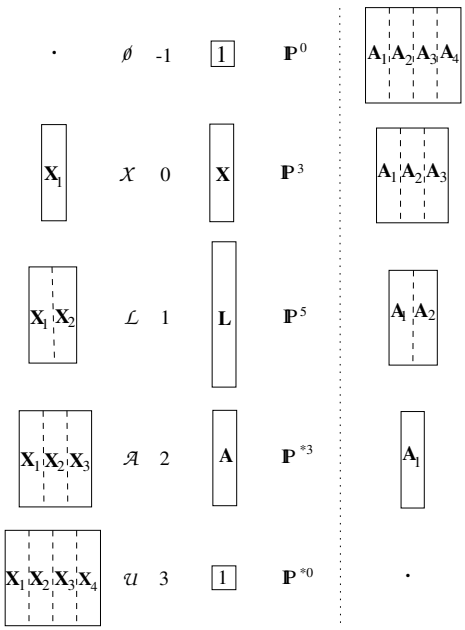


Table 5.7 Visualization of Plücker coordinates with dual representations for 3D elements. Six columns from left to right: (1) The space spanned by the join of 0, 1, 2, 3 and 4 3D points, (2) the resulting entities, the empty space $\emptyset$, the point $\mathcal{X}$, the 3D line $\mathcal{L}$ joining two points, the plane $\mathcal{A}$ through three points, and the complete projective plane, the universe $\mathcal{U}$, (3) the dimension of these manifolds, (4) the representation with Plücker coordinates, (5) the dimension of the projective space of the Plücker coordinates, and (6) the space spanned by the intersection of 4, 3, 2, 1, and 0 planes

to the geometric calculus based on determinants, discussed in the previous section and realized by Browne (2009).

5.6.2 The Dual 3D Line

Given a 3D line by the join of two points $\mathcal{X}, \mathcal{Y}$, its dual line is the intersection of the two planes $\mathcal{A}, \mathcal{B}$ dual to the given points. Obviously, we may define the dual 3D line

$$\overline{\mathcal{L}} = \overline{\mathcal{X} \wedge \mathcal{Y}} = \overline{\mathcal{X}} \barwedge \overline{\mathcal{Y}} = \mathcal{A} \cap \mathcal{B} \tag{5.132}$$

either

1. by the dual Plücker coordinates using (5.65), p. 217, (5.131), p. 230, *dual Plücker coordinates*

$$\overline{\mathcal{L}}(\overline{\mathbf{L}}): \quad \overline{\mathbf{L}} = \mathsf{D}\mathbf{L} = \begin{bmatrix} \boldsymbol{L}_0 \\ \boldsymbol{L}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_0 \times \boldsymbol{Y}_0 \\ X_h \boldsymbol{Y}_0 - Y_h \boldsymbol{X}_0 \end{bmatrix}, \tag{5.133}$$

   thus by exchanging the homogeneous and the Euclidean parts of the coordinate vector $\mathbf{L}$ of $\mathcal{L}$, or *dual Plücker matrix*
2. by the *dual Plücker matrix* as a function of $\mathbf{L}$ and the homogeneous coordinates $\mathbf{A}$ and $\mathbf{B}$ of two generating planes,

$$\overline{\mathcal{L}}(\overline{\mathsf{\Gamma}}(\mathbf{L})): \quad \overline{\mathsf{\Gamma}}(\mathbf{L}) = \mathsf{\Gamma}(\overline{\mathbf{L}}) = \begin{bmatrix} -\mathsf{S}(\boldsymbol{L}_h) & -\boldsymbol{L}_0 \\ \boldsymbol{L}_0^\mathsf{T} & 0 \end{bmatrix} = \mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T}. \tag{5.134}$$

   The first expression results from (5.68), p. 219 by exchanging vectors $\boldsymbol{L}_0$ and $\boldsymbol{L}_h$. The second results from (5.122), p. 228 and (5.125), p. 228.

Since we distinguish between the name of an entity and its representation, we may also represent a line by the coordinates of the dual line, thus $\mathcal{L}(\overline{\mathbf{L}})$, by the dual Plücker matrix, thus $\mathcal{L}(\overline{\mathsf{\Gamma}}(\mathbf{L}))$, or by making the underlying construction explicit: e.g., $\mathcal{L} = \mathcal{A} \cap \mathcal{B} = \mathcal{L}(\mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T})$.[7]

5.6.3 Geometric Relations Between Dual Entities

The geometric relation between dual entities is a very intimate one and also closely related to the concept of polarity at the unit circle, cf. Figs. 5.20 to 5.22. *duality and polarity*

In all presented cases we observe the following relations between an entity and its dual:

1. The entity and its dual lie on opposite sides of the origin.
2. The product of their distances to the origin gives 1.
3. The shortest distance between both passes through the origin.

As a special case take the dual to a 2D line l through the origin. The dual point

$$\mathcal{x} = \overline{\mathit{l}}: \quad \mathbf{x} = \bar{\mathbf{l}} = \begin{bmatrix} \boldsymbol{l}_h \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\phi \\ \sin\phi \\ 0 \end{bmatrix} \tag{5.135}$$

is a point at infinity in the direction of the normal of the line, or equivalently, the point at infinity v_{n_l} of a line n_l perpendicular to l. Analogously, given a plane $\mathcal{A}$ passing through the origin, its dual 3D point $\mathcal{X} = \overline{\mathcal{A}}$,

$$\mathcal{X} = \overline{\mathcal{A}}: \quad \mathbf{X} = \overline{\mathbf{A}} = \begin{bmatrix} \boldsymbol{N}_A \\ 0 \end{bmatrix}, \tag{5.136}$$

is the point at infinity, $\mathcal{V}_{N_A}$, of a line $\mathcal{N}_A$ perpendicular to the plane, see Fig. 5.21.

[7] As above, a 3D line $\mathcal{L}$ is a geometric entity in $\mathbb{P}^3$ and represented as a point $\mathbf{L}$ in $\mathbb{P}^5$. But a dual line $\overline{\mathcal{L}} = \mathsf{D}\mathbf{L} = \mathcal{M}$ is an element of $\mathbb{P}^{*3}$ and represented as point $\overline{\mathbf{L}}$ in $\mathbb{P}^{*5}$ or when interpreted as $\mathcal{M} = \mathsf{D}\mathbf{L}$ in $\mathbb{P}^3$ represented as a point $\mathbf{M}$ in $\mathbb{P}^5$. We always work with 3D points $\mathcal{X}$, 3D lines $\mathcal{L}$, and $\mathcal{A}$ in $\mathbb{P}^3$.

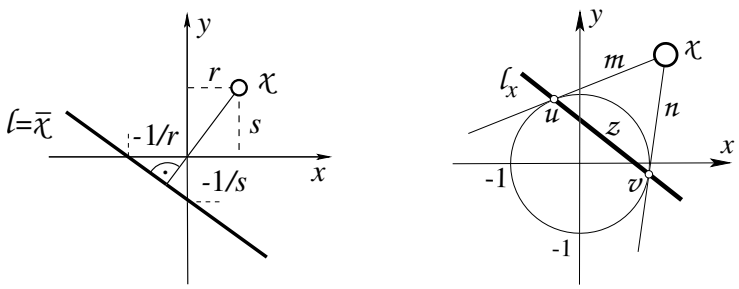


Fig. 5.20 Duality and polarity w.r.t. the unit circle. **Left:** *Duality* of point and line in the plane. The point $\mathcal{X}$ and the line $\mathcal{l} = \overline{\mathcal{X}}$ have the same homogeneous coordinates $[r, s, t]^\mathsf{T}$, in this figure $t = 1$. **Right:** *Polarity* of point and line w.r.t. the unit circle; the line $\mathcal{l}_x$, which is the polar line to the point $\mathcal{X}$, is the line connecting the two tangent points u and v of the two tangents m and n of $\mathcal{X}$ at the unit circle, or dually, the point $\mathcal{X}$, which is the pole to the line $\mathcal{l}_x$, is the intersection of the two tangents m and n of the intersection points u and v of the line $\mathcal{l}_x$ with the unit circle. We have the following relation: The dual line $\mathcal{l}$ (left) and the polar line $\mathcal{l}_x$ (right) with respect to the unit circle are point symmetric to the origin. This geometric relation transfers to 3D, see the next two figures. More on polarity is given in Sect. 5.7.1.4, p. 238, cf. exercise 8, p. 245

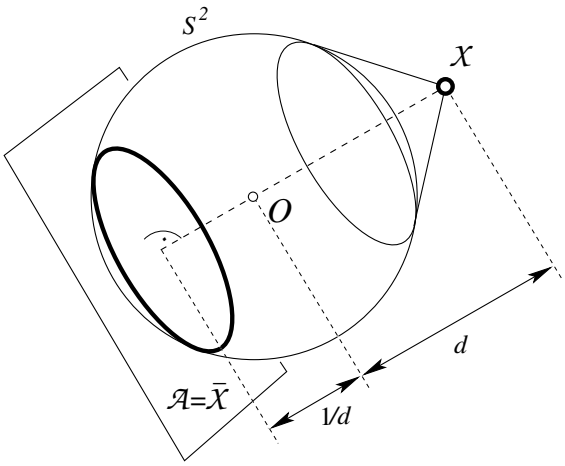


Fig. 5.21 Duality of point and plane in 3D. The point $\mathcal{X}$ and the plane $\mathcal{A}$ have the same homogeneous coordinates, their distances from the origin O multiply to one. The tangent lines from $\mathcal{X}$ at the unit circle in 2D (cf. Fig. 5.20, right) transfer here to a tangent cone at a unit sphere in 3D; the tangent points here form a small circle. The line joining the tangent points in 2D, cf. Fig. 5.20, right, transfers in 3D to the plane containing the small circle not shown here. The symmetrical counterpart of this small circle, shown as bold circle, lies in A

5.6.4 Relation to Grassmann–Cayley and Geometric Algebra

Grassmann–Cayley algebra

There is a close relationship between Plücker coordinates and what is called the *Grassmann algebra*, which can be used to represent them in a natural way. The prominent feature of Grassmann algebra is that the basic vectors of the vector space are made explicit in the representations. Faugeras and Papadopoulo (1998) showed how to represent the geometry of one, two and three images using Grassmann–Cayley algebra. The book of Browne (2009) develops the Grassmann algebra in arbitrary dimensions using basic concepts from linear algebra.

As an example, take the three-dimensional space $\mathbb{R}^3$ with base vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$. We define an exterior product "$\wedge$" of vectors $\mathbf{x} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + x_3\boldsymbol{e}_3$ with associativity and distributivity, and the following property:

$$\mathbf{x} \wedge \mathbf{y} = -\mathbf{y} \wedge \mathbf{x}\,, \tag{5.137}$$

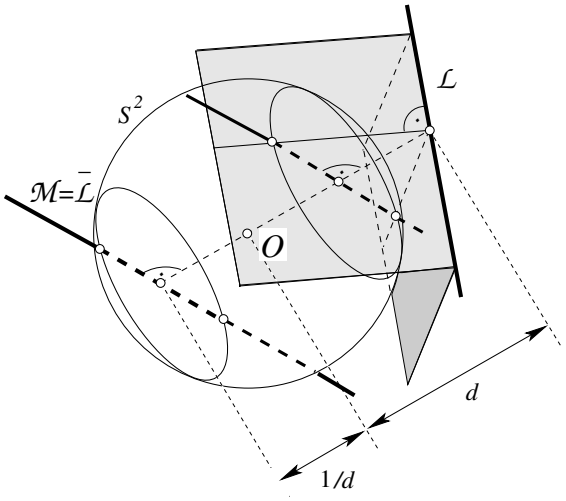


Fig. 5.22 Duality of two 3D lines. The two lines $\mathcal{L}$ and $\mathcal{M}$ have the same homogeneous coordinates, except for the exchange of their homogeneous and Euclidean part. The dual line $\mathcal{M}$ is the join of the points opposite to the tangent points at the unit sphere of the two tangent planes of the line $\mathcal{L}$. The line $\mathcal{L}$ and its dual line $\mathcal{M} = \bar{\mathcal{L}}$ are mutually orthogonal as $\boldsymbol{L}_0 \perp \boldsymbol{L}_h$

which is equivalent to the property

$$\mathbf{x} \wedge \mathbf{x} = 0\,. \tag{5.138}$$

The exterior product of two vectors $\mathbf{x}$ and $\mathbf{y}$ leads to what is called a *bivector*,

$$\begin{aligned}\mathbf{l} = \mathbf{x} \wedge \mathbf{y} &= (x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + x_3\boldsymbol{e}_3) \wedge (y_1\boldsymbol{e}_1 + y_2\boldsymbol{e}_2 + y_3\boldsymbol{e}_3) \\ &= x_1y_1\boldsymbol{e}_1 \wedge \boldsymbol{e}_1 + x_1y_2\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + x_1y_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 \\ &\quad + x_2y_1\boldsymbol{e}_2 \wedge \boldsymbol{e}_1 + x_2y_2\boldsymbol{e}_2 \wedge \boldsymbol{e}_2 + x_2y_3\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \\ &\quad + x_3y_1\boldsymbol{e}_3 \wedge \boldsymbol{e}_1 + x_3y_2\boldsymbol{e}_3 \wedge \boldsymbol{e}_2 + x_3y_3\boldsymbol{e}_3 \wedge \boldsymbol{e}_3 \\ &= (x_2y_3 - x_3y_2)\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 + (x_3y_1 - x_1y_3)\boldsymbol{e}_3 \wedge \boldsymbol{e}_1 + (x_1y_2 - x_2y_1)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2\,,\end{aligned} \tag{5.139}$$

having a basis consisting of three bivectors, namely $\boldsymbol{e}_2 \wedge \boldsymbol{e}_3$, $\boldsymbol{e}_3 \wedge \boldsymbol{e}_1$, and $\boldsymbol{e}_1 \wedge \boldsymbol{e}_2$, and having coordinates that are the minors of the matrix $[\mathbf{x}, \mathbf{y}]$. In our context, bivectors can be interpreted geometrically as lines $\mathcal{l}$.

The exterior product $\mathbf{x} \wedge \mathbf{y} \wedge \mathbf{z}$ of three vectors leads to the trivector

$$\mathbf{x} \wedge \mathbf{y} \wedge \mathbf{z} = D_x\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3\,, \tag{5.140}$$

with $D_x = |\mathbf{x}, \mathbf{y}, \mathbf{z}|$, which is the determinant of the three vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. The basis of this trivector is

$$I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3\,. \tag{5.141}$$

In this Grassmann algebra with three basis vectors, all exterior products with four or more vectors are zero.

The concept of duality is intimately related to the duality of the corresponding bases. The basis $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ for points is dual to the basis $(\boldsymbol{e}_2 \wedge \boldsymbol{e}_3, \boldsymbol{e}_3 \wedge \boldsymbol{e}_1, \boldsymbol{e}_1 \wedge \boldsymbol{e}_2)$ in the following sense: the join of corresponding base elements yields $I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$.

The concept can be extended to the Grassmann–Cayley algebra by introducing a regressive product, equivalent to the cap product used for the intersection of 2D lines or planes, and an inner product. The developments in this and the following chapters can be understood as the coordinate version of the approach of Grassmann–Cayley algebra, fixing the sequence of the basis vectors in a meaningful manner.

Representing the geometric entities by their coordinates is motivated by the possibility of integrating statistical reasoning into the geometric reasoning, which requires an attachment of uncertainties to measurable quantities which are functions of the coordinates of the geometric entities.

geometric algebra

The integration of the inner (dot) and the exterior product in the *geometric algebra* by Hestenes and Ziegler (1991), by defining the geometric product as $\mathbf{xy} = \mathbf{x}.\mathbf{y}+\mathbf{x}\wedge\mathbf{y}$, further simplifies the algebraic treatment (Dorst et al., 2009). The construction and constraints presented here actually have been verified using the geometric algebra package by Ashdown (1998).

However, in the case of integrating statistics, it also requires falling back on the coordinates (Perwass, 2009) which is even more relevant for uncertain geometric reasoning with the more general conformal geometric algebra (Gebken, 2009).

5.7 Conics and Quadrics

5.7.1 Conics . . . 236
5.7.2 Quadrics . . . 239
5.7.3 Singular Conics and Quadrics . . . 241
5.9.1 Visualization of $\mathbb{P}^2$ Using the Stereographic Projection . . . 243
5.9.2 Elements of a 2D Coordinate System . . . 243
5.9.3 Elements of a 3D Coordinate System . . . 244

Conics and quadrics are second-order curves and surfaces, respectively, which frequently occur in image analysis, especially when observing circles, cylinders, or spheres. They can easily be represented using homogeneous coordinates and be used to describe invariants of geometric transformations. We consider conics defined by points and also their duals defined by straight lines.

5.7.1 Conics

5.7.1.1 General Form

Conics are intersections of a circular cone with a plane. The general form is

$$C(x,y) = a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + 2a_{13}x + 2a_{23}y + a_{33} = 0 \tag{5.142}$$

or

$$C(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T}\mathsf{A}\boldsymbol{x} + 2\boldsymbol{a}^\mathsf{T}\boldsymbol{x} + a = 0\,, \tag{5.143}$$

with

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathsf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \qquad \boldsymbol{a} = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \qquad a = a_{33}. \tag{5.144}$$

Substituting $x = u/w$ and $y = v/w$ we obtain

$$C(u,v,w) = a_{11}u^2 + 2a_{12}uv + a_{22}v^2 + 2a_{13}uw + 2a_{23}vw + a_{33}w^2 = 0\,, \tag{5.145}$$

motivating the indices of the coefficients a_{ij}.

The quadratic form can be written as a homogeneous form,

$$\mathcal{C}: \quad C(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0 \tag{5.146}$$

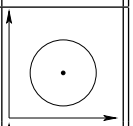
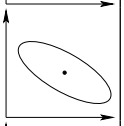
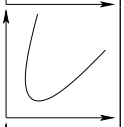
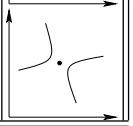
with the symmetric and homogeneous matrix[8]

$$\mathsf{C} = \left[\begin{array}{cc|c} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \hline a_{31} & a_{32} & a_{33} \end{array}\right] = \begin{bmatrix} \mathsf{C}_{hh} & \boldsymbol{c}_{h0} \\ \boldsymbol{c}_{0h}^{\mathsf{T}} & c_{00} \end{bmatrix} = \begin{bmatrix} A & \boldsymbol{a} \\ \boldsymbol{a}^{\mathsf{T}} & a \end{bmatrix}, \tag{5.147}$$

motivating the factors 2. We again partition the representing algebraic entity, the 3×3 matrix C, according to the homogeneous and the Euclidean parts of the homogeneous vector $\mathbf{x}$.

We have collected the following important special curves in Table 5.8.

Table 5.8 Special regular conics with $|\mathsf{C}| \neq 0$

name	constraints	
circle	$\mathsf{C}_{hh} = I_2$	
ellipse	$\lvert\mathsf{C}_{hh}\rvert > 0$	
parabola	$\lvert\mathsf{C}_{hh}\rvert = 0$	
hyperbola	$\lvert\mathsf{C}_{hh}\rvert < 0$	

5.7.1.2 Point of Symmetry

Interestingly conics, which are the intersection of an arbitrary plane with a cone, are point symmetric if C_{hh} is regular. The point of symmetry $\mathbf{x}_0$ can be determined by rearranging (5.143). We arrive at the central form of a symmetric conic:

$$C(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathsf{T}} \mathsf{C}_{hh} (\boldsymbol{x} - \boldsymbol{x}_0) + \overline{c_{00}} = 0\,, \tag{5.148}$$

with

$$\boldsymbol{x}_0 = -\mathsf{C}_{hh}^{-1} \boldsymbol{c}_{h0} \qquad \overline{c_{00}} = c_{00} - \boldsymbol{c}_{h0}^{\mathsf{T}} C_{hh}^{-1} \boldsymbol{c}_{h0}\,, \tag{5.149}$$

as can be verified easily. Thus, if a point $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{d}$ lies on the conic, the point $\boldsymbol{x} = \boldsymbol{x}_0 - \boldsymbol{d}$ lies on the conic as well.

5.7.1.3 Parametric Form of Regular Conics

For plotting purposes a parametric representation is necessary. This can easily be achieved for regular conics, i.e., ellipses and hyperbola, by using the form (5.148) and translating and rotating the coordinate system. The eigen decomposition of $\mathsf{C}_{hh}^e = -\mathsf{C}_{hh}/\overline{c_{00}} = R \Lambda R^{\mathsf{T}}$ leads to eigenvectors $R = [\boldsymbol{r}_1, \boldsymbol{r}_2]$ and the eigenvalues in $\Lambda = \mathrm{Diag}([\lambda_1, \lambda_2])$. Then we have the transformed coordinates

$$\boldsymbol{y} = R^{\mathsf{T}} (\boldsymbol{x} - \boldsymbol{x}_0)\,, \tag{5.150}$$

[8] Thereby switching the variable names from a's to c's.

for which the conic has the form,

$$\frac{y_1^2}{\frac{1}{\lambda_1}} + \frac{y_2^2}{\frac{1}{\lambda_2}} = 1\,. \tag{5.151}$$

If both eigenvalues are positive, we obtain an ellipse. The ellipse can thus be written in parametric form,

$$y_1(t) = \sqrt{\frac{1}{\lambda_1}}\cos t \qquad y_2(t) = \sqrt{\frac{1}{\lambda_2}}\sin t \qquad t \in [0, 2\pi)\,. \tag{5.152}$$

If the eigenvalues have different signs, we assume $\lambda_2 < 0$, and obtain a hyperbola which can be written parametrically,

$$y_1(t) = \pm\sqrt{\frac{1}{\lambda_1}}\frac{1}{\cos t} \qquad y_2(t) = \sqrt{\frac{1}{-\lambda_2}}\tan t\,. \tag{5.153}$$

The two signs in y_1 correspond to the two parts of the hyperbola. If $\lambda_1 < 0$, we have to exchange the eigenvalues and the corresponding eigenvectors in R.

If the range of the hyperbola in the y_2-direction is given by $[-a, a]$, e.g., when plotting an error band of a line segment of length $2a$, then the range of the parameter t is $[-\mathrm{atan}(a/\sqrt{-\lambda_2}), \mathrm{atan}(a/\sqrt{-\lambda_2})]$.

Both curves can then be plotted in the original coordinate system using $\boldsymbol{x}(t) = \boldsymbol{x}_0 + R\boldsymbol{y}(t)$.

5.7.1.4 Polarity and Tangent Line

Each point $\mathcal{X}$ has a corresponding *polar line* $\mathcal{l}_x$ with respect to a conic $\mathcal{C}$, cf. Fig. 5.23,

$$\mathcal{l}_x : \quad \mathbf{l}_x = \mathsf{C}\mathbf{x}\,. \tag{5.154}$$

Vice versa, each line $\mathcal{l}$ has a corresponding *pole*,

$$\mathcal{X}_l : \quad \mathbf{x}_l = \mathsf{C}^\circ\,\mathbf{l}\,, \tag{5.155}$$

which can be constructed in a similar manner or can be derived just by dualizing (5.154), cf. the next section. Instead of using the (transpose) inverse $C^{-\mathsf{T}}$ for defining the pole of a line, we use the cofactor matrix C° in order to allow for singular conics with $|C| = 0$, cf. Sect. 5.7.3, p. 241.[9]

If the pole is on the outside the conic, i.e., on the convex side, cf. $\mathcal{X}_3$, the polar is obtained by (1) taking the tangents at $\mathcal{C}$ through $\mathcal{X}_3$, say $\mathcal{m}_1$ and $\mathcal{m}_2$, and (2) joining the two tangent points, say $\mathcal{y}_1$ and $\mathcal{y}_2$, yielding the polar line $\mathcal{l}_3$ of the point $\mathcal{X}_3$. If the pole is inside the conic, i.e., on the concave side, cf. $\mathcal{X}_1$, we obtain the polar by (1) joining $\mathcal{X}_1$ with the centre $\mathcal{z}$, obtaining the line $\mathcal{m}$, and taking the intersection point $\mathcal{X}_2$ of $\mathcal{m}$ with the conic closest to $\mathcal{X}_1$, (2) intersecting the conic with a line $\mathcal{l}_3$ through $\mathcal{X}_1$ and parallel to the tangent $\mathcal{l}_2$ at $\mathcal{X}_2$, leading to points $\mathcal{y}_1$ and $\mathcal{y}_2$, and (3) taking the line $\mathcal{l}_1$ through the intersection point $\mathcal{X}_3$ of the two tangents at $\mathcal{y}_1$ and $\mathcal{y}_2$ parallel to $\mathcal{l}_3$.

Exercise 5.16

This point-to-line correspondence, called *polarity*, is closely related to duality, as we saw in Fig. (5.20), p. 234. It is also a particular example of other projective point–line transformations, which are called *correlations*, cf. Sect. 6.6, p. 282.

Exercise 5.8

If the pole $\mathcal{X}$ lies on the conic, the polar line is its tangent (cf. Hartley and Zisserman, 2000, Sect. 2.2). If the point $\mathcal{X}$ is positively directed, the normal of $\mathcal{l}_x$ in (5.154) points to

tangent at conic

[9] Often the dual conic matrix is defined as the adjugate matrix C^* (cf. Hartley and Zisserman, 2000, Sect. 2.2); since C is symmetric the cofactor and the adjugate matrix are identical, and we can take C° as the dual transformation of C, cf. (6.50), p. 259.

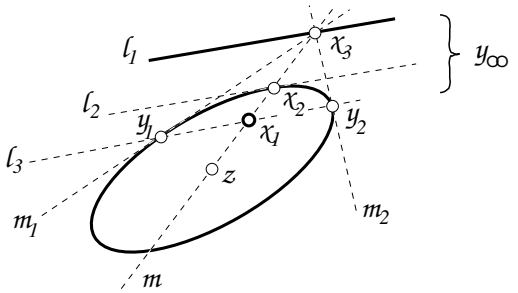


Fig. 5.23 Polarity at a conic. The line l_1 is the polar of x_1 w.r.t. the ellipse. The point x_1 is the pole of l_1 w.r.t. the ellipse. If the point lies on the conic, as does the point x_2, the polar is the tangent at the conic in that point, here the line l_2. If the point is at infinity, i.e., the intersection point y_∞ of the parallel lines l_i, then the polar is the line through the centre z passing through the tangent point with tangent parallel to l_i, here the line m, which is at the same time the line joining the three collinear points x_i. Similarly, the point x_3 and the line l_3 are related by polarity

the positive area of the conic; for the sign of points and conics, cf. Sect. 9.1.1.7, p. 348. The mutual relation of pole, polar line, and tangents is the topic of an exercise. Exercise 5.9

If the conic has the form Exercise 5.10

$$\mathsf{C} = I_3\,, \tag{5.156}$$

we obtain the relation for the polar line $\mathbf{l}_x = \mathbf{x}$. But this has been the relation between the point and its dual, cf. Sect. 5.6.3. Thus the duality is the standard polarity, namely the polarity w.r.t. the conic $\mathsf{C} = \mathrm{Diag}([1, 1, 1])$. It is an improper conic: it does not contain any real points but only imaginary points, namely with coordinates $[i\cos\phi, i\sin\phi]$ with the imaginary unit $i = \sqrt{-1}$.

5.7.1.5 The Line Conic

Instead of defining a conic by its set of points, we can also define it by the envelope of its tangents,

$$\mathbf{l}^\mathsf{T}\mathsf{C}^\circ\mathbf{l} = 0\,, \tag{5.157}$$

see Fig. 5.24, where C° is the cofactor of matrix C. If it is regular, then due to symmetry $\mathsf{C}^\circ = |\mathsf{C}|\,\mathsf{C}^{-1}$.

This can be shown as follows: an arbitrary point x on the conic with its tangent is $\mathbf{l} = \mathsf{C}\mathbf{x}$. Due to the symmetry of C, we have $\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = (\mathsf{C}^\circ\mathbf{l})^\mathsf{T}\mathsf{C}(\mathsf{C}^\circ\mathbf{l}) = \mathbf{l}^\mathsf{T}\mathsf{C}^\circ\mathbf{l} = 0$.

When we represent a conic $\mathcal{C}$ by C°, we call it a *line conic*. Since it refers to lines, it often is also called the *dual conic*.[10] *line conic, dual conic*

5.7.2 Quadrics

Quadrics $\mathcal{Q}$, second-order surfaces in 3D, have the general form

$$\mathcal{Q}: \quad \mathsf{Q} = \begin{bmatrix} \mathsf{Q}_{hh} & \boldsymbol{q}_{h0} \\ \boldsymbol{q}_{0h}^\mathsf{T} & q_{00} \end{bmatrix} \qquad \mathbf{X}^\mathsf{T}\mathsf{Q}\mathbf{X} = 0\,, \tag{5.158}$$

with the symmetric 4×4 matrix Q and the homogeneous coordinates $\mathbf{X}$ of a 3D point $\mathcal{X}$ sitting on $\mathcal{Q}$. There are several important special cases. For regular quadrics with $|\mathsf{Q}| \neq 0$ and assuming $q_{00} < 0$, we have as a function of the upper left matrix Q_{hh}

[10] If the matrix C° is applied to points, it represents the dual conic $\overline{\mathcal{C}}$, thus $\overline{\mathcal{C}}: \mathbf{x}^\mathsf{T}\mathsf{C}^\circ\mathbf{x} = 0$. However, we do not use the name $\overline{\mathcal{C}}$ in the following.

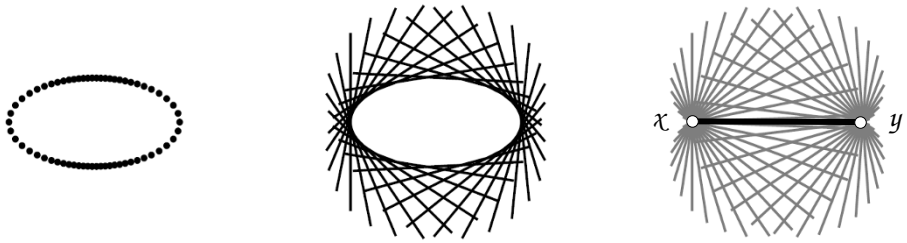


Fig. 5.24 Regular and singular point and line conics. **Left**: Regular point conic (5.146), $\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0$, here $\mathsf{C} = \mathrm{Diag}([1, 4, -4])$. **Middle**: Regular line conic, cf. Sect. 5.7.1.5, $\mathbf{l}^\mathsf{T}\mathsf{C}^\circ\mathbf{l} = 0$, $\mathsf{C}^\circ = \mathrm{Diag}([4, 1, -1])$ up to a scale factor of -4. **Right**: Singular line conic (5.162), $\mathbf{l}^\mathsf{T}\mathsf{C}^\circ\mathbf{l} = 0$, $\mathsf{C}^\circ = \mathrm{Diag}([4, 0, -1]) = \mathbf{x}\mathbf{y}^\mathsf{T} + \mathbf{y}\mathbf{x}^\mathsf{T}$ with points $\mathcal{X}([-2, 0, 1]^\mathsf{T})$ and $\mathcal{Y}([2, 0, 1]^\mathsf{T})$ representing a flat ellipse, which consist of two bundles of concurrent lines

- the sphere: $Q_{hh} = \lambda I_3$,
- the ellipsoid: the eigenvalues $\lambda(Q_{hh})$ are positive,
- the hyperboloid of two sheets: one of the eigenvalues of Q_{hh} is negative, and
- the hyperboloid of one sheet: two of the eigenvalues of Q_{hh} are negative.

Among the singular quadrics, we have the hyperbolic paraboloid, e.g., $xy - z - 1 = 0$; the cone, e.g., $x^2 + y^2 - z^2 = 0$; and especially the circular cylinder, e.g., $x^2 + y^2 - 1 = 0$, which we characterize in more detail below due to its practical relevance, see Fig. 5.25.[11]

Fig. 5.25 Quadrics. Sphere: $\mathsf{Q} = \mathrm{Diag}([1, 1, 1, -1])$, ellipsoid: $\mathsf{Q} = \mathrm{Diag}([1, 4, 12, -12])$, hyperboloid of two sheets: $\mathsf{Q} = \mathrm{Diag}([-1, -4, 1, -1])$, hyperboloid of one sheet: $\mathsf{Q} = \mathrm{Diag}([3, 1, -1, -1])$, hyperbolic paraboloid: $z = xy$, cone: $\mathsf{Q} = \mathrm{Diag}([1, 1, -1, 0])$, cylinder: $\mathsf{Q} = \mathrm{Diag}([1, 1, 0, -1])$

We again have the following relations:

- Quadrics, like conics, partition 3D space into a positive and a negative region defined by the sign of $Q(\mathbf{X}) = \mathbf{X}^\mathsf{T}\mathsf{Q}\mathbf{X}$, where $\mathcal{X}(\mathbf{X})$ is a point in the region not on the quadric.

point of symmetry of a regular quadric

- If the upper left 3×3 submatrix Q_{hh} is regular, the *point of symmetry* $\mathcal{X}_0$ is, cf. (5.149), p. 237

$$\boldsymbol{X}_0 = -Q_{hh}^{-1}\boldsymbol{q}_{h0}\,. \tag{5.159}$$

- The *tangent plane* $\mathcal{A}_X$ at point $\mathcal{X}$ to a quadric is given by, cf. (5.154), p. 238

$$\mathcal{A}_X: \quad \mathbf{A}_x = \mathsf{Q}\mathbf{X}. \tag{5.160}$$

tangent plane

The normal of the plane points into the positive region if $X_h > 0$.

- All tangent planes $\mathcal{A}$ of $\mathcal{Q}$ form an envelope of it. This yields the *plane quadric*, cf. (5.157), p. 239 (for using the cofactor matrix, cf. the footnote on p. 238)

$$\mathbf{A}^\mathsf{T}\mathsf{Q}^\circ\mathbf{A} = 0\,, \tag{5.161}$$

where Q° is the cofactor matrix of Q. Also here, Q° is sometimes called the dual quadric.

[11] A further classification of the different special cases can be found in the 30th Edition of the CRC Standard Mathematical Tables and Formulas (CRC Press).

- The intersection of a 3D line $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ with the quadric can be determined in the same way as in 2D, to be discussed later, cf. ((7.7), p. 293), again leading to two intersection points at maximum.

5.7.3 Singular Conics and Quadrics

A singular conic generally is a line pair. If it consists of the lines l and m it is given by

$$\mathsf{C} = \mathbf{l}\mathbf{m}^\mathsf{T} + \mathbf{m}\mathbf{l}^\mathsf{T}, \tag{5.162}$$

as all points x either on l or m lead to $\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0$.

Analogously, a singular line conic is the flat ellipse between two given points x and y. It is represented by the line conic (for using the cofactor matrix, cf. the footnote on p. 238)

$$\mathsf{C}^\circ = \mathbf{x}\mathbf{y}^\mathsf{T} + \mathbf{y}\mathbf{x}^\mathsf{T}, \tag{5.163}$$

as all lines $\mathbf{l}$ passing through x or y satisfy $\mathbf{l}^\mathsf{T}\mathsf{C}^\circ\mathbf{l} = 0$, cf. Fig. 5.24.

We especially use the singular line conic

singular line conic
singular dual conic

$$\mathsf{C}_\infty^\circ = \begin{bmatrix} \mathit{I}_2 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \tag{5.164}$$

for deriving angles between lines, as the upper left unit matrix selects the normal direction $\boldsymbol{l}_h$ from a line $\mathit{l}(\mathbf{l})$, cf. (7.1.3), p. 297. It is also an invariant conic for planar motions, cf. (6.4.5.2), p. 274. The singular dual conic C_∞° can be defined by

$$\mathsf{C}_\infty^\circ = \mathbf{i}\mathbf{j}^\mathsf{T} + \mathbf{j}\mathbf{i}^\mathsf{T}, \tag{5.165}$$

with the two *absolute points*

absolute points

$$\mathbf{i} = \begin{bmatrix} i \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{j} = \begin{bmatrix} -i \\ 1 \\ 0 \end{bmatrix} \qquad \text{with} \qquad i = \sqrt{-1}. \tag{5.166}$$

We will follow convention and refer to C_∞° as the singular dual conic.

Regular 3D conics can be represented as flat quadrics. In normalized position, namely when they are flat in the Z-direction, they have the form

$$\mathsf{Q}_0^\circ = \mathrm{Diag}([\lambda_1, \lambda_2, 0, -1]). \tag{5.167}$$

For example, a 3D circle with radius R lying in the XY plane has the quadric representation $\mathrm{Diag}([R^2, R^2, 0, -1])$. In 3D, we also use the singular dual quadric

$$\mathsf{Q}_\infty^\circ = \begin{bmatrix} \mathit{I}_3 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \tag{5.168}$$

for deriving the angles between planes.

5.8 Normalizations of Homogeneous Vectors

The ambiguity of homogeneous coordinates w.r.t. scaling needs to be eliminated when visualizing or comparing geometric entities. This can be achieved by proper normalization. As already shown for 2D entities in Sect. 5.1.2.2, p. 198, Euclidean normalization discloses the Euclidean properties of an element, either the inhomogeneous coordinates or

the distance to the origin. Spherical normalization forces the homogeneous vector to lie on the unit sphere.

Euclidean Normalization By *Euclidean normalization* the vector is normalized such that the homogeneous part has Euclidean norm 1.

For the 2D point $\mathcal{x}$ and the 2D line $\mathcal{l}$, we obtain

$$\mathbf{x}^e = \frac{\mathbf{x}}{x_h} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}, \qquad \mathbf{l}^e = \frac{\mathbf{l}}{|\boldsymbol{l}_h|} = \begin{bmatrix} \boldsymbol{n} \\ -d \end{bmatrix}. \tag{5.169}$$

Similarly, we obtain in 3D,

$$\mathbf{X}^e = \frac{\mathbf{X}}{X_h} = \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix}, \quad \mathbf{A}^e = \frac{\mathbf{A}}{|\boldsymbol{A}_h|} = \begin{bmatrix} \boldsymbol{N} \\ -S \end{bmatrix}, \quad \mathbf{L}^e = \frac{\mathbf{L}}{|\boldsymbol{L}_h|} = \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix}. \tag{5.170}$$

Obviously, entities at infinity cannot be represented by this Euclidean normalization, as their homogeneous part is 0.

Spherical Normalization In *spherical normalization*, all coordinates of a homogeneous vector are processed the same way, and the complete vector is normalized to 1. Thus for any entity $\mathbf{g}$ we have

$$\mathbf{g}^s = \mathsf{N}(\mathbf{g}) = \frac{\mathbf{g}}{|\mathbf{g}|} \tag{5.171}$$

using the normalization operator $\mathsf{N}(.)$.

For example, we have the spherically normalized homogeneous coordinates of a 2D point $\mathcal{x}$,

$$\mathbf{x}^s = \mathsf{N}(\mathbf{x}) = \mathsf{N}\left(\begin{bmatrix} u \\ v \\ w \end{bmatrix}\right) = \frac{1}{\sqrt{u^2+v^2+w^2}} \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{5.172}$$

Thus, the spherically normalized homogeneous coordinates of all 2D points and 2D lines build the unit sphere S^2 in $\mathbb{R}^3$, cf. Fig. 5.3, p. 201. By analogy, the spherically normalized homogeneous coordinates of all 3D points and planes form the unit sphere S^3 in $\mathbb{R}^4$.

5.9 Canonical Elements of Coordinate Systems

We now want to give an example of the use of homogeneous representations. We discuss the canonical elements of coordinate systems in 2D and 3D. Canonical geometric elements of a coordinate system are its origin, its axes, and the points at infinity in the direction of the axes. In 2D, we additionally have the line at infinity. In 3D, we additionally have the coordinate planes, the plane at infinity, and the lines at infinity in the coordinate planes.

These canonical elements turn out to be interpretations of the basic unit vectors $\boldsymbol{e}_i^{[n]}$ in $\mathbb{R}^3$, $\mathbb{R}^4$ and $\mathbb{R}^6$. In Sect. 7.3.2, p. 312, we will use them for interpreting the columns and rows of the matrices representing linear mappings and for finding representative points on lines and planes.

5.9.1 Visualization of the Projective Plane Using the Stereographic Projection

We already visualized some of these elements on the celestial sphere in Fig. 5.10, p. 211. There we mapped the projective space $\mathbb{P}^3$ into/onto the unit ball $B^3 = \{\boldsymbol{X}| \; |\boldsymbol{X}| \leq 1\}$ and could visualize the zenith and the horizon. We now make this mapping explicit.

We start from the spherically normalized $n+1$-vectors $\mathbf{x}^s$ and map them via a stereographic projection from $[\mathbf{0}_n^\mathsf{T}, -1]^\mathsf{T}$ onto the hyperplane $x_h = 0$, interpreted as the projective space $\mathbb{P}^n$ containing the result of the stereographic projection.

In 2D (cf. Fig. 5.26), we take the south pole i.e., the point $\mathcal{S}([0,0,-1]^\mathsf{T})$ on the unit sphere, as the centre of the stereographic projection and map the spherically normalized points, $\mathbf{x}^s$ into the equator plane, i.e., the (uv)-plane. This leads to a point $\mathcal{X}^\sigma$ with in-

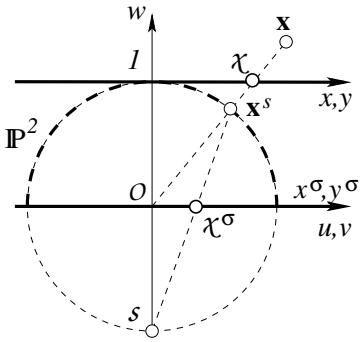


Fig. 5.26 Stereographic mapping of the projective plane for visualization of oriented entities, cf. Chap. 9, p. 343. Mapping the unit sphere S^2 representing $\mathbb{P}^2$ onto the unit disc in the equator plane using a stereographic projection. Each point $\mathcal{X}$ is mapped to $\mathcal{X}^\sigma$ via the spherically normalized point $\mathbf{x}^s$ on the upper half of the unit sphere, seen from the south pole s of the unit sphere. We will come back to this visualization in Sect. 9.4, p. 347 on oriented projective geometry

homogeneous coordinates $\boldsymbol{x}^\sigma = [x^\sigma, y^\sigma]$ in a (x^σ, y^σ)-coordinate system, which is identical to the (uv)-System. All points $\mathcal{X} \in \mathbb{R}^2$ are mapped onto the open unit disc, i.e., the two-dimensional ball without its boundary. The horizon is mapped to the equator, a unit circle S^1. Observe, the vectors $\mathbf{x}$ and $\mathbf{y} = -\mathbf{x}$, which represent the same 2D point, stereographically map to two different points, ${}^s\mathbf{x}$ and ${}^s\mathbf{y}$. This is why we can use the stereographic projection to visualize oriented entities, cf. Chap. 9, p. 343. Exercise 5.11

In 3D, we therefore obtain a mapping from $\mathbb{R}^3$ onto the open unit ball B^3 in a 3D coordinate system $(X^\sigma, Y^\sigma, Z^\sigma)$, whereas the plane at infinity is mapped to the surface of S^2. The formalization is shown in Example 11 for the 2D case.

3D lines cannot directly be visualized this way, as they lie on a four-dimensional subspace of the five-dimensional unit sphere S^5 in six-dimensional space $\mathbb{R}^6$; the subspace is defined by the Plücker constraint.

5.9.2 Elements of a 2D Coordinate System

The canonical elements of a 2D coordinate system can be described by its axes $\mathcal{l}_x$ and $\mathcal{l}_y$, seen as straight lines $\mathcal{l}$, and its origin $\mathcal{X}_O$,

$$\mathbf{l}_x = \boldsymbol{e}_2^{[3]}, \qquad \mathbf{l}_y = \boldsymbol{e}_1^{[3]}, \qquad \mathbf{x}_O = \boldsymbol{e}_3^{[3]}. \tag{5.173}$$

The dual interpretation of these vectors leads to the directions $\mathcal{X}_{\infty x}$ and $\mathcal{X}_{\infty y}$ of the x and the y axes and the horizon, i.e., the line $\mathcal{l}_\infty$ at infinity

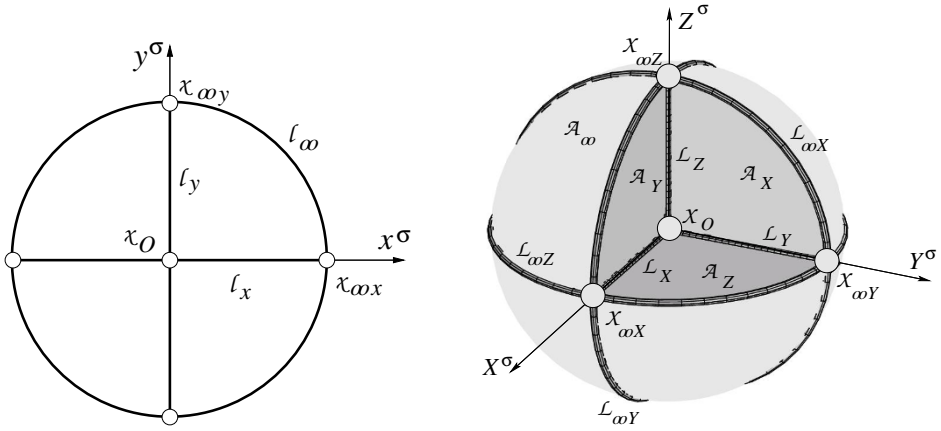


Fig. 5.27 Entities of 2D and 3D coordinate systems, including the origin, coordinate axes, coordinate planes, and the corresponding entities at infinity. They are shown on the unit disk and in the unit sphere, respectively. **Left:** the elements of the 2D coordinate system. The circle is a visualization of the line at infinity or the horizon of the plane with real points lying in the interior of the sphere. **Right:** the elements of the 3D coordinate system. The sphere represents the plane at infinity with the entities at infinity on it

$$\mathbf{x}_{\infty x} = \boldsymbol{e}_1^{[3]}, \qquad \mathbf{x}_{\infty y} = \boldsymbol{e}_2^{[3]}, \qquad \mathbf{l}_\infty = \boldsymbol{e}_3^{[3]}. \tag{5.174}$$

Using the spherical representation, we now visualize these entities in Fig. 5.27.

Obviously, the canonical elements are interpretations of the unit vectors in $\mathbb{R}^3$.

5.9.3 Elements of a 3D Coordinate System

The 3D coordinate system is now described, again using homogeneous coordinates. It is composed of its principal planes, $\mathcal{A}_X$, $\mathcal{A}_Y$, $\mathcal{A}_Z$, perpendicular to the axes $\mathcal{L}_X$, $\mathcal{L}_Y$, $\mathcal{L}_Z$, and its origin $\mathcal{X}_0$. For these entities, we have the coordinates

$$\mathbf{A}_X = \boldsymbol{e}_1^{[4]}, \qquad \mathbf{A}_Y = \boldsymbol{e}_2^{[4]}, \qquad \mathbf{A}_Z = \boldsymbol{e}_3^{[4]}, \qquad \mathbf{X}_O = \boldsymbol{e}_4^{[4]}, \tag{5.175}$$

and in the parameter space of lines (not shown in Fig. 5.27),

$$\mathbf{L}_X = \boldsymbol{e}_1^{[6]}, \qquad \mathbf{L}_Y = \boldsymbol{e}_2^{[6]}, \qquad \mathbf{L}_Z = \boldsymbol{e}_3^{[6]}. \tag{5.176}$$

The interpretation of the other unit vectors leads to points, lines, and planes at infinity, namely the three points at infinity in the direction of the three axes and the plane at infinity,

$$\mathbf{X}_{\infty X} = \boldsymbol{e}_1^{[4]}, \qquad \mathbf{X}_{\infty Y} = \boldsymbol{e}_2^{[4]}, \qquad \mathbf{X}_{\infty Z} = \boldsymbol{e}_3^{[4]}, \qquad \mathbf{A}_\infty = \boldsymbol{e}_4^{[4]}, \tag{5.177}$$

and the lines at infinity that are the horizons of the three coordinate planes,

$$\mathbf{L}_{\infty X} = \boldsymbol{e}_4^{[6]}, \qquad \mathbf{L}_{\infty Y} = \boldsymbol{e}_5^{[6]}, \qquad \mathbf{L}_{\infty Z} = \boldsymbol{e}_6^{[6]}. \tag{5.178}$$

Obviously, these entities at infinity, (5.177) and (5.178), are dual to the real entities, first mentioned in (5.175) and (5.176), respectively.

Also, the interpretations of the unit vectors in $\mathbb{R}^4$ and $\mathbb{R}^6$ here lead to the canonical elements of a 3D coordinate system.

5.10 Exercises

Basics

1. (2) Using Cramer's determinant rule for solving 2×2 equation systems, show that the intersection point $\mathcal{X}$ of two lines $\mathcal{l}_i([a_{i1}, a_{i2}, a_{i3}]^\mathsf{T}), i = 1, 2$, is given by (5.1.2.4), p. 201. What happens if the two lines are parallel? What happens if they are identical? Does (5.14) lead to a meaningful result?
2. (1) Show that three points on the line at infinity are collinear; thus, the line at infinity is a straight line. *Hint:* Determine whether the third point lies on the line passing through the first two.
3. (2) Verify the numbers in Fig. 5.7, p. 208.
4. (2) Write the mapping $x' = 1/x$ of points on the projective line with homogeneous coordinates. Write them in the form $\mathbf{x}' = \mathsf{H}\mathbf{x}$ with a suitable 2×2 matrix. What result for $\mathbf{x}'$ do you obtain for $x = 0$?
5. (1) a: What happens to the coordinates of a plane through three points when you exchange them cyclically? b: What happens to the volume of a tetrahedron if you exchange its four points cyclically?
6. (3) Find the definition of the Plücker coordinates in the following sources: Faugeras (1993), Hartley and Zisserman (2000), Stolfi (1991), Pottmann and Wallner (2010), the introduction by Jones (2000) and Wikipedia; compare the representations of the Plücker coordinates of the 3D line as a sequence of the g_{ij} of the Plücker matrix $\mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T} = [g_{ij}]$ as a permutation of the elements of $\mathbf{L}$ in (7.38), p. 301. Discuss the differences. What effect do the representations have on the ease of remembering the Plücker constraint?
7. (2) Take the geographic coordinates of your home city and describe the path of the sun during the equinox (i.e., on March 21 or September 21) as a line at infinity, assuming the sun lies on the celestial sphere. Use a local coordinate system with the X-axis pointing towards south, the Y-axis pointing east, and the Z-axis pointing towards the zenith.
8. (1) Given the line $\mathcal{l}$, its dual point $\mathcal{X}$, and $\mathcal{l}_x$, the polar of $\mathcal{X}$ w.r.t. the unit circle, show that $\mathcal{l}_x$and $\mathcal{l}$ are related by reflection at the origin.
9. (2) Given a conic $\mathcal{C}(\mathsf{C})$ and two points $\mathcal{y}$ and $\mathcal{z}$ on the conic, show that the intersection point $\mathcal{X} = \mathcal{l}_y \cap \mathcal{l}_z$ of the tangents in $\mathcal{y}$ and $\mathcal{z}$ is the pole of the line $\mathcal{l} = \mathcal{y} \wedge \mathcal{z}$ joining the two points.
10. (3) The following exercise is to show the generality of the polarity relation at a conic, especially if the polar line does not intersect the conic.
    Refer to Fig. 5.23, p. 239. Given the line $\mathcal{l}_3([1, 1, -2])$ and the ellipse $\mathcal{C}(2x^2+y^2-1 = 0)$, determine the coordinates of the intersection points $\mathcal{y}_1$ and $\mathcal{y}_2$. Allow complex numbers. Discuss the result of the previous exercise for points with complex numbers.
    Use the intersection points $\mathcal{y}_1$ and $\mathcal{y}_2$ and determine $\mathcal{x}_3$ as the intersection of the two tangent lines, the line $\mathcal{l}_3$ as the join of the two tangent points and $\mathcal{l}_1$ as the join of $\mathcal{x}_3$ with the point at infinity of the line $\mathcal{l}_3$. Determine $\mathcal{x}_3$. Check whether $\mathcal{x}_3$ is the pole of $\mathcal{l}_3$.
    Give a simple rule for how to geometrically construct (1) the polar of a point $\mathcal{X}$ inside an ellipse and (2) the pole of a line $\mathcal{l}$ not intersecting the ellipse.
11. (3) Refer to Fig. 5.26, p. 243:

    a. Show, that using homogeneous coordinates the relation between a point $\mathcal{X}$ with $\mathbf{x} = [u, v, w]^\mathsf{T}$ and its image $\mathcal{X}^\sigma$ with stereographic coordinates $\mathbf{x}^\sigma = [u^\sigma, v^\sigma, w^\sigma]^\mathsf{T}$ is given by the mapping

$$u^\sigma = u \qquad v^\sigma = v \qquad w^\sigma = w + \sqrt{u^2 + v^2 + w^2}. \tag{5.179}$$

b. Where do the points $\pm \boldsymbol{e}_i^{[3]}, i = 1, 2, 3$ map to?
c. Let $\mathcal{X}$ be a point with inhomogeneous coordinates $\boldsymbol{x} = [x, y]^\mathsf{T}$, which may be represented by $\mathbf{x}_+ = [x, y, 1]^\mathsf{T}$ or by $\mathbf{x}_- = [-x, -y, -1]^\mathsf{T}$. Where do the points $\mathbf{x}_+^\sigma$ and $\mathbf{x}_-^\sigma$ lie? What is the product of their signed distances from the origin?
d. Is there a point $\mathcal{X}$ which maps to $\mathbf{x}^\sigma = [0, 0, 0]^\mathsf{T}$? What are its homogeneous and inhomogeneous coordinates?
e. What is the image $\mathcal{l}^\sigma$ of a line $\mathcal{l}$?
f. Generalize the result to 3D and show that the meridian at Greenwich, i.e., the line at infinity in the XZ-plane, is mapped to a unit circle in the XZ-plane.

12. (1) Given is a 3D line by a point $\mathcal{X}(\boldsymbol{X})$ and the direction $\boldsymbol{D}$. What are its Plücker coordinates?

Proofs

13. (2) Using Fig. 5.2, right, p. 200, show geometrically that the w-coordinate of the vector $\mathbf{l}^e$ is identical to the distance of the line $\mathcal{l}$ to the origin. *Hint:* Compare the two triangles $(O_2 z_l O_3)$ and $(s\mathbf{l}^e O_3)$.
14. (1) Prove (5.19), p. 203.
15. (2) Prove (5.79), p. 220, using the relation between the vectors in a plane spanned by the two normals of the planes, see the Fig. 5.28. *Hint:* Assume the normal vectors $\boldsymbol{A}_h$ and $\boldsymbol{B}_h$ of the planes and the direction vector $\boldsymbol{L}_h$ of the line are normalized to 1, set $\boldsymbol{L}_0 = \alpha \boldsymbol{A}_h + \beta \boldsymbol{B}_h$ and determine α and β from the constraints $\boldsymbol{N} \cdot \boldsymbol{A}_h = |\boldsymbol{A}_h|$ and $\boldsymbol{N} \cdot \boldsymbol{B}_h = |\boldsymbol{B}_h|$.

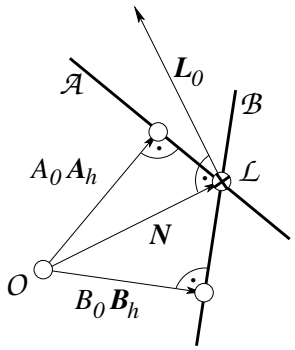


Fig. 5.28 3D line $\mathcal{L}$ seen in the direction $\boldsymbol{L}_h$ of the line and generated by intersection of the planes $\mathcal{A}$ and $\mathcal{B}$. The normal $\boldsymbol{N}$ to the line has the same length as $\boldsymbol{L}_0$

16. (3) Prove that (5.154) yields the tangent line $\mathcal{l}(\mathcal{X})$ by establishing the constraint that the two intersection points of a line $\mathcal{l}$ with a conic are identical.

Chapter 6
Transformations

6.1 Structure of Projective Collineations 248
6.2 Basic Transformations 250
6.3 Concatenation and Inversion of Transformations 261
6.4 Invariants of Projective Mappings 266
6.5 Perspective Collineations 277
6.6 Projective Correlations 282
6.7 Hierarchy of Projective Transformations and Their Characteristics 284
6.8 Normalizations of Transformations 285
6.9 Conditioning 286
6.10 Exercises 287

This chapter discusses transformations of geometric entities. The most general transformation we need in our context is that caused by imaging an object with a pinhole camera. It is straight line preserving – here neglecting distortions due to imperfection of the lens manufacturing. Such projective transformations can be represented by matrices. The projection itself is a matrix-vector multiplication of the homogeneous coordinates of the geometric entity. This simplifies concatenation and inversion, simply reducing to multiplying and inverting matrices. Special cases of a projective mapping are affinities, similarities, motions, rotations and translations, which are omnipresent in the processing of geometric entities. We discuss the representation of these transformations in 2D and 3D and analyse invariant geometric entities or properties. The mapping from 3D object space to 2D image space, discussed in Part III, exploits the results derived in this chapter.

We often translate or rotate coordinate systems, leading to changes in the homogeneous coordinates of points, lines, and planes. Changing points in a fixed reference system may also be of concern. Affine or projective transformations are essential for modelling cameras. Predicting for a given image point the line locus of a corresponding point in two-view analysis (epipolar line) supports considerably stereo matching procedures. In pursuing all these transformations, and also for the concatenation of transformations, their inversion, and the mapping of entities at infinity, we make substantial use of homogeneous representations. The term "transformation", meaning projective transformation, has been given many names in the literature – and also in this book – each stressing one of its specific properties: projective mapping, projectivity or projective collineation, homography, or simply DLT for *direct linear transformation*.

direct linear transformation

We distinguish between different categories of projective mappings:

- *Collineations* generally map points or hyperplanes to points or hyperplanes; they are straight line preserving and characteristically have what is called the *cross ratio* as invariant. A very prominent example is the 2D collineation mapping a plane onto a plane, e.g., when taking an image of a planar object with a pinhole camera. Among collineations, rotations require special treatment due to the diversity of their representations. These will be discussed in Chap. 8.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_6

- *Correlations* map points to hyperplanes, e.g., points in 2D to 2D lines and in 3D to planes.[1] Correlations also preserve the cross ratio. The most prominent example is the above-mentioned determination of what is called the *epipolar line* when observing a 3D scene with two pinhole cameras. Given the image point of a scene point in one image, the epipolar line is the locus for the image point in the other image. Projective correlations are discussed in Sect. 6.6.
- Collineations may be *regular* or *singular*. Regular collineations, such as the above-mentioned 2D collineation, are frequently used to map points, lines, or planes into a space of the same dimension. These mappings are one-to-one, thus can be inverted.
  Singular collineations occur when modelling cameras, since usually, a higher-dimensional space is mapped onto a lower-dimensional space. Naturally, no inversion is possible. Singular collineations appear when modelling the geometry of perspective cameras in Sect. 12.1.3.7.
- When categorizing collineations, it is necessary to distinguish whether the two spaces, which are related by the transformation, are treated as independent or whether they are embedded into a common, possibly higher-dimensional space; in other words, the domain and the range of a collineation may be treated as *independent*, may be *embedded* into a higher dimensional space or may be an *identical space*. *Perspective collineations* are characterized by the existence of what is called a *projection centre* so that object point, image point, and projection centre are collinear. Only if the perspective collineation is a mapping of one space onto itself, called *autocollineation*, do fixed elements (fixed points, lines, or planes) exist geometrically. Invariants of collineations are the topic of Sect. 6.4.
  Perspective mappings can be regarded as a special case of more general, not straight line preserving mappings with a projection centre, where the two spaces, especially the image space, need not to be a projective space. More about these mappings can be found in Part III in the context of modelling cameras.
- The regular projective mappings form a *Lie group*, i.e., a differentiable group. Their specializations, such as the motion or the perspectivity mappings, also form Lie groups. These specializations can be represented by fewer parameters, thus need fewer correspondences for their determination and therefore allow a more stable estimation when searching for outliers, given the same number of observations. Sect. 6.7 presents the hierarchy of projective mappings and their subgroups.
- Finally, we will distinguish between transformations which preserve orientation of an entity and those which reverse orientation. We will find that projective collineations are not orientation preserving, in contrast to affine collineations. Therefore, we introduce the concept of quasi-affine collineations, which are orientation preserving in a restricted domain. This will be discussed in Chap. 9.

6.1 Structure of Projective Collineations

We first discuss collineations, i.e., spatial transformations which preserve straight lines. Their basic definition refers to coordinates of points.

Definition 6.1.12: ***Projective collineation.*** Given two projective spaces $\mathbb{P}^n$ and $\mathbb{P}^m$. A mapping $\mathcal{H} : \mathbb{P}^n \to \mathbb{P}^m$ is called a collineation or a homography if it preserves collinearity of three points. ◇

Collineations may compactly be written with homogeneous coordinates as matrix vector multiplication.

[1] The notion "correlation" is not to be confused with the one in image processing or statistics.

Theorem 6.1.1: ***Representation of projective collineation.*** A projective collineation is a linear mapping of the homogeneous coordinates

$$\mathcal{H} : \mathbb{P}^m \to \mathbb{P}^n \qquad \mathbf{x} \mapsto \mathsf{H}\mathbf{x} \qquad \underset{(n+1)\times 1}{\mathbf{x}'} = \underset{(n+1)\times(m+1)}{\mathsf{H}} \underset{(m+1)\times 1}{\mathbf{x}} . \tag{6.1}$$

This mapping is also called a *direct linear transformation* (DLT). It has $(n+1)(m+1)-1$ degrees of freedom

direct linear transformation (DLT)

We denote homographies by $\mathcal{H}$ and their transformation matrices with H. The transformation matrix is homogeneous: its scaling leads to a scaling of $\mathbf{x}'$ which does not change the transformed point. Thus the matrices H and $\lambda\mathsf{H}$ with $\lambda \neq 0$ represent the same transformation, and therefore H is homogeneous. In oriented projective geometry, $\lambda > 0$ is required for two matrices to represent the same oriented mapping (cf. Chap. 9).

When mapping a projective plane $\mathbb{P}^2$ to another plane $\mathbb{P}^2$ or to itself, the transformation matrix is a homogeneous 3×3 matrix. When mapping a projective space $\mathbb{P}^3$ to another projective space $\mathbb{P}^3$, it is a homogeneous 4×4 matrix. Analogously, a mapping of the projective line $\mathbb{P}^1$ to another projective line $\mathbb{P}^1$ onto itself is represented by a 2×2 matrix H. When modelling cameras, we will have the case $n < m$ ($n = 2$, $m = 3$ and $n = 1$, $m = 2$), where one dimension is lost during the projection. The case $n > m$ occurs when backprojecting points and lines into 3D space, which leads to projection lines and projection planes.

It is easy to prove that if H is regular, the mapping (6.1) actually is straight line-preserving, thus a projective collineation. We prove this for $n = 2$: Let us assume $\mathcal{H} = \mathbb{P}^2 \to \mathbb{P}^2$ is represented by a regular 3×3 matrix and three collinear points $\mathcal{X}_i, i = 1, 2, 3$ are given. Then a threefold application of (6.1) yields a compact representation with the coordinate vectors aggregated to 3×3 matrices,

$$[\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3] = \mathsf{H}[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3] . \tag{6.2}$$

Taking determinants we see that if the given three points are collinear, their determinant $|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3|$ is 0, and then also the determinant $|\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3|$ of the mapped points is 0; thus they are also collinear. The proof generalizes to general $\mathbb{P}^n, n \geq 2$, as in that case points sitting on hyperplanes map to points on hyperplanes. Thus straight lines, which are the intersection of hyperplanes, are preserved. It can be shown that singular mappings and mappings with $m \neq n$ are also straight line preserving, as they can be decomposed into regular mappings and straight line-preserving projections onto hyperplanes. Exercise 6.5

Since a homography is a linear mapping, it is continuous, preserves incidences, is differentiable, and preserves tangents at curves. We will get to additional properties of regular collineations when discussing invariants in Sect. 6.4, p. 266.

The mapping (6.1) allows two distinct interpretations, see Fig. 6.1:

1. The point $\mathcal{X}$ is displaced within a fixed given coordinate system; thus, changing its position, namely from $\mathbf{x}$ to $\mathbf{x}'$, results in a different point $\mathcal{X}'(\mathbf{x}')$. Special cases are translations, rotations, general motions of a point, or – in the case of imaging – the mapping of a point from object to image space.
2. The coordinate system is displaced, i.e., changed such that the same point $\mathcal{X}$ has coordinates $\mathbf{x}$ in one system and $\mathbf{x}'$ in the other, thus $\mathcal{X}(\mathbf{x}) = \mathcal{X}(\mathbf{x}')$. Classical coordinate transformations may again be translations, rotations or motions, but also affine or projective transformations of the coordinate system.

In the following chapters we will introduce *transformations as displacements* of points within a fixed given reference system, thus following item 1. The meaning of the other transformations is later made explicit in the context.

transformations are displacements

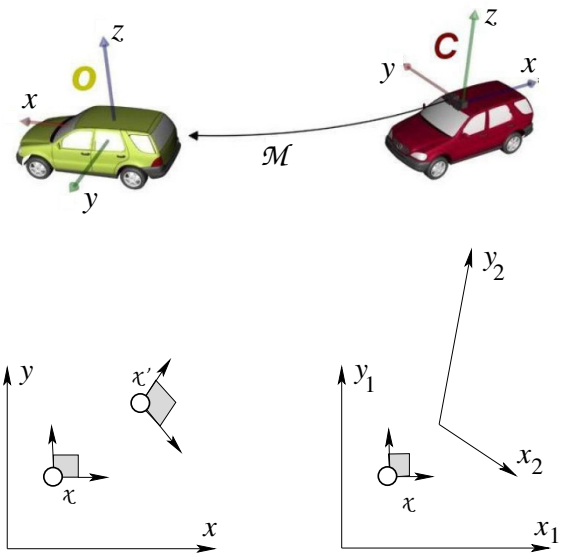


Fig. 6.1 Interpretation of transformations. **Top:** Two moving cars, car C taken as reference and car O taken as object, with relative motion M and differently fixed coordinate systems S_O and S_C. We may describe the relative motion in either coordinate system. Seen from C, the car O turns right and is at a different position. When seen from O, subjectively treated as fixed, the reference system C is moving backwards. The two motions are inverse to each other. Observe, the coordinate systems of the two cars need not be in the same relation to the car, e.g., when being from two different producers. **Bottom left:** The point $\mathcal{X}(x, y)$ together with its own coordinate system is displaced to $\mathcal{X}'(x', y')$ which is expressed in the fixed reference frame, thus $\mathsf{H} : \mathcal{X}(x, y) \to \mathcal{X}'(x', y')$. The displacement may contain a shear of the coordinate axes. **Bottom right:** The point $\mathcal{X}(x_1, y_1)$ together with its own coordinate system is represented in the displaced reference frame $\mathcal{X}(x_2, y_2)$, thus $\mathsf{H} : \mathcal{X}(x_1, y_1) \to \mathcal{X}(x_2, y_2)$

6.2 Basic Transformations

6.2.1 Transformation of 2D Points 250
6.2.2 Transformation of 3D Points 255
6.2.3 Transformation of 1D Points 257
6.2.4 Transformation of Lines and Hyperplanes 258
6.2.5 Transformation of Conics and Quadrics 260
6.2.6 Summary of Transformations 260

We consider the classification of projective collineations of points first in 2D, and then in 3D and 1D, all defined as displacements of points. Taking these relationships as a basis, we subsequently transform other geometric entities, namely lines and hyperplanes, and finally conics and quadrics.

6.2.1 Transformation of 2D Points

We discuss the transformation of 3-vectors, representing the homogeneous coordinates of 2D points, and show specializations. The most general *projective mapping* in 2D is

$$\mathcal{H}(\mathsf{H}): \quad \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \boldsymbol{x}'_0 \\ x'_h \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & 1/\lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix}. \tag{6.3}$$

The number of free parameters, i.e., the number of degrees of freedom of this transformation, is 8, as the mapping is homogeneous and the scaling can be chosen arbitrarily.

Before we give an example of this type of transformation, we present the most important special cases of this general mapping:

- *Translation* $\mathcal{T}(\boldsymbol{t})$ with two parameters $\boldsymbol{t} = [t_x, t_y]^\mathsf{T}$:

$$\mathcal{T}(\boldsymbol{t}): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{6.4}$$

  or compactly

$$\mathcal{T}(\boldsymbol{t}): \quad \mathbf{x}' = \mathsf{T}\mathbf{x} \quad \text{with} \quad \mathsf{T} = \begin{bmatrix} \mathsf{I}_2 & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.5}$$

  It can easily be shown that scaling the matrix T with an arbitrary scalar $\lambda \neq 0$ does not change the mapping. Exercise 6.2

- *Mirroring* at the y-axis:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{6.6}$$

  or, compactly,

$$\mathbf{x}' = \mathsf{Z}\mathbf{x} \quad \text{with} \quad \mathsf{Z} = \begin{bmatrix} \mathsf{Z} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.7}$$

- *Rotation* $\mathcal{R}(\varphi)$ around the origin with one parameter φ:

$$\mathcal{R}(\varphi): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{6.8}$$

  or, compactly,

$$\mathcal{R}(\varphi): \quad \mathbf{x}' = \mathsf{R}\mathbf{x} \quad \text{with} \quad \mathsf{R} = \begin{bmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.9}$$

  Here R is an orthonormal matrix with determinant $+1$.
  We will discuss rotations in more detail later (cf. Chap. 8).

- *Planar motion* or congruency $\mathcal{M}(\boldsymbol{t}, \varphi)$ with three parameters, consisting of rotation φ and translation $\boldsymbol{t}$:

$$\mathcal{M}(\boldsymbol{t}, \varphi): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{6.10}$$

  or, compactly,

$$\mathcal{M}(\boldsymbol{t}, \varphi): \quad \mathbf{x}' = \mathsf{M}\mathbf{x} \qquad \text{with} \quad \begin{bmatrix} \cos\varphi & -\sin\varphi & t_x \\ \sin\varphi & \cos\varphi & t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.11}$$

- *Common scaling* $\mathcal{D}(\lambda)$ with a parameter λ:

$$\mathcal{D}(\lambda): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}, \tag{6.12}$$

or, compactly,

$$\mathcal{D}(\lambda): \quad \mathbf{x}' = \mathsf{D}\mathbf{x} \quad \text{with} \quad \mathsf{D} = \begin{bmatrix} \lambda I_2 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{6.13}$$

For $\lambda > 1$, we obtain a magnification, for $0 < \lambda < 1$ a reduction. For $\lambda < 0$, in addition to the scaling we obtain a mirroring at the origin, thus a rotation by 180° around the origin.

- *Similarity transformation* $\mathcal{S}(a, b, c, d)$ with four parameters:

$$\mathcal{S}(a, b, c, d): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} , \tag{6.14}$$

or, compactly,

$$\mathcal{S}(a, b, c, d): \quad \mathbf{x}' = \mathsf{S}\mathbf{x} \quad \text{with} \quad \mathsf{S} = \begin{bmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda R & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{6.15}$$

Here $a = \lambda \cos\varphi$ and $b = \lambda \sin\varphi$ contain the components of rotation φ and scaling λ; the parameters c and d are the translation parameters. The transformation matrix S should not be confused with the skew symmetric matrix $\mathsf{S}(\mathbf{x}) = \mathsf{S}_x$ (cf. Chap. 8, p. 325).

- *Scale difference between x and y coordinates* with a parameter m:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 - m/2 & 0 \\ 0 & 1 + m/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} , \tag{6.16}$$

or, with homogeneous coordinates,

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} 1 - m/2 & 0 & 0 \\ 0 & 1 + m/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} . \tag{6.17}$$

- *Shear of axes symmetrically* with a parameter s:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & s/2 \\ s/2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} , \tag{6.18}$$

or, with homogeneous coordinates,

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & s/2 & 0 \\ s/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} . \tag{6.19}$$

- *Asymmetric shear* with parameter s':

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & s' & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} , \tag{6.20}$$

resulting in an upper triangular matrix.

- *Affine transformation* $\mathcal{A}(\mathsf{A})$ with six parameters, two translations, one rotation, two individual scalings, and one shear of the axes:

$$\mathcal{A}(\mathsf{A}): \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} , \tag{6.21}$$

or, compactly,

$$\mathcal{A}(\mathsf{A}): \quad \mathbf{x}' = \mathsf{A}\mathbf{x} \quad \text{with} \quad \mathsf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.22}$$

The parameters a_{11}, a_{12}, a_{21}, and a_{22} are responsible for the rotation, the scalings, and the shear; the parameters a_{13} and a_{23} for the translation.

We will avoid the name $\mathcal{A}$ for affinities as the same name is used for planes.

- *Projective transformation* $\mathcal{H}(\mathsf{H})$ depending on a general 3×3 matrix, due to homogeneity, having only 8 degrees of freedom:

$$\mathcal{H}(\mathsf{H}): \quad \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \boldsymbol{x}'_0 \\ x'_h \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & 1/\lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix}. \tag{6.23}$$

In addition to the parameters of the affine transformation, we have the parameters g and h which cause the typical effects of a projective transformation, namely mapping parallel lines to converging lines (Table 6.1). This projective transformation will be extensively applied in the analysis of one and two views.

It can easily be verified that the more general transformation matrices can be composed of some of the simpler transformation matrices by multiplication. For example, the affine transformation matrix can be generated from the product of the matrices (6.5), (6.9), (6.13), (6.17), and (6.20). However, these compositions are not unique and can be flexibly tailored to the particular application.

Table 6.1 collects all these transformations and shows their effects on the unit square.

Example 6.2.17: ***Pinhole camera.*** This is an example of a homography showing the interpretation of parameters in the last row of its matrix H. It also demonstrates that a plane-to-plane mapping with a pinhole camera can be written as a homography.

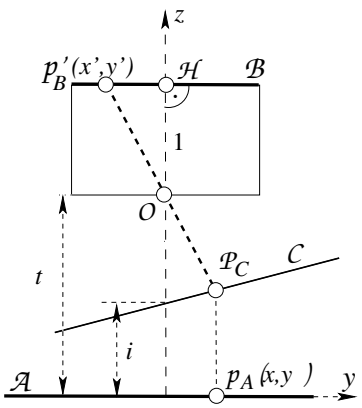
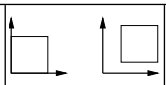
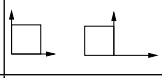
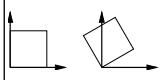
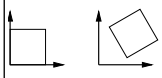
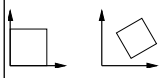
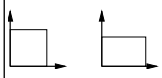
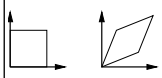
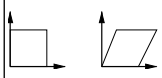
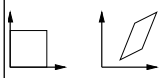
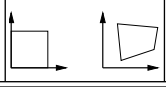


Fig. 6.2 Example for a homography using a pinhole camera (with its principal point $\mathcal{H}$, cf. Fig. 12.2, p. 458). The 3D point $\mathcal{P}_C$ lying on the plane $\mathcal{C}$ is mapped to the point $p'_B(x', y')$ on the image plane $\mathcal{B}$ of a pinhole camera. The coordinates of the 3D point depend on the 2D coordinates of the point $p_A(x, y)$ on the reference plane $\mathcal{A}$. We obtain a straight line-preserving mapping $\mathcal{H} : \mathcal{A} \to \mathcal{B}$ from the reference plane $\mathcal{A}$ to the image plane $\mathcal{B}$. The point $\mathcal{H}$ is the centre of the coordinate system in the image

A pinhole camera maps 3D space onto a 2D space, both in one common coordinate system. Here we want to show that a mapping of a reference plane $\mathcal{A}$ to the image plane $\mathcal{B}$ via a tilted plane $\mathcal{C}$ yields a homography where the parameters of the plane will show up in the last row of matrix H, cf. Fig. 6.2.

The reference plane $\mathcal{A}$ containing the point $p(x, y)$ is the xy-plane of a 3D coordinate system. The tilted plane $\mathcal{C}$ is assumed to be $z = cx+fy+i$. Points $\mathcal{P}_C$ on $\mathcal{C}$ thus have coordinates $[x, y, cx+fy+i]^\mathsf{T}$. They are projected from $\mathcal{A}$ by a simple orthographic transformation, a specialization of the affine transformation.

Table 6.1 Planar straight line-preserving transformations of points, effect onto a square, number of free parameters (degrees of freedom) and transformation matrices

2D Transformation	Figure	d.o.f.	H	H
Translation		2	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} I & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Mirroring at y-axis		0	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Rotation		1	$\begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Motion		3	$\begin{bmatrix} \cos\varphi & -\sin\varphi & t_x \\ \sin\varphi & \cos\varphi & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Similarity		4	$\begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \lambda R & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Scale difference		1	$\begin{bmatrix} 1+m/2 & 0 & 0 \\ 0 & 1-m/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} D & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Shear		1	$\begin{bmatrix} 1 & s/2 & 0 \\ s/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Asym. shear		1	$\begin{bmatrix} 1 & s' & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} S' & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Affinity		6	$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} A & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
Projectivity		8	$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$	$\begin{bmatrix} A & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & 1/\lambda \end{bmatrix}$

The pinhole camera with projection centre O at $[0, 0, t]^\mathsf{T}$ and focal length 1 looks downwards. The image point $p'_B(x', y')$ has coordinates

$$x' = \frac{x}{z-t} \qquad y' = \frac{y}{z-t} \tag{6.24}$$

on the plane $\mathcal{B}$, with

$$z = cx + fy + i\,. \tag{6.25}$$

The mapping $\mathcal{H} : \mathcal{A} \to \mathcal{B}$ therefore reads as (6.24) or

$$x' = \frac{x}{cx + fy + i - t} \qquad y' = \frac{y}{cx + fy + i - t}\,. \tag{6.26}$$

These relations are linear in the numerator and denominator, and the denominators are equal. Thus the mapping of the coordinates $[x, y]$ of the point p_A on the reference plane $\mathcal{A}$ to the coordinates $[x', y']$ of the point p'_B in the image plane $\mathcal{B}$ is a homography and is given by

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} x'w' \\ y'w' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & f & i-t \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\,. \tag{6.27}$$

This is an example with the off-diagonal elements in the last row of the mapping matrix containing values $\neq 0$. If $i - t = 1$, we obtain a homography with free parameters c and f only in the last row. ◇

Thus the example shows:

1. The parameters c and f are related to the slope of the plane $\mathcal{C}$.
2. The mapping from a plane in object space to the image plane via a pinhole camera is a homography.

6.2.2 Transformation of 3D Points

In analogy to 2D transformations, we obtain the transformations of 3D space; the transformation matrices now have size 4×4. Its general form is

$$\mathcal{H}(\mathsf{H}): \begin{bmatrix} U' \\ V' \\ W' \\ T' \end{bmatrix} = \begin{bmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ d & h & l & p \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \boldsymbol{X}'_0 \\ X'_h \end{bmatrix} = \begin{bmatrix} A & \boldsymbol{T} \\ \boldsymbol{P}^\mathsf{T} & 1/\lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_0 \\ X_h \end{bmatrix}. \tag{6.28}$$

This homogeneous mapping has 15 degrees of freedom. We again start with the simple cases.

- *Translation* $\mathcal{T}(\boldsymbol{T})$ with three parameters in $\boldsymbol{T}$:

$$\mathcal{T}(\boldsymbol{T}): \quad \boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{T}, \tag{6.29}$$

  or, in homogeneous coordinates,

$$\mathcal{T}(\boldsymbol{T}): \quad \mathbf{X}' = \mathsf{T}(\boldsymbol{T})\mathbf{X} \quad \text{with} \quad \mathsf{T}(\boldsymbol{T}) = \begin{bmatrix} I_3 & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.30}$$

  The homogeneous coordinate vectors $\mathbf{X}$ and $\mathbf{X}'$ need not be normalized, i.e., the last coordinate does not need to be $X_4 = 1$.
- *Rotation* $\mathcal{R}(R)$ with three independent parameters of a rotation matrix R:

$$\mathcal{R}(R): \quad \boldsymbol{X}' = R\boldsymbol{X}, \tag{6.31}$$

  or, in homogeneous coordinates,

$$\mathcal{R}(R): \quad \mathbf{X}' = \mathsf{R}\mathbf{X} \quad \text{with} \quad \mathsf{R} = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \tag{6.32}$$

  with some rotation matrix $RR^\mathsf{T} = I_3$, $R^\mathsf{T} = R^{-1}$. We will discuss rotation matrices in detail later in Chap. 8, p. 325.
- *Spatial motion* or *rigid body motion* $\mathcal{M}(R, \boldsymbol{T})$ with six parameters, translation and rotation:

$$\mathcal{M}(R, \boldsymbol{T}): \quad \boldsymbol{X}' = R\boldsymbol{X} + \boldsymbol{T}, \tag{6.33}$$

  or, in homogeneous coordinates,

$$\mathcal{M}(R, \boldsymbol{T}): \quad \mathbf{X}' = \mathsf{M}(R, \boldsymbol{T})\mathbf{X} \quad \text{with} \quad \mathsf{M}(R, \boldsymbol{T}) = \begin{bmatrix} R & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.34}$$

- *Spatial similarity transformation* $\mathcal{M}(R, \boldsymbol{T}, \lambda)$ with seven parameters, translation (3), rotation (3) and common scale (1):

$$\mathcal{M}(R, \boldsymbol{T}, \lambda): \quad \boldsymbol{X}' = \lambda R\boldsymbol{X} + \boldsymbol{T}, \tag{6.35}$$

  or, in homogeneous coordinates,

$$\mathcal{M}(R, \boldsymbol{T}, \lambda): \quad \mathbf{X}' = \mathsf{M}(R, \boldsymbol{T}, \lambda)\mathbf{X} \quad \text{with} \quad \mathsf{M}(R, \boldsymbol{T}, \lambda) = \begin{bmatrix} \lambda R & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.36}$$

- *Affine transformation* $\mathcal{A}(A, \boldsymbol{T})$ with 12 parameters, translation (3), rotation (3), three scales, and three shears:

$$\mathcal{A}(A, \boldsymbol{T}): \quad \begin{bmatrix} \boldsymbol{X}' \\ 1 \end{bmatrix} = \begin{bmatrix} A & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix}. \tag{6.37}$$

It can easily be shown that a pure 3D affinity requires six parameters. When there is no translation and no rotation, the three unit vectors may be scaled (λ_i) and pairwise symmetrically sheared ($[a, b, c]$):

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}' = \begin{bmatrix} \lambda_x & a & b & 0 \\ a & \lambda_y & c & 0 \\ b & c & \lambda_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \tag{6.38}$$

This representation with a symmetric matrix, called *Cauchy's strain tensor* (cf. Sokolnikov, 1956, p. 14), can be used for visualizing the affine properties of a general affinity since a general 3D affine mapping can easily be partitioned into this pure affine mapping and a 3D motion (6.34).

- *Projective transformation* $\mathcal{H}(\mathsf{H})$ with 15 parameters:

$$\mathcal{H}(\mathsf{H}): \quad \mathbf{X}' = \mathsf{H}\mathbf{X} \quad \text{with} \quad \mathsf{H} = \begin{bmatrix} \mathsf{A} & \boldsymbol{T} \\ \boldsymbol{P}_{\text{pr}}^{\mathsf{T}} & s \end{bmatrix}, \tag{6.39}$$

where vector $\boldsymbol{P}_{\text{pr}} \neq 0$ is typical for the projectivity. The number of free parameters is 15, since the transformation matrix is homogeneous.

The following examples show the practical relevance of 3D collineations.

Example 6.2.18: ***Projection with an ideal lens.*** This example of a 3D homography demonstrates the usefulness of projective geometry for modelling the mapping of thin lenses, since a sharp optical projection with a thin lens is a homography.

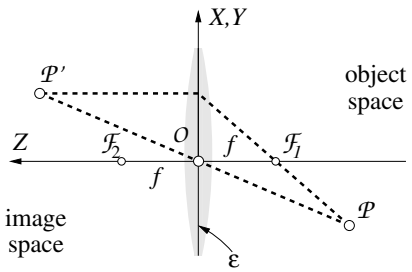


Fig. 6.3 Sharp optical projection with a thin lens. Rays through the focal point $\mathcal{F}_1$ are parallel to the optical axis in the image space. Together with the collinearity of $\mathcal{P}$, $\mathcal{P}'$ and the projection centre O, the mapping is straight line-preserving. The sign of f is negative here, as the coordinates of $\mathcal{F}_1$ are assumed to be $[0, 0, f]^{\mathsf{T}}$. The principal plane is ε

There is one coordinate system for both spaces (Fig. 6.3); thus, the mapping is an autocollineation. Let the origin of the coordinate system be in the centre of the lens, called O. Further, let the optical axis of the thin lens be the Z axis. The principal plane ε of the lens is the XY plane and the focal length is f. Although it is an optical constant, the scalar f is treated here as a coordinate with sign. The point $\mathcal{P}$ in object space is mapped to the point $\mathcal{P}'$ in image space, fulfilling the following two conditions:

1. The three points $\mathcal{P}$, $\mathcal{P}'$ and O are collinear.
2. Rays in object space passing through the focal point $\mathcal{F}_1$ bend at the principal plane ε and are parallel to the optical axis in image space.

From the figure, we directly obtain the relations:

$$X' = f\frac{X}{f-Z} \qquad Y' = f\frac{Y}{f-Z} \qquad Z' = f\frac{Z}{f-Z}\,. \tag{6.40}$$

This mapping is linear in the numerators and denominators, and has identical denominators; thus, it can be written as a homography:

$$\begin{bmatrix} U' \\ V' \\ W' \\ T' \end{bmatrix} = \begin{bmatrix} X'T' \\ Y'T' \\ Z'T' \\ T' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & -1 & f \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \tag{6.41}$$

This proves that the ideal optical projection is straight line-preserving. ◇

From the two examples (the mapping with the pinhole camera (6.27) and the mapping with the thin lens (6.41), we conclude that the pure projectivity is characterized by up to two or three parameters in the last row of (6.27) and (6.41), respectively.

There is yet another remarkable feature of the projection (6.41) with the ideal lens: points on the principal plane ε map to themselves; they are fixed points of this particular 3D homography. The existence of both the projection centre and the fixed plane specify this homography as a perspective collineation; for more detail, cf. Sect. 6.5, p. 277.

6.2.3 Transformation of 1D Points

The mapping of the projective line can easily be derived by specialization of (6.23), p. 253. We obtain

$$\mathcal{H}(\mathsf{H}): \quad \mathbb{P} \to \mathbb{P} \qquad \mathbf{x}' = \mathsf{H}\mathbf{x} \tag{6.42}$$

with a general 2×2 matrix H with three free parameters.

The two basic transformations, which do not contain a projective part, are translation and dilation (or scaling with the transformation matrices):

$$\mathsf{T} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \qquad \mathsf{D} = \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} . \tag{6.43}$$

The most important mapping of the projective line onto itself, not including translation or scaling, is the inversion $x' = 1/x$, which for the projective line reads as

$$I: \mathbb{P} \to \mathbb{P} \qquad \begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} . \tag{6.44}$$

The mapping is defined for all points on the projective line, as the origin $O([0,1]^\mathsf{T})$ is mapped to infinity $O'([1,0]^\mathsf{T})$ and vice versa. This is in contrast to the mapping $\mathbb{R} \setminus 0 \to \mathbb{R} : x \mapsto 1/x$, where the origin has no image.

Example 6.2.19: ***Inverse depth.*** If we want to represent points $\mathcal{X} \in \mathbb{P} \setminus 0$ (i.e., with the exception of the origin) with their distance from the origin, but want to include the points at infinity, then we may represent the points by the inverse distance $1/x \in \mathbb{R}$. When modelling 3D points in front of a camera, the property of this mapping is exploited and applied to the Z coordinate and is called *inverse depth-representation*. If the camera points in the Z-direction and the 3D coordinates are given in the coordinate system of the camera, the 3D transformation reads as

$$\begin{bmatrix} X' \\ Y' \\ I \\ 1 \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \\ 1/Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} X \\ Y \\ 1 \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \tag{6.45}$$

While the depth Z is mapped to the inverse depth $I = 1/Z$, the X and the Y coordinates are mapped to $\tan\alpha = X/Z$ and $\tan\beta = Y/Z$, i.e., the tangents of the horizontal and the vertical viewing angles α and β of the viewing ray, respectively. This mapping preserves the collinearity and coplanarity of points and is therefore of practical advantage (Montiel, 2006).[2] ◇

[2] We will encounter this mapping again when analysing the geometry of the camera pair: As the inverse depth is proportional to the parallax in a normal stereo image pair, representing the parallax image is equivalent to representing the inverse depth image.

6.2.4 Transformation of Lines and Hyperplanes

The transformation of hyperplanes, especially of 2D lines and 3D planes, is closely related to the transformation of points. As hyperplanes are dual to points, the transformation of hyperplanes sometimes is called *dual collineation*, cf. Ressl (2003, Sect. 6.2).

We derive the transformation without explicitly referring to the dimension, 2 or 3. Then the transformations for points and hyperplanes are related by the following proposition.

Proposition 6.2.1: ***Transformation of hyperplanes.*** If points $\mathbf{X} \in \mathbb{P}^n$ are transformed according to $\mathbf{X}' = \mathsf{H}\mathbf{X}$, hyperplanes $\mathbf{A} \in \mathbb{P}^{*n}$ transform according to

$$\mathcal{H}_A: \quad \mathbf{A}' = \mathsf{H}_A\mathbf{A} \quad \text{with} \quad \mathsf{H}_A = \mathsf{H}^\circ, \tag{6.46}$$

where H° denotes the cofactor matrix $\mathsf{H}^\circ = |\mathsf{H}|\,\mathsf{H}^{-\mathsf{T}}$, cf. App. (A.19), p. 769.

Exercise 6.20

The proof is left as an exercise. This general result (cf. also Browne, 2009, Eq. (2.30)) confirms the well-known relation in $\mathbb{P}^2$: When generating a line with two points via $\mathbf{l} = \mathbf{x} \times \mathbf{y}$ and transforming both points with (A.46), p. 772, we obtain the transformed line $\mathbf{l}' = \mathsf{H}^\circ\mathbf{l} = (\mathsf{H}\mathbf{x}) \times (\mathsf{H}\mathbf{y})$.

The mapping of hyperplanes with the cofactor matrix requires an explanation. In classical textbooks, such as Hartley and Zisserman (2000), the mapping of lines (hyperplanes in $\mathbb{P}^2$) is given by

$$\mathbf{l}' = \mathsf{H}^{-\mathsf{T}}\mathbf{l} \quad \text{or} \quad \mathsf{H}^\mathsf{T}\mathbf{l}' = \mathbf{l}. \tag{6.47}$$

This differs from our definition by the factor $|\mathsf{H}|$, which appears irrelevant, as the transformation matrix is homogeneous. The scale factor becomes relevant in two cases: (1) If we perform variance propagation, the homogeneous vector and its covariance matrix need to be consistent. This is only the case with the transformation matrix H°, not with $\mathsf{H}^{-\mathsf{T}}$. (2) If the determinant of H is negative, the orientation of the transformed line will change, cf. Sect. 9, p. 343. An important property of (6.46) is its validity if H is singular, which can be proven by using an adequate standardized point set. However, the transformation (6.47) is of advantage if the scale factor is not relevant, especially when estimating homographies from line correspondences using the second relation of (6.47) in the form $\mathsf{S}(\mathbf{l})\mathsf{H}^\mathsf{T}\mathbf{l}' = \mathbf{0}$, cf. (7.114), p. 315.

Example 6.2.20: ***Singular projection from $\mathbb{P}^3$ to a plane in $\mathbb{P}^3$.*** Let a point $\mathcal{X} \in \mathbb{P}^3$ be projectively mapped to a point $\mathcal{X}'$ in the plane $\mathcal{B}$, specifically the XY-plane with $\mathbf{B} = [0, 0, 1, 0]$, see Fig. 6.4.

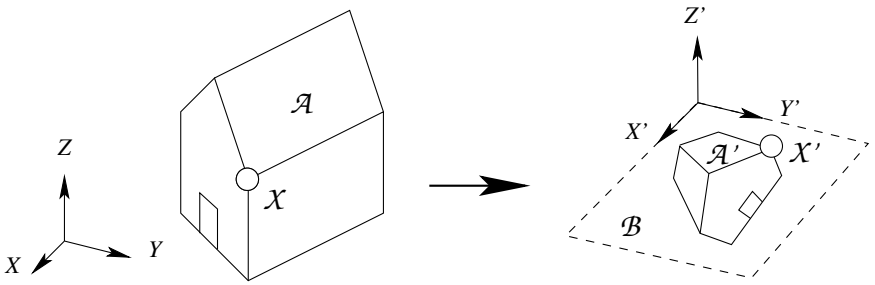


Fig. 6.4 Singular projective mapping from $\mathbb{P}^3$ to a plane $\mathcal{B}$ in $\mathbb{P}^3$, namely the XY-plane

This mapping has the general form

$$\mathbf{X}' = \begin{bmatrix} U' \\ V' \\ 0 \\ T' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ 0 & 0 & 0 & 0 \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1^\mathsf{T} \\ \mathbf{H}_2^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \\ \mathbf{H}_4^\mathsf{T} \end{bmatrix} \mathbf{X} = \mathsf{H}\mathbf{X}, \tag{6.48}$$

where we assume its rank is 3. The homography matrix H is singular and guarantees the transformed point to have Z'-coordinate 0. Mapping an arbitrary plane $\mathbf{A} = [A, B, C, D]^\mathsf{T}$ uses the cofactor matrix H°,

which is also defined for singular matrices, cf. (A.19), p. 769: As the third row of H is 0, all rows except the third row of H° are 0. Therefore we obtain

$$\mathbf{B} = \mathsf{H}^\circ \mathbf{A} = \begin{bmatrix} \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \\ \mathbf{Z}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \end{bmatrix} \mathbf{A} = \begin{bmatrix} 0 \\ 0 \\ C' \\ 0 \end{bmatrix}, \tag{6.49}$$

where $\mathbf{Z} = \mathbf{H}_1 \cap \mathbf{H}_2 \cap \mathbf{H}_4 \neq \mathbf{0}$ and $C' = \mathbf{Z}^\mathsf{T}\mathbf{A} \neq 0$. This confirms the coordinates $\mathbf{B}$ to be that of the XY-plane $\mathcal{B}$. The mapping (6.48) is the basis for modelling straight line-preserving cameras. ◇

We may interpret the relation between the transformations H and H° as mutually dual transformations,

$$\overline{\mathsf{H}} = \mathsf{H}^\circ . \tag{6.50}$$

Thus the cofactor matrix H° is the dual of H. This allows us, as an example, to infer $\mathbf{l}' = \overline{\mathsf{H}}\mathbf{l}$ from $\mathbf{x}' = \mathsf{H}\mathbf{x}$ using the duality principle.

We are now left with the transformation of 3D lines.

Proposition 6.2.2: ***Transformation of 3D lines.*** Given the 3D homography for points

$$\mathbf{X}' = \begin{bmatrix} \boldsymbol{X}'_0 \\ X'_h \end{bmatrix} = \begin{bmatrix} A & \boldsymbol{T} \\ \boldsymbol{P}^\mathsf{T} & s \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_0 \\ X_h \end{bmatrix} = \mathsf{H}\mathbf{X}, \tag{6.51}$$

a 3D line $\mathcal{L}$ is transformed according to *transformation of 3D points*

$$\mathbf{L}' = \mathsf{H}_L \, \mathbf{L}, \tag{6.52}$$

with the 6×6 line transformation matrix

$$\underset{6\times6}{\mathsf{H}_L} = \begin{bmatrix} sA - \boldsymbol{T}\boldsymbol{P}^\mathsf{T} & A\mathsf{S}^\mathsf{T}(\boldsymbol{P}) \\ \mathsf{S}(\boldsymbol{T})A & A^\circ \end{bmatrix}. \tag{6.53}$$

This can be proven by transforming the points $\mathbf{X}$ and $\mathbf{Y}$ generating the line $\mathbf{L} = \mathbf{X} \wedge \mathbf{Y}$. Observe, the line transformation matrix H_L is quadratic in the elements of H. Exercise 6.19

The motion matrix M_L for 3D lines follows from (6.53) by simplification using the spatial motion matrix for points (6.34):

$$\underset{6\times6}{\mathsf{M}_L} = \begin{bmatrix} R & \mathbf{0} \\ \mathsf{S}(\boldsymbol{T})R & R \end{bmatrix} = \begin{bmatrix} R & \mathbf{0} \\ -\mathsf{S}^\mathsf{T}(\boldsymbol{T})R & R \end{bmatrix}. \tag{6.54}$$

The second version of M_L is given due to its similarity to the matrix of a 3D motion of a plane,

$$\mathsf{M}^{-\mathsf{T}} = \begin{bmatrix} R & \mathbf{0} \\ -\boldsymbol{T}^\mathsf{T}R & 1 \end{bmatrix}. \tag{6.55}$$

If a 3D line $\mathcal{L}$ is represented by its Plücker matrix $\mathsf{\Gamma}(\mathbf{L})$, the Plücker matrix $\mathsf{\Gamma}(\mathbf{L}')$ of the transformed line $\mathcal{L}'$ is given by

$$\mathsf{\Gamma}(\mathbf{L}') = \mathsf{H}\mathsf{\Gamma}(\mathbf{L})\mathsf{H}^\mathsf{T}. \tag{6.56}$$

This can be seen directly when representing the Plücker matrix as $\mathsf{\Gamma}(\mathbf{L}') = \mathbf{X}'\mathbf{Y}'^\mathsf{T} - \mathbf{Y}'\mathbf{X}'^\mathsf{T}$ with some arbitrary distinct points $\mathcal{X}, \mathcal{Y} \in \mathcal{L}$ and substituting $\mathbf{X}' = \mathsf{H}\mathbf{X}$ and $\mathbf{Y}' = \mathsf{H}\mathbf{Y}$. The transformation is again quadratic in the elements of H.

6.2.5 Transformation of Conics and Quadrics

The representation of conics and quadrics has already been discussed in Sect. 5.7, p. 236. Given the conic $\mathcal{C}(\mathsf{C}) : \mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0$ and the mapping $\mathcal{H}$ of points $\mathbf{x}' = \mathsf{H}\mathbf{x}$, the transformed point conic $\mathcal{C}'(\mathsf{C}')$ is given by

$$\mathsf{C}' = \mathsf{H}^\circ \mathsf{C} \mathsf{H}^{\circ\mathsf{T}} \tag{6.57}$$

with (A.19), p. 769 the cofactor matrix $\mathsf{H}^\circ = |\mathsf{H}|\mathsf{H}^{-\mathsf{T}}$ if H is regular. This holds, since for every point $\mathcal{X} \in \mathcal{C}$,

$$\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = (\mathbf{x}'^\mathsf{T}\mathsf{H}^{-\mathsf{T}})\,((\mathsf{H}^\circ)^{-1}\mathsf{C}'(\mathsf{H}^\circ)^{-\mathsf{T}})\,(\mathsf{H}^{-1}\mathbf{x}') = 1/|\mathsf{H}|^2\,\mathbf{x}'^\mathsf{T}\,\mathsf{C}'\,\mathbf{x}' = 0\,; \tag{6.58}$$

thus the transformed points $\mathbf{x}'$ also lie on the transformed conic. Observe, we do *not* propose the transformation $\mathsf{C}' = \mathsf{H}^{-\mathsf{T}}\mathsf{C}\mathsf{H}^{-1}$, as it would be inconsistent for singular conics of the form $\mathsf{C} = \mathbf{l}\mathbf{m}^\mathsf{T} + \mathbf{m}\mathbf{l}^\mathsf{T}$ (cf. (5.162), p. 241) when transforming both lines via $\mathbf{l}' = \mathsf{H}^\circ\mathbf{l}$ and $\mathbf{m}' = \mathsf{H}^\circ\mathbf{m}$ and applying variance propagation, cf. the discussion below (6.46), p. 258.

Similarly, dual conics or line conics C°, cf. Sect. 5.7.1.5, therefore transform as

$$\mathsf{C}'^\circ = \mathsf{H}\mathsf{C}^\circ\mathsf{H}^\mathsf{T}\,. \tag{6.59}$$

Also, the transformed point-quadric $\mathcal{Q}'(\mathsf{Q}')$ of a given quadric $\mathcal{Q}(\mathsf{Q})$ is obtained from

$$\mathsf{Q}' = \mathsf{H}^\circ\mathsf{Q}\mathsf{H}^{\circ\mathsf{T}}. \tag{6.60}$$

and a dual or plane-quadric Q° is transformed by

$$\mathsf{Q}'^\circ = \mathsf{H}\mathsf{Q}^\circ\mathsf{H}^\mathsf{T}\,. \tag{6.61}$$

E.g. a general 3D circle can be represented as a transformed unit circle in the XY-plane with normal $\boldsymbol{e}_3^{[3]}$ using the dual quadric $\mathsf{Q}_0^\circ = \mathrm{Diag}([-1,-1,0,1])$. You have to scale it by its radius R, rotate it from $\boldsymbol{e}_3^{[3]}$ to $\boldsymbol{N}$ and move it to its centre $\boldsymbol{X}_0$.

6.2.6 Summary of Transformations

The transformations of points, lines, planes, conics, and quadrics together with their transformation matrices are given in Tables 6.2 and 6.3. The transformation matrices for conics and 2D lines, and for planes and quadrics, are identical, taking the homogeneity of the matrices into account.

Observe the similarity of the expressions in the transformations of lines and planes.

Table 6.2 Motions in 2D of points and lines with 3×3 transformation matrices M and M_l

2D entity / transformation		translation	rotation	motion
2D point	$\underset{3\times3}{\mathsf{M}} =$	$\begin{bmatrix} I_2 & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
2D line	$\underset{3\times3}{\mathsf{M}_l} =$	$\begin{bmatrix} I_2 & \mathbf{0} \\ -\boldsymbol{t}^\mathsf{T} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ -\boldsymbol{t}^\mathsf{T}R & 1 \end{bmatrix}$

Table 6.3 Motions in 3D of points, lines, and planes with transformation matrices M, M_L, and M_A. Note: $-\mathcal{S}^{\mathsf{T}}(\boldsymbol{T}) = \mathcal{S}(\boldsymbol{T})$

3D entity transformation		translation	rotation	motion
3D point	$\underset{4\times4}{\mathsf{M}} =$	$\begin{bmatrix} I_3 & \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$
3D line	$\underset{6\times6}{\mathsf{M}_L} =$	$\begin{bmatrix} I_3 & \mathbf{0} \\ -\mathcal{S}(\boldsymbol{T})^{\mathsf{T}} & I_3 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & R \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ -\mathcal{S}(\boldsymbol{T})^{\mathsf{T}}R & R \end{bmatrix}$
plane	$\underset{4\times4}{\mathsf{M}_A} =$	$\begin{bmatrix} I_3 & \mathbf{0} \\ -\boldsymbol{T}^{\mathsf{T}} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ -\boldsymbol{T}^{\mathsf{T}}R & 1 \end{bmatrix}$

6.3 Concatenation and Inversion of Transformations

6.3.1 Inversion 261
6.3.2 Concatenation 261

Since projective collineations can be represented as linear mappings when using homogeneous coordinates, inversion of mappings corresponds to matrix inversion and concatenation of mappings corresponds to matrix multiplication.

6.3.1 Inversion

We start with a transformation $\mathcal{H}(\mathsf{H})$ of a point $\mathcal{X}_1(\mathbf{x}_1)$ yielding point $\mathcal{X}_2(\mathbf{x}_2)$,

$$\mathcal{H}: \quad \mathbf{x}_2 = \mathsf{H}\,\mathbf{x}_1\,. \tag{6.62}$$

The inverse transformation is achieved by multiplication from left with H^{-1}, which yields $\mathbf{x}_1 = \mathsf{H}^{-1}\,\mathbf{x}_2$. Thus the inversion of a transformation is given by

$$\mathcal{H}^{-1}: \quad \mathbf{x}_1 = \mathsf{H}^{-1}\mathbf{x}_2\,, \tag{6.63}$$

again for arbitrary regular transformations.

6.3.2 Concatenation

6.3.2.1 Concatenation of Basic Transformations

Let us have two homographies

$$\mathcal{H}_1: \quad \mathbf{x}_1 = \mathsf{H}_1\,\mathbf{x}_0 \qquad \text{and} \qquad \mathcal{H}: \quad \mathbf{x}_2 = \mathsf{H}_2\,\mathbf{x}_1\,. \tag{6.64}$$

The *concatenation* results from direct matrix multiplication

$$\mathbf{x}_2 = \mathsf{H}_2\mathsf{H}_1\,\mathbf{x}_0 = \mathsf{H}\,\mathbf{x}_0\,, \tag{6.65}$$

thus

$$\mathcal{H} = \mathcal{H}_2 \circ \mathcal{H}_1: \quad \mathsf{H} = \mathsf{H}_2\mathsf{H}_1\,. \tag{6.66}$$

This is valid independently of the type of transformation.

6.3.2.2 Concatenations with Displacements of Objects and Coordinate Systems

Concatenation of transformations needs a more detailed discussion. We have started with transformations which displace one object represented by points into another object.

coordinate transformation

The situation changes if the object is fixed and the reference system changes. This is usually called a coordinate transformation. Now, a positive translational displacement of the reference system makes the object appear with diminished coordinates; a positive rotation of the reference system makes the object appear to be rotated in the reverse direction. We therefore must apply the inverse transformation matrices. If the coordinate system is displaced according to H, the coordinates of a point are transformed by

$$\mathbf{x}' = \mathsf{H}^{-1}\,\mathbf{x}\,. \tag{6.67}$$

In addition, we need to distinguish whether the second transformation refers to the original coordinate system or to the coordinate system after the first transformation.

The four possible combinations reflect different conventions or different physical situations. We first give the result of the four cases A to D and then discuss each separately. Finally, we discuss a notation which allows us to make the different cases explicit.

The four cases are collected in Table 6.4:

Table 6.4 Rules for the concatenation of two transformations

	refer to fixed system	refer to displaced system
Displacement of object	A: $\mathbf{x}_2 = \mathsf{H}_2\mathsf{H}_1\mathbf{x}_0$	B: $\mathbf{x}_2 = \mathsf{H}_1\mathsf{H}_2\mathbf{x}_0$
Displacement of reference frame	C: $\mathbf{x}'' = \mathsf{H}_1^{-1}\mathsf{H}_2^{-1}\mathbf{x}$	D: $\mathbf{x}'' = \mathsf{H}_2^{-1}\mathsf{H}_1^{-1}\mathbf{x}$

A If the transformation of the *object* is described in the *original reference system*, taken to be fixed, the concatenation is performed with the *original matrices* by multiplication from the *left*, (6.70).

B If the transformation of the *object* is described in the *coordinate system of the transformed object*, the concatenation is performed with *the original matrices* by multiplication from the *right*, (6.71).

C If the transformation of the *reference system* is described in the *original reference system*, the concatenation is performed with the *inverse transformation matrices* by multiplication from the *right*, (6.72).

D If the transformation of the *reference system* is described in the *transformed reference system*, the concatenation is performed with the *inverse transformation matrices* by multiplication from the *left*, (6.73).

We now discuss the four different concatenations for motions in detail; however, the discussion refers to general transformations $\mathcal{H}$ as well.

Concatenation A: The first possibility is the one already mentioned, cf. Fig. 6.5, left.

1. We move the object together with the point $\mathcal{X}_0$ according to the first transformation, $\mathcal{H}_1$. It is described in the original coordinate system, which is attached to the object. This leads to the new point $\mathcal{X}_1$,

$$\mathbf{x}_1 = \mathsf{H}_1\mathbf{x}_0. \tag{6.68}$$

2. The second motion $\mathcal{H}_2$ of the object with point $\mathcal{X}_1$ from position 1 to position 2 yields point $\mathcal{X}_2$,

$$\mathbf{x}_2 = \mathsf{H}_2\mathbf{x}_1\,, \tag{6.69}$$

or, together,

$$\mathbf{x}_2 = \mathsf{H}_2\mathsf{H}_1\mathbf{x}_0\,. \tag{6.70}$$

The second motion of the object was expressed in the original coordinate system, which is assumed to be fixed when performing this type of concatenation.

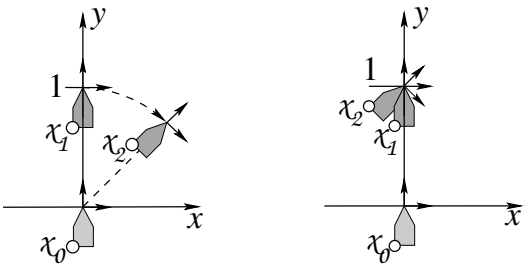


Fig. 6.5 Concatenation of motions of an object. The reference system is fixed. **Left:** (A) the second motion refers to original system of the object. **Right:** (B) the second motion refers to system of the moved object. The original point χ_0 has coordinates $[-1/12, -1/3]^\mathsf{T}$. The first motion is a translation by 1 in the y-direction, the second is a rotation by -45^o

Concatenation B: For constructive or physical reasons we might like to describe the second motion in the new coordinate system, which is assumed to be attached to the moved object, see Fig. 6.5, right.

We can achieve this in the following manner:

1. Motion of χ_0 with the first transformation, as above,

$$\mathbf{x}_1 = \mathsf{H}_1\mathbf{x}_0.$$

2. In order to express the motion in the new system, we perform three additional motions:

   a. We first undo the first motion, expressing it in the original system.
   b. We perform the second motion.
   c. We now perform the first motion; this carries the effect of the second motion.

   Together, we obtain

$$\begin{aligned}\mathbf{x}_2 &= \mathsf{H}_1\mathsf{H}_2\mathsf{H}_1^{-1}\mathbf{x}_1\\ &= \mathsf{H}_1\mathsf{H}_2\underbrace{\mathsf{H}_1^{-1}\mathsf{H}_1}_{I_3}\mathbf{x}_0\,,\end{aligned}$$

   thus

$$\mathbf{x}_2 = \mathsf{H}_1\mathsf{H}_2\mathbf{x}_0\,. \tag{6.71}$$

   Obviously, we need to concatenate the transformation matrices in the reverse order.

Concatenation C: The effect of displacements of the reference system described in the original reference system onto the coordinates $\mathbf{x}$ of a point χ yields the inverse transformation to (6.70), see Fig. 6.6, left:

$$\mathbf{x}'' = \mathsf{H}_1^{-1}\mathsf{H}_2^{-1}\mathbf{x}\,. \tag{6.72}$$

Here $\mathbf{x}''$ is the coordinate vector of the point χ in the new reference system, which is obtained by two consecutive coordinate transformations from $\mathbf{x}$ to $\mathbf{x}'$ to $\mathbf{x}''$, described in the original reference system.

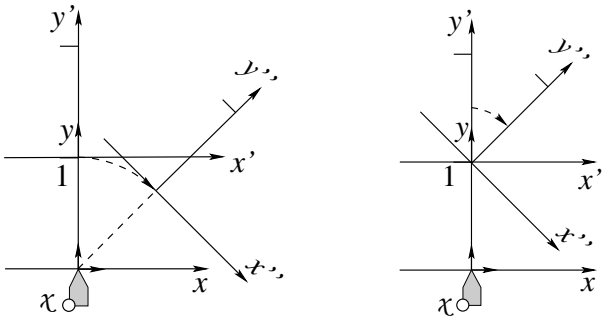


Fig. 6.6 Concatenation of displacements of the coordinate system. The object is fixed. **Left:** (C) the second displacement refers to original coordinate system. **Right:** (D) the second displacement refers to displaced coordinate system. The original coordinates and the two motions, there of the object, here of the coordinate system, are the same as in Fig. 6.5, p. 263

Concatenation D: Finally, we may describe the effect of the second displacement of the reference system on the coordinate system reached after the first displacement, see Fig. 6.6, right. Then we obtain the inverse of (6.72), now referring to the coordinates $\mathbf{x}$ of the unchanged point χ

$$\mathbf{x}'' = \mathsf{H}_2^{-1}\mathsf{H}_1^{-1}\mathbf{x}\,. \tag{6.73}$$

Thus, coordinates $\mathbf{x}''$ of the point χ are expressed in the reference system after its second displacement.

Example 6.3.21: ***Denavit–Hartenberg parameters.*** In robotics, a complete arm of a robotic machine is composed of a sequence of rigid links. The links can perform articulated motions at joints, realizing rotations or translations. Here, we only discuss rotatory joints. As the axes can be interpreted as 3D lines, and 3D lines have four degrees of freedom, the relation between two successive links can be represented with four parameters, except for singularities. The complete coordinate transformation from the basic coordinate system to the coordinate system of the gripper or end effector can be derived by concatenating these local transformation matrices. With the four parameters per pair of consecutive axes we use the fact that, except if consecutive axes are parallel, there exists a unique 3D line perpendicular to z_{n-1} and z_n, cf. Fig. 6.7.[3]

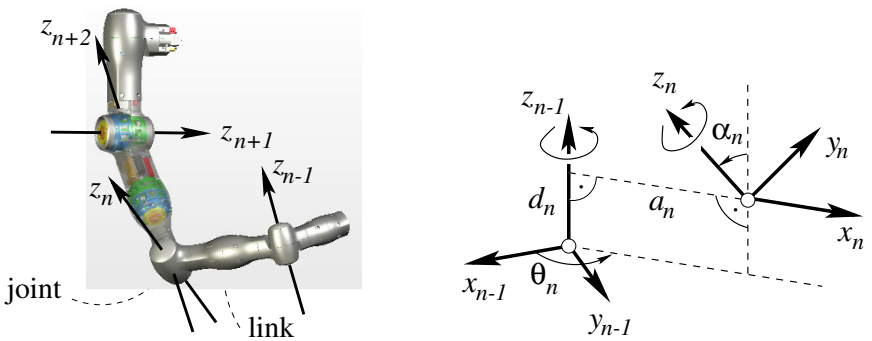


Fig. 6.7 Denavit–Hartenberg representation of the mutual transformation of two consecutive rotational joints of a robot link. **Left:** robot arm with rotation axes. **Right:** Parameters of Denavit–Hartenberg representation

Let the two consecutive rotation axes be represented by the z-axis of the right-handed coordinate systems S_{n-1} and S_n. The coordinate system of the second link is defined as follows:

- The origin is at the point of the z_n-axis closest to the z_{n-1} axis.
- The x_n axis by construction is perpendicular to the z_{n-1} axis and points away from it.

[3] From http://www.leros-f.iat.uni-bremen.de/, last visited 4.5.2015.

- The y_n-axis completes the frame to a right-handed system.

We now need four parameters to specify the motion of the frame S_{n-1} to the frame S_n:

1. We rotate the coordinate system S_{n-1} by the angle θ_n around the z_{n-1} axis:

$$\mathsf{M}_1 = \begin{bmatrix} R_z(\theta_n) & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{6.74}$$

2. We shift the coordinate system by d_n along the z_{n-1}-axis:

$$\mathsf{M}_2 = \begin{bmatrix} I_3 & \boldsymbol{t}_z(d_n) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{6.75}$$

3. We shift the coordinate system along the rotated and shifted x_{n-1} axis by a_n:

$$\mathsf{M}_3 = \begin{bmatrix} I_3 & \boldsymbol{t}_x(a_n) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{6.76}$$

4. We rotate the coordinate system around the new x-axis by α_n:

$$\mathsf{M}_4 = \begin{bmatrix} R_x(\alpha_n) & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{6.77}$$

Since the transformations always refer to the moved coordinate systems, the complete motion is described by:

$$\mathsf{M}(\theta_n, d_n, a_n\alpha_n) = \mathsf{M}_1(\theta_n)\mathsf{M}_2(d_n)\mathsf{M}_3(a_n)\mathsf{M}_4(\alpha_n) \tag{6.78}$$

$$= \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & a_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \cos\theta_n & a_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d \\ 0 & 0 & 0 & 1\,. \end{bmatrix} \tag{6.79}$$

◇

6.3.2.3 A Notation for Object and Coordinate Transformations

Providing a transformation $\mathcal{H}(\mathsf{H})$ requires us to describe its function as object or coordinate transformation and its reference coordinate system. If many such transformations are involved, it might be useful to use a notation which makes these descriptions explicit. We discuss one possibility for such a notation.

We assume that a matrix performs a transformation of an object as in (6.64), p. 261. We now add indices to the transformation to make the two point names i and j explicit and write:

$$_j\mathcal{H}^i : \qquad \mathbf{x}_j = {}_j\mathsf{H}^i\, \mathbf{x}_i\,. \tag{6.80}$$

The indices are placed such that the name i of the given point $\mathcal{X}_i$ appears as right upper index at $\mathcal{H}$, such that they can be cancelled on the right-hand side of the equation, leaving the lower index j on both sides. Inversion of this transformation now is

$$_i\mathcal{H}^j = (\,_j\mathcal{H}^i)^{-1} : \qquad {}_i\mathsf{H}^j = (\,_j\mathsf{H}^i)^{-1}\,. \tag{6.81}$$

Thus, exchanging the indices is equivalent to inversion, symbolically as well as algebraically. Concatenation of two such transformations, $_j\mathcal{H}^i$ from i to j and $_k\mathcal{H}^j$ from j to k, yields the concatenated transformation $_k\mathcal{H}^i$ from i to k,

$$_k\mathcal{H}^i = {}_k\mathcal{H}^j \circ {}_j\mathcal{H}^i : \qquad {}_k\mathsf{H}^i = {}_k\mathsf{H}^j\ {}_j\mathsf{H}^i\,. \tag{6.82}$$

Observe the cancelling of the index j.

We write the coordinate system for a vector as left upper superscript, e.g., ${}^1\mathbf{x}$, if the coordinate vector refers to coordinate system S_1. Thus, for the point i the coordinate transformation from the coordinate system l to the coordinate system m is written as

$$ {}^{m}\mathcal{H}_{l}: \quad {}^{m}\mathbf{x}_{i} = {}^{m}\mathsf{H}_{l}\, {}^{l}\mathbf{x}_{i}\,. \tag{6.83} $$

Observe, the indices at the transformation matrix now sit differently, which allows us to distinguish this from the previous case. Inversion and concatenation of coordinate transformations work the same way:

$$ {}^{l}\mathsf{H}_{m} = ({}^{m}\mathsf{H}_{l})^{-1} \quad \text{and} \quad {}^{n}\mathsf{H}_{l} = {}^{n}\mathsf{H}_{m}\, {}^{m}\mathsf{H}_{l}\,. \tag{6.84} $$

Finally, we need to express the relation between displacement, ${}_{j}\mathcal{H}^{i}$, of a coordinate system and resulting transformation, ${}^{j}\mathcal{H}_{i}$, of the coordinates; that is why we need to take the same indices i and j. Following (6.67), p. 262 we therefore have the relation

$$ {}^{j}\mathcal{H}_{i} = ({}_{j}\mathcal{H}^{i})^{-1}: \qquad {}^{j}\mathsf{H}_{i} = ({}_{j}\mathsf{H}^{i})^{-1}\,. \tag{6.85} $$

Observe, we have:

$$ {}^{j}\mathcal{H}_{i} = {}_{i}\mathcal{H}^{j}: \qquad {}^{j}\mathsf{H}_{i} = {}_{i}\mathsf{H}^{j}\,, \tag{6.86} $$

i.e., the coordinate transformation matrix for points is the inverse displacement matrix for the coordinate system, as is to be expected. We will use this type of specification of transformations when necessary.

6.4 Invariants of Projective Mappings

6.4.1 Invariants and Equivariant Functions 266
6.4.2 Invariants of Collineations and of Its Specializations 268
6.4.3 The Cross Ratio 268
6.4.4 Invariants of More General Configurations in 2D 271
6.4.5 Fixed Points, Lines, and Planes of Autocollineations 272

Invariants play a central role in science. Invariants characterize objects and provide a basis for the recognition or the comparison of objects. For example, imagine a photo of a person, and name three properties which are necessary for recognizing the person under varying pose and illumination. Early work on geometric image analysis extensively used invariants (Mundy and Zisserman, 1992; Gros and Quan, 1992).

After some basic definitions, we discuss the invariants of collineations. We distinguish between

1. Invariants of geometric configurations.
2. Fixed geometric elements, thus invariants of the (respective) projective space as a whole.

6.4.1 Invariants and Equivariant Functions

We start with the definition of an invariant.

Definition 6.4.13: ***Invariant.*** A property I of an entity E is an invariant w.r.t. a transformation $\mathcal{g}$ of the entity if for all elements $\mathcal{g}$ of the transformation group $\mathcal{G}$,

$$ I(E) = I(\mathcal{g}(E)) \qquad \mathcal{g} \in \mathcal{G}\,. \tag{6.87} $$

Thus the transformation $\mathcal{g}$ applied to the entity E does not change the property I . ◇

Example 6.4.22: ***Invariants of a polygon.*** Let the entity E be a polygon and $\mathcal{g} = \mathcal{M}$ a planar motion. Then the number of points, the lengths of the sides, the angles, the length ratios, the area, and the perimeter are invariant. The property of parallel opposite sides, if it is present, is also an invariant, cf.

Fig. 6.8. If we again have a polygon as entity E but now $\mathcal{g} = \mathcal{A}$ is an affine transformation, then angles,

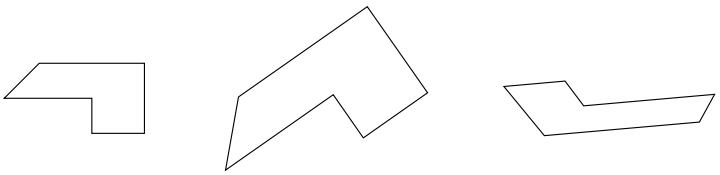


Fig. 6.8 Invariants of a polygon (**left**) w.r.t. similarity (**centre**) and affinity (**right**)

lengths and areas are no longer invariant; however, the property that parallel lines map to parallel lines is still valid. ◇

Obviously, the type of invariant is arbitrary. Invariants highly depend on the assumed transformation. If we do not specify the type of invariant, there is no general automatic procedure to find invariants.

The notion of invariance should not be confused with the situation in which the centroid of a figure is mapped to the centroid of the transformed figure. Here, determining the centroid of a set of points and determining the transformation of this point may be exchanged. This property of the function f is called *equivariance*, cf. Reisert and Burkhardt (2007) and Sebbar and Sebbar (2012).

Definition 6.4.14: ***Equivariant function.*** A function f acting on a set S of points is called equivariant if it can be exchanged with the elements $\mathcal{g}$ of the group G acting on the elements of the set

$$f(\mathcal{g}(S)) = \mathcal{g}(f(S))\,. \tag{6.88}$$

◇

Example 6.4.23: ***Equivariant function.*** Let the set S of points be a rectangle, $\mathcal{g} = \mathcal{T}$ a planar translation, and the function f an affine scaling with respect to the centroid μ_x of the rectangle in the x-direction: $x' - \mu'_x = \lambda(x - \mu_x)$. Then this affine scaling can be applied before or after the translation, and, therefore, the mapping f is an equivariant function with respect to the affine scaling, cf. the examples in Fig. 6.9, left. If the transformation is a planar motion or even a homography, the operator and the transformation cannot be switched. Also, local scaling w.r.t. homography is not an operator homomorphism.

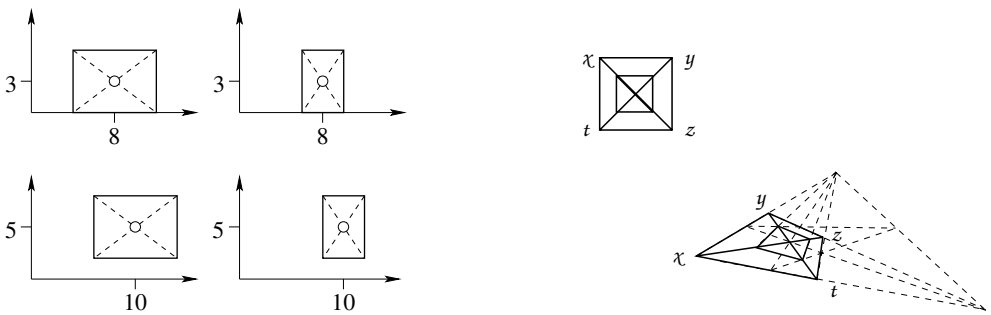


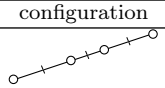
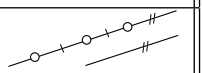
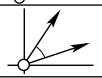
Fig. 6.9 Equivariant functions. **Left:** Centroid related scaling in the x-direction w.r.t. translation. **Right:** Intersection of diagonals w.r.t. homography

Alternatively, let the entity be a quadrangle $(\chi\ y\ z\ t)$, the transformation $\mathcal{g}$ be a homography, and the operator the intersection of the diagonals $u = (\chi \wedge z) \cap (y \wedge t)$. The intersection (of the diagonals) and the homography may be exchanged; thus, the intersection is an equivariant function, see Fig. 6.9. However, a scaling with respect to the intersection of the diagonals is not an equivariant function. ◇

6.4.2 Invariants of Collineations and of Its Specializations

The following table contains the characteristic invariants of collineations and their specializations. The invariants hold for a transformation and its specializations (top-down), but not for its generalization (bottom-up). The invariants of affinities and its specializations

Table 6.5 Invariants of spatial configurations under collineations and their specializations. A direction is the angle between a straight line and one of the coordinate axis

transformation	invariant	configuration
collineation	incidence, straight lines cross ratio of collinear points	
affinity	distance ratio of collinear points parallelity	
similarity	angle between lines, ratio of distances	
motion	distance between points	
rotation around origin	direction differences from origin	
translation	directions	

are well-known (cf. Table 6.5):

1. Translations preserve directions.
2. Rotations around the origin preserve angles as seen from the origin.
3. Motions preserve distances between points.
4. Similarities preserve arbitrary angles and distance ratios of arbitrary point pairs, the ratio of two distances in a triangle. Distances are not preserved.
5. Affinities preserve parallelities (see the proof in Sect. 6.4.5.2, p. 274) and ratios of distances between parallel lines. Angles or arbitrary distance ratios are not preserved.

6.4.3 The Cross Ratio

Angles between lines and the ratio of distances between collinear point pairs are not preserved under collineations. This can be seen in the example of Fig. 6.10.

Exercise 6.8

However, four collinear points have an invariant under projective transformation: the cross ratio, which can be transferred to four concurrent lines and also to a pencil of four concurrent planes. The cross ratio can be used to describe more general configurations by their invariants.

6.4.3.1 Cross Ratio of Four Collinear Points

The basic configuration for the cross ratio is four collinear points.

Definition 6.4.15: ***Cross ratio of four collinear points.*** The cross ratio $\mathrm{CR}(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4)$ of four collinear points with line coordinates (x_1, x_2, x_3, x_4) is defined as

$$\mathrm{CR}(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4) = \frac{x_1 - x_3}{x_2 - x_3} : \frac{x_1 - x_4}{x_2 - x_4} \tag{6.89}$$

If a point is at infinity, the rule $\infty/\infty = 1$ is used. ◇

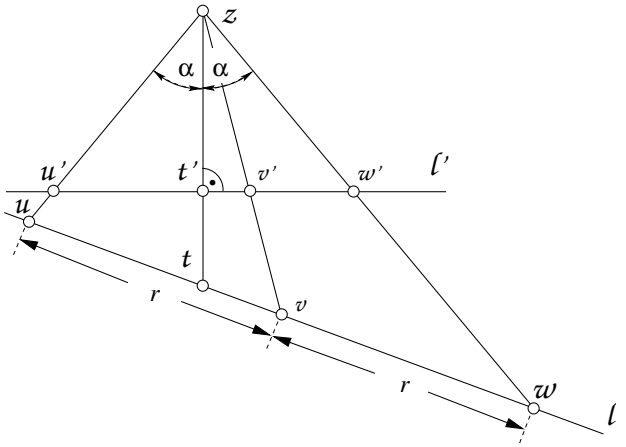


Fig. 6.10 Non-invariance of the distance ratio under perspectivity. The three collinear points u, v, and w on line l having the same distance $r = \overline{uv} = \overline{vw}$ are mapped to the line l' via the perspective projection with projection centre z, leading to the points u', v', and w'. Obviously, the distances between the image points are different: $\overline{u'v'} \neq \overline{v'w'}$. The midpoint v of $\overline{uw}$ is not mapped to the midpoint v' of $\overline{u'w'}$. Now, imagine the point v is the centre of a circle with radius $\overline{uv}$ lying in the plane through l perpendicular to the drawing plane. Its image in the plane through l', again orthogonal to the drawing plane, will be an ellipse: Obviously, the centre of an ellipse, which is the image of a circle, is not the image of the centre of that circle

We now have the following theorem:

Theorem 6.4.2: ***Invariance of cross ratio.*** The cross ratio $\mathrm{CR}(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4)$ of four collinear points is invariant under collineations $\mathcal{H}$. Thus if $\mathcal{X}_i' = \mathcal{H}(\mathcal{X}_i), i = 1, 2, 3, 4$, then

$$\mathrm{CR}(\mathcal{X}_1', \mathcal{X}_2', \mathcal{X}_3', \mathcal{X}_4') = \mathrm{CR}(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4)\,. \tag{6.90}$$

The proof (cf. Exercise 21, p. 289) exploits the fact that each of the four indices appears twice in the cross ratio, once in the numerator, once in the denominator.

Given four points, there are 24=4! permutations for their sequence. Thus it is possible to define 24 different cross ratios. However, six of them are distinct generally, but mutually functionally dependent. If one cross ratio is λ, we have the six different values for cross ratios of four points:

$$\lambda, \quad \frac{1}{\lambda}, \quad 1-\lambda, \quad \frac{1}{1-\lambda}, \quad \frac{\lambda}{1-\lambda}, \quad \frac{1-\lambda}{\lambda}\,. \tag{6.91}$$

Example 6.4.24: ***Inferring distances between collinear points.*** Assume, in Fig. 6.11, we have

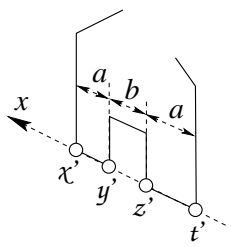


Fig. 6.11 Example for the use of the invariance of the cross ratio. Due to the assumed symmetry – the distance of the door from its two neighbouring facade borders is identical to some unknown value a and the assumed knowledge about the width of the building we can infer the true width of the door in the scene from the image points

observed the collinear image points x', y', z' and t' and know that the door is in the centre of the facade, which has a width of $w = 10$ m. Then, using the cross ratio we can determine the width b of the door and its distance a from the right and left wall from the two equations:

$$\mathrm{CR}(x,y,z,t) = \frac{a+b}{b} : \frac{2a+b}{a+b} = \mathrm{CR}(x',y',z',t') \qquad 2a+b = 10\ [\mathrm{m}] \tag{6.92}$$

where the cross ratio $\mathrm{CR}(x',y',z',t')$ can be determined from image measurements. ◇

Mirror symmetric configurations in a plane are characterized by (1) the existence of a symmetry axis and (2) the property that lines through symmetric point pairs are parallel. Now, we regard points on such a line: a point $\mathcal{X}$, its mirror point $\mathcal{X}'$, the mid point $\mathcal{Y}$ of both, which is the point of symmetry, and the point $\mathcal{X}_\infty$. Then their cross ratio is $\mathrm{CR}(\mathcal{X},\mathcal{X}',\mathcal{Y},\mathcal{X}_\infty) = -1$ and we say the configuration is *harmonic*. We therefore use the definition:

Definition 6.4.16: ***Harmonic points.*** Four points on a line are harmonic if their cross ratio is -1. ◇

6.4.3.2 Cross Ratio of Four Concurrent Lines

The cross ratio transfers to a pencil of four rays.

Definition 6.4.17: ***Cross ratio of four concurrent lines.*** The cross ratio $\mathrm{CR}(\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3,\mathcal{L}_4)$ of four concurrent lines with directions $(\phi_1,\phi_2,\phi_3,\phi_4)$ is defined as

$$\mathrm{CR}(\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3,\mathcal{L}_4) = \frac{\sin(\phi_1-\phi_3)}{\sin(\phi_2-\phi_3)} : \frac{\sin(\phi_1-\phi_4)}{\sin(\phi_2-\phi_4)}\,. \tag{6.93}$$

◇

We can see this from Fig. 6.12: The coordinate differences $x_i - x_j$ on line m and the sine of the direction differences $\sin(\phi_i - \phi_j)$ are related by the area F of the triangle $(z\ \mathcal{X}_i\ \mathcal{X}_j)$ with sides s_i, s_j via $2F = h(x_i - x_j) = s_i s_j \sin(\phi_i - \phi_j)$, which allows us to develop (6.93) from (6.89).

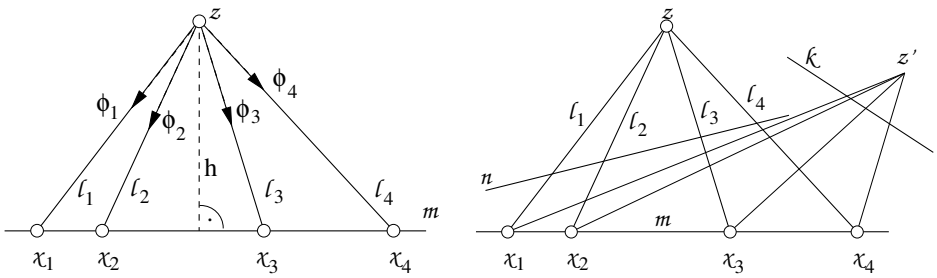


Fig. 6.12 Cross ratio of points and lines. **Left:** The collinear points $\mathcal{X}_i, i = 1,2,3,4$ and the concurrent lines $\mathcal{L}_i, i = 1,2,3,4$, are related by the central point z, having the distance h from the line m. The set of points and the set of lines have the same cross ratio. **Right:** Concurrent lines allow us to transfer the cross ratio from the original points $\mathcal{X}_i, i = 1,2,3,4$ to the intersection points of the lines $\mathcal{L}_i, i = 1,2,3,4$ with the line n or via the lines through z' to the intersection points with the line k

Given a fifth line m not identical to $\mathcal{L}_i$ and not passing through their intersection point, the cross ratio of the four concurrent lines $\mathcal{L}_i$ can be computed by

$$\mathrm{CR}(\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3,\mathcal{L}_4) = \frac{|\mathbf{m},\mathbf{l}_1,\mathbf{l}_2|}{|\mathbf{m},\mathbf{l}_3,\mathbf{l}_2|} : \frac{|\mathbf{m},\mathbf{l}_1,\mathbf{l}_4|}{|\mathbf{m},\mathbf{l}_3,\mathbf{l}_4|}\,. \tag{6.94}$$

This cross ratio is also the cross ratio $\mathrm{CR}(\mathcal{X}_1,\mathcal{X}_2,\mathcal{X}_3,\mathcal{L}_4)$ of the four intersection points $\mathcal{X}_i = \mathcal{L}_i \cap m$ of the lines $\mathcal{L}_i$ with m.

Proof: Without loss of generality, we can choose the line $\mathbf{m} = [0,1,0]$ to be the x-axis, and the intersection point of the lines not to lie on the y-axis, e.g. at $\mathbf{x} = [0,1,1]^\mathsf{T}$. The intersection points $\mathcal{X}_i = \mathcal{L}_i \cap m$ of the lines with the x-axis are assumed to have coordinates x_i. Then the determinants are

$$|\mathbf{m}, \mathbf{l}_i, \mathbf{l}_j| = \left|\mathbf{m}, \mathsf{S}(\mathbf{x}) \begin{bmatrix} x_i \\ 0 \\ 1 \end{bmatrix}, \mathsf{S}(\mathbf{x}) \begin{bmatrix} x_j \\ 0 \\ 1 \end{bmatrix}\right| = \begin{vmatrix} 0 & 1 & 1 \\ 1 & x_i & x_j \\ 0 & -x_i & -x_j \end{vmatrix} = x_j - x_i \,, \tag{6.95}$$

which completes the proof. ◇

Example 6.4.25: ***Image of the horizon from the image of three equidistant parallel lines.*** Given is an image with three lines $\ell_i', i = 1, 2, 3$, which in 3D are coplanar, parallel, and equidistant, see Fig. 6.13. The task is to determine the image h' of the line at infinity of the plane.

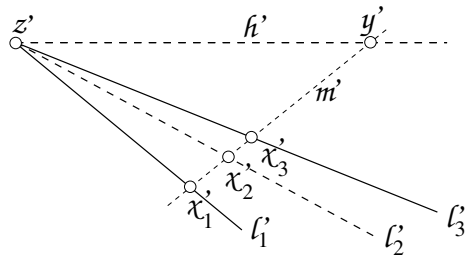


Fig. 6.13 Horizon of a plane with equidistant coplanar lines

We give two solutions, a constructive one and an algebraic one:

1. We first determine the intersection point $z' = \ell_1' \cap \ell_2'$ of two of the image lines, ℓ_i'. z' is the image of the point at infinity of the set of 3D lines. Then we take an arbitrary line m' passing through the three lines ℓ_i', leading to three intersection points $x_i' = m' \cap \ell_i'$. We now construct a point $y' \in m'$ such that $\mathrm{CR}(x_1', x_3', x_2', y') = -1$. Then the sought line is $h' = z' \wedge y'$.
2. The construction can be used to derive a closed form solution (Schaffalitzky and Zisserman, 2000),

$$\mathbf{h}' = [\mathbf{m}', \mathbf{l}_1', \mathbf{l}_2']\mathbf{l}_3' - [\mathbf{m}', \mathbf{l}_2', \mathbf{l}_3']\mathbf{l}_1' \,, \tag{6.96}$$

where m' is an arbitrary line not identical to ℓ_i and not passing through the intersection point of the three lines. Exercise 6.22

◇

6.4.4 Invariants of More General Configurations in 2D

A more general configuration in comparison to a line may be characterized by more than one invariant.

For this we need to distinguish between a description of the configuration within a coordinate system and a description of the form of the configuration which might be the subject of an arbitrary transformation of some specified type.

If we are able to describe a geometric configuration within a coordinate system with a minimum number, o, of parameters, we are able to give the number of functionally independent invariants with respect to a group of transformations.

If this transformation is specified by t parameters, the number i of invariants is given by

$$i = o - t \,. \tag{6.97}$$

The reason is the following: Let us assume $o > t$, then we can use t parameters o_t to describe a subpart of the object, and this is also valid for the transformed figure. Based on the correspondence between the original and the transformed subparts of the object, we are able to derive the t parameters of the transformation. We can now apply the transformation to the other $o - t$ parameters of the object, necessarily leading to the corresponding parameters of the transformed object.

Example 6.4.26: ***Invariants of a rectangle.*** A rectangle $(ABCD)$ under planar motion has two functionally independent invariants.

The specification of a rectangle in the plane requires $o = 5$ parameters: e.g., the coordinates of two points A and B and the distance of the opposite side (CD) from (AB). The planar motion can be specified by the two parameters of the translation of one of the points, say A, and one parameter of the rotation of one of the sides, say AB. Then we have used $t = 3$ of the five parameters for specifying the motion. The other $i = 2 = 5 - 3$ parameters, e.g., the sides $\overline{AB}$ and $\overline{AC}$, are completely independent of the motion, thus invariants. ◇

We now can transfer this reasoning to invariances of configurations under projective transformations.

Example 6.4.27: ***Invariants of a quintuple.*** A quintuple of 2D points has two functionally independent invariants under a collineation. This is in accordance with (6.97), as a quintuple is described by $o = 10$ coordinates and a collineation requires $t = 8$ parameters, leaving $i = o - t = 2$ invariants. These can easily be defined the following way: Let the first four points be used to determine a homography. Then we need projective invariants to determine the fifth point. These could be the two cross ratios of the two sets of concurrent lines through two of the points. Two cross ratios can be used to identify a specific five-point configuration of a coded target consisting of several circular discs, cf. Fig. 1.4, p. 5. They also can be exploited to determine the image p' of a point p if the homography is given by four points. Let the four points x, y, z, t and their images x', y', z', t' be given. Then the relation of p w.r.t. the first four points can be characterized by the two cross ratios $\mathrm{CR}(yx, yz, yt, yp)$ and $\mathrm{CR}(zx, zy, zt, zp)$, which can then be used to identify the point p' using the *paper-strip construction*, cf. Fig. 6.14. ◇

Exercise 6.11

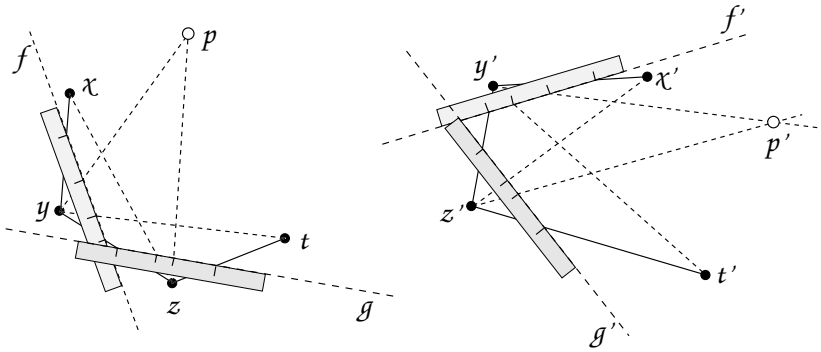


Fig. 6.14 Paper strip construction: Five points are characterized by two invariants w.r.t. a collineation, namely two cross ratios. Each cross ratio may be realized by four points on a paper strip. Given the five points x, y, z, t and p on the left, and the points x', y', z', t' on the right, point p can be transfered to p' using a paper strip

The inverse problem arises when four points and the cross ratio of the pencil of lines through the four points are given. What is the locus of the vertex of the pencil if it is moved in $\mathbb{P}^2$ with constant cross ratio? The answer is given by Chasles' theorem, see Fig. 6.15.

Theorem 6.4.3: ***Chasles' theorem.*** Given five points $x_1, \ldots x_5$ on a nondegenerate conic forming a pencil of lines with vertex x_5, the cross ratio of the pencil is independent of the position of x_5 on the conic.

The proof projectively maps the configuration on a circle, where the cross ratio depends on the sine of the angles at x_5 which are invariant of the choice of x_5 on the circle.

6.4.5 Fixed Points, Lines, and Planes of Autocollineations

In the following, we discuss invariants of autocollineations, which are mappings of a projective space onto itself, i.e., together with its entities, points, lines, or planes. The invariant

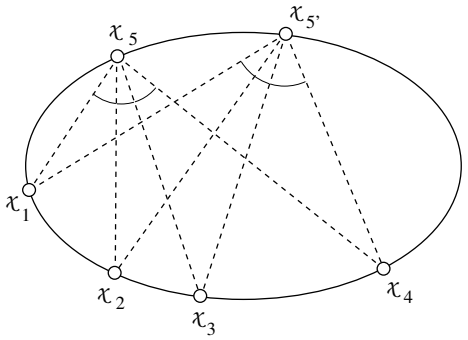


Fig. 6.15 Chasles' theorem: The cross ratio of a pencil of lines through four points on a conic is invariant of the choice of the vertex $\mathcal{X}_5$ on that conic

entities are accordingly called fixed points, fixed lines, and fixed planes. They may be used to characterize a mapping.

For example, planar rotations have the origin as a fixed point and the line at infinity as a fixed line, i.e., a point on the line at infinity is mapped to another point at infinity. For spatial translations, by contrast, the plane at infinity is a fixed plane. Here, in contrast to rotations, a point at infinity is mapped to itself, i.e., all points at infinity are fixed points. Therefore, we need to distinguish fixed lines and fixed planes which are mapped to themselves point by point from those fixed entities on which points are displaced.

6.4.5.1 Number of Fixed Points and Hyperplanes

For an arbitrary regular mapping $\mathcal{H} : \mathbb{P}^n \to \mathbb{P}^n$, fixed points $\mathcal{X}^f$ are defined by

$$\mathbf{x}^f = \lambda \mathsf{H} \mathbf{x}^f \,, \tag{6.98}$$

where we intentionally made the proportionality of the left and right sides of the expression explicit. Obviously, the homogeneous coordinates of the fixed points are the right eigenvectors of the matrix H.

Due to the characteristic polynomial of matrix H, the maximum number of fixed points is $n+1$. As the eigenvectors are either real or pairs of complex numbers, we may have less than $n+1$ or even no real fixed points. This depends on the dimension $n+1$ of the mapping matrix: If the dimension is odd, we have an odd number of real fixed points, thus at least one. If the dimension is even, we have an even number of real fixed points, and possibly no fixed point. We count double roots of the characteristic polynomial of H as two roots.

The discussion directly transfers to the mapping of hyperplanes. Using the notation from 2D, cf. Sect. 6.2.4, fixed hyperplanes $\mathcal{l}^f$ are defined by $\mathbf{l}^f = \mathsf{H}^{\circ}\mathbf{l}^f$, or, equivalently, (if H is regular),

$$\lambda \mathsf{H}^{\mathsf{T}} \mathbf{l}^f = \mathbf{l}^f \,. \tag{6.99}$$

The fixed hyperplanes are determined by the left eigenvectors of H. Therefore, the number of real fixed hyperplanes is identical to the number of fixed points.

If there are multiple real eigenvalues with eigenvectors, say $\mathbf{x}_i^f$, $i = 1, ..., k$, then the complete space $\mathbf{x}^f = \sum_{i=1}^{k} \alpha_i \mathbf{x}_i^f$ spanned by the eigenvectors is mapped to itself pointwise.

We may categorize all homographies as a function of the number of real roots and possibly the number of double roots. In the following, however, we discuss the fixed elements of the most important mappings only.

6.4.5.2 Fixed Entities of Planar Homographies

Planar homographies with $n = 2$ have at least one fixed point and at least one fixed line.

Fixed Elements of Planar Translations. For translations with $\boldsymbol{t} = [t_x, t_y]^\mathsf{T}$ we find $\lambda_i = 1, i = 1, 2, 3$. The fixed points are

$$\mathbf{x}_1^f = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2^f = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{thus} \quad \mathbf{x}^f = \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} \tag{6.100}$$

for a translation, where (α, β) can be chosen arbitrarily. They span the line at infinity. Thus all points at infinity are fixed points.

The fixed lines are

$$\mathbf{l}_1^f = \begin{bmatrix} -t_y \\ t_x \\ 0 \end{bmatrix}, \quad \mathbf{l}_2^f = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{thus} \quad \mathbf{l}^f = \begin{bmatrix} -t_y \\ t_x \\ \alpha \end{bmatrix} \tag{6.101}$$

with arbitrary α. They span all lines parallel to the translation vector, including the line at infinity. Thus all lines parallel to the translation vector are fixed lines, and the line at infinity is mapped to itself pointwise.

Fixed Elements of Pure Planar Rotations. For pure rotations, we find the only real eigenvalue $\lambda = 1$. Thus, there is only one fixed point, the origin, and one fixed line, the line at infinity,

$$\mathbf{x}^f = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{l}^f = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{6.102}$$

Fixed Elements of Planar Motions. Also, for motions $\boldsymbol{x}' = R\boldsymbol{x} + \boldsymbol{t}$, we only have one real eigenvalue of the homogeneous transformation matrix, thus a single fixed point and a single fixed line. They are given by

$$\boldsymbol{x}^f = (I_2 - R)^{-1}\boldsymbol{t}, \qquad \mathbf{l}^f = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{6.103}$$

Exercise 6.23 Each motion can be represented as a rotation around a point. We can characterize motions as special collineations:

Theorem 6.4.4: ***Motion and the singular dual conic.*** A planar motion is a homography which preserves the singular dual conic $\mathsf{C}_\infty^* = \mathrm{Diag}([1, 1, 0])$, cf. (5.164), p. 241. The two points $\mathbf{i} = [i, 1, 0]^\mathsf{T}$ and $\mathbf{j} = [-i, 1, 0]^\mathsf{T}$ on this conic are fixed points.

Exercise 6.10 The proof is direct.

Fixed Elements of 2D Affinities. Affine mappings have one or three real eigenvalues. One real eigenvalue is $\lambda = 1$. The fixed line $\ell^f = \ell_\infty$ is the line at infinity. This characterizes affine mappings. The property can be used to show that parallel lines are preserved. The converse also is true.

Proposition 6.4.3: ***Affinities and parallelity.*** A collineation is an affinity if and only if the parallelity of straight lines is preserved.

Proof: (1) Given an affinity, we need to show that parallel lines are mapped to parallel lines. Let these lines be ℓ and m. Their intersection point $\chi = \ell \cap m$ is a point at infinity on the line at infinty, say n_∞, thus $\chi \in n_\infty$. The lines are mapped to ℓ' and m', their intersection point to $\chi' = \ell' \cap m'$. As affinities map points at infinity to points at infinity, hence $\chi' \in n_\infty$: the mapped lines ℓ' and m' are parallel.

(2) Given a collineation which preserves parallelism, we need to show it is an affinity. An affinity maps all points at infinity again to infinity. Thus the line at infinity must be mapped to itself: $\mathbf{n}_\infty = \lambda \mathsf{H}^\mathsf{T} \mathbf{n}_\infty$. This is equivalent to

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} g \\ h \\ i \end{bmatrix} . \tag{6.104}$$

This is only true if $g = h = 0$. Thus $\mathcal{H}$ is an affinity. ◇

The invariants for 2D motions are collected in Table 6.6.

Table 6.6 Fixed points and fixed lines of general 2D motions. $\boldsymbol{t}^\perp$ is the vector perpendicular to $\boldsymbol{t}$. The scalar α and the 2-vector $\boldsymbol{\beta}$ are arbitrary

	fixed points	fixed lines
2D translation $\boldsymbol{t}$	$\begin{bmatrix} \boldsymbol{\beta} \\ 0 \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{t}^\perp \\ \alpha \end{bmatrix}$
2D rotation R	$\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$	$\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$
2D motion $(R, \boldsymbol{t})$	$\begin{bmatrix} (I_2 - R)^{-1} \boldsymbol{t} \\ 1 \end{bmatrix}$	$\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$

6.4.5.3 Fixed Entities of Spatial Homographies

As the characteristic polynomial of spatial homographies has degree 4, these homographies have no, two, or four real eigenvalues and therefore the same number of fixed points. Moreover, we are also interested in fixed lines and planes.

The analysis is quite parallel to the analysis in 2D.

Fixed Elements of Spatial Translations. For a translation with $\boldsymbol{T} = [T_1, T_2, T_3]^\mathsf{T}$, we have a fourfold eigenvalue $\lambda = 1$. The space of eigenvectors is three-dimensional. The three *fixed points* span the plane at infinity, in full analogy to the situation in 2D, cf. (6.100):

$$\mathbf{X}^f = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ 0 \end{bmatrix} . \tag{6.105}$$

The *fixed planes* are all linear combinations of the three different right eigenvectors of the translation matrix M_A, cf. Table 6.3, p. 261, hence

$$\mathbf{A}^f = \begin{bmatrix} S(\boldsymbol{T})\boldsymbol{\alpha} \\ \delta \end{bmatrix} , \tag{6.106}$$

with arbitrary scalars α, β, and γ or arbitrary 3-vector $\boldsymbol{\alpha}$ and scalar δ. These are all planes which are parallel to the translation vector or, equivalently, their normal is perpendicular to the translation.

The general representation of the space of planes parallel to $\boldsymbol{T}$ is left as an exercise. Exercise 6.12

The fixed lines are defined by

$$\mathbf{L}^f = \lambda \mathsf{H}_L \, \mathbf{L}^f , \tag{6.107}$$

with the line mapping $\mathcal{H}_L$ from (6.53), p. 259. For a translation, we have

$$\mathsf{H}_L = \begin{bmatrix} I_3 & 0_{3\times 3} \\ S(\boldsymbol{T}) & I_3 \end{bmatrix} , \tag{6.108}$$

from which we obtain the sixfold eigenvalue $\lambda = 1$. The space of fixed lines is four-dimensional and spans all lines parallel to the translation ($\alpha \neq 0$) and all lines at infinity ($\alpha = 0$),

$$\mathbf{L}^f = \begin{bmatrix} \alpha \boldsymbol{T} \\ \boldsymbol{\beta} \end{bmatrix} \tag{6.109}$$

with arbitrary scalar α and 3-vector $\boldsymbol{\beta}$.

Fixed Elements of Spatial Motions. For a motion with rotation axis $\boldsymbol{r}$, angle ω, and translation vector $\boldsymbol{T}$, the homogeneous motion matrix has one double eigenvalue, $\lambda = 1$. The only fixed point is the point at infinity in the direction of the rotation vector. The only fixed plane is the plane at infinity.

Also, the line mapping matrix for 3D motion has a double eigenvalue $\lambda = 1$. There are two fixed lines.

1. The first fixed line is parallel to the rotation axis. It indicates that each motion with six parameters can be realized as a screw motion, i.e., a rotation around a given 3D line with four parameters, by angle ω (one parameter) and a translation along this line (one parameter).
   The position of the 3D line can be determined in the following manner: Rotate the coordinate system such that the direction of the rotation axis is parallel to the Z-axis; this requires two parameters. The resulting rotation together with the translation in the XY-plane, together three parameters, can be realized as a rotation around a point in the XY-plane. The remaining parameter is the shift along the Z-direction.

Exercise 6.24

2. (2) The second fixed line is the ideal line in the plane perpendicular to the rotation axis.

Analogously, a spatial motion can be characterized by its invariant quadric.

Theorem 6.4.5: ***Spatial motion and the singular dual quadric.*** A 3D collinearity is a motion only if it has the singular dual quadric Q_∞° as an invariant, cf. (5.168), p. 241.

The proof is similar to that in 2D. The fixed elements of 3D motions are collected in the Table 6.7.

Table 6.7 Fixed points, fixed lines, and fixed planes of general 3D motions. The 3-vector $\boldsymbol{r}$ is a vector parallel to the rotation axis satisfying $\boldsymbol{r} = \mathsf{R}\boldsymbol{r}$. The scalar α and the 3-vector $\boldsymbol{\beta}$ are arbitrary

	fixed points	fixed lines	fixed planes
3D translation $\boldsymbol{T}$	$\begin{bmatrix} \boldsymbol{\beta} \\ 0 \end{bmatrix}$	$\begin{bmatrix} \alpha\boldsymbol{T} \\ \boldsymbol{\beta} \end{bmatrix}$	$\begin{bmatrix} \mathsf{S}_T\boldsymbol{\beta} \\ \alpha \end{bmatrix}$
3D rotation $\mathsf{R}(\boldsymbol{r})$	$\begin{bmatrix} \boldsymbol{r} \\ \alpha \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{r} \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{r} \\ \alpha \end{bmatrix}$
3D motion $(\mathsf{R}(\boldsymbol{r}), \boldsymbol{T})$	$\begin{bmatrix} \boldsymbol{r} \\ 0 \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{r} \\ -\mathsf{S}(\boldsymbol{r})(\mathsf{I}_3 - \mathsf{R})^{-1}\boldsymbol{T} \end{bmatrix}, \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{r} \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix}$

Fixed Elements of Spatial Affine Mappings. At least one real eigenvalue is 1. The plane at infinity is a fixed plane for all affinities. Again, a collineation is an affinity only if it preserves parallelism of lines.

6.4.5.4 Fixed Elements of 1D Homographies

Homographies in 1D have either no or two fixed points. The inversion $x' = 1/x$, which we showed to be a 1D-homography in Sect. 6.2.3, p. 257, has two fixed points, whereas the

negative inversion $x' = -1/x$ has no fixed point. As in the previous cases, multiple real eigenvalues may occur, as, e.g., for the mapping $x' = 1/(2-x)$. Exercise 6.14

6.5 Perspective Collineations

6.5.1 Perspective Autocollineations 277
6.5.2 Conjugate Rotations 281

The central perspective mapping with a pinhole camera with planar sensor as it is straight line-preserving is evidently a strong motivation for using projective geometry. The main characteristic of a perspective mapping is the existence of a *projection centre*. However, as mentioned in the introduction, a collineation may refer to two different spaces or to the same space. For collineations between two different spaces the existence of a projection centre is only meaningful if the two spaces are embedded into a common space, for example the 3D space of an object and the 2D space of its image. They are both embedded into the same space during image capture; thus, image points are treated as 3D points, where the third coordinate is zero. As the modelling of cameras is a topic of Part III, we do not discuss this case here.

Perspective collineations, i.e., those collineations which have a projection centre, are a special case of collineations.

Definition 6.5.18: ***Perspective collineation.*** A perspective collineation is a collineation with a projection centre called $\mathcal{Z}$ such that a point $\mathcal{X}$, its image $\mathcal{X}'$, and the projection centre $\mathcal{Z}$ are collinear. ◇

Collineations referring to the same space are denoted as autocollineations and will be presented next.

6.5.1 Perspective Autocollineations

We now discuss perspective autocollineations $\mathbb{P}^n \to \mathbb{P}^n$, i.e., mappings of $\mathbb{P}^n$ onto itself. The general definition is the following.

Proposition 6.5.4: ***Perspective autocollineation, homology.*** The following two statements regarding a perspective collineation $\mathcal{H} : \mathbb{P}^2 \to \mathbb{P}^2$ are equivalent: (1) The collineation is a perspective collineation. (2) There exists a fixed point $\mathcal{Z}$ and a hyperplane $\mathcal{A}$ where all points $\mathcal{X} \in \mathcal{A}$ are mapped to themselves. Autocollineations have the general form

$$\mathsf{H} = \mathit{I}_n - \alpha \mathbf{Z}\mathbf{A}^\mathsf{T}\,, \tag{6.110}$$

where the point $\mathcal{Z}(\mathbf{Z})$, the hyperplane $\mathcal{A}(\mathbf{A})$ and the *modulus* α can be chosen arbitrarily. A perspective autocollineation is also called *homology*.

The name homology for a perspective mapping can be traced back to Poncelet 1822 (cf. Cremona, 1885, page IX, para. 5).

We discuss autocollineations in more detail only for $n = 2$ and $n = 3$.

6.5.1.1 Perspective Autocollineations in 2D

We start with the proof of the proposition for 2D autocollineations $\mathcal{H} : \mathbb{P}^2 \to \mathbb{P}^2$, thereby referring to its two statements.

Proof: Given statement (1), we prove (2): Given a collineation with fixed point z, we prove that there exists a line l whose points are projected through z to themselves. First, we observe that lines through z are mapped to themselves. As each line m through z is mapped to itself, this mapping of lines is projective, thus has either zero or two fixed points. As z is one fixed point, there is always a second

fixed point z_m on m. This also holds for a second line k through z, with a second fixed point z_k. The line l joining the two fixed points z_m and z_k is mapped point by point to itself, as the intersection point of any third line through z with l is mapped to itself.

Given (2), we now prove (1): Let a fixed point z and a fixed line l, in general not in coincidence with z, be given. All points on l are mapped to themselves. Then we need to show that the collineation

$$\mathsf{H} = \mathsf{I}_3 - \alpha \mathbf{z}\mathbf{l}^\mathsf{T} \tag{6.111}$$

with its modulus α is a perspective collineation. First, the point z is mapped to itself, as $\mathsf{H}\mathbf{z} = (\mathsf{I}_3 - \alpha\mathbf{z}\mathbf{l}^\mathsf{T})\mathbf{z} = (1 - \alpha(\mathbf{l}^\mathsf{T}\mathbf{z}))\mathbf{z} = \lambda\mathbf{z}$. Second, if a point x lies on l, we have $\mathbf{l}^\mathsf{T}\mathbf{x} = 0$ and again $\mathsf{H}\mathbf{x} = \mathbf{x}$. ◇

It is useful to distinguish whether the fixed point lies on the fixed hyperplane or not, since the number of parameters specifying the collineation is different in each case.

The general perspectivity in 2D has five degrees of freedom. They are specified by the fixed point z, the fixed line l, and one pair of corresponding points (y, y') to determine the modulus α.

Definition 6.5.19: ***Elation.*** A perspective autocollineation is an elation if the fixed point lies on the fixed line. It is represented by (6.111) with the constraint $\mathbf{z}^\mathsf{T}\mathbf{l} = 0$. ◇

The degrees of freedom of this mapping in 2D are four. They are specified by the fixed point z, the direction of the fixed line l, and one pair of points (y, y') to determine the modulus α.

There are numerous applications of perspective autocollineations. Important examples for autocollineations are repeated structures, the mirroring or the transformation between an object and its shadow when viewed in an image. We start with repeated structures.

An object plane may be composed of translated, identical structures, such as windows of a facade. The relation $\mathcal{H}_t$ between such repeated objects, say points x and x_t, is an elation: Its fixed point z is a point at infinity $\mathbf{z} = [\cos\phi, \sin\phi, 0]^\mathsf{T}$ in the direction ϕ of the translation, the fixed line is the line at infinity $\mathbf{l} = [0, 0, 1]^\mathsf{T}$, the modulus α is the size of the translation. With (6.111), p. 278 we obtain the transformation

$$\mathbf{x}_t = \mathsf{H}_t\mathbf{x} \tag{6.112}$$

translation as elation

with the transformation matrix

$$\mathsf{H}_t = \mathsf{I}_3 - \alpha \begin{bmatrix} \cos\phi \\ \sin\phi \\ 0 \end{bmatrix} [0\ 0\ 1] = \begin{bmatrix} 1 & 0 & -\alpha\cos\phi \\ 0 & 1 & -\alpha\sin\phi \\ 0 & 0 & 1 \end{bmatrix} . \tag{6.113}$$

Now, let us assume this pair of translated objects is projected to another plane by a homography H. How does the relation H_t between the equal structures change by this projection? Starting from the two projections $\mathbf{x}' = \mathsf{H}\mathbf{x}$ and $\mathbf{x}'_t = \mathsf{H}\mathbf{x}_t$, together with (6.112), we find

$$\mathbf{x}'_t = \mathsf{H}'_t\mathbf{x}' \tag{6.114}$$

conjugate transformation

with the transformation matrix

$$\mathsf{H}'_t = \mathsf{H}\mathsf{H}_t\mathsf{H}^{-1} . \tag{6.115}$$

Taking the transformation of points and lines into account, we see that the type of a given perspective autocollineation, namely an elation, is preserved.[4] The transformation H'_t is called the *conjugate transformation* of H_t; specifically it is called a *conjugate translation*. Therefore, we have the following corollary of the definition above.

Corollary 6.5.6: ***Conjugate translation.*** The homography of a translation (elation) H_t is a conjugate translation, i.e., it can be written as $\mathsf{H}'_t = \mathsf{H}\mathsf{H}_t\mathsf{H}^{-1}$, where H_t is a translation and H an arbitrary regular homography. The matrix H'_t results from H_t by a similarity transformation or conjugation with H, hence the name. The eigenvalues of H'_t are the same as those of H_t. The conjugate translation is an elation. ◇

[4] Here we can use $\mathbf{l}' = \mathsf{H}^{-\mathsf{T}}\mathbf{l} = \mathsf{H}^\ominus/|\mathsf{H}|\mathbf{l}$ as the factor α does not have a geometric meaning after applying H.

The following example demonstrates its use.

Example 6.5.28: ***Repeated structures on a facade.*** The basic elements of a planar object showing repeated structures are mutually related by an elation, see Fig. 6.16. As the relation between a plane in 3D and the image plane of a pinhole camera is a general straight line-preserving homography, we can use the properties of the translations at the object for identifying the elation in the image and for analysing elements of the repetitive patterns in the image, e.g. for architectural tasks. For this purpose, we need to identify the fixed points and the fixed line: The fixed line is the image[5] of the line at infinity of the plane in object space, the fixed point is the image[6] of the point at infinity of the spatial translation. The modulus can be determined from one pair of corresponding points. ◇

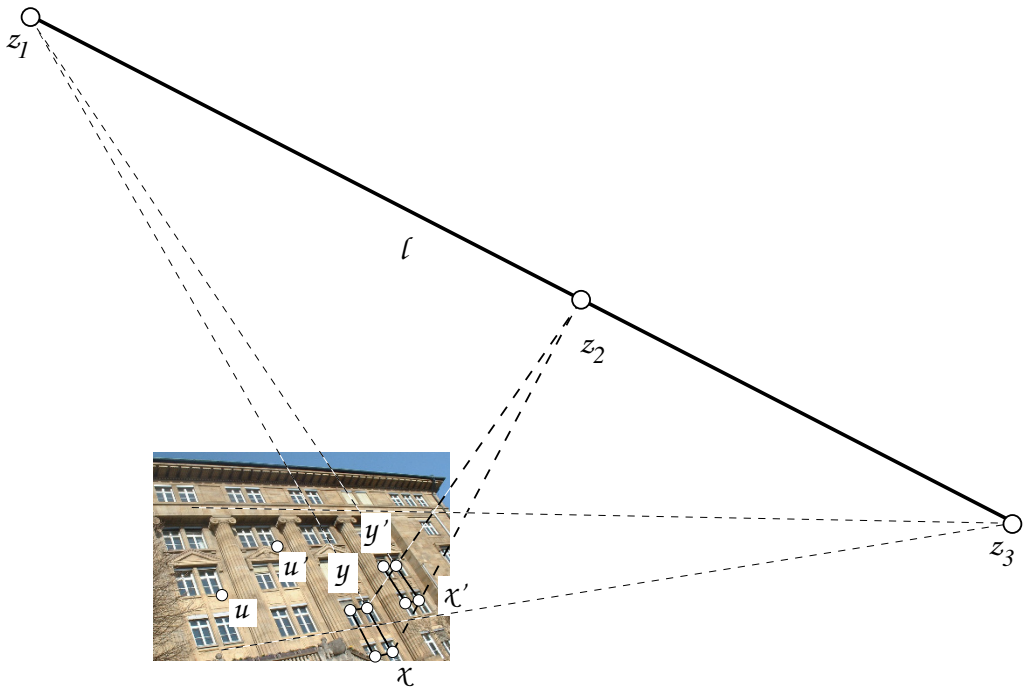


Fig. 6.16 Elation of repeated structures on a facade. Except for the top row and the right column, the windows together with the window crosses are within a plane. They are repeated in columns and rows but also in diagonals. They define the fixed points z_1, z_2, z_3 and the fixed line ℓ for three translations 1, 2, and 3. Given two pairs of points, say (x, x') and (y, y'), allows us to determine $z_2 = (x \wedge x') \cap (y \wedge y')$ and ℓ via z_1. This defines the conjugate translation H'_t which can be used to transfer another point, say u, to its diagonal neighbour u', as on the facade the two translations $x \to x'$ and $u \to u'$ are identical

Example 6.5.29: ***Reflectively symmetric object.*** Mirroring at the y-axis (cf. (6.7), p. 251) is a special perspectivity. Its fixed point is at infinity in the x-direction, $\mathbf{x} = [1, 0, 0]^\mathsf{T}$; its fixed line is the y-axis, $\mathbf{l} = [1, 0, 0]^\mathsf{T}$. ◇

Choosing $\alpha = 2$ and normalizing the vectors, we have the representation

$$\mathsf{H} = I_3 - 2\frac{\mathbf{x}\mathbf{l}^\mathsf{T}}{|\mathbf{x}|\,|\mathbf{l}|} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.116}$$

This special homology is called a *harmonic homology* (Mendonça et al., 2001), since the four points, the point z, its mirror point z', the midpoint t of z, z', and the fixed point x_∞ are

[5] The vanishing line of the plane, for vanishing elements; cf. Sect. 12.3.4, p. 529.

[6] The vanishing point of the 3D line, for vanishing elements; cf. Sect. 12.3.4, p. 529.

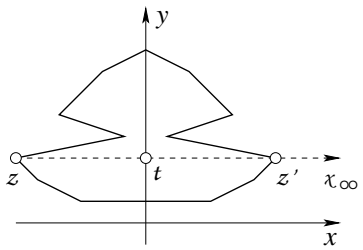


Fig. 6.17 Harmonic homology: Mirroring in 2D at an axis, here the y-axis, establishes a harmonic homology where the cross ratio $\mathrm{CR}(z, z'; t, \chi_\infty)$ is -1, cf. Sect. 6.4.3.1, p. 268

in harmonic position (see Fig. 6.17): the cross ratio, cf. (6.89), p. 268, is $\mathrm{CR}(z, z', t, \chi_\infty) = (x_z - x_t)/(x_{z'} - x_t) : (x_z - x_\infty)/(x_{z'} - x_\infty) = x_z/x_{z'} = -1$ using $\infty/\infty = 1$.

Example 6.5.30: ***Image of a reflectively symmetric planar object.*** The images of corresponding points of a reflectively symmetric object are related by a homology. The fixed point is the image of the point at infinity of the generating parallel lines, the fixed line is the image of the symmetry axis, cf. Fig. 6.18, left. ◇

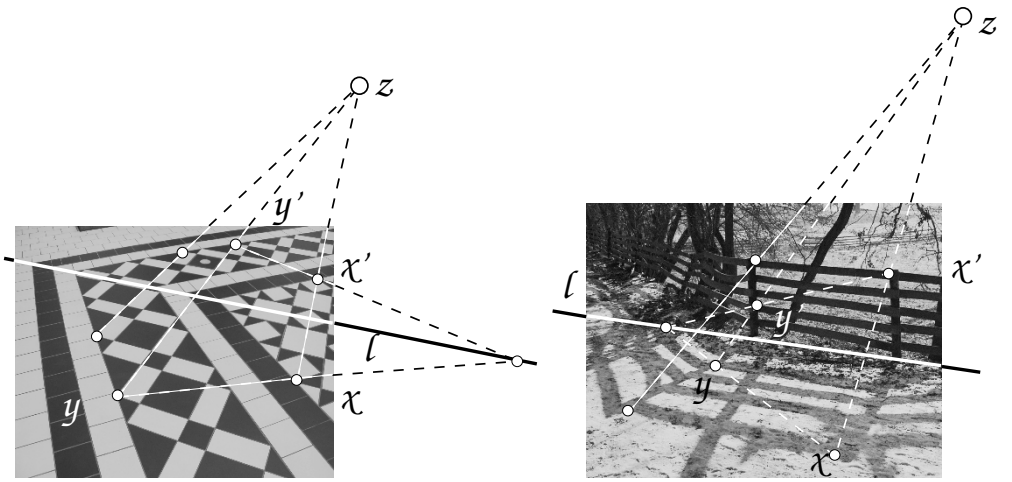


Fig. 6.18 Examples for 2D homologies. **Left:** Image of a symmetric figure. **Right:** Image of a planar object and its shadow cast onto a plane. Observe: for both examples with recovered fixed point z, fixed line ℓ, and a pair of points (χ, χ'), we can easily construct the image of an arbitrary point y, as the recovered lines joining $\chi \wedge y$ and $\chi' \wedge y'$ need to intersect on ℓ

Example 6.5.31: ***Shadow of a planar object.*** The images of corresponding points of a planar object and its shadow cast onto a plane are related by a homology. The fixed point is the image of the light source. The fixed line is the image of the intersection line of the two planes, cf. Fig. 6.18, right. ◇

Exercise 6.15

6.5.1.2 Perspective Autocollineations in 3D

In 3D, we have the perspective 3D autocollineation with fixed point $\mathcal{X}$ and fixed plane $\mathcal{A}$,

$$\mathsf{H} = I_4 - \alpha \mathbf{X}\mathbf{A}^\mathsf{T}. \tag{6.117}$$

mirroring at a plane

It is again a homology if the fixed point does not lie on the fixed plane; otherwise, it is an elation. A special homology is the mirroring at a plane $\mathcal{A}(\mathbf{A})$ with normal $\boldsymbol{N} = \boldsymbol{A}_h/|\boldsymbol{A}_h|$

and distance $S = -A_0/|\boldsymbol{A}_h$ to the origin:

$$\mathsf{H} = \begin{bmatrix} I_3 - 2\boldsymbol{N}\boldsymbol{N}^\mathsf{T} & 2S\boldsymbol{N} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \quad \text{with} \quad |\boldsymbol{N}| = 1\,. \tag{6.118}$$

The 3D homology has seven degrees of freedom, the 3D elation has six degrees of freedom. One pair of given points $(\mathscr{Y}, \mathscr{Y}')$ may determine the modulus α. Exercise 6.25

Example 6.5.32: ***Mapping with an ideal lens.*** Mapping with an ideal lens, see Sect. 6.2.2, p. 256, is an elation with the fixed point $\mathscr{X}([0,0,0,1])$, the centre of the lens, and the XY-plane $\mathscr{A}([0,0,1,0])$, the fixed plane. The mapping reads

$$\mathsf{H} = I_4 - \frac{1}{f}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}[0\,0\,1\,0] = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&-1/f&1\end{bmatrix} \tag{6.119}$$

where $f = \frac{1}{\alpha}$ is the focal length of the lens.

Although it is a special collineation, an ideal lens maps lines and planes to lines and planes, respectively. Furthermore, the depth Z of all points are transformed according to the well-known formula by Gauss for a thin lens:

$$\frac{1}{f} = \frac{1}{Z} - \frac{1}{Z'}\,. \tag{6.120}$$

◇

6.5.2 Conjugate Rotations

Lastly, we will discuss the relation between two perspective images taken from the same position in 3D space. This is the classical set-up for generating panoramas via stitching, see Fig. 6.19.

Fig. 6.19 Example for conjugate rotation. Images for stitching taken from the same position are related by a conjugate rotation. **Left and centre:** given images. **Right:** stitched image. Used software: Microsoft ICE

Without loss of generality, we assume the hole of the pinhole camera is at the origin of the coordinate system. As the distances of the scene points with coordinates $[X, Y, Z]$ from the camera centre do not affect their image, we assume that all points $\mathscr{X}_\infty$ lie on the plane at infinity, with 3D-directions $\mathbf{x} = [X, Y, Z]^\mathsf{T}$, see Fig. 6.20. The image coordinate system has its origin in the point $\mathscr{H}$ closest to the projection centre $\mathscr{Z}$; its axes are parallel to X, Y, Z. Image points have homogeneous coordinates $\mathbf{x}' = [x', y', 1]^\mathsf{T}$. We assume the pinhole camera is characterized by the distance c of the image plane from the hole $\mathscr{Z}$.

The first image is taken with the nonrotated pinhole camera. Then the mapping can be written as

$$x' = c\frac{X}{Z} \qquad y' = c\frac{Y}{Z} \tag{6.121}$$

or as

$$\mathbf{x}' = \mathsf{K}\mathbf{x}\,, \tag{6.122}$$

with the matrix, later termed camera matrix,

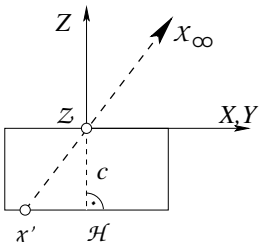


Fig. 6.20 Conjugate rotation during camera rotation. Observing points X_∞ at infinity with direction $[X, Y, Z]$ using a pinhole camera

$$\mathsf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{6.123}$$

The second image is taken with the rotated camera. If the rotation is R, we first rotate the directions $\mathbf{x}$ and then project; thus, we have the mapping

$$\mathbf{x}'' = \mathsf{K}R\mathbf{x} . \tag{6.124}$$

Thus the two images are related by

$$\mathbf{x}'' = \mathsf{H}\mathbf{x}' \tag{6.125}$$

with

$$\mathsf{H} = \mathsf{K}R\mathsf{K}^{-1} . \tag{6.126}$$

This special type of collineation is a *conjugate rotation*. It has four degrees of freedom: three rotation angles and the camera constant c. The eigenvalues of H, which is the similarity matrix transform of R, are the same as that of the pure rotation with rotation angle ω, namely $\{1, \mathrm{e}^{i\omega}, \mathrm{e}^{-i\omega}\}$. Consequently, we get the same fixed elements as with the rotation.

When K is a general homography,[7] the matrix $\mathsf{K}R\mathsf{K}^{-1}$ represents a general conjugate rotation. It has seven degrees of freedom, as can be shown using the Jacobian of the homography with respect to the twelve parameters, nine of which represent the matrix K and the remaining three the rotation R. This Jacobian has rank 7, indicating only seven of these twelve parameters are independent. On the other hand, if the parameter c of the camera is 1, the 3×3 homography matrix is a general 3×3 rotation matrix!

If the image contains a building we can derive the rotation of the camera w.r.t. the building coordinate system, cf. Sect. 12.3, p. 523, which can be used to rectify the image on one of the facades.

6.6 Projective Correlations

Projective correlations map points $\mathbf{X}$ to lines or, generally, to hyperplanes; thus, they are dualizing transformations, cf. Sect. 5.6.1, p. 229.

They have the general form

$$\mathcal{B} : \mathbb{P}^n \to \mathbb{P}^{*m} \quad \mathbf{A} = \mathsf{B}\mathbf{X} , \tag{6.127}$$

where $\mathbf{X} \in \mathbb{P}^n$ is a point mapped to the hyperplane with coordinates $\mathbf{A} \in \mathbb{P}^{*m}$. The $(m+1) \times (n+1)$ matrix B may be regular or singular.

[7] We will see that a general camera matrix is a special affinity with five parameters, namely without a rotation component.

We have already discussed two special cases:

- Determining the *polar* $\mathcal{A}$ of a point $\mathcal{X}$ w.r.t. a conic C or quadric Q, e.g., *polar plane of a 3D point*

$$\mathbf{A} = \mathsf{Q}\mathbf{X}\,. \tag{6.128}$$

  This transformation is called a polar correlation, where matrix $\mathsf{B} := \mathsf{Q}$ is regular and symmetrical.
- Determining the *dual* of a 3D point $\mathcal{X}$ what is a hyperplane. Here the mapping is represented by the unit matrix, e.g.,

$$\mathbf{A} = \mathsf{I}_4\mathbf{X}\,. \tag{6.129}$$

We introduce here two additional forms of correlation which will be used later:

- A special case in 2D is the classical Hough transformation in image processing. It may be used to automatically identify the line $\mathcal{l}$ of collinear points $\mathcal{x}_i, i = 1, ..., I$. The idea is to replace each point $\mathcal{x}_i(\mathbf{x}_i), \mathbf{x}_i \in \mathbb{P}^2$ by its dual line $\mathcal{l}_i$ in the space $\mathbb{P}^{*2}$ of lines. These lines intersect in a point $\mathcal{x}(\mathbf{x}), \mathbf{x} \in \mathbb{P}^{*2}$, which is the dual of the sought line in the primary space $\mathbb{P}^2$. In Fig. 6.21, p. 283, three points $\mathcal{x}_i$ on a line $\mathcal{l}$ are mapped to the lines $\mathcal{l}_i$ in the parameter space of lines. This allows us to identify collinear points $\mathcal{x}_i$ by clustering in the Hough space, i.e., by finding the point in the Hough space where the largest number of lines $\mathcal{l}_i$ meet. One way to realize the Hough transformation is the following: Given the point $\mathcal{x}_i(x_i, y_i)$, all lines through it satisfy $y_i = mx_i + k$ or $k = -x_i m + y_i$ or $x_i m + k_i - y = [x_i, 1, -y_i][m, k, 1]^\mathsf{T}$, which is a linear function $k = f(m)$. With the homogeneous coordinates $\mathbf{l}_i = [x_i, 1, -y_i]^\mathsf{T}$ of this line, we may represent it as (omitting the index i)

$$\mathcal{T} : \mathbb{P}^2 \to \mathbb{P}^{*2} \qquad \mathbf{l} = \begin{bmatrix} x \\ 1 \\ -y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathsf{T}\mathbf{x}\,. \tag{6.130}$$

  Though this mapping is mathematically appealing, it has the disadvantage that vertical lines cannot be represented. Therefore, in applications the Hesse form of a line is often preferred.

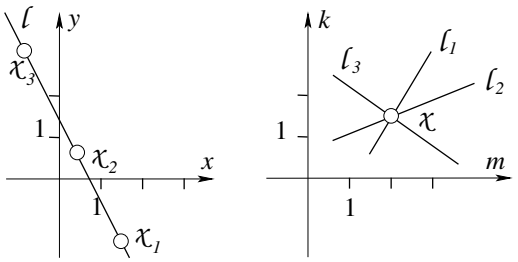


Fig. 6.21 Projective correlation: The Hough transformation. All points $\mathcal{x}_i$ on the line $y = mx + k = -2x + 3/2$ map to lines $\mathcal{l}_i$ with representation $k = -x_i m + y_i$ through the point $(m, k) = (2, 3/2)$

- When we discuss the geometry of the image pair, we will find the epipolar line in one image to be a singular correlation of the corresponding point in the other image, cf. Sect. 13.2.5, p. 562.

6.7 Hierarchy of Projective Transformations and Their Characteristics

We now want to provide a compact overview of the different projective transformations discussed so far, which are of concern in various contexts of geometric image analysis, especially for transferring points and lines between 3D spaces and their images.

All transformations have group structure, with the matrix multiplication of their matrix representation as the group operation. Moreover, the transformations form Lie groups, which are differentiable. This is important when estimating their parameters.

Since all of them are special projective transformations, they have the following characteristics in common: they preserve straight lines, incidences, the tangency, and the cross ratio.

The set of all transformations – shown in their hierarchy in Figs. 6.22 and 6.23 – are mappings of a projective space onto itself or onto its dual. They can be distinguished by their type and by their fixed elements; note the following comments.

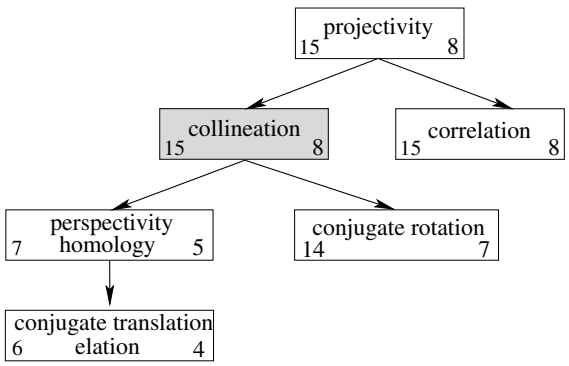


Fig. 6.22 Hierarchy of projective transformations in decreasing order of their degrees of freedom. They are shown in the boxes: on its left side for 3D and on the right for 2D space. For embedded perspectivities, additional degrees of freedom are necessary. We will discuss this aspect in Part III, p. 439.

- *Collineations* map points to points, such as $\mathbf{x}' = \mathsf{H}\mathbf{x}$, and lines to lines. They have the property that they transform every figure into a projectively equivalent figure, leaving all its projective properties invariant. Mappings of hyperplanes to hyperplanes, such as $\mathbf{l}' = \mathsf{H}^{\circ}\mathbf{l}$, sometimes are called *dual collineations*, cf. Ressl (2003). We find 2D collineations when describing mappings between planar objects.
- *Correlations* map points to hyperplanes, and hyperplanes to points. For example, determining the polar plane w.r.t. a quadric $\mathbf{A} = \mathsf{Q}\mathbf{X}$ is a correlation. Mappings of hyperplanes to points sometimes are called *dual correlations*, such as the one determining the pole w.r.t. a quadric $\mathbf{X} = \mathsf{Q}^{\circ}\mathbf{A}$, cf. Ressl (2003) and Pottmann and Wallner (2010). We will find correlations when analysing the image pair and the image triplet (Part III).
- *Conjugate collineations* appear in two specializations, namely conjugate rotations $\mathsf{K}\mathsf{R}\mathsf{K}^{-1}$ for the generation of a panorama and conjugate translations for analysing regular patterns or symmetries in an image. The degrees of freedom depend on the type of collineation.
- *Perspectivities*, i.e., collineations with a projection centre z and a fixed line ℓ or hyperplane in general, which are also called *homologies* (cf. Cremona, 1885, Page IX, para. 5). In general we have z distinct from ℓ. If z is on ℓ we have a special homology, namely an *elation*. We will show that all full rank projective mappings from $\mathbb{P}^3$ to $\mathbb{P}^2$, which are used to model straight line-preserving cameras, are perspectivities, since we always can derive the projection centre (cf. (12.45), p. 475).

- *Polarities* are correlations which map geometric elements to their dual and vice versa (Sect. 5.7.1.4, p. 238).

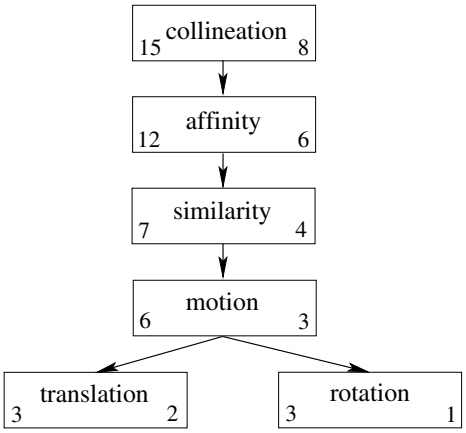


Fig. 6.23 Specialization hierarchy of collineations, together with their degree of freedom in 3D and 2D

We distinguish between the basic collineations by their invariants, starting from the general collineations, whose complete characteristics are given above:

- collineations preserve the cross ratio,
- collineations are *affinities* if and only if they preserve parallelism,
- collineations are *similarities* if and only if they preserve angles or distance ratios,
- collineations are *motions* if they preserve distances,
- collineations are *rotations around the origin* if direction differences as seen from the origin are preserved,
- collineations are *translations* if and only if directions of lines or between points are preserved.

Other special projectivities may occur depending on the application, especially when chaining two of the transformations listed above.

6.8 Normalizations of Transformations

Similarly to homogeneous vectors for representing geometric entities, homogeneous matrices can be normalized. We distinguish between the following three normalizations: Euclidean normalization, spherical normalization, and spectral normalization.

Euclidean Normalization of Transformations. Euclidean normalization of a homography $\mathcal{H} : \mathbb{P}^n \to \mathbb{P}^m$ refers to a partitioning of the matrix in accordance with the partitioning of the homogeneous vectors,

$$\mathbf{x}' = \mathsf{H}\mathbf{x}\,, \quad \text{or} \quad \begin{bmatrix} \boldsymbol{x}'_0 \\ x'_h \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & s \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix} . \tag{6.131}$$

Euclidean normalization of the homography normalizes the last element s to $+1$:

$$\mathsf{H}^e = \frac{1}{s}\mathsf{H}\,. \tag{6.132}$$

Such a normalization is very useful in case of affinities $\mathcal{A}$ where the projective part $\boldsymbol{p}$ of the transformation is zero, as then Euclideanly normalized homogeneous vectors are mapped

to Euclideanly normalied homogeneous vectors $\mathbf{x}'^e = \mathsf{A}^e\mathbf{x}^e$, cf. Sect. 6.2.1, p. 250. Since affinities form a group, Euclidean normalization is a natural choice, as for $\mathsf{A} = \mathsf{A}_1\mathsf{A}_2$ we have $\mathsf{A}^e = \mathsf{A}_1^e\mathsf{A}_2^e$.

Spherical Normalization of Transformations. When representing transformations H using the vector $\mathbf{h} = \mathrm{vec}\mathsf{H}$ of their elements, it is an intuitive choice to spherically normalize this vector. As the vector norm $|\boldsymbol{h}|^2 = \sum_k h_k^2$ is identical to the Frobenius norm $||\mathsf{H}||^2 = \sum_{ij} H_{ij}^2$, we have the spherical normalization

$$\mathsf{H}^s = \mathsf{N}(\mathsf{H}) = \frac{1}{||\mathsf{H}||}\mathsf{H}\,. \tag{6.133}$$

This spherical normalization is automatically realized in estimation procedures where the vector $\mathbf{h}$ is an eigenvector or singular vector of a problem-specific matrix.

The disadvantage of this normalization is that there is no group structure for spherically normalized matrices, since for $\mathsf{H} = \mathsf{H}_1\mathsf{H}_2$, we have $\mathsf{H}^s \neq \mathsf{H}_1^s\mathsf{H}_2^s$.

Spectral Normalization of Transformations. If homographies are regular, thus $\mathcal{H} : \mathbb{P}^n \leftrightarrow \mathbb{P}^n$, the determinant of the matrix can be enforced to be ± 1. Spectrally normalized homographies H^σ form a group, since $1 = |\mathsf{H}_1^\sigma\mathsf{H}_2^\sigma| = |\mathsf{H}_1^\sigma|\,|\mathsf{H}_2^\sigma|$. In general we have $|c\mathsf{H}| = c^{n+1}|\mathsf{H}|$, so the normalization can be realized as

$$\mathsf{H}^\sigma = \frac{1}{\sqrt[n+1]{\mathrm{abs}(|\mathsf{H}|)}}\mathsf{H}\,, \tag{6.134}$$

which takes into account that the determinant may be negative and preserves the sign of the determinant.

6.9 Conditioning

It is generally recommended to improve the condition of any geometric computation, especially if the absolute coordinate values are far from 1, e.g., when geocentric coordinates are involved. As an example, take the join of two points $\mathcal{X}$ and $\mathcal{Y}$ far from the origin:

$$\mathbf{l} = \mathbf{x} \times \mathbf{y} = \begin{bmatrix} 10^k + 1 \\ 10^k \\ 1 \end{bmatrix} \times \begin{bmatrix} 10^k \\ 10^k - 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 10^k - (10^k - 1) \\ -(10^k + 1) + 10^k \\ (10^k + 1)(10^k - 1) - 10^k\,10^k \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \tag{6.135}$$

which, due to the difference $(10^k + 1)(10^k - 1) - 10^k\,10^k$ of numbers with $2k + 1$ digits in the third element, only leads to the correct result if at least $2k + 1$ digits are used for calculation.

The effect of rounding errors needs to be taken into account in all geometric computations, for constructions, and transformations, or during estimation. The effect can drastically be diminished by suitably transforming the geometric entities into another coordinate system, performing the geometric operation, and then transforming them back.

The transformation $\breve{\mathbf{x}} = \mathsf{T}\mathbf{x}$ for conditioning should guarantee that the points approximately have centroid $\mathbf{0}$ and their average distance from the origin is ≤ 1. This decreases the condition number of the matrices and thus stabilizes the results, cf. Golub and van Loan (1996) and Hartley (1997a).

For example, assume we have two points $\mathcal{X}_1$ and $\mathcal{X}_2$ from a set $\{\mathcal{X}_j\}$ of points not containing a point at infinity, and we want to determine the joining line $\mathcal{l} = \mathcal{X}_1 \wedge \mathcal{X}_2$. Then we may use the transformation

$$\breve{\mathbf{x}}_i = \mathsf{T}\mathbf{x}_i\,, \quad i = 1, 2\,, \tag{6.136}$$

with

$$\mathsf{T} = \begin{bmatrix} 1 & 0 & -\mu_x \\ 0 & 1 & -\mu_y \\ 0 & 0 & \max s_j \end{bmatrix}, \tag{6.137}$$

with the centroid μ_x, μ_y of the point set and the distances s_j of the points from the centroid. The joining line is determined in the transformed coordinate system by $\breve{\mathbf{l}} = \breve{\mathbf{x}}_1 \times \breve{\mathbf{x}}_2$ and transformed back into the original coordinate system, yielding $\mathbf{l} = \mathsf{T}/|\mathsf{T}|\,\breve{\mathbf{l}}$. Exercise 6.16

This type of transformation of coordinates, with the goal of increasing the numerical stability and the condition of the resulting matrices, is called *conditioning*.[8]

If we determine transformation matrices, the transformation needs to be expressed with conditioned matrices. For example, determining a homography $\mathbf{x}' = \mathsf{H}\mathbf{x}$ from a set of point pairs starts with conditioning the points with two matrices T and T' for the two point sets $\{\mathbf{x}_i\}$ and $\{\mathbf{x}'_i\}$. The equivalent transformation is $\breve{\mathbf{x}}'_i = \breve{\mathsf{H}}\breve{\mathbf{x}}_i$, with the conditioned transformation matrix

$$\breve{\mathsf{H}} = \mathsf{T}'\mathsf{H}\mathsf{T}^{-1}. \tag{6.138}$$

Its determination uses the conditioned points $\{\breve{\mathbf{x}}_i\}$ and $\{\breve{\mathbf{x}}'_i\}$. Finally the original transformation is determined from (6.138). If 2D lines, planes, or possibly 3D lines are to be used together with 2D points or 3D points, the same conditioning transformations must be used.

The effect of this type of conditioning onto the normal equations is demonstrated in Sect. 15.3.3.1, p. 657 on block adjustment.

6.10 Exercises

Basics

1. (1) Name four examples of affine transformations which are not specializations of each other.
2. (1) Show that the translation matrix T in (6.5), p. 251 actually is a homogeneous transformation matrix, i.e., scaling of T does not change the mapping $(x \to x', y \to y')$.
3. (2) Show that, in 2D, mirroring of a point at a line through the origin, represented by the normalized direction $\mathbf{d}$, is achieved by the homography

$$\mathsf{H} = I_3 - 2\mathbf{d}\mathbf{d}^\mathsf{T}. \tag{6.139}$$

4. (2) Show that mirroring a point $\mathcal{X}$ at a line $\mathcal{l}$ is achieved by the mapping

$$\mathsf{H} = \begin{bmatrix} 1 + 2\,l_2{}^2 & -2\,l_2 l_1 & -2\,l_3 l_1 \\ -2\,l_2 l_1 & 1 + 2\,l_1{}^2 & -2\,l_3 l_2 \\ 0 & 0 & 1 + 2\,l_2{}^2 + 2\,l_1{}^2 \end{bmatrix}. \tag{6.140}$$

   Give all fixed points and fixed lines. *Hint:* use (7.18), p. 295.
5. (2) Prove that the mapping

$$\mathcal{H} : \mathbb{P}^2 \to \mathbb{P}^1 \quad \mathbf{x}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}, \tag{6.141}$$

   with the homogeneous coordinates $\mathbf{x} = [u, v, w]^\mathsf{T}$ and $\mathbf{x}' = [u, v]^\mathsf{T}$, can be represented as a singular collineation by embedding the points $\mathcal{X}'$ into a 2D projective space. Show

[8] In Hartley's paper (1997a) he calls this procedure *normalization*. In view of the classic concepts used in numerical analysis, we use the term *conditioning*, reserving the term normalization for the standardization of the length of homogeneous vectors.

that collinear points are mapped to collinear points. How can you generalize the proof to a mapping $\mathbf{x}' = \mathsf{P}\mathbf{x}$ with a general 2×3 matrix P?

6. (1) Given the line $l(x/6 + y/3 = 1)$, show that translating the line by $\boldsymbol{t} = [2, 1]^\mathsf{T}$ leads to $l'(x/10 + y/5 = 1)$ using (6.46).
7. (1) Determine the coordinates of the points shown in the figures on pp. 263 and 264 and verify the figure.
8. (1) Show that the centre $\mathbf{x}_0$ of a conic $\mathcal{C}$ is the pole of the polar line at infinity, $\mathbf{l}_\infty$. Now assume this conic $\mathcal{C}$ is mapped to $\mathcal{C}'$ via a projective mapping. Determine the image $\mathbf{x}'_0$ of the centre $\mathbf{x}_0$ using only $\mathcal{C}'$ and the image $\mathbf{l}'_\infty$ of the line at infinity.
9. (3) Explain why there exists a 2D homography with three fixed points and with three distinct eigenvalues. How would you construct such a homography with the fixed points $\boldsymbol{x}_1^f = [0, 0]^\mathsf{T}$, $\boldsymbol{x}_2^f = [1, 0]^\mathsf{T}$ and $\boldsymbol{x}_3^f = [0, 1]^\mathsf{T}$?
10. (2) Prove that a collineation is a motion if it preserves the singular dual conic $\mathsf{C}_\infty^* = \mathrm{Diag}([1, 1, 0])$. *Hint:* Use (6.59), p. 260.
11. (1) Transfer the point p to p' and p'' in the following figure using the paper strip construction, see Figs. 6.14 and 6.24.

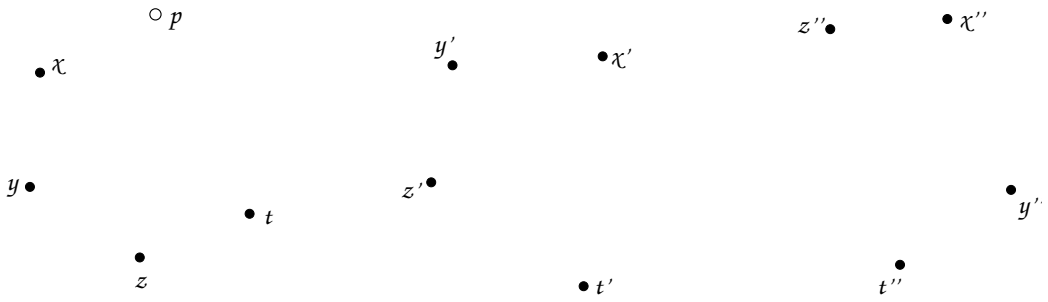


Fig. 6.24 Exercise: Paper strip construction. Transfer point p to construct p' and p''

12. (3) Equation (6.106), p. 275 is only valid for nonhorizontal planes, as, otherwise, the space is only two-dimensional, since the second vector $[-T_2, T_1, 0, 0]^\mathsf{T}$ vanishes. *Hint:* Use the fact $\mathcal{S}(\boldsymbol{T})\boldsymbol{T} = \mathbf{0}$ to identify the three rows of $\mathcal{S}(\boldsymbol{T})$ as vectors perpendicular to $\boldsymbol{T}$.
13. (3) Construct a spatial motion by a screw motion. The given 3D line is $\mathbf{L} = [-3/5, 0, 4/5, 4/5, 0, 0]^\mathsf{T}$, the rotation angle is $\omega = 90°$, and the translation along the 3D line is 2. Verify the motion: (1) the 3D line intersects the XY-plane in the point $[0, 1, 0]^\mathsf{T}$ and this point is moved to $[-6/5, 0, 8/5]^\mathsf{T}$, (2) the origin is moved to $\mathcal{O}'$ with coordinates $[X, Y, Z]^\mathsf{T}$, where X is the distance of the 3D line from the origin, and the point $[0, Y, Z]^\mathsf{T}$ is the point on the 3D line closest the origin. Determine the coordinates of $\mathcal{O}'$.
14. (1) Express the mapping $x' = 1/(2 - x)$ (see Sect. 6.4.5.4, p. 276) using homogeneous coordinates and show that the homography matrix has the double eigenvalue 1. Draw a graph, and explain the situation.
15. (1) Given a perspectivity of a planar scene with a regular pattern, such as a facade, which allows us to determine the image $\mathbf{l}'_\infty = [l'_1, l'_2, l'_3]^\mathsf{T}$ of the vanishing line of the plane, show that

$$\mathsf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l'_1 & l'_2 & l'_3 \end{bmatrix}$$

maps the image such that lines which are parallel in 2D are mapped to parallel lines. What type is the remaining transformation of the image to the true planar scene?

16. (1) Using the example of (6.135) with $k = 8$, show that, without conditioning, 16 digits are not sufficient to obtain the correct solution (e.g., using MATLAB), and that

conditioning both point vectors $\mathbf{x}$ and $\mathbf{y}$ with some suitable T, with $\mu_x = \mu_y = 10^8$ and $\max_s = 2$, yields the correct result.

17. (1) Can you generate a regular homography with four 2D points in the general position as fixed points which is not the identity transformation? Give a reason and, if possible, a solution.
18. (3) Two bumper cars $\mathcal{A}$ and $\mathcal{B}$ wait for the beginning of their ride. Together with that of the cashier $\mathcal{C}$, their positions are shown in Fig. 6.25. Both drivers are in the centres of the coordinate systems of the two cars. The task is to determine the direction in which each driver sees the other one and the cashier.

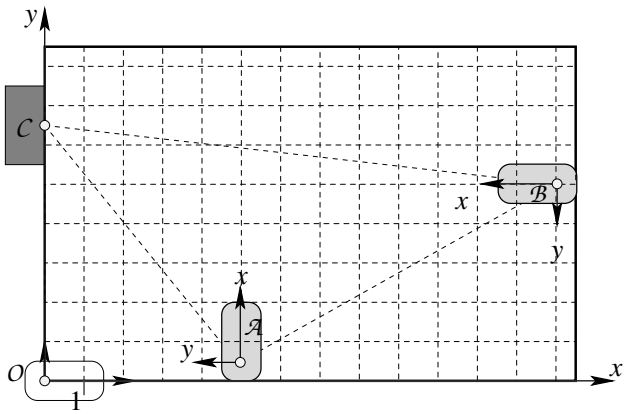


Fig. 6.25 Two bumper cars $\mathcal{A}$ and $\mathcal{B}$ viewing the cashier $\mathcal{C}$ (in scale)

a. Determine the displacements $\mathcal{M}_A$ and $\mathcal{M}_B$ of the reference coordinate system in the two car systems.
b. Determine the directions in which both drivers see the cashier, by expressing the coordinates of the cashier in the car systems. Compare the result with the directions in the figure.

Now both cars move. Turns are performed on the spot. Car $\mathcal{A}$ performs the following three moves: (1) 2 m ahead, (2) right turn by 45°, (3) 1 m ahead. Person $\mathcal{B}$ performs the following six moves: (1) right turn by 45°, (2) 1 m ahead, (3) right turn by 45°(4) 3 m backwards, (5) left turn by 90°, (6) 3 m ahead. Answer the following questions:

c. Determine the composite motion of both cars.
d. In what directions do the drivers see the cashier?
e. In what directions do the two drivers see each other?
f. What is the distance between the drivers now?

Proofs

19. (3) Prove (6.53), p. 259. *Hint:* Use (A.47), p. 772.
20. Prove (6.46), p. 258. *Hint:* Generalize the proof for the transformation of the cross product in (A.46), p. 772.
21. (2) Prove (6.90). *Hint:* Use the 1D homography $\mathbf{x}'_i = \mathsf{H}\mathbf{x}_i$ with homogeneous coordinates $x_i = [u_i, v_i]^\mathsf{T}$ and write the coordinate differences in (6.89), p. 268, as determinants, e.g.,

$$x_i - x_j = \begin{vmatrix} x_i & x_j \\ 1 & 1 \end{vmatrix} = \begin{vmatrix} u_i & u_j \\ v_i & v_j \end{vmatrix} . \tag{6.142}$$

22. (2) Prove ((6.96), p. 271). *Hint:* Assume $\mathbf{h}' = \mathbf{l}'_3 + \alpha \mathbf{l}'_1$ and determine α using the cross ratio.
23. (1) Prove (6.103), p. 274 and show that each motion can be represented as a rotation around the fixed point.
24. (2) Prove the expression for the fixed line of a 3D motion in Table 6.7, p. 276.
    a. Check $\mathbf{L}^f = \mathsf{H}_L \mathbf{L}^f$. *Hint:* Use $RS(\boldsymbol{r}) = S(\boldsymbol{r})R$ (Why does this hold?)
    b. (Constructive proof) Define a motion M, such that the rotation with rotation axis $\boldsymbol{r}$ is $R(\boldsymbol{r})$ and that the point $\boldsymbol{T}_0$ is a fixed point.
    c. How many degrees of freedom does M have? Why? An additional translation by $k\boldsymbol{r}$ changes the rotation axis into a screw axis. How many degrees of freedom does this combined motion therefore have?
    d. Relate the fixed point $\boldsymbol{T}_0$ of M to the translation vector $\boldsymbol{T}$ in the standard motion matrix.
    e. Determine the rotation axis.
25. (1) Prove (6.118), p. 281.

Chapter 7
Geometric Operations

7.1 Geometric Operations in 2D Space 292
7.2 Geometric Operations in 3D Space 299
7.3 Vector and Matrix Representations for Geometric Entities 311
7.4 Minimal Solutions for Conics and Transformations 316
7.5 Exercises 322

This chapter discusses geometric operations of geometric entities. It covers a wide range of constructions, constraints, and functions based on points, lines, planes, conics, and quadrics, including elements at infinity. Geometric entities are assumed to be certain and the results to exist and to be unique, or at least to be finite in number. We tackle *constructions* and *constraints* in 2D more extensively to illustrate the ways to express them as functions of the basic homogeneous coordinates, which then can be generalized to 3D, for which we only discuss the basic relations. Nearly all expressions are multilinear forms. They allow us to directly derive the Jacobians necessary for deriving the uncertainty of the constructed entities or residuals of the constraints in Chap. 9, p. 343, on uncertain geometry. The Jacobians themselves can be treated as matrix representations of geometric entities, their rows and vectors giving a clear interpretation of the geometric entity w.r.t. the given coordinate system and allowing us to select linearly independent constraints. The chapter ends with closed form solutions for determining conics and transformations from a minimal set of geometric entities.

constructions and constraints

Spatial reasoning covers a large range of operations. Examples are

- testing certain spatial relations, such as incidence, parallelism, or orthogonality, say of lines and planes,
- constructing new entities, e.g., intersections or joins from given ones, such as the intersection of a plane with a 3D line or the join of a 3D point with a line,
- determining distances or angles between given entities, say between two 3D lines.

Transformations of geometric entities were discussed in the previous chapter. Checking qualitative relations between two entities, such as whether a point lies left or right of a line, are discussed in the context of oriented projective geometry in Chap. 9, p. 343.

Many of the operations, especially constraints and constructions, are linear in the given elements, as for the intersection of two lines $\mathbf{x} = \mathbf{l} \cap \mathbf{m} = \mathbf{l} \times \mathbf{m} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l}$, or the constraint of a point which is incident to a line $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$. Generally, for two generating elements $\mathbf{a}, \mathbf{b}$, we will obtain the bilinear form

$$\mathbf{n} = \mathbf{a} \circ \mathbf{b} = \mathsf{A}(\mathbf{a})\mathbf{b} = \mathsf{B}(\mathbf{b})\mathbf{a}\,, \tag{7.1}$$

or for three elements the trilinear form

$$\mathbf{m} = \mathbf{c} \circ_1 \mathbf{d} \circ_2 \mathbf{e} = \mathsf{C}(\mathbf{c}, \mathbf{d})\mathbf{e} = \mathsf{D}(\mathbf{d}, \mathbf{e})\mathbf{c} = \mathsf{E}(\mathbf{e}, \mathbf{c})\mathbf{d}\,. \tag{7.2}$$

bilinear and multilinear forms provide Jacobians

The matrices in these relations can be interpreted as Jacobians, e.g.,

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_7

$$\mathsf{A}(\mathbf{a}) = \frac{\partial \mathbf{n}}{\partial \mathbf{b}} = \frac{\partial(\mathbf{a} \circ \mathbf{b})}{\partial \mathbf{b}} \quad \text{or} \quad \mathsf{C}(\mathbf{c}, \mathbf{d}) = \frac{\partial \mathbf{m}}{\partial \mathbf{e}} = \frac{\partial(\mathbf{c} \circ_1 \mathbf{d} \circ_2 \mathbf{e})}{\partial \mathbf{e}} \,. \tag{7.3}$$

Therefore, we write all these relations such that the Jacobians can be seen directly in the algebraic expression as in (7.1) and (7.2). This will simplify uncertainty propagation when using the Taylor expansion of the multi-linear expression for the constructed entity.

Though the expressions for distances and angles are nonlinear in the given entities, they contain multilinear forms, which make the determination of the Jacobians easier.

We first discuss operations in 2D space thoroughly and then generalize to 3D. As the skew symmetric matrix of the coordinates of a 2D point or a 2D line is used to support operations between points and lines, we will develop similar matrices depending on the homogeneous coordinate vectors of 3D points and planes supporting the construction of 3D entities. The Plücker matrix of the 3D line will play a central role when expressing relations containing 3D lines.

The chapter closes with algorithms for determining transformations from a minimum number of other entities. These algorithms are useful for detecting outliers in situations with more than the minimum number of given entities.

The constructions and relations given in this chapter are developed such that they can be used for oriented as well as for nonoriented entities. Occasionally, we make this explicit, e.g., for distances and angles. In Chap. 9, p. 343 we will discuss oriented entities in more detail.

7.1 Geometric Operations in 2D Space

7.1.1 Constructions in 2D 292
7.1.2 Geometric Relations in 2D 295
7.1.3 Distances and Angles in 2D 297

We start with constructions of new entities from given ones, as this is the basis for all further operations such as checking spatial relations and determining distances and angles.

7.1.1 Constructions in 2D

We already discussed the intersection of two lines and the join of two points, cf. Sect. (5.1.2.4), p. 201. We also need to describe the geometric entities associated with lines and points, e.g., the *point on a line closest to the origin* or to another point, and the *line through the origin or through a general point perpendicular to a given line*. The operations which give us these constructions are collected in Table 7.1, p. 293; they are derived in the following sections.

Exercise 7.14

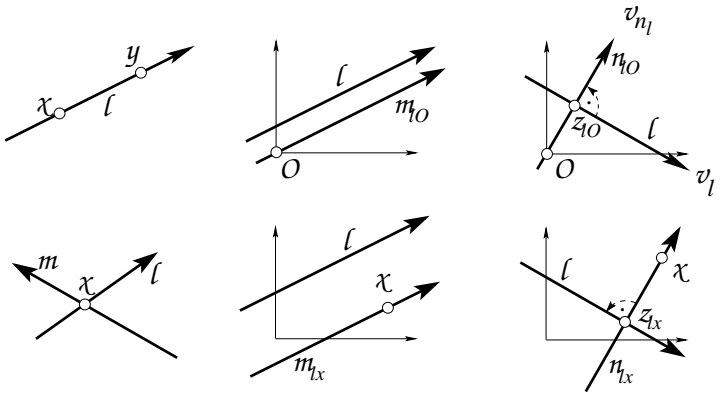
7.1.1.1 Intersection and Join

Intersection of two Lines. The *intersection* point $\mathcal{X}$ of two lines $\mathcal{l}$ and $\mathcal{m}$ has to fulfil $\mathbf{x}^\mathsf{T}\mathbf{l} = \mathbf{x}^\mathsf{T}\mathbf{m} = 0$; thus, the 3-vector $\mathbf{x}$ is perpendicular to $\mathbf{l}$ and to $\mathbf{m}$, leading to the result

$$\mathcal{X} = \mathcal{l} \cap \mathcal{m} : \quad \mathbf{l} \times \mathbf{m} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathbf{m} \times \mathbf{l} = -\mathsf{S}(\mathbf{m})\mathbf{l} \,. \tag{7.4}$$

Two parallel lines $[a, b, c_1]^\mathsf{T}$ and $[a, b, c_2]^\mathsf{T}$ intersect in the point at infinity $[b, -a, 0]^\mathsf{T}$ which is the direction of the lines. Thus, unlike with inhomogeneous coordinates, this situation does not lead to a singularity. If the two lines are identical, the intersection yields the

Table 7.1 Construction of 2D geometric entities. The first six forms are linear in the coordinates of the given entities, allowing simple variance propagation. The relations use the matrix $\mathsf{G}_3 = \mathrm{Diag}([1, 1, 0])$



given 2D entities	new entity	geometric construction	algebraic construction	eqn.
points $\mathcal{x}$, y	join $\mathcal{l}$	$\mathcal{l} = \mathcal{x} \wedge y$	$\mathbf{l} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x}$	(7.5)
lines $\mathcal{l}$, m	intersection $\mathcal{x}$	$\mathcal{x} = \mathcal{l} \cap m$	$\mathbf{x} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l}$	(7.4)
line $\mathcal{l}$	parallel m_{lO}	$m_{lO} \parallel \mathcal{l}$, $m_{lO} \ni O$	$\mathbf{m}_{lO} = \mathsf{G}_3\mathbf{l}$	(7.14)
line $\mathcal{l}$, point $\mathcal{x}$	parallel m_{lx}	$m_{lx} \parallel \mathcal{l}$, $m_{lx} \ni \mathcal{x}$	$\mathbf{m}_{lx} = -\mathsf{S}(\mathsf{S}_3\mathbf{l})\mathbf{x} = \mathsf{S}(\mathbf{x})\mathsf{S}_3\mathbf{l}$	(7.15)
line $\mathcal{l}$	normal n_{lO}	$n_{lO} \perp \mathcal{l}$, $n_{lO} \ni O$	$\mathbf{n}_{lO} = -\mathsf{S}_3\mathbf{l}$	(7.13)
line $\mathcal{l}$, point $\mathcal{x}$	normal n_{lx}	$n_{lx} \perp \mathcal{l}$, $n_{lx} \ni \mathcal{x}$	$\mathbf{n}_{lx} = -\mathsf{S}(\mathsf{G}_3\mathbf{l})\mathbf{x} = \mathsf{S}(\mathbf{x})\mathsf{G}_3\mathbf{l}$	(7.16)
line $\mathcal{l}$	foot point z_{lO}	$z_{lO} = \mathcal{l} \cap n_{lO} =$	$\mathbf{z}_{lO} = -\mathsf{S}(\mathbf{l})\mathsf{S}_3\mathbf{l}$	(7.17)
line $\mathcal{l}$, point $\mathcal{x}$	foot point z_{lx}	$z_{lx} = \mathcal{l} \cap n_{lx}$	$\mathbf{z}_{lx} = \mathsf{S}(\mathbf{l})\mathsf{S}(\mathbf{x})\mathsf{G}_3\mathbf{l}$	(7.18)

0-vector, as $\mathbf{x} = \mathbf{l} \times \mathbf{m} = \mathbf{0}$; thus, it is undefined. This fact will be used to check the identity of two lines.

Joining Two Points. By duality, we obtain the line $\mathcal{l}$ *joining* two points $\mathcal{x}$ and y,

$$\mathcal{l} = \mathcal{x} \wedge y : \quad \mathbf{x} \times \mathbf{y} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x}\,. \tag{7.5}$$

Identical points lead to an indefinite line.

The line, when constructed from two points, has a direction $\mathbf{d}_l$ pointing from $\mathcal{x}$ to y. If the points are exchanged we obtain the negative line parameters, reversing the direction of the line.

Intersection of Line and Conic. A line intersects a conic in up to two points. Let the intersection point $\mathbf{z} = \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$ be on the line $\mathcal{l} = \mathcal{x} \wedge y$, with $\mathcal{x}$ and y not identical, then we have the condition

$$\mathsf{C}(\mathbf{z}) = (\alpha\mathbf{x} + (1-\alpha)\mathbf{y})^\mathsf{T}\mathsf{C}(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) = 0 \tag{7.6}$$

or

$$(\mathbf{x}-\mathbf{y})^\mathsf{T}\mathsf{C}(\mathbf{x}-\mathbf{y})\alpha^2 + 2(\mathbf{x}-\mathbf{y})^\mathsf{T}\mathsf{C}\mathbf{y}\alpha + \mathbf{y}^\mathsf{T}\mathsf{C}\mathbf{y} = 0\,, \tag{7.7}$$

which is a quadratic equation in the unknown parameter α. We have three cases:

1. two distinct real solutions; then we have two intersection points,

2. two identical real solutions; then we have one tangent point counting as two intersection points, and
3. no real solution; then we have two complex intersection points.

7.1.1.2 Points at Infinity, Directions, Normals, and Parallel Lines

Direction and Normal of a Line. Each line ℓ has a point at infinity v_l which lies in the direction of the line. The point at infinity v_l is the intersection of the line ℓ with the line at infinity, $\ell_\infty([0,0,1]^\mathsf{T})$, hence $v_l = \ell \cap \ell_\infty$. So we obtain $\mathbf{v}_l = \mathbf{l} \times \mathbf{l}_\infty = -\mathbf{l}_\infty \times \mathbf{l} = -\mathsf{S}(\mathbf{l}_\infty)\mathbf{l}$, or

$$v_l : \quad \mathbf{v}_l = -\mathsf{S}_3\mathbf{l} \qquad \text{with} \qquad \mathsf{S}_3 \doteq \mathsf{S}(\mathbf{l}_\infty) = \mathsf{S}(\boldsymbol{e}_3^{[3]}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{7.8}$$

The skew matrix S_3 will be used for determining angles between lines in Sect. 7.1.3, p. 298.

The direction $\boldsymbol{l}_h$ of the normal of the line can be extracted from the line by

$$\boldsymbol{l}_h = \mathsf{P}_3\mathbf{l} \tag{7.9}$$

using the 2×3 projection matrix

$$\mathsf{P}_3 = [I_2 \,|\, \mathbf{0}] . \tag{7.10}$$

Therefore, the point at infinity, v_{n_l}, in the direction of the normal n_l of a line which lies perpendicular to ℓ is given by

$$v_{n_l} : \quad \mathbf{v}_{n_l} = \mathsf{G}_3\mathbf{l} = \begin{bmatrix} \boldsymbol{l}_h \\ 0 \end{bmatrix} , \tag{7.11}$$

with the matrix

$$\mathsf{G}_3 \doteq \mathsf{P}_3^\mathsf{T}\mathsf{P}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{7.12}$$

The directions $\mathbf{v}_l$ of the line ℓ and $\mathbf{v}_{nl}$ of its normal differ by a rotation of $+90°$, so that $\mathbf{v}_{n_l} = R_{+90°}\mathbf{v}_l$ or $\mathbf{v}_{n_l} = -R_{+90°}\mathsf{S}_3\mathbf{l} = \mathsf{G}_3\mathbf{l}$.

Observe, G_3 is equivalent to the singular dual conic C_∞^*, cf. (5.164), p. 241. We use the short notation G_3 for the product of the two projection matrices, as we will later generalize to cases where the conic is not the natural representation.

Parallel and Normal Lines Through a Point The line n_{lO} passing through the origin O and which is normal to a given line ℓ is the dual to the point at infinity of the line, cf. (5.135), p. 233 and Table 5.3, p. 230,

$$n_{lO} = \overline{v}_l : \quad \mathbf{n}_{lO} = \overline{\mathbf{v}}_l = -\mathsf{S}_3\mathbf{l} . \tag{7.13}$$

The line m_{lO} through O and parallel to ℓ is the dual of the point at infinity v_{nl} of ℓ,

$$m_{lO} = \overline{v}_{n_l} : \quad \mathbf{m}_{lO} = \overline{\mathbf{v}}_{n_l} = \mathsf{G}_3\mathbf{l} . \tag{7.14}$$

Parallel and Normal Line Through a Point. The line m_{lx} through $\mathcal{X}$ and *parallel* to ℓ is given by the join of this point with the point at infinity of this line,

$$m_{lx} : \quad \mathbf{m}_{lx} = \mathbf{x} \times \mathbf{v}_l = \mathsf{S}(\mathbf{x})\mathsf{S}_3\mathbf{l} = -\mathsf{S}(\mathsf{S}_3\mathbf{l})\mathbf{x} . \tag{7.15}$$

The line n_{lx} *normal* to the line ℓ and passing through the point $\mathcal{X}$ is given by the join of the point with the point at infinity in direction of the normal

$$n_{lx} : \quad \mathbf{n}_{lx} = \mathbf{x} \times \mathbf{v}_{n_l} = \mathsf{S}(\mathbf{x})\mathsf{G}_3\mathbf{l} = -\mathsf{S}(\mathsf{G}_3\mathbf{l})\mathbf{x} . \tag{7.16}$$

7.1.1.3 Foot Points

Foot Point on a Line. The point z_{lO} on a line ℓ closest to the origin O is obtained by the intersection of the line ℓ with its normal line n_{lO} passing through the origin,

$$z_{lO}: \qquad \mathbf{z}_{lO} = \mathbf{l} \times \mathbf{n}_{lO} = -\mathsf{S}(\mathbf{l})\mathsf{S}_3\mathbf{l}\,, \tag{7.17}$$

which does *not* depend linearly on the line parameters.

The point z_{lx} on a line ℓ closest to a point χ is given by the intersection of the line ℓ with the line n_{lx} normal to the line and passing through χ,

$$z_{lx}: \quad \mathbf{z}_{lx} = \mathbf{l} \times \mathbf{n}_{lx} = \mathsf{S}(\mathbf{l})\mathsf{S}(\mathbf{x})\mathsf{G}_3\mathbf{l} = -\mathsf{S}(\mathbf{l})\mathsf{S}(\mathsf{G}_3\mathbf{l})\mathbf{x}\,, \tag{7.18}$$

which is linearly dependent only on $\mathbf{x}$.

Foot Point on a Conic. Given a conic $\mathcal{C}$ and a point χ, we determine $y \in \mathcal{C}$ closest to χ. We determine two constraints for the unknown point y. We use the tangent line ℓ at the unknown point y: $\mathbf{l} = \mathsf{C}\mathbf{y}$ and the line n joining the given point χ and y, namely $\mathbf{n} = \mathsf{S}(\mathbf{x})\mathbf{y}$. We then obtain two constraints for the unknown point y:

1. The tangent ℓ must be perpendicular to n,

$$\mathbf{n}\cdot\mathbf{l} = 0: \qquad \mathbf{y}^\mathsf{T}\mathsf{S}(\mathbf{x})\mathsf{C}\mathbf{y} = 0\,. \tag{7.19}$$

   This constrains the point y to lie on the conic $\mathsf{S}(\mathbf{x})\mathsf{C}$.
2. The unknown point must lie on the conic

$$y \in \mathcal{C}: \qquad \mathbf{y}^\mathsf{T}\mathsf{C}\mathbf{y} = 0\,. \tag{7.20}$$

The unknown point is one of the up to four intersection points of the two conics $\mathsf{S}(\mathbf{x})\mathsf{C}$ and C from the two quadratic equations (7.19) and (7.20) for $\mathbf{y}$. Therefore, the solution requires us to determine the zeros of a fourth degree polynomial. A more efficient solution is to parametrize the point y on the conic and to find the point y^* with the shortest distance from χ numerically. Exercise 7.15

7.1.2 Geometric Relations in 2D

The relations in 2D, which we discuss next, are collected in Table 7.2.

Incidence. A line ℓ and a point χ are incident if and only if

$$\iota(\chi, \ell) \quad \Leftrightarrow \quad \mathbf{x}^\mathsf{T}\mathbf{l} = au + bv + cw = 0\,, \tag{7.21}$$

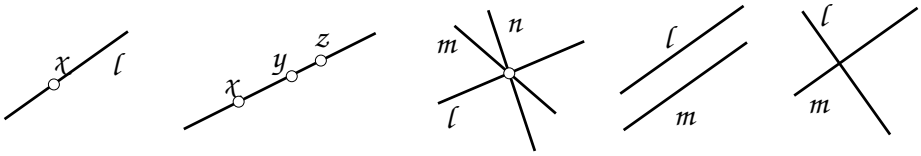
which immediately results from the implicit equation $ax + by + c = 0$ of a line. It shows the two 3-vectors $\mathbf{x}$ and $\mathbf{l}$ to be perpendicular. Given the line parameters, it gives the *implicit line equation*, i.e., constraints for all points sitting on the line. Given the point parameters, by analogy, it provides the *implicit point equation*, i.e., constraints for all lines passing through the point. Observe the symmetry of the relation (7.21), which results from the duality principle (Sect. 5.1.2.5, p. 203).

We will actually perform a statistical test of the residual,

$$w = \mathbf{x}^\mathsf{T}\mathbf{l}\,, \tag{7.22}$$

of the constraint, which should be 0. We express this as $w \stackrel{!}{=} 0$. Then we can directly extract the Jacobians $\partial w/\partial\mathbf{l} = \mathbf{x}^\mathsf{T}$ and $\partial w/\partial\mathbf{x} = \mathbf{l}^\mathsf{T}$ from (7.22), thus also directly from (7.21).

Table 7.2 Constraints between 2D geometric entities together with the degrees of freedom necessary for statistical testing. The complete plane (universe) is denoted by U, for G_3, cf. (7.11), p. 294



2D entities	name	relation	constraint	d.o.f.	eqn.
point $\mathcal{x}$, line $\mathcal{l}$	incidence	$\mathcal{x} \in \mathcal{l}$	$\mathbf{x}^\mathsf{T}\mathbf{l} = 0$	1	(7.21)
point $\mathcal{x}$, point $\mathcal{y}$	identity	$\mathcal{x} \equiv \mathcal{y}$	$\mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x} = \mathbf{0}$	2	(7.23)
line $\mathcal{l}$, line $\mathcal{m}$	identity	$\mathcal{l} \equiv \mathcal{m}$	$\mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l} = \mathbf{0}$	2	(7.24)
three points	collinearity	$\mathcal{x} \wedge \mathcal{y} \wedge \mathcal{z} \neq U$	$\lvert\mathbf{x}, \mathbf{y}, \mathbf{z}\rvert = 0$	1	(7.25)
three lines	concurrence	$\mathcal{l} \cap \mathcal{m} \cap \mathcal{n} \neq \emptyset$	$\lvert\mathbf{l}, \mathbf{m}, \mathbf{n}\rvert = 0$	1	(7.26)
line $\mathcal{l}$, line $\mathcal{m}$	parallelism	$\mathcal{l} \parallel \mathcal{m}$	$\mathbf{l}^\mathsf{T}\mathsf{S}_3\mathbf{m} = 0$	1	(7.27)
line $\mathcal{l}$, line $\mathcal{m}$	orthogonality	$\mathcal{l} \perp \mathcal{m}$	$\mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m} = 0$	1	(7.28)

Therefore, in the following, we will restrict ourselves to only giving either the constraint or the residual of the constraint.

Identity of Two 2D Entities. Two points $\mathcal{x}$ and $\mathcal{y}$ are identical if and only if the joining line is indefinite, thus, using the overloading of the wedge operator, cf. (5.93), p. 223,

$$\mathcal{x} \equiv \mathcal{y} \quad \Leftrightarrow \quad \mathbf{x} \wedge \mathbf{y} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x} = \mathbf{0}\,. \tag{7.23}$$

Only two of these constraints are linearly independent. Analogously, we have the constraint for the identity of two lines using (5.100), p. 224,

$$\mathcal{l} \equiv \mathcal{m} \quad \Leftrightarrow \quad \mathbf{l} \cap \mathbf{m} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l} = \mathbf{0}\,. \tag{7.24}$$

Collinearity and Concurrence of 2D Points and Lines. Three points $\mathcal{x}_i$ are collinear if the 3×3 determinant or the triple product of the three homogeneoeus 3-vectors vanishes:

$$\mathsf{collinear}(\mathcal{x}_1, \mathcal{x}_2, \mathcal{x}_3) \;\Leftrightarrow\; |\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3| = \langle \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \rangle = \mathbf{x}_1^\mathsf{T}(\mathbf{x}_2 \times \mathbf{x}_3) = 0\,. \tag{7.25}$$

The reason is that the determinant is the volume of the parallelepiped spanned by the three 3-vectors $\mathbf{x}_i$ starting at the origin in $[u, v, w]$-space. It is equivalent to requiring the first point $\mathbf{x}_1$ to be incident with the line $\mathbf{x}_2 \times \mathbf{x}_3$ through the other two points.

the line at infinity is a straight line

Observe: Points $[u, v, 0]^\mathsf{T}$ at infinity are lying on the line $[0, 0, 1]^\mathsf{T}$ at infinity. Three points at infinity are collinear; thus, the line at infinity is a straight line!

The collinearity of three points may also be expressed as $\mathcal{x} \wedge \mathcal{y} \wedge \mathcal{z} \neq U$, where U is the *universe*, i.e., the complete plane. This is because, following the definition of the Plücker coordinates in ((5.91), p. 222), $\mathcal{x} \wedge \mathcal{y} \wedge \mathcal{z}$ consists of all points which are a linear combination $\alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z}$ of the three coordinate vectors. This linear combination covers a straight line only if the three points are collinear.

By duality, three lines $\mathcal{l}_i$ intersect in one point if the 3×3 determinant or the triple product vanishes:

$$\mathsf{concurrent}(\mathcal{l}_1, \mathcal{l}_2, \mathcal{l}_3) \;\Leftrightarrow\; |\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3| = \langle \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3 \rangle = \mathbf{l}_1^\mathsf{T}(\mathbf{l}_2 \times \mathbf{l}_3) = 0\,. \tag{7.26}$$

This is the same as requiring the first line to pass through the intersection point of the other two.

Parallelism and Orthogonality of Two Lines. Two lines $\mathcal{l}$ and $\mathcal{m}$ are *parallel* $\mathcal{l} \parallel \mathcal{m}$

- if their intersection point $\mathcal{x} = \mathcal{l} \cap \mathcal{m}$ is at infinity: $(\mathbf{l} \times \mathbf{m}) \cdot \mathbf{l}_\infty = 0$, or
- if the point at infinity $\mathcal{v}_m$ of the line $\mathcal{m}$ is incident to $\mathcal{l}$: $\mathbf{l}^\mathsf{T}\mathbf{v}_m = 0$, or
- if the three lines $\mathcal{l}$, $\mathcal{m}$ and $\mathcal{l}_\infty = \boldsymbol{e}_3^{[3]}$ are concurrent: $|\mathbf{l}, \mathbf{m}, \mathbf{l}_\infty| = 0$.

All constraints are linear in the coordinates of $\mathcal{l}$ and $\mathcal{m}$. The last expression is the easiest to derive, cf. (7.8), p. 294, and leads to a constraint of the following bilinear form

$$\mathcal{l} \parallel \mathcal{m} \quad \Leftrightarrow \quad \mathbf{l} \cdot (\mathbf{m} \times \mathbf{l}_\infty) = -\mathbf{l}^\mathsf{T}\mathsf{S}_3\mathbf{m} = 0\,. \tag{7.27}$$

The minus sign is not relevant for testing.
Two lines $\mathcal{l}$ and $\mathcal{m}$ are *orthogonal* $\mathcal{l} \perp \mathcal{m}$

- in case their normals span an angle of 90°: $\boldsymbol{l}_h \cdot \boldsymbol{m}_h = 0$ or
- in case the points at infinity of their normals span an angle α of 90°: $\mathbf{v}_{nl} \cdot \mathbf{v}_{nm} = 0$ or
- in case the point at infinity of the normal of one line, say $\mathcal{v}_{nl}$, cf. (7.11), lies on the other line, say $\mathcal{m}$. This can be expressed with a bilinear constraint,

$$\mathcal{l} \perp \mathcal{m} \quad \Leftrightarrow \quad \mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m} = 0\,. \tag{7.28}$$

7.1.3 Distances and Angles in 2D

The calculation of distances between points is a bit more complex when using homogeneous than when using inhomogeneous coordinates. But the distances of points and lines from the origin demonstrate the similarities in the representation, namely the distinction between the Euclidean and the homogeneous parts of the coordinate vectors.

Distances between points and lines may obtain a sign, e.g., when we want to distinguish between the distance of points right and left of a line. Of course, the unsigned distance is obtained by taking the absolute value of the signed distance. We will give the signed distances, but discuss the meaning and the use of the signs in Chap. 9, p. 343. The expressions for signed distances are collected in Table 7.3.

Distance of Lines and Points from the Origin. The distance d_{lO} of a line $\mathcal{l}$ from the origin is given by

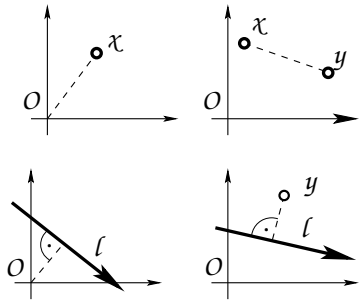
$$d_{lO} = \frac{l_0}{|\boldsymbol{l}_h|}\,. \tag{7.29}$$

This directly follows from the relation between the Hessian form of the line and the homogeneous coordinates of the line, cf. (5.37), p. 207. If the line is Euclideanly normalized, cf. (5.9), the distance d_{lO} reduces to the Euclidean part.[1]

The absolute value of the distance, $|l_0|/|\boldsymbol{l}_h|$, is the ratio of the absolute values of the Euclidean to the homogeneous parts of the homogeneous vector. This regularity transfers to all other distances from the origin, cf. Table 7.3, and was the reason for the particular partitioning of the homogeneous vectors into homogeneous and Euclidean parts, cf. Sect. 5.2, p. 205.

[1] Without mentioning we assumed the coordinates of the origin to be $[0, 0, 1]^\mathsf{T}$, thus to have positive third coordinate. If we would have chosen $[0, 0, -1]^\mathsf{T}$ as coordinates the distance would have been negative, cf. Sect. 9, p. 343.

Table 7.3 Signed distances between different 2D geometric entities. For the interpretation of their signs, cf. Chap. 9, p. 343: If the points have positive sign, i.e., if $\text{sign}(x_h) = \text{sign}(y_h) > 0$, all distances in the figures are positive



distances of	from origin O	from point y
2D point x	$d_{xO} = \dfrac{\lvert\boldsymbol{x}_0\rvert}{x_h}$	$d_{xy} = \dfrac{\lvert x_h\boldsymbol{y}_0 - y_h\boldsymbol{x}_0\rvert}{x_h y_h}$
2D line l	$d_{lO} = \dfrac{l_0}{\lvert\boldsymbol{l}_h\rvert}$	$d_{ly} = \dfrac{\mathbf{y}^\mathsf{T}\mathbf{l}}{\lvert y_h\boldsymbol{l}_h\rvert}$

Distances from Point to Point and from Point to Line. The distance d_{xy} between two points x and y is given by

$$d_{xy} = \frac{|x_h\boldsymbol{y}_0 - y_h\boldsymbol{x}_0|}{x_h y_h}\,, \tag{7.30}$$

which in the case of Euclidean normalization of the two coordinate vectors reduces to the absolute value of the difference of the Euclidean parts.

The distance d_{yl} of a point y from a line l is given by

$$d_{yl} = d_{ly} = \frac{\mathbf{y}\cdot\mathbf{l}}{|y_h\boldsymbol{l}_h|} \tag{7.31}$$

which in the case of Euclidean normalization of the two vectors reduces to their inner product.

direction angle

Angles. We start with the anticlockwise angle between the x-axis and a direction vector $\boldsymbol{d} = [x, y]^\mathsf{T}$. It is called the *direction angle* of $\boldsymbol{d}$ and lies in the range $[0, 2\pi)$, see Fig. 7.1a.

The classical arctan function yields $\phi_{Ox} = \arctan(y/x)$, which only is unique in the range $(-\pi/2, +\pi/2)$, thus in the first and fourth quadrants. In order to achieve unique values for all quadrants we need to exploit the signs of x and y, or use the two-argument function $\text{atan2}\,(y, x)$, which lies in the range $(-\pi, +\pi]$, and can be mapped to the desired range $[0, 2\pi)$. For positive x the two-argument function $\text{atan2}\,(\bullet, \bullet)$ specializes to the one-argument function $\arctan(\bullet)$, namely $\arctan(y/x) = \text{atan2}\,(y, x)$. Thus we have the direction angle

$$\phi_{Ox} = \mod\,(\text{atan2}\,(y, x) + 2\pi, 2\pi) \in [0, 2\pi)\,. \tag{7.32}$$

We now discuss angles between lines and between points as seen from the origin. An intuitive direct solution would be to determine the angle from the inner product of the two directions and take the arc-cosine. For small angles this solution is numerically unstable.

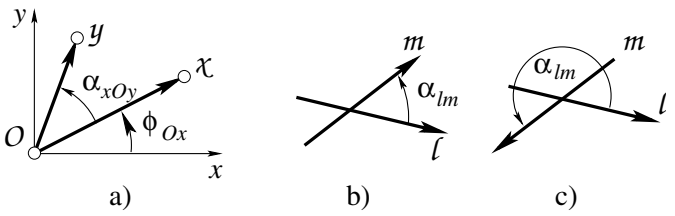


Fig. 7.1 Directions and angles between two lines. **a)** The direction angle of the direction $O\chi$ is the anticlockwise angle between the x-axis and the direction. It lies in the range $[0, 2\pi)$. **b)** The smallest angle between two nonoriented lines is an acute angle. In the figure it is also the anticlockwise angle between the directed lines. **c)** If we take the angle as the anticlockwise rotation of the first line to the second oriented line it will be in a range $[0, 2\pi)$. Similarly, we can determine the angle between the directions to two 2D points as seen from the origin, see c); the angle will lie in the range $[0, 2\pi)$

Furthermore, we only obtain the acute angle in the range $[0, \pi/2]$. If we define the angle as the anticlockwise rotation of the first to the second line, not taking their directions into account, we obtain an angle in the range $[0, \pi)$. If we furthermore take the directions of the lines into account, the resulting angle lies in the range $[0, 2\pi)$. We give the solution to the last case, as it is both numerically stable and easily allows us to derive the angles for the two previous cases.

The angle between two directed lines, say l and m, can be determined as the angle between the directions of their normal vectors. This requires deriving the direction angles ϕ_{n_l} and ϕ_{n_m} of normals, cf. (5.7), p. 198, leading to

$$\alpha_{lm} = \mod(\phi_{n_m} - \phi_{n_l} + 2\pi, 2\pi) \in [0, 2\pi)\,. \tag{7.33}$$

A simpler – also numerically stable – expression for the angle, however, as an explicit function of the given homogeneous coordinates, uses the two-argument version atan2 $(\cdot, \cdot)$ of the arctan function just once. With the matrices S_3 and G_3 introduced above (cf. (7.8) and (7.11)), we obtain

$$\alpha_{lm} = \alpha(\mathit{l}, \mathit{m}) = \mod\left(\text{atan2}\left(\mathbf{l}^\mathsf{T}\mathsf{S}_3\mathbf{m}, \mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m}\right) + 2\pi, 2\pi\right) \in [0, 2\pi). \tag{7.34}$$

The proof is left as an exercise. The angle α_{xOy} between the directions to two points χ and y, as seen from the origin O, is analogously given by

Exercise 7.13
Exercise 7.1

$$\alpha_{xOy} = \alpha(\chi, \mathit{y}) = \text{atan2}\left(\mathbf{x}^\mathsf{T}\mathsf{S}_3\mathbf{y}, \mathbf{x}^\mathsf{T}\mathsf{G}_3\mathbf{y}\right). \tag{7.35}$$

7.2 Geometric Operations in 3D Space

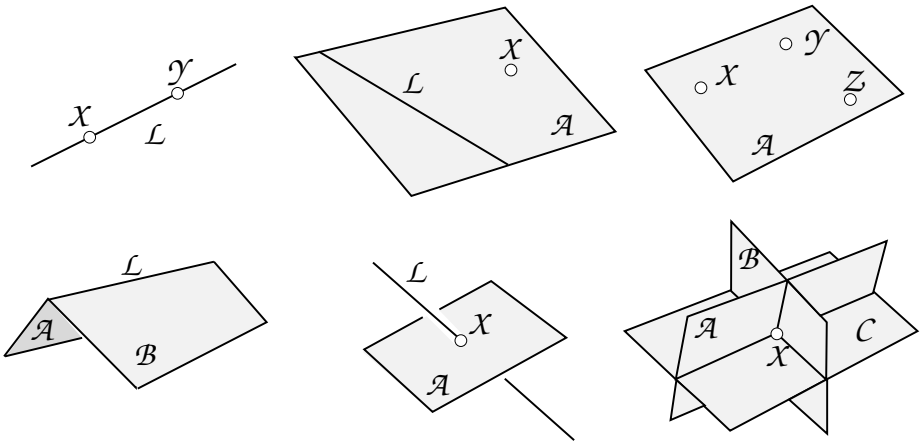
7.2.1 Constructions in 3D 300
7.2.2 Geometric Relations in 3D 304
7.2.3 Distances and Angles in 3D 308

Operations in 3D are more complex and show a richer structure. However, we only introduce the basic relations and leave the more specific ones to the exercises, p. 322. We again start with constructions and discuss constraints later.

7.2.1 Constructions in 3D

The main constructions are collected in the Table 7.4. They are again multilinear forms, and we want to make the Jacobians explicit by writing them as matrix-vector products. This will lead to a matrix named $\mathsf{\Pi}(\bullet)$ depending on a 4-vector representing the homogeneous coordinates either of a 3D point or of a plane, hence the name of the matrix: $\mathsf{\Pi}$ for *p*oints or *p*lanes.

Table 7.4 Constructions of 3D entities



3D entities	new element	construction	Eq.
points $\mathcal{X}$, $\mathcal{Y}$	line $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$	$\mathbf{L} = \mathsf{\Pi}(\mathbf{X})\mathbf{Y} = -\mathsf{\Pi}(\mathbf{Y})\mathbf{X}$	(7.38)
planes $\mathcal{A}$, $\mathcal{B}$	line $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$	$\mathbf{L} = \overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B} = -\overline{\mathsf{\Pi}}(\mathbf{B})\mathbf{A}$	(7.44)
line $\mathcal{L}$, point $\mathcal{X}$	plane $\mathcal{A} = \mathcal{L} \wedge \mathcal{X}$	$\mathbf{A} = \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathbf{L})\mathbf{X} = \overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X})\mathbf{L}$	(7.48)
line $\mathcal{L}$, plane $\mathcal{A}$	point $\mathcal{X} = \mathcal{L} \cap \mathcal{A}$	$\mathbf{X} = \mathsf{\Gamma}^{\mathsf{T}}(\mathbf{L})\mathbf{A} = \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\mathbf{L}$	(7.45)
points $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$	plane $\mathcal{A} = \mathcal{X} \wedge \mathcal{Y} \wedge \mathcal{Z}$	$\overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{X})\mathbf{Y})\mathbf{Z} = \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{Y})\mathbf{Z})\mathbf{X}$ $= \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{Z})\mathbf{X})\mathbf{Y}$	(7.49)
planes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$	point $\mathcal{X} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$	$\overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{A})\mathbf{B})\mathbf{C} = \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{B})\mathbf{C})\mathbf{A}$ $= \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathsf{\Pi}(\mathbf{C})\mathbf{A})\mathbf{B}$	(7.50)

7.2.1.1 Line from 3D Points and from Planes

Line Joining Two 3D Points. The line $\mathcal{L}$ passing through two points $\mathcal{X}$ and $\mathcal{Y}$ is given by

$$\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}: \quad \mathbf{L} = \mathbf{X} \wedge \mathbf{Y} = \begin{bmatrix} X_h \boldsymbol{Y}_0 - Y_h \boldsymbol{X}_0 \\ \boldsymbol{X}_0 \times \boldsymbol{Y}_0 \end{bmatrix} = -\mathbf{Y} \wedge \mathbf{X}\,, \tag{7.36}$$

following the definition of the Plücker coordinates for 3D lines in (5.65), p. 217. Observe the overloading of the wedge operator, now applied to two 4-vectors.

With the 6×4 matrix

$$\mathsf{\Pi}(\mathbf{X}) = \frac{\partial(\mathbf{X} \wedge \mathbf{Y})}{\partial \mathbf{Y}} = \begin{bmatrix} X_h I_3 & -\boldsymbol{X}_0 \\ \mathsf{S}(\boldsymbol{X}_0) & \mathbf{0} \end{bmatrix} = \left[\begin{array}{ccc|c} T & 0 & 0 & -U \\ 0 & T & 0 & -V \\ 0 & 0 & T & -W \\ \hline 0 & -W & V & 0 \\ W & 0 & -U & 0 \\ -V & U & 0 & 0 \end{array}\right] \tag{7.37}$$

depending on the 4-vector $\mathbf{X}^\mathsf{T} = [\boldsymbol{X}_0^\mathsf{T}, X_h] = [U, V, W, T]$, we can write the construction of a line as

$$\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}: \quad \mathbf{L} = \mathbf{X} \wedge \mathbf{Y} = \mathsf{\Pi}(\mathbf{X})\mathbf{Y} = -\mathsf{\Pi}(\mathbf{Y})\mathbf{X}\,. \tag{7.38}$$

If the two points $\mathcal{X}$ and $\mathcal{Y}$ are Euclideanly normalized, the line $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ is directed from $\mathcal{X}$ to $\mathcal{Y}$, as then $\boldsymbol{L}_h = \boldsymbol{Y} - \boldsymbol{X}$, cf. (5.65), p. 217.

The matrix $\mathsf{\Pi}(\mathbf{X})$ has rank 3, as

$$\mathsf{\Pi}(\mathbf{X})\mathbf{X} = \mathbf{0}, \tag{7.39}$$

or as

$$\mathsf{\Pi}^\mathsf{T}(\mathbf{X})\mathsf{\Pi}(\mathbf{X}) = \mathbf{X}^\mathsf{T}\mathbf{X}\left(I_4 - \frac{\mathbf{X}\mathbf{X}^\mathsf{T}}{\mathbf{X}^\mathsf{T}\mathbf{X}}\right), \tag{7.40}$$

the 4×4 matrix in brackets, which is an idempotent projection matrix of rank 3. In Sect. 7.3 we will show that $\mathsf{\Pi}(\mathbf{X})$ can be interpreted as a matrix representation of a 3D point, similarly to how $\mathsf{\Gamma}(\mathbf{L})$ is a matrix representation of a 3D line.

Line as Intersection of Two Planes. As points in 3D are duals of planes, the definition of 3D lines based on planes is closely related to that based on points. Following (5.123), p. 228, we directly see

$$\mathcal{L} = \mathcal{A} \cap \mathcal{B}: \quad \mathbf{L} = \mathbf{A} \cap \mathbf{B} = \begin{bmatrix} \boldsymbol{A}_h \times \boldsymbol{B}_h \\ A_0 \boldsymbol{B}_h - B_0 \boldsymbol{A}_h \end{bmatrix} = -\mathbf{B} \cap \mathbf{A}\,, \tag{7.41}$$

with the 6×4 matrix

$$\overline{\mathsf{\Pi}}(\mathbf{A}) \doteq \frac{\partial(\mathbf{A} \cap \mathbf{B})}{\partial \mathbf{B}} = \begin{bmatrix} \mathsf{S}(\mathbf{A}_h) & \mathbf{0} \\ A_0 I_3 & -\boldsymbol{A}_h \end{bmatrix} \tag{7.42}$$

$$= \left[\begin{array}{ccc|c} 0 & -C & B & 0 \\ C & 0 & -A & 0 \\ -B & A & 0 & 0 \\ \hline D & 0 & 0 & -A \\ 0 & D & 0 & -B \\ 0 & 0 & D & -C \end{array}\right] = D\,\mathsf{\Pi}(\mathbf{A})\,, \quad D = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix} \tag{7.43}$$

depending on the 4-vector $\mathbf{A}^\mathsf{T} = [\boldsymbol{A}_h^\mathsf{T}, A_0] = [A, B, C, D]$, thus

$$\mathcal{L} = \mathcal{A} \cap \mathcal{B}: \quad \mathbf{L} = \mathbf{A} \cap \mathbf{B} = \overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B} = -\overline{\mathsf{\Pi}}(\mathbf{B})\mathbf{A}\,. \tag{7.44}$$

The direction $\boldsymbol{L}_h = \boldsymbol{A}_h \times \boldsymbol{B}_h$ of line $\mathcal{L}$ depends only on the normals of the two planes.

7.2.1.2 Constructions with a 3D Line

Intersection Point of a Line and a Plane. The intersection point $\mathcal{X} = \mathcal{L} \cap \mathcal{A} = \mathcal{A} \cap \mathcal{L}$ of a line $\mathcal{L}$ and a plane $\mathcal{A}$ results from

$$\mathbf{X} = \mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{A} = \begin{bmatrix} \mathsf{S}(\mathbf{L}_0) & \boldsymbol{L}_h \\ -\boldsymbol{L}_h^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_h \\ A_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_0 \times \boldsymbol{A}_h + \boldsymbol{L}_h A_0 \\ -\boldsymbol{L}_h^\mathsf{T} \boldsymbol{A}_h \end{bmatrix} \tag{7.45}$$

using the expression for $\mathsf{\Gamma}(\mathbf{L})$ in (5.68), p. 219. This is because (see Fig. 7.2) an arbitrary plane $\mathcal{B}$ through $\mathcal{X}$ intersects $\mathcal{A}$ in a line $\mathcal{M}$ passing through $\mathcal{L}$, which can be assumed to be the join of $\mathcal{X}$ and another point $\mathcal{Y}$. Therefore $\mathbf{B}^\mathsf{T}\mathbf{X} = \mathbf{B}^\mathsf{T}\mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{A} = -\mathbf{B}^\mathsf{T}\mathsf{\Gamma}(\mathbf{L})\mathbf{A} = -\mathbf{B}^\mathsf{T}(\mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T})\mathbf{A} = 0$ must hold for arbitrary $\mathbf{B}$ passing through $\mathcal{X}$, cf. (7.62).

By rearrangement we also find

$$\mathbf{X} = \begin{bmatrix} A_0 I & -\mathsf{S}(\boldsymbol{A}_h) \\ -\boldsymbol{A}_h^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix} = \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\mathbf{L}\,. \tag{7.46}$$

Altogether, we obtain the intersection point $\mathcal{X}$ of a line $\mathcal{L}$ and a plane $\mathcal{A}$,

$$\mathcal{X} = \mathcal{L} \cap \mathcal{A} = \mathcal{A} \cap \mathcal{L}: \quad \mathbf{X} = \mathbf{L} \cap \mathbf{A} = \mathbf{A} \cap \mathbf{L} = \mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{A} = \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\mathbf{L}\,. \tag{7.47}$$

Observe, there is no sign change when exchanging the order of $\mathcal{L}$ and $\mathcal{A}$.

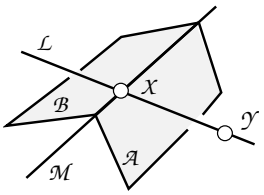


Fig. 7.2 Proof of the relation for the intersection point of a line and a plane. Intersection point $\mathcal{X}$ of plane $\mathcal{A}$ and line $\mathcal{L}$ constructed as the join of $\mathcal{X}$ and another point $\mathcal{Y}$. An arbitrary plane $\mathcal{B}$ through $\mathcal{X}$ intersects $\mathcal{A}$ in $\mathcal{M}$, which is the basis for the proof of $\mathbf{X} = \mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{A}$

Plane Joining a Line and a Point. Dually to the last relation, we obtain the plane passing through the line $\mathcal{L}$ and the point $\mathcal{X}$,

$$\mathcal{A} = \mathcal{L} \wedge \mathcal{X} = \mathcal{X} \wedge \mathcal{L}: \quad \mathbf{A} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathbf{L})\mathbf{X} = \overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{X})\mathbf{L}\,, \tag{7.48}$$

again with no sign change when interchanging $\mathcal{X}$ and $\mathcal{L}$.

7.2.1.3 Three and More Entities

Plane Through Three Points. The plane $\mathcal{A} = \mathcal{X} \wedge \mathcal{Y} \wedge \mathcal{Z}$ passing through three points $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ is given by joining a point with the line joining the other two.

$$\mathcal{A} = \mathcal{X} \wedge \mathcal{Y} \wedge \mathcal{Z}: \ \mathbf{A} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{X})\mathbf{Y})\mathbf{Z} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{Y})\mathbf{Z})\mathbf{X} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{Z})\mathbf{X})\mathbf{Y}\,. \tag{7.49}$$

The three points can be exchanged cyclically without changing the sign of the plane vector.

Point Through Three Planes. The intersection point $\mathcal{X} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$ of three planes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ is the intersection point of one plane with the intersection line of the other two planes, e.g., $(\mathbf{A} \cap \mathbf{B}) \cap C = \mathsf{\Gamma}^\mathsf{T}(\mathbf{A} \cap \mathbf{B})\mathbf{C} = \mathsf{\Gamma}^\mathsf{T}(\overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B})\mathbf{C} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{A})\mathbf{B})\mathbf{C}$:

$$\mathcal{X} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}: \ \mathbf{X} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{A})\mathbf{B})\mathbf{C} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{B})\mathbf{C})\mathbf{A} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathsf{\Pi}(\mathbf{C})\mathbf{A})\mathbf{B}\,. \tag{7.50}$$

Line Passing Through Four 3D Lines. Given four lines $\mathcal{L}_i$, $i = 1, ...4$ that are pairwise skew, there are at most two lines, $\mathcal{M}_1$ and $\mathcal{M}_2$, passing through all these four.

Formally, we have four incidence constraints $\mathbf{L}_i^\mathsf{T} D \mathbf{M} = 0$, since we saw that in (5.117), p. 227, if the determinant D_x is zero, the two lines $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ and $\mathcal{M} = \mathcal{Z} \wedge \mathcal{T}$ intersect. The four constraints can be collected in the homogeneous equation system

$$\underset{4\times 6}{\mathsf{C}^{\mathsf{T}}}\ \underset{6\times 1}{\mathbf{M}} = \underset{4\times 1}{\mathbf{0}} \tag{7.51}$$

with the 6×4 matrix

$$\mathsf{C} = [D\mathbf{L}_1, D\mathbf{L}_2, D\mathbf{L}_3, D\mathbf{L}_4]\,. \tag{7.52}$$

It has rank 4; thus, there is a two-dimensional null space spanned by two different 6-vectors, say $\mathbf{N}_1$ and $\mathbf{N}_2$. Every linear combination $\mathbf{M} = \alpha\mathbf{N}_1 + (1-\alpha)\mathbf{N}_2$, depending on α, is a solution. But only those fulfilling the Plücker constraint $\mathbf{M}^{\mathsf{T}} D\mathbf{M} = 0$ are valid. This leads to

$$\mathbf{M}^{\mathsf{T}} D\mathbf{M} = (\mathbf{N}_2 - \mathbf{N}_1)^{\mathsf{T}} D(\mathbf{N}_2 - \mathbf{N}_1)\alpha^2 + 2(\mathbf{N}_2 - \mathbf{N}_1)^{\mathsf{T}} D\mathbf{N}_2\alpha + \mathbf{N}_2^{\mathsf{T}} D\mathbf{N}_2 = 0 \tag{7.53}$$

which is quadratic in α with up to two solutions,

$$\mathbf{M}_i = \alpha_i\mathbf{N}_1 + (1-\alpha_i)\mathbf{N}_2, \quad i = 1, 2\,. \tag{7.54}$$

Example 7.2.33: ***Observing a Moving Vehicle.*** This procedure may be used to determine the path of a vehicle moving along a straight line on a plane at constant velocity if the directions $\mathcal{L}_i$ to this vehicle are observed at four different times from four different positions, cf. Teller and Hohmeyer (1999) and Avidan and Shashua (2000). ◇ Exercise 7.4

The situation can easily be visualized, see Fig. 7.3. The first three lines $\mathcal{L}_i, i = 1, 2, 3$,

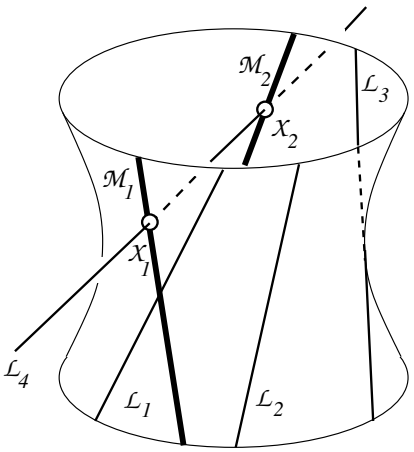


Fig. 7.3 Line through four given lines. The two lines $\mathcal{M}_1$ and $\mathcal{M}_2$ meet the given four lines $\mathcal{L}_i$, $i = 1, ...4$, the first three spanning a hyperboloid of one sheet and the last one intersecting the hyperboloid

define a ruled surface of degree 2 (cf. Weber, 2003a) namely a hyperboloid of one sheet $\mathbf{X}^{\mathsf{T}}\mathsf{Q}\mathbf{X} = 0$ with

$$\mathsf{Q} = \overline{\Gamma}(\mathbf{L}_1)\Gamma(\mathbf{L}_2)\overline{\Gamma}(\mathbf{L}_3) - \overline{\Gamma}(\mathbf{L}_3)\Gamma(\mathbf{L}_2)\overline{\Gamma}(\mathbf{L}_1)\,. \tag{7.55}$$

This can be shown as follows. Due to (7.48), all points $\mathcal{X}$ on either $\mathcal{L}_1$ or $\mathcal{L}_3$ lead to $\mathbf{X}^{\mathsf{T}}\mathsf{Q}\mathbf{X} = 0$, and for any point $\mathcal{X}$ on $\mathcal{L}_2$, we have $\mathbf{X}^{\mathsf{T}}\mathsf{Q}\mathbf{X} = \mathbf{X}^{\mathsf{T}}\overline{\Gamma}(\mathbf{L}_1)\Gamma(\mathbf{L}_2)\overline{\Gamma}(\mathbf{L}_3)\mathbf{X} - \mathbf{X}^{\mathsf{T}}\overline{\Gamma}(\mathbf{L}_3)\Gamma(\mathbf{L}_2)\overline{\Gamma}(\mathbf{L}_1)\mathbf{X} = 0$, since both terms vanish: e.g., we can write the first term as

$$\mathbf{X}^{\mathsf{T}}\Bigg(\underbrace{\mathbf{L}_1 \wedge (\mathbf{L}_2 \cap \underbrace{(\mathbf{L}_3 \wedge \mathbf{X})}_{\mathbf{A}})}_{\mathbf{X}}\Bigg) \tag{7.56}$$

(cf. Table 7.4). As the plane $\mathcal{A} = \mathcal{L}_3 \wedge \mathcal{X}$ and the line $\mathcal{L}_2$ meet in $\mathcal{X}$, the plane $\mathcal{L}_1 \wedge \mathcal{X}$ contains $\mathcal{X}$, so the term is zero (see Courant et al., 1996, Fig. 108).

This kind of hyperboloid is composed of two infinite sets of lines (the generating thin lines and the thick lines on the hyperboloid in the figure), each line of each set intersects all lines of the other set, thus satisfying the constraints (7.51). If the fourth line $\mathcal{L}_4$ intersects the hyperboloid in two points $\mathcal{X}_1$ and $\mathcal{X}_2$, the sought lines $\mathcal{M}_i$ pass through these points and belong to the second set of generating lines (shown here as thick lines).

More constructions in 3D, namely those involving more than one entity or parallelisms and orthogonalities, are addressed in the exercises, p. 322.

7.2.2 Geometric Relations in 3D

Table 7.5 collects basic spatial relations between 3D entities, in particular the entities involved, the geometric relations, and the algebraic constraints. In contrast to 2D, the constraints in 3D show different degrees of freedom. They are relevant for selecting independent algebraic constraints and for statistical testing.

7.2.2.1 Incidences

Incidence of Point and Plane. A point $\mathcal{X}$ and a plane $\mathcal{A}$ are incident if

$$\iota(\mathcal{X}, \mathcal{A}) \quad \Leftrightarrow \quad \langle \mathbf{X}, \mathbf{A} \rangle = \mathbf{X}^\mathsf{T}\mathbf{A} = \mathbf{A}^\mathsf{T}\mathbf{X} = 0\,, \tag{7.57}$$

Compare this with the definition of the plane in (5.47), p. 211.

Incidence of Two Lines. The incidence or coplanarity of two lines occurs in various forms, depending on how the two lines are given.

- Two lines $\mathcal{L}$ and $\mathcal{M}$ are coplanar or incident if

$$\iota(\mathcal{L}, \mathcal{M}) \quad \Leftrightarrow \quad \langle \mathbf{L}, \mathbf{M} \rangle_D = \mathbf{L}^\mathsf{T} D \mathbf{M} = \mathbf{L}^\mathsf{T} \overline{\mathbf{M}} = 0\,, \tag{7.58}$$

  where D is the dualizing operator for 3D lines, cf. (5.115), p. 227.
- In case the two lines are given by points, thus $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ and $\mathcal{M} = \mathcal{Z} \wedge \mathcal{T}$ (cf. (5.117), p. 227), then

$$\langle \mathbf{X} \wedge \mathbf{Y}, \mathbf{Z} \wedge \mathbf{T} \rangle = -|\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}| = (\mathbf{X} \wedge \mathbf{Y})^\mathsf{T} \overline{(\mathbf{Z} \wedge \mathbf{T})} = 0\,. \tag{7.59}$$

- In case the lines are given by planes $\mathcal{L} = \mathcal{A} \cap \mathcal{B}$ and $\mathcal{M} = \mathcal{C} \cap \mathcal{D}$, we have the constraint

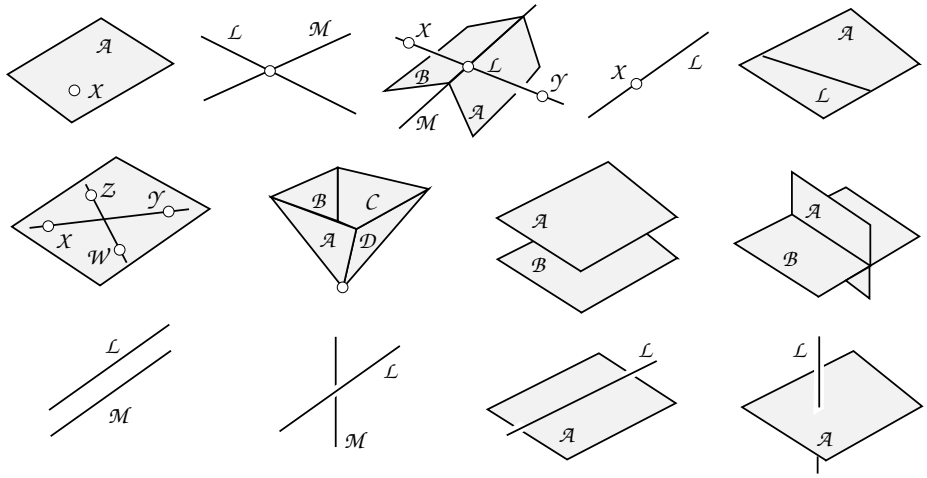
$$\langle \mathbf{A} \cap \mathbf{B}, \mathbf{C} \cap \mathbf{D} \rangle = -|\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}| = (\mathbf{A} \cap \mathbf{B})^\mathsf{T} \overline{(\mathbf{C} \cap \mathbf{D})} = 0\,. \tag{7.60}$$

- In case one line is given by two points $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ and the other by two planes $\mathcal{M} = \mathcal{A} \cap \mathcal{B}$, we have

$$\begin{aligned} \langle \mathbf{X} \wedge \mathbf{Y}, \mathbf{A} \cap \mathbf{B} \rangle &= \mathbf{X}^\mathsf{T}(\mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T})\mathbf{Y} && (7.61)\\ &= \mathbf{A}^\mathsf{T}(\mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T})\mathbf{B} && (7.62)\\ &= (\mathbf{X} \wedge \mathbf{Y})^\mathsf{T} \overline{(\mathbf{A} \cap \mathbf{B})} = 0\,. && (7.63) \end{aligned}$$

  This can be seen from Fig. 7.4. We will exploit this relation observing $\Gamma(\mathbf{X} \wedge \mathbf{Y}) = \mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T}$ and $\overline{\Gamma}(\mathbf{A} \cap \mathbf{B}) = \mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T}$ in the next relation.

Table 7.5 Relations between 3D entities, degrees of freedom. The whole space is U



3D-entities	relation	constraint	d.o.f.	Eq.
point $\mathcal{X}$, plane $\mathcal{A}$	$\mathcal{X} \in \mathcal{A}$	$\mathbf{X}^\mathsf{T}\mathbf{A} = 0$	1	(7.57)
two lines $\mathcal{L}$, $\mathcal{M}$	$\mathcal{L} \cap \mathcal{M} \neq \emptyset$	$\mathbf{L}^\mathsf{T}\overline{\mathbf{M}} = 0$	1	(7.58)
two lines $\mathcal{X} \wedge \mathcal{Y}$, $\mathcal{A} \cap \mathcal{B}$	$(\mathcal{X} \wedge \mathcal{Y}) \cap (\mathcal{A} \cap \mathcal{B}) \neq \emptyset$	$\mathbf{X}^\mathsf{T}(\mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T})\mathbf{Y} = 0$	1	(7.61)
point $\mathcal{X}$, line $\mathcal{L}$	$\mathcal{X} \in \mathcal{L}$	$\overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{X} = \mathbf{0}$	2	(7.67)
line $\mathcal{L}$, plane $\mathcal{A}$	$\mathcal{L} \in \mathcal{A}$	$\mathsf{\Gamma}(\mathbf{L})\mathbf{A} = \mathbf{0}$	2	(7.64)
point $\mathcal{X}$, point $\mathcal{Y}$	$\mathcal{X} \equiv \mathcal{Y}$	$\overline{\mathsf{\Pi}}(\mathbf{X})\mathbf{Y} = -\overline{\mathsf{\Pi}}(\mathbf{Y})\mathbf{X} = \mathbf{0}$	3	(7.70)
plane $\mathcal{A}$, plane $\mathcal{B}$	$\mathcal{A} \equiv \mathcal{B}$	$\overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B} = -\overline{\mathsf{\Pi}}(\mathbf{B})\mathbf{A} = \mathbf{0}$	3	(7.71)
line $\mathcal{L}$, line $\mathcal{M}$	$\mathcal{L} \equiv \mathcal{M}$	$\overline{\mathsf{\Gamma}}(\mathbf{L})\mathsf{\Gamma}(\mathbf{M}) = \mathsf{0}$	4	(7.73)
four points	$\mathcal{X} \wedge \mathcal{Y} \wedge \mathcal{Z} \wedge \mathcal{W} \neq U$	$\lvert\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}\rvert = 0$	1	(7.76)
four planes	$\mathcal{A} \cap \mathcal{B} \cap \mathcal{C} \cap \mathcal{D} \neq \emptyset$	$\lvert\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\rvert = 0$	1	(7.77)
two planes $\mathcal{A}$, $\mathcal{B}$	$\mathcal{A} \parallel \mathcal{B}$	$\mathsf{S}(\mathsf{P}_4\mathbf{B})\mathsf{P}_4\mathbf{A} = \mathbf{0}$	2	(7.83)
	$\mathcal{A} \perp \mathcal{B}$	$\mathbf{A}^\mathsf{T}\mathsf{G}_4\mathbf{B} = 0$	1	(7.82)
two lines $\mathcal{L}$, $\mathcal{M}$	$\mathcal{L} \parallel \mathcal{M}$	$\mathsf{S}(\mathbf{QL})\mathbf{QM} = \mathbf{0}$	2	(7.88)
	$\mathcal{L} \perp \mathcal{M}$	$\mathbf{L}^\mathsf{T}\mathsf{G}_6\mathbf{M} = 0$	1	(7.87)
line and plane $\mathcal{L}$, $\mathcal{A}$	$\mathcal{L} \parallel \mathcal{A}$	$\mathbf{L}^\mathsf{T}\mathsf{G}_{64}\mathbf{A} = 0$	1	(7.92)
	$\mathcal{L} \perp \mathcal{A}$	$\mathsf{S}(\mathbf{QL})\mathsf{P}_4\mathbf{A} = \mathbf{0}$	2	(7.93)

7.2.2.2 Incidence of Plane and Line.

A plane $\mathcal{A}$ and a line $\mathcal{L}$ are incident if

$$\iota(\mathcal{L}, \mathcal{A}) \quad \Leftrightarrow \quad \mathsf{\Gamma}(\mathbf{L})\mathbf{A} = \mathbf{0}\,. \tag{7.64}$$

This can be seen as follows: Let the line be generated as a join of two points $\mathcal{X}, \mathcal{Y} \in \mathcal{A}$; then, due to $\mathsf{\Gamma}(\mathcal{X} \wedge \mathcal{Y}) = \mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T}$ and $\mathbf{X}^\mathsf{T}\mathbf{A} = 0$ and $\mathbf{Y}^\mathsf{T}\mathbf{A} = 0$, (7.64) holds. Using (5.68), p. 219, the constraint (7.64) can be written as

$$\boldsymbol{L}_0 \times \boldsymbol{A}_h + \boldsymbol{L}_h A_0 = \mathbf{0} \tag{7.65}$$

$$\boldsymbol{L}_h^\mathsf{T} \boldsymbol{A}_h = 0\,. \tag{7.66}$$

As the Plücker matrix has rank 2, only two constraints are linearly independent.

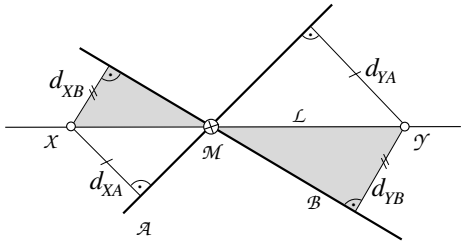


Fig. 7.4 Proof of the incidence constraint of two lines, one given as join of two points, the other given as intersection of two planes (7.61). Line $\mathcal{M}$, which is the intersection of the planes $\mathcal{A}$ and $\mathcal{B}$, is vertical to the viewing plane. Line $\mathcal{L}$ joining the two points $\mathcal{X}$ and $\mathcal{Y}$, lying in general position before or behind the viewing plane, meets $\mathcal{M}$ in the viewing plane. It can be observed in the two similar quadrangles: The distances between $\mathcal{X}$ and $\mathcal{Y}$ from the planes $\mathcal{A}$ and $\mathcal{B}$, intersecting in $\mathcal{M}$, are proportional: $d_{XA}/d_{YA} = d_{XB}/d_{YB}$ or $d_{XA}d_{YB} = d_{YA}d_{XB}$. As the inner products are equal to the distances after Euclidean normalization, we have $\mathbf{X}^\mathsf{T}\mathbf{A}\,\mathbf{Y}^\mathsf{T}\mathbf{B} = \mathbf{Y}^\mathsf{T}\mathbf{A}\,\mathbf{X}^\mathsf{T}\mathbf{B}$

Incidence of Point and Line. A point $\mathcal{X}$ and a line $\mathcal{L}$ are incident if

$$\iota(\mathcal{L}, \mathcal{X}) \quad \Leftrightarrow \quad \overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{X} = \mathbf{0} \tag{7.67}$$

This results from dualizing (7.64). The constraint (7.67) can also be written as

$$\boldsymbol{L}_h \times \boldsymbol{X}_0 + \boldsymbol{L}_0 X_h = \mathbf{0} \tag{7.68}$$

$$\boldsymbol{L}_0^\mathsf{T} \boldsymbol{X}_0 = 0\,. \tag{7.69}$$

Again, only two constraints are linearly independent.

7.2.2.3 Identity of Two 3D Entities

Two points $\mathcal{X}$ and $\mathcal{Y}$ are identical if the joining line is indefinite, thus

$$\mathcal{X} \equiv \mathcal{Y} \quad \Leftrightarrow \quad \mathbf{X} \wedge \mathbf{Y} = \mathsf{\Pi}(\mathbf{X})\mathbf{Y} = \mathbf{0}_6\,. \tag{7.70}$$

Only three of these constraints are linearly independent.

Analogously, two planes $\mathcal{A}$ and $\mathcal{B}$ are identical if the common line is indefinite, thus

$$\mathcal{A} \equiv \mathcal{B} \quad \Leftrightarrow \quad \mathbf{A} \cap \mathbf{B} = \mathsf{\Pi}(\mathbf{A})\mathbf{B} = \mathbf{0}_6\,. \tag{7.71}$$

Only three of these constraints are linearly independent.

Two 3D lines $\mathcal{L}$ and $\mathcal{M}$ are identical if at least two points on $\mathcal{L}$, say $\mathcal{X}$ and $\mathcal{Y}$, lie on two planes generating $\mathcal{M}$, say $\mathcal{A}$ and $\mathcal{B}$. Then we need to check

$$\mathbf{X}^\mathsf{T}\mathbf{A} = 0 \qquad \mathbf{Y}^\mathsf{T}\mathbf{A} = 0 \qquad \mathbf{X}^\mathsf{T}\mathbf{B} = 0 \qquad \mathbf{Y}^\mathsf{T}\mathbf{B} = 0\,. \tag{7.72}$$

This only holds if $(\mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T})(\mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T}) = \mathsf{0}_{4\times 4}$ and leads to the constraint

$$\mathcal{L} \equiv \mathcal{M} \quad \mathsf{\Gamma}(\mathbf{L})\overline{\mathsf{\Gamma}}(\mathbf{M}) = \mathsf{0}_{4\times 4}\,. \tag{7.73}$$

Only four of these 16 constraints are linearly independent.

7.2.2.4 Three and Four Entities

Collinearity of 3D Points and Concurrence of Planes. Three 3D points $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ are incident to a 3D line, thus collinear, if the matrix, whose columns are the three homogeneous vectors, has rank 2:

$$\mathsf{collinear}(\mathcal{X},\mathcal{Y},\mathcal{Z}) \quad \Leftrightarrow \quad \mathrm{rk}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = 2\,. \tag{7.74}$$

This is because $\mathcal{Z}$ must lie on the line through $\mathcal{X}$ and $\mathcal{Y}$, and due to $\mathbf{Z} = \alpha\mathbf{X} + (1-\alpha)\mathbf{Y}$, $\mathbf{Z}$ must be linearly dependent on $\mathbf{X}$ and $\mathbf{Y}$. Exercise 7.2

By duality, three planes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are incident to a 3D line, thus concurrent, if the matrix of their homogeneous coordinates has rank 2:

$$\mathsf{concurrent}(\mathcal{A},\mathcal{B},\mathcal{C}) \quad \Leftrightarrow \quad \mathrm{rk}(\mathbf{A},\mathbf{B},\mathbf{C}) = 2\,. \tag{7.75}$$

Coplanarity of Four Points and Concurrence of Four Planes. We have the following constraint for four coplanar points: Exercise 7.3

$$\mathsf{coplanar}(\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{U}) \quad \Leftrightarrow \quad |\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{U}| = 0\,. \tag{7.76}$$

Similarly the constraint for four concurrent planes is

$$\mathsf{concurrent}(\mathcal{A},\mathcal{B},\mathcal{C},\mathcal{D}) \quad \Leftrightarrow \quad |\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}| = 0\,. \tag{7.77}$$

7.2.2.5 Orthogonality and Parallelism of Lines and Planes

Orthogonality and Parallelism of Two Planes. The two planes $\mathcal{A}([\boldsymbol{A}_h^\mathsf{T}, A_0])$ and $\mathcal{B}([\boldsymbol{B}_h^\mathsf{T}, B_0])$ are orthogonal or parallel if the homogeneous parts of their coordinates, thus their normals, are orthogonal or parallel, respectively. The constraints can be written as

$$\mathcal{A} \perp \mathcal{B} \;\Leftrightarrow\; \langle \boldsymbol{A}_h, \boldsymbol{B}_h \rangle \;=\; \boldsymbol{A}_h^\mathsf{T}\boldsymbol{B}_h \;= 0 \tag{7.78}$$

$$\mathcal{A} \parallel \mathcal{B} \;\Leftrightarrow\; \boldsymbol{A}_h \times \boldsymbol{B}_h \;=\; \mathsf{S}(\boldsymbol{A}_h)\boldsymbol{B}_h = \mathbf{0}\,. \tag{7.79}$$

The normal of $\mathcal{A}$ can be determined from

$$\boldsymbol{A}_h = \mathsf{P}_4\mathbf{A}\,, \tag{7.80}$$

with the 3×4 projection matrix and its square, the matrix G_4,

$$\mathsf{P}_4 = [\boldsymbol{I}_3 \,|\, \mathbf{0}] \qquad \mathsf{G}_4 = \mathsf{P}_4^\mathsf{T}\mathsf{P}_4\,, \tag{7.81}$$

which is identical to the singular dual quadric Q_∞^*, cf. (5.168), p. 241. Therefore, we also can write these constraints as functions of the full homogeneous vectors,

$$\mathcal{A} \perp \mathcal{B} \;\Leftrightarrow\; \mathbf{A}^\mathsf{T}\mathsf{G}_4\mathbf{B} = 0 \tag{7.82}$$

$$\mathcal{A} \parallel \mathcal{B} \;\Leftrightarrow\; \mathsf{S}(\mathsf{P}_4\mathbf{A})\mathsf{P}_4\mathbf{B} = \mathbf{0}\,. \tag{7.83}$$

As the skew symmetric matrix has rank 2, only two of the three constraints for the parallelism are linearly independent.

Orthogonality and Parallelism of Lines. Two lines $\mathcal{L}([\boldsymbol{L}_h^\mathsf{T}, \boldsymbol{L}_0^\mathsf{T}])$ and $\mathcal{M}([\boldsymbol{M}_h^\mathsf{T}, \boldsymbol{M}_0^\mathsf{T}])$ are orthogonal or parallel if the homogeneous components of their coordinates are orthogonal or parallel, respectively,

$$\mathcal{L} \perp \mathcal{M} \;\Leftrightarrow\; \langle \boldsymbol{L}_h, \boldsymbol{M}_h \rangle \;= \boldsymbol{L}_h^\mathsf{T}\boldsymbol{M}_h = 0 \tag{7.84}$$

$$\mathcal{L} \parallel \mathcal{M} \;\Leftrightarrow\; \boldsymbol{L}_h \times \boldsymbol{M}_h \;= \mathsf{S}(\boldsymbol{L}_h)\boldsymbol{M}_h = \mathbf{0}\,. \tag{7.85}$$

Again, using a 3×6 projection matrix Q_6 and its square G_6,

$$\mathsf{Q}_6 = [\boldsymbol{I}_3 | \mathsf{0}_{3\times 3}] \qquad \mathsf{G}_6 = \mathsf{Q}^\mathsf{T}\mathsf{Q} = \begin{bmatrix} \boldsymbol{I}_3 & \mathsf{0}_{3\times 3} \\ \mathsf{0}_{3\times 3} & \mathsf{0}_{3\times 3} \end{bmatrix}\,, \tag{7.86}$$

as functions of the full homogeneous vectors, we obtain

$$\mathcal{L} \perp \mathcal{M} \;\Leftrightarrow\; \mathbf{L}^{\mathsf{T}}\mathsf{G}_6\mathbf{M} = 0 \tag{7.87}$$
$$\mathcal{L} \parallel \mathcal{M} \;\Leftrightarrow\; \mathsf{S}(\mathsf{Q}\mathbf{L})\mathsf{Q}\mathbf{M} = \mathbf{0}\,. \tag{7.88}$$

Again, only two constraints are linearly independent.

Orthogonality and Parallelism of Lines and Planes. A line $\mathcal{L}([\boldsymbol{L}_h^{\mathsf{T}}, \boldsymbol{L}_0^{\mathsf{T}}])$ and a plane $\mathcal{A}([\boldsymbol{A}_h^{\mathsf{T}}, A_0])$ are orthogonal or parallel if the homogeneous components of their coordinates are orthogonal or parallel, respectively.

$$\mathcal{L} \parallel \mathcal{A} \;\Leftrightarrow\; \langle \boldsymbol{L}_h, \boldsymbol{A}_h \rangle = \boldsymbol{L}_h^{\mathsf{T}} \boldsymbol{A}_h = 0 \tag{7.89}$$
$$\mathcal{L} \perp \mathcal{A} \;\Leftrightarrow\; \boldsymbol{L}_h \times \boldsymbol{A}_h = \mathsf{S}(\boldsymbol{L}_h)\boldsymbol{A}_h = \mathbf{0}. \tag{7.90}$$

We also can write these constraints with the projection matrices P_4 and Q and their product

$$\mathsf{G}_{64} = \mathsf{Q}^{\mathsf{T}}\mathsf{P}_4 = \begin{bmatrix} I_3 & \mathbf{0} \\ 0_{3\times 3} & \mathbf{0} \end{bmatrix} \tag{7.91}$$

as functions of the full homogeneous vectors

$$\mathcal{L} \parallel \mathcal{A} \;\Leftrightarrow\; \mathbf{L}^{\mathsf{T}}\mathsf{G}_{64}\mathbf{A} = 0 \tag{7.92}$$
$$\mathcal{L} \perp \mathcal{A} \;\Leftrightarrow\; \mathsf{S}(\mathsf{Q}\mathbf{L})\mathsf{P}_4\mathbf{A} = \mathbf{0}. \tag{7.93}$$

Again, only two constraints are linearly independent.

7.2.3 Distances and Angles in 3D

We give the signed distances in Table 7.6, p. 309.

7.2.3.1 Distances to the Origin

Distance of a Point to the Origin. The distance of a point $\mathcal{X}([\boldsymbol{X}_0^{\mathsf{T}}, X_h])$ from the origin is

$$d_{XO} = \frac{|\boldsymbol{X}_0|}{X_h}\,. \tag{7.94}$$

The sign of the distance is identical to the sign $\operatorname{sign}(X_h)$ of the point.[2]

Distance of a Plane to the Origin. The signed distance of the plane $\mathcal{A}([\boldsymbol{A}_h^{\mathsf{T}}, A_0])$ from the origin is

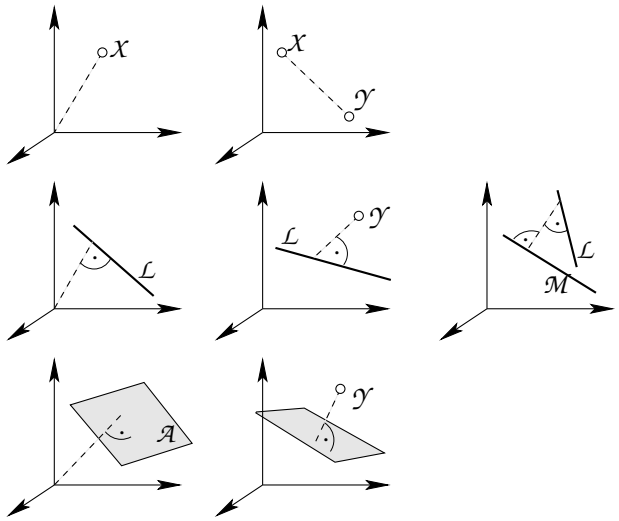
$$d_{AO} = \frac{A_0}{|\boldsymbol{A}_h|}\,. \tag{7.95}$$

The distance is positive if the origin is on the negative side of the plane, again assuming the origin to have positive fourth coordinate.

Distance of a Line to the Origin. The distance of the line $\mathcal{L}([\mathbf{L}_h^{\mathsf{T}}, \boldsymbol{L}_0^{\mathsf{T}}])$ from the origin is

[2] This is valid if the origin has positive fourth coordinate.

Table 7.6 Distances between 3D entities



distance	from origin O	from point $\mathcal{Y}$	from line $\mathcal{M}$
3D point $\mathcal{X}$	$d_{XO} = \dfrac{\lvert\boldsymbol{X}_0\rvert}{X_h}$	$d_{XY} = \dfrac{\lvert X_h\boldsymbol{Y}_0 - X_h\boldsymbol{X}_0\rvert}{X_hY_h}$	$d_{XM} = \dfrac{\lvert\boldsymbol{X}_0 \times \boldsymbol{M}_h - X_h\boldsymbol{M}_0\rvert}{X_h\lvert\boldsymbol{M}_h\rvert}$
3D line $\mathcal{L}$	$d_{LO} = \dfrac{\lvert\boldsymbol{L}_0\rvert}{\lvert\boldsymbol{L}_h\rvert}$	$d_{LY} = \dfrac{\lvert\boldsymbol{Y}_0 \times \boldsymbol{L}_h - Y_h\boldsymbol{L}_0\rvert}{Y_h\lvert\boldsymbol{L}_h\rvert}$	$d_{LM} = \dfrac{\overline{\mathbf{L}}^{\mathsf{T}}\mathbf{M}}{\lvert\boldsymbol{L}_h \times \boldsymbol{M}_h\rvert}$
plane $\mathcal{A}$	$d_{AO} = \dfrac{A_0}{\lvert\boldsymbol{A}_h\rvert}$	$d_{AY} = \dfrac{\mathbf{A}^{\mathsf{T}}\mathbf{Y}}{\lvert Y_h\boldsymbol{A}_h\rvert}$	–

$$d_{LO} = \frac{|\boldsymbol{L}_0|}{|\boldsymbol{L}_h|}\,, \tag{7.96}$$

which we already proved above, cf. (5.67), p. 218.

Remark: Observe the nice analogy between the derived distances of entities from the origin, which are the ratios of the absolute values of the Euclidean and the homogeneous parts of the homogeneous vector, which motivated the naming of the parts of the homogeneous vectors of points, lines, and planes (cf. Brand, 1966).

7.2.3.2 Distance of a Point to a Point, a Line, and a Plane

Distance Between Two Points. The distance between two points $\mathcal{X}$ and $\mathcal{Y}$ is

$$d_{XY} = \frac{|X_h\boldsymbol{Y}_0 - Y_h\boldsymbol{X}_0|}{X_hY_h}\,. \tag{7.97}$$

The distance is positive if the two points have the same sign, thus $\text{sign}(X_h) = \text{sign}(Y_h)$.

Distance of a Point to a Line. The distance of a point $\mathcal{X}$ from a line $\mathbf{L}$ is given by

$$d_{XL} = \frac{|\boldsymbol{X}_0 \times \boldsymbol{L}_h - X_h\boldsymbol{L}_0|}{X_h|\boldsymbol{L}_h|}\,. \tag{7.98}$$

The sign of the distance is identical to the sign of the point.

Exercise 7.5

Proof: We shift both point and line such that the shifted point is in the origin and apply (5.67). Shifting a line by $\boldsymbol{T}$ yields $\mathbf{L}' = [\boldsymbol{L}_h, \boldsymbol{L}_0 - \boldsymbol{T} \times \boldsymbol{L}_h]$. Here we have the shift $\boldsymbol{T} = -\boldsymbol{X}_0/X_h$, which yields the shifted line $[\boldsymbol{L}_h, \boldsymbol{L}_0 - \boldsymbol{X}_0/X_h \times \boldsymbol{L}_h]$, from which we obtain the distance $d_{L'0} = (|-\boldsymbol{X}_0/X_h \times \boldsymbol{L}_h + \boldsymbol{L}_0|)/(|\boldsymbol{L}_h|)$ which yields (7.98). ◇

Distance of a Plane to a Point. The signed distance of a plane $\mathcal{A}$ from a point $\mathcal{X}$ is given by

$$d_{XA} = \frac{\langle \boldsymbol{X}, \boldsymbol{A} \rangle}{|X_h \boldsymbol{A}_h|}, \tag{7.99}$$

with the inner product $\langle \boldsymbol{X}, \boldsymbol{A} \rangle = \mathbf{X}^\mathsf{T}\mathbf{A}$. The sign of the distance is positive if the point $\mathcal{Y}$ lies on the positive side of the plane $\mathcal{A}$.[3]

7.2.3.3 Distance Between Two Lines

The distance between two lines $\mathcal{L}$ and $\mathcal{M}$ is given by

$$d_{LM} = \frac{\langle \mathbf{L}, \mathbf{M} \rangle_D}{|\boldsymbol{L}_h \times \boldsymbol{M}_h|} \tag{7.100}$$

using the inner product $\langle \mathbf{L}, \mathbf{M} \rangle_D = \mathbf{L}^\mathsf{T} D \mathbf{M} = \overline{\mathbf{L}}^\mathsf{T} \mathbf{M} = \mathbf{L}^\mathsf{T} \overline{\mathbf{M}}$. The sign of the distance is identical to the sign of $\langle \mathbf{L}, \mathbf{M} \rangle_D$ and will be discussed below.

Proof: Let us assume the lines are defined by four points $\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2)$ and $\mathcal{M}(\mathcal{X}_3, \mathcal{X}_4)$, see Fig. 7.5, p. 310. The numerator can then be expressed as

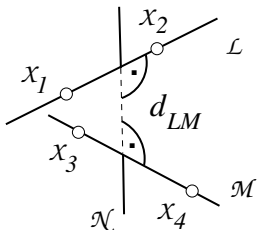


Fig. 7.5 The distance between two 3D lines $\mathcal{L}$ and $\mathcal{M}$

$$\begin{aligned}
\langle \mathbf{L}, \mathbf{M} \rangle_D &= \mathbf{L}^\mathsf{T} \overline{\mathbf{M}} \\
&= \boldsymbol{L}_h^\mathsf{T} \boldsymbol{M}_0 + \boldsymbol{M}_h^\mathsf{T} \boldsymbol{L}_0 \\
&= (\boldsymbol{X}_2 - \boldsymbol{X}_1)^\mathsf{T} (\boldsymbol{X}_3 \times \boldsymbol{X}_4) + (\boldsymbol{X}_4 - \boldsymbol{X}_3)^\mathsf{T} (\boldsymbol{X}_1 \times \boldsymbol{X}_2) \\
&= \langle \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_4 \rangle - \langle \boldsymbol{X}_1, \boldsymbol{X}_3, \boldsymbol{X}_4 \rangle + \langle \boldsymbol{X}_4, \boldsymbol{X}_1, \boldsymbol{X}_2 \rangle - \langle \boldsymbol{X}_3, \boldsymbol{X}_1, \boldsymbol{X}_2 \rangle \\
&= \langle (\boldsymbol{X}_2 - \boldsymbol{X}_1), (\boldsymbol{X}_4 - \boldsymbol{X}_3), (\boldsymbol{X}_1 - \boldsymbol{X}_3) \rangle \\
&= (\boldsymbol{L}_h \times \boldsymbol{M}_h)^\mathsf{T} (\boldsymbol{X}_1 - \boldsymbol{X}_3) .
\end{aligned}$$

The line $\mathcal{N}$ orthogonal to both given lines has the direction $\boldsymbol{L}_h \times \boldsymbol{M}_h$. After division of $\langle \mathbf{L}, \mathbf{M} \rangle_D$ by $|\boldsymbol{L}_h \times \boldsymbol{M}_h|$, we obtain the length of the projection of the segment $(\mathcal{X}_1, \mathcal{X}_3)$ onto the line $\mathcal{N}$, which is the desired distance. ◇

[3] This assumes that X_h is positive; otherwise, the distance is negative.

7.2.3.4 Angle Between Two Directions

To determine the angle between two lines, two planes, or a line and a plane, we may use their directions or normals, respectively. Therefore, we only need to give an expression for the angle between two directions. Again, in order to obtain numerically stable results, we prefer to express the angle by the atan2 $(\bullet, \bullet)$ function with two arguments. Given two directions $\boldsymbol{N}$ and $\boldsymbol{M}$, we therefore obtain

$$\alpha_{NM} = \operatorname{atan2}\left(|\boldsymbol{N} \times \boldsymbol{M}|, \boldsymbol{N} \cdot \boldsymbol{M}\right) \in [0, \pi)\,. \tag{7.101}$$

If we want to obtain the acute angle between nonoriented lines $\mathcal{N}$ and $\mathcal{M}$, we take $\min(\alpha_{NM}, \pi - \alpha_{NM})$.

7.3 Vector and Matrix Representations for Geometric Entities and Their Interpretation

7.3.1 Motivation 311
7.3.2 Matrix Representations of Points, Lines, and Planes 312
7.3.3 Vector Representations of Transformations and Conics 315
7.4.1 Selection of Independent Constraints 317
7.4.2 Affinity and Similarity 320
7.4.3 General Homographies in 2D and 3D 321

7.3.1 Motivation

Geometric entities generally are represented by homogeneous vectors, while transformations are represented by homogeneous matrices. However, there are several reasons why matrix representations for geometric entities and vector representations for transformations may be useful:

- A 3D line was represented with a 6-vector and also with a skew symmetric 4×4 matrix, both containing the Plücker coordinates. The two representations were useful in the preceding chapters when constructing new geometric elements.
  Obviously, the other constructions also contained matrices depending on the Plücker coordinates of the elements, the 3×3 skew symmetric matrix $\mathsf{S}(\mathbf{x})$ of a 3-vector and the 6×4 matrix $\mathsf{\Pi}(\mathbf{X})$ of a 4-vector.
- All constructions were represented as matrix-vector multiplications, where the matrix nearly always depends linearly on the coordinates of the corresponding geometric element. Thus, these constructions were bilinear forms in the generating vectors. For example, instead of $\mathbf{l} = \mathbf{x} \times \mathbf{y}$, we can write

  $$l_k = \epsilon_{ijk}\, x_i\, y_j \tag{7.102}$$

  with the ϵ-tensor, cf. Sect. A.A.14, p. 782, which is $+1$ if (ijk) is an even permutation, -1 if (ijk) is an odd permutation of $\{1, 2, 3\}$ and 0 otherwise. This may be achieved with all other bilinear constructions.
- The representation

  $$\mathbf{l} = \mathsf{S}(\mathbf{x})\mathbf{y} \tag{7.103}$$

  for the construction of a line from two points can be interpreted as a singular correlation, mapping the point y to the line ℓ. This observation eliminates the distinction between constructions and transformations.

- If we want to determine the elements h_{ij} of a transformation matrix H from corresponding points in an estimation procedure, we would like to have the unknown parameters h_{ij} collected in a vector.
- Finally, given an entity (say a 3D line $\mathcal{L}$), we have the problem of finding a minimum set of other entities (here two points on the line), which we want to use for further processing. This task turns out to be equivalent to the task of choosing independent constraints when checking spatial relations (here, in the case of a 3D line passing through a plane $\mathcal{A}$, two out of the four constraints $\overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{A} = \mathbf{0}_{4\times 1}$).

For these reasons, we want to collect the already derived vector and matrix representations of geometric entities and transformations. In order to be able to interpret the matrix representations of geometric entities, we analyse the geometric significance of their rows and columns using the unit vectors of the basic elements of the underlying coordinate frame, cf. Sect. 5.9, p. 242.

7.3.2 Matrix Representations of Points, Lines, and Planes

7.3.2.1 Matrix Representations of 2D Entities

2D Point. A 2D point is represented either by its homogeneous coordinate vector or by the skew symmetric matrix of this vector.

$$\mathcal{X}: \ \mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad \mathcal{X}: \ \mathsf{S}(\mathbf{x}) \,. \tag{7.104}$$

We now make the columns of the skew matrix explicit. We obtain with the unit vectors $\mathbf{e}_i^{[3]} \in \mathbb{R}^3$,

$$\begin{aligned} \mathsf{S}(\mathbf{x}) &= \begin{bmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{bmatrix} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] = \mathsf{S}(\mathbf{x}) \left[\boldsymbol{e}_1^{[3]}, \boldsymbol{e}_2^{[3]}, \boldsymbol{e}_3^{[3]} \right] \\ &= \left[\mathbf{x} \times \mathbf{e}_1^{[3]}, \mathbf{x} \times \mathbf{e}_2^{[3]}, \mathbf{x} \times \mathbf{e}_3^{[3]} \right] = [\mathbf{x} \wedge \mathbf{x}_{\infty x}, \mathbf{x} \wedge \mathbf{x}_{\infty y}, \mathbf{x} \wedge \mathbf{x}_O] \ . \end{aligned} \tag{7.105}$$

We now interpret the unit vectors $\mathbf{e}_i^{[3]}$ as the canonical points of the coordinate system, namely as the points at infinity $\mathbf{x}_{\infty x}$ and $\mathbf{x}_{\infty y}$ of the x- and y-axes of the coordinate frame and as its origin $\mathbf{x}_O$, p. 243. The columns of $\mathsf{S}(\mathbf{x})$ thus are the coordinates of three lines, namely the join of the point $\mathcal{X}$ with the three canonical points, see Fig. 7.6, left:

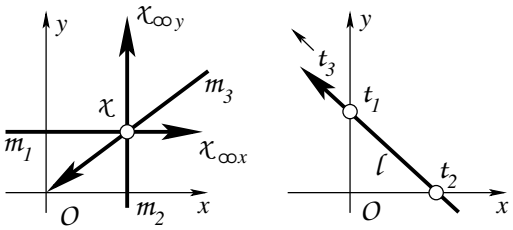


Fig. 7.6 Visualization of skew symmetric matrix of a 2D point. **Left**: Visualization of columns $[\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3]$ of a skew symmetric matrix $\mathsf{S}(\mathbf{x})$ representing point $\mathcal{X}$. **Right**: Columns $[\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]$ of a skew symmetric matrix $\mathsf{S}(\mathbf{l})$ representing line ℓ

- The first column $\mathbf{m}_1$ of $\mathsf{S}(\mathbf{x})$ represents the line m_1 passing through $\mathcal{X}$ and the point at infinity in the direction of the x-axis.

- The second column $\mathbf{m}_2$ of $\mathsf{S}(\mathbf{x})$ represents the line m_2 passing through χ and the point at infinity in the direction of the y-axis.
- The third column $\mathbf{m}_3$ of $\mathsf{S}(\mathbf{x})$ represents the line m_3 passing through χ and the origin.

If the point is Euclideanly normalized, the directions follow from the sequence of points, cf. Chap. 9, p. 343.

2D Line. A similar reasoning applied to the skew symmetric matrix $\mathsf{S}(\mathbf{l})$ of a 2D line (cf. (5.3), p. 198) and the axes ℓ_x and ℓ_y of the coordinate system (cf. (5.173), p. 243),

$$\mathsf{S}(\mathbf{l}) = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3] = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} = \mathsf{S}(\mathbf{l})[\boldsymbol{e}_1^{[3]}, \boldsymbol{e}_2^{[3]}, \boldsymbol{e}_3^{[3]}] = [\mathbf{l} \cap \mathbf{l}_y, \mathbf{l} \cap \mathbf{l}_x, \mathbf{l} \cap \mathbf{l}_\infty] \quad (7.106)$$

shows the columns $\mathbf{t}_i$ of $\mathsf{S}(\mathbf{l})$ to be the intersection points of line ℓ with the basic lines $\mathbf{e}_i^{[3]}$ of the coordinate frame, see Fig. 7.6, right:

- The first column $\mathbf{t}_1$ of $\mathsf{S}(\mathbf{l})$ represents the intersection point t_1 of ℓ with the y-axis $[1, 0, 0]^\mathsf{T}$.
- The second column $\mathbf{t}_2$ of $\mathsf{S}(\mathbf{l})$ represents the intersection point t_2 of ℓ with the x-axis $[0, 1, 0]^\mathsf{T}$.
- The third column $\mathbf{t}_3$ of $\mathsf{S}(\mathbf{l})$ represents the intersection point t_3 of ℓ with the line at infinity, thus represents the direction of the line.

7.3.2.2 Matrix Representations of 3D Entities

3D Point. A 3D point can be represented by the matrix $\mathsf{\Pi}(\mathbf{X})$. It was derived (cf. (7.38), p. 301) for the join of two points, yielding a 3D line or for the join of a point with a 3D line, yielding a plane (cf. (7.48), p. 302),

$$\mathcal{L} = \mathcal{X} \wedge \mathcal{Y} : \ \mathbf{L} = \mathsf{\Pi}(\mathbf{X})\mathbf{Y} \quad \text{and} \quad \mathcal{A} = \mathcal{X} \wedge \mathcal{L} : \ \mathbf{A} = \overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{X})\mathbf{L}\,. \quad (7.107)$$

For interpreting the columns of $\mathsf{\Pi}$, as before with 2D entities, we introduce the canonical elements $\mathbf{e}_i^{[4]}$ of the coordinate frame (cf. (5.177), p. 244) as particular point vectors into (7.107) to directly disclose the geometric image of its columns and rows. We choose the point $\mathcal{Y}$ to be one of the canonical points $\mathbf{e}_i^{[4]}$ of the 3D coordinate system and obtain for $\mathbf{L}$ the ith column of the matrix $\mathsf{\Pi}(\mathbf{X})$. These column vectors then are the join of $\mathcal{X}$ with the canonical points. Thus the columns of $\mathsf{\Pi}(\mathbf{X})$ are the 3D lines $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ parallel to the coordinate axes and the 3D line $\mathcal{L}_4$ through the origin, see Fig. 7.7, left.

Now, in a 6D parameter coordinate frame we choose the canonical lines $\boldsymbol{e}_i^{[6]}$ as 3D lines $\mathcal{L}_i$ in order to obtain the columns of $\mathsf{\Pi}^\mathsf{T}(\mathbf{X})$ or, equivalently, the rows of $\mathsf{\Pi}^\mathsf{T}(\mathbf{X})$. They then are visualized in the (XYZ)-System $\mathbb{R}^3$. We obtain the planes $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ parallel to the coordinate planes as the first three rows of $\mathsf{\Pi}(\mathbf{X})$ from (7.107), right (see Fig. 7.7, centre), and the planes $\mathcal{A}_4$, $\mathcal{A}_5$, and $\mathcal{A}_6$ through $\mathcal{X}$ and the three coordinate axes as the last three rows of $\mathsf{\Pi}(\mathbf{X})$, see Fig. 7.7, right.

3D Line. As discussed above, we may represent a 3D line by its Plücker matrix $\mathsf{\Gamma}(\mathbf{L}) = -\mathsf{\Gamma}^\mathsf{T}(\mathbf{L})$ or by its dual Plücker matrix $\overline{\mathsf{\Gamma}}(\mathbf{L}) = -\overline{\mathsf{\Gamma}}^\mathsf{T}(\mathbf{L})$. They are used when intersecting the line with a plane $\mathcal{B}$ or when joining it with a point $\mathcal{Y}$,

$$\mathbf{X} = \mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{B} \quad \text{and} \quad \mathbf{A} = \overline{\mathsf{\Gamma}}^\mathsf{T}(\mathbf{L})\mathbf{Y}\,. \quad (7.108)$$

This allows us – as before – to interpret the columns of the two matrices geometrically (see Fig. 7.8).

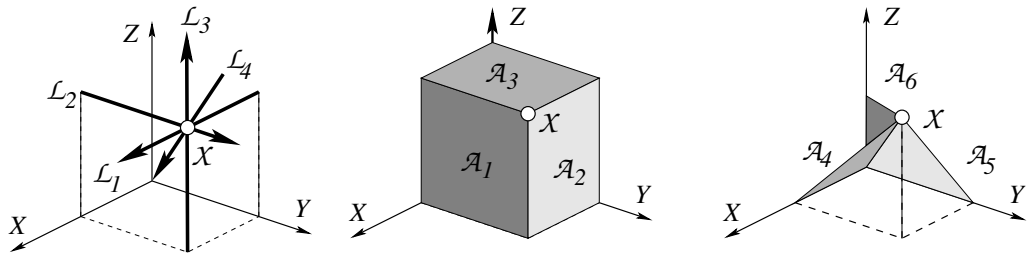


Fig. 7.7 Visualization of columns and rows of the matrix $\mathsf{\Pi}(\mathbf{X})$ representing point $\mathcal{X}$. **Left:** the four columns of $\mathsf{\Pi}(\mathbf{X})$ are lines. **Centre:** The first three rows of $\mathsf{\Pi}(\mathbf{X})$ are planes. **Right:** The last three rows of $\mathsf{\Pi}(\mathbf{X})$ are planes

We choose the basic planes $\boldsymbol{e}_i^{[4]}$ of the coordinate system one after the other as plane $\mathcal{B}$ and obtain as columns of $\mathsf{\Gamma}(\mathbf{L})$ the intersection points $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ of $\mathcal{L}$ with the coordinate planes, and with the plane at infinity, the point at infinity $\mathcal{X}_4$ of the line $\mathcal{L}$.

If we choose the basic points of the coordinate system as point $\mathcal{Y}$ we obtain as columns of $\overline{\mathsf{\Gamma}}(\mathbf{L})$ the join of $\mathcal{L}$ with the points at infinity of the three coordinate axes, thus the projection planes $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ of $\mathcal{L}$ perpendicular to the coordinate planes. Finally, we obtain the plane $\mathcal{A}_4$ through the line $\mathcal{L}$ and the origin.

The rows of the two skew matrices are the negative vectors of the columns; thus, they represent the antipodal points and antipodal planes of the points and planes given above, cf. Sect. 9.1, p. 344.

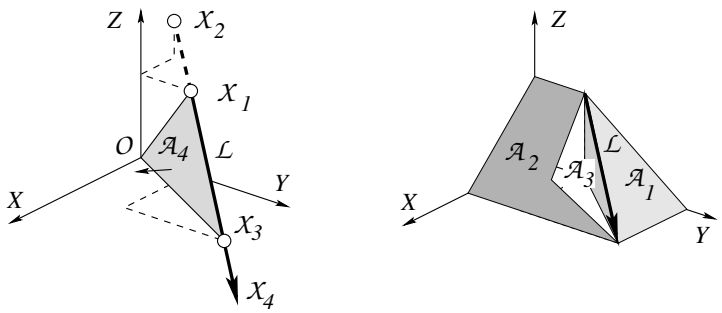


Fig. 7.8 Visualization of columns of the Plücker matrix $\mathsf{\Gamma}(\mathbf{L})$ as points, and of its dual $\overline{\mathsf{\Gamma}}(\mathbf{L})$ as planes. **Left:** The columns of $\mathsf{\Gamma}(\mathbf{L})$ and the last column of $\overline{\mathsf{\Gamma}}(\mathbf{L})$, see $\mathcal{A}_4$. **Right:** The first three columns of $\overline{\mathsf{\Gamma}}(\mathbf{L})$

Plane. Finally, we may represent a plane $\mathcal{A}$ by the matrix $\overline{\mathsf{\Pi}}(\mathbf{A})$, which is used for intersection with another plane $\mathcal{B}$ and with a line $\mathcal{L}$, respectively,

$$\mathbf{L} = \overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B} \quad \text{and} \quad \mathbf{X} = \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\mathbf{M}\,, \tag{7.109}$$

which again allows us to extract and interpret the columns and rows of $\overline{\mathsf{\Pi}}(\mathbf{A})$.

If we consecutively choose the plane $\mathcal{B}$ to be one of the basic planes, we obtain as columns of $\overline{\mathsf{\Pi}}(\mathbf{A})$ the intersection lines $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ of $\mathcal{A}$ with the coordinate planes and the line at infinity $\mathcal{L}_4$, see Fig. 7.9. For interpreting the rows of $\overline{\mathsf{\Pi}}$ or the columns of $\overline{\mathsf{\Pi}}^{\mathsf{T}}$, we now consecutively choose the base lines of the coordinate axes to be the lines $\mathcal{M}$. We start with $\overline{\mathbf{M}} = \boldsymbol{e}_1^{[6]}$ or $\mathbf{M} = \boldsymbol{e}_4^{[6]}$, the line at infinity of the YZ-plane. Thus the first row is the point at infinity $\mathcal{X}_1$ of the line $\mathcal{L}_1$. Similarly, we obtain as the second and third rows the points at infinity $\mathcal{X}_2$ and $\mathcal{X}_3$ of the lines $\mathcal{L}_2$ and $\mathcal{L}_3$. The last three unit vectors $\boldsymbol{e}_i^{[6]}, i = 1, 2, 3$, after dualizing, correspond to the X-, the Y-, and the Z-axes. Thus the last three rows of $\overline{\mathsf{\Pi}}(\mathbf{A})$ are the intersections of $\mathcal{A}$ with the axes of the coordinate system.

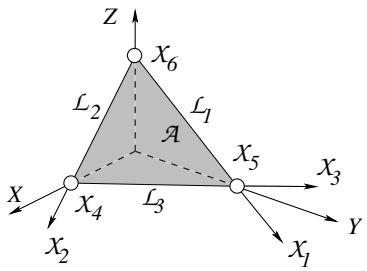


Fig. 7.9 Visualization of columns and rows of $\overline{\Pi}(\mathbf{A})$. The first three columns are the intersections $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ of $\mathcal{A}$ with the coordinate planes. The last column is the line at infinity $\mathcal{L}_4$ of the plane $\mathcal{A}$, not shown. The first three rows are the points at infinity $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ of these three intersection lines. The last three rows of $\overline{\Pi}(\mathbf{A})$ are the intersection points of $\mathcal{A}$ with the coordinate axes.

7.3.3 Vector Representations of Transformations and Conics

Vector representations of transformation matrices or of conics are needed when estimating the parameters of the matrices. For this, we make use of the two operators vec(A), for stacking the columns of general matrices, and vech(C), for stacking the lower triangular submatrix of a symmetric matrix, and of the Kronecker product $\mathsf{C} = \mathsf{A} \otimes \mathsf{B} = [a_{ij}\mathsf{B}]$ of two matrices, cf. App. A.7, p. 775.

7.3.3.1 Vector Representations of Transformations

The general projective collineation or homography in 2D may be written in forms

$$\mathbf{x}' = \mathsf{H}\mathbf{x} = \begin{bmatrix} \mathbf{l}_1^\mathsf{T} \\ \mathbf{l}_2^\mathsf{T} \\ \mathbf{l}_3^\mathsf{T} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{l}_1^\mathsf{T}\mathbf{x} \\ \mathbf{l}_2^\mathsf{T}\mathbf{x} \\ \mathbf{l}_3^\mathsf{T}\mathbf{x} \end{bmatrix} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{7.110}$$

or as

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} \mathbf{x}^\mathsf{T} & \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} & \mathbf{x}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} & \mathbf{x}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix} = [x_1 I_3,\, x_2 I_3,\, x_3 I_3] \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}. \tag{7.111}$$

This yields the three compact forms

$$\mathbf{x}' = \mathsf{H}\mathbf{x} = (I_3 \otimes \mathbf{x}^\mathsf{T})\mathrm{vec}(\mathsf{H}^\mathsf{T}) = (\mathbf{x}^\mathsf{T} \otimes I_3)\mathrm{vec}(\mathsf{H}). \tag{7.112}$$

The last form with vec(H) matches the MATLAB code vec(H) = H(:), and we prefer it for that reason.

We usually need the constraint $\mathcal{X}' \equiv \mathcal{H}(\mathcal{X})$, which also requires fixing the scaling factor. This can be achieved implicitly by using the constraint $\mathbf{x}' \times \mathsf{H}\mathbf{x} = \mathbf{0}$ in one of the following forms

$$\mathsf{S}(\mathbf{x}')\,\mathsf{H}\,\mathbf{x} = \mathbf{0} \qquad -\mathsf{S}(\mathsf{H}\mathbf{x})\,\mathbf{x}' = \mathbf{0} \qquad (\mathbf{x}^\mathsf{T} \otimes \mathsf{S}(\mathbf{x}'))\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}\,, \tag{7.113}$$

each containing one of the three elements $\mathbf{x}, \mathbf{x}'$, and vec(H) as the last vector.

Transformations of 2D lines can also be written in this way. From $\mathbf{l} \times \mathsf{H}^\mathsf{T}\mathbf{l}' = \mathbf{0}$ we have (cf. (6.46), p. 258 neglecting the factor $|\mathsf{H}|$)

$$\mathsf{S}(\mathbf{l})\,\mathsf{H}^\mathsf{T}\,\mathbf{l}' = \mathbf{0} \qquad -\mathsf{S}(\mathsf{H}^\mathsf{T}\mathbf{l}')\,\mathbf{l} = \mathbf{0} \qquad (\mathsf{S}(\mathbf{l}) \otimes \mathbf{l}'^\mathsf{T})\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}\,. \tag{7.114}$$

We obtain similar relations for points and planes in 3D, cf. Table 7.7, p. 317.

Though the transformation matrix H_L of 3D lines is quadratic in the entries of the homography matrix H for points (cf. (6.53), p. 259 and (6.56), p. 259), we can arrive at a constraint for two 3D lines $\mathcal{L}$ and $\mathcal{L}'$ via a given homography $\mathcal{H}$ which is trilinear. This follows from (6.56), p. 259 and (7.73), p. 306:

$$\overline{\mathsf{\Gamma}}(\mathbf{L}')\mathsf{\Gamma}(\mathsf{H}_L\mathbf{L}) = \overline{\mathsf{\Gamma}}(\mathbf{L}')\mathsf{H}^\mathsf{T}\mathsf{\Gamma}(\mathbf{L})\mathsf{H} = \mathsf{0}\,. \tag{7.115}$$

Since H can be assumed to be regular, we obtain

$$\overline{\mathsf{\Gamma}}(\mathbf{L}')\mathsf{H}^\mathsf{T}\mathsf{\Gamma}(\mathbf{L}) = \mathsf{0}\,. \tag{7.116}$$

7.3.3.2 Vector Representation of Conics and Quadrics

Conics and quadrics are represented with symmetric matrices, cf. Sect. 5.7, p. 236. Using the vec operator and the Kronecker product, we can write a point conic as

$$\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = (\mathbf{x}^\mathsf{T} \otimes \mathbf{x}^\mathsf{T})\mathrm{vec}\mathsf{C} = 0\,. \tag{7.117}$$

However, as matrix C is symmetric, the entries appear twice in $\mathrm{vec}\mathsf{C}$. This is not favourable if we want to estimate the parameters. However, the constraint $a_{11}u^2 + 2a_{12}uv + 2a_{13}uw + a_{22}v^2 + 2a_{23}vw + a_{33}w^2 = 0$ can be written as

$$[u^2, uv, uw, v^2, vw, w^2] \begin{bmatrix} 1&0&0&0&0&0 \\ 0&2&0&0&0&0 \\ 0&0&2&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&2&0 \\ 0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{22} \\ a_{23} \\ a_{33} \end{bmatrix} = \mathrm{vech}^\mathsf{T}(\mathbf{x}\mathbf{x}^\mathsf{T})\mathsf{W}_3\mathrm{vech}\mathsf{C} = 0\,, \tag{7.118}$$

applying the vech operator to the dyad $\mathbf{x}\mathbf{x}^\mathsf{T}$ and the conic C, and weighting the off-diagonal elements with a factor of 2. The generalization to quadrics is straightforward. The relations are collected in Table 7.7.

7.4 Determining Transformations and Conics from a Minimum of Geometric Entities

This section collects minimal solutions for determining transformations and conics from a minimum number of given geometric entities. Minimal solutions can be seen as constructions of transformations from geometric entities.

Minimal solutions are of utmost importance when searching for good data in a set of contaminated data, e.g., using a RANSAC scheme (cf. Fischler and Bolles, 1981).

Most of the transformations lead to a linear equation system. Under certain conditions, it may be of advantage to work with transformed parameters.

We start with homographies, specialize to affinities and similarities, discuss perspectivities and conjugate rotations, and end with the determination of conics, especially ellipses. Many of the vector-valued constraints are linearly dependent algebraically; for instance the identity constraint $\mathsf{S}(\mathbf{x})\mathbf{y} = \mathbf{0}_{3\times1}$ of two points x and y only contains two linearly independent constraints. We discuss the suitable selection of independent constraints first.

Table 7.7 Geometric relations including transformations, conics, and quadrics useful for testing and estimating geometric entities and transformations

transformation	entity	d.o.f.	relations	Eq.
2D homography	point	2	$\mathsf{S}(\mathbf{x}')\,\mathsf{H}\,\mathbf{x} = \mathbf{0}$	(7.113)
			$\mathsf{S}(\mathsf{H}\mathbf{x})\,\mathbf{x}' = \mathbf{0}$	
			$(\mathbf{x}^\mathsf{T} \otimes \mathsf{S}(\mathbf{x}'))\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}$	
	line	2	$\mathsf{S}^\mathsf{T}(\mathsf{H}^\mathsf{T}\mathbf{l}')\,\mathbf{l} = \mathbf{0}$	(7.114)
			$\mathsf{S}(\mathbf{l})\,\mathsf{H}^\mathsf{T}\,\mathbf{l}' = \mathbf{0}$	
			$(\mathsf{S}(\mathbf{l}) \otimes \mathbf{l}'^\mathsf{T})\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}$	
3D homography	point	3	$\mathsf{\Pi}(\mathbf{X}')\,\mathsf{H}\,\mathbf{X} = \mathbf{0}$	
			$\mathsf{\Pi}(\mathsf{H}\mathbf{X})\,\mathbf{X}' = \mathbf{0}$	(7.113)
			$(\mathbf{X}^\mathsf{T} \otimes \mathsf{\Pi}(\mathbf{X}'))\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}$	
	plane	3	$\overline{\mathsf{\Pi}}(\mathsf{H}^\mathsf{T}\mathbf{A}')\,\mathbf{A} = \mathbf{0}$	
			$\overline{\mathsf{\Pi}}(\mathbf{A})\,\mathsf{H}^\mathsf{T}\,\mathbf{A}' = \mathbf{0}$	(7.114)
			$(\overline{\mathsf{\Pi}}(\mathbf{A}) \otimes \mathbf{A}'^\mathsf{T})\,\mathrm{vec}(\mathsf{H}) = \mathbf{0}$	
	line	4	$\overline{\mathsf{\Gamma}}(\mathbf{L}')\mathsf{H}^\mathsf{T}\mathsf{\Gamma}(\mathbf{L}) = \mathbf{0}$	(7.116)
			$(\overline{\mathsf{\Gamma}}(\mathbf{L}) \otimes \mathsf{\Gamma}(\mathbf{L}'))\mathrm{vec}(\mathsf{H}) = \mathbf{0}$	
conic	point	1	$\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0$	(5.146)
			$\mathrm{vech}(\mathbf{x}^\mathsf{T} \otimes \mathbf{x}^\mathsf{T})\boldsymbol{W}_3\mathrm{vech}\mathsf{C} = 0$	(7.117)
	line	1	$\mathbf{l}^\mathsf{T}\mathsf{C}^*\mathbf{l} = 0$	(5.157)
			$\mathrm{vech}(\mathbf{l}^\mathsf{T} \otimes \mathbf{l}^\mathsf{T})\boldsymbol{W}_3\mathrm{vech}\mathsf{C}^* = 0$	(7.117)
quadric	point	1	$\mathbf{X}^\mathsf{T}\mathsf{Q}\mathbf{X} = 0$	(5.158)
			$\mathrm{vech}(\mathbf{X}^\mathsf{T} \otimes \mathbf{X}^\mathsf{T})\boldsymbol{W}_4\mathrm{vech}\mathsf{Q} = 0$	(7.117)
	plane	1	$\mathbf{A}^\mathsf{T}\mathsf{Q}^*\mathbf{A} = 0$	(5.161)
			$\mathrm{vech}(\mathbf{A}^\mathsf{T} \otimes \mathbf{A}^\mathsf{T})\boldsymbol{W}_4\mathrm{vech}\mathsf{Q}^* = 0$	(7.117)

7.4.1 Selection of Independent Constraints in Identity and Incidence Relations

We discuss the principle of selecting independent constraints, first using the identity of two points in 2D and 3D and then the incidence of a line with a plane.

7.4.1.1 Selecting Independent Rows in $\mathsf{S}(\mathbf{x})$

We interpret the rows of $\mathsf{S}(\mathbf{x})$ as joins of the point $\mathcal{X}$ with the three canonical points of the coordinate system (cf. Sect. 7.3.2.1, p. 312) and thus obtain for the identity constraint $\mathsf{S}(\mathbf{x})\mathbf{y} = \mathbf{0}_{3\times 1}$

$$\mathcal{X} \equiv y: \quad \mathsf{S}(\mathbf{x})\mathbf{y} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \mathbf{y} = \begin{bmatrix} \mathbf{l}_1^\mathsf{T} \\ \mathbf{l}_2^\mathsf{T} \\ \mathbf{l}_3^\mathsf{T} \end{bmatrix} \mathbf{y} = \begin{bmatrix} \mathbf{l}_1^\mathsf{T}\mathbf{y} \\ \mathbf{l}_2^\mathsf{T}\mathbf{y} \\ \mathbf{l}_3^\mathsf{T}\mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{7.119}$$

The three constraints $\mathbf{l}_i^\mathsf{T}\mathbf{y} = 0, i = 1, 2, 3$, collected in the last two terms are linearly dependent, since multiplying the stacked coefficients $[\mathbf{l}_i^\mathsf{T}]$ for $\mathbf{y}$ with $\mathbf{x}^\mathsf{T}$ from the left yields zero.

We now select the index j such that $|x_j|$ is the largest of the three entries of $\mathbf{x}$. The element x_j appears in the two rows $\mathbf{l}_{j_1}$ and $\mathbf{l}_{j_2}$ of $\mathsf{S}(\mathbf{x})$. The corresponding constraints are linearly independent, due to the zeros on the diagonal.

Therefore, we arrive at the following scheme:

1. Determine j as index with maximum $|x_j|$ in $\mathbf{x}$.
2. Determine the matrix

$$\mathsf{S}^{(j)}(\mathbf{x}) = \underbrace{\begin{bmatrix} \boldsymbol{e}_{j_1}^{[3]\mathsf{T}} \\ \boldsymbol{e}_{j_2}^{[3]\mathsf{T}} \end{bmatrix}}_{M^{(s)}(\mathbf{x})} \mathsf{S}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{l}_{j_1}^{\mathsf{T}} \\ \boldsymbol{l}_{j_2}^{\mathsf{T}} \end{bmatrix}, \tag{7.120}$$

where

$$j \to (j_1, j_2): \quad \{1 \to (2,3),\ 2 \to (3,1),\ 3 \to (1,2)\} \tag{7.121}$$

independently of the index j, and the 2×3 matrix with two selected independent rows is denoted by $\mathsf{S}^{(s)}$. Then the two constraints in

matrix $\mathsf{S}^{(s)}(\mathbf{x})$ with selected independent rows

$$\mathsf{S}^{(s)}(\mathbf{x})\,\mathbf{y} = \mathbf{0}_{2\times 1} \tag{7.122}$$

are linearly independent.[4] It is useful to have a function which, for a given vector $\mathbf{x}$, yields both matrices, $M^{(s)}(\mathbf{x})$ and $\mathsf{S}^{(s)}(\mathbf{x})$, as this makes it possible to select *the same constraints* in $\mathsf{S}(\mathbf{x})\mathbf{y} = \mathbf{0}$ and $-\mathsf{S}(\mathbf{y})\mathbf{x} = \mathbf{0}$ by multiplication of the two forms with $M^{(s)}(\mathbf{x})$ from the left:

$$\mathsf{S}^{(s)}(\mathbf{x}) = M^{(s)}(\mathbf{x})\,\mathsf{S}(\mathbf{x}) \qquad \mathsf{S}^{(s)}(\mathbf{y}) = M^{(s)}(\mathbf{x})\,\mathsf{S}(\mathbf{y})\,. \tag{7.123}$$

Moreover, the two lines $\mathcal{l}_{j_1}$ and $\mathcal{l}_{j_2}$ can be used to represent the point $\mathcal{x}$,

$$\mathcal{x}: \quad \{\mathcal{l}_{j_1}, \mathcal{l}_{j_2}\}, \tag{7.124}$$

as they are distinct and intersect in $\mathcal{x}$. Similarly, a line can be represented by two distinct points

$$\mathcal{l}: \quad \{\mathcal{x}_{j_1}, \mathcal{x}_{j_2}\}, \tag{7.125}$$

by selecting two independent (j_1, j_2) rows from $\mathsf{S}(\mathbf{l})$.

This selection principle also works for points or lines at infinity. All constraints containing a skew symmetric matrix $\mathsf{S}(\cdot)$ can be handled this way.

7.4.1.2 Selecting Independent Rows in $\mathsf{\Pi}(\mathbf{X})$ and $\overline{\mathsf{\Pi}}(\mathbf{A})$

Checking the identity of two points or of two planes involves the constraint matrices $\mathsf{\Pi}(\mathbf{X})$ and $\overline{\mathsf{\Pi}}(\mathbf{A})$, respectively. They have three degrees of freedom. We only discuss the situation for points, cf. (7.107), p. 313. Thus, only three out of the following six constraints are linearly independent:

$$\mathcal{X} \equiv \mathcal{Y}: \quad \mathsf{\Pi}(\mathbf{X})\mathbf{Y} = \left[\begin{array}{ccc|c} X_4 & 0 & 0 & -X_1 \\ 0 & X_4 & 0 & -X_2 \\ 0 & 0 & X_4 & -X_3 \\ \hline 0 & -X_3 & X_2 & 0 \\ X_3 & 0 & -X_1 & 0 \\ -X_2 & X_1 & 0 & 0 \end{array}\right] \mathbf{Y} = \mathbf{0}_{6\times 1}\,. \tag{7.126}$$

We need to select the three rows corresponding to the indices with the largest value in $\mathbf{X}$. Thus, we use the reduced matrix

[4] It would be consistent to call this matrix $\mathsf{S}^{(s)\mathsf{T}}(\mathbf{x})$, as it has fewer rows than columns. However, the notation is too cumbersome and does not easily generalize to the cases discussed below.

$$\mathsf{\Pi}^{(s)}(\mathbf{X}) := \mathsf{\Pi}^{(j)}(\mathbf{X}) = \begin{bmatrix} \boldsymbol{e}_{j_1}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{j_2}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{j_3}^{[4]\mathsf{T}} \end{bmatrix} \mathsf{\Pi}(\mathbf{X}) = \begin{bmatrix} \boldsymbol{A}_{j_1}^{\mathsf{T}} \\ \boldsymbol{A}_{j_2}^{\mathsf{T}} \\ \boldsymbol{A}_{j_3}^{\mathsf{T}} \end{bmatrix}, \tag{7.127}$$

with

$$j \to (j_1, j_2, j_3): \quad \{1 \to (1,5,6),\ 2 \to (2,4,6),\ 3 \to (3,4,5),\ 4 \to (1,2,3)\}. \tag{7.128}$$

The selection leads to three planes, $\mathcal{A}_{j_1}$, $\mathcal{A}_{j_2}$, $\mathcal{A}_{j_3}$, representing the point $\mathcal{X}$,

$$\mathcal{X}: \quad \{\mathcal{A}_{j_1}, \mathcal{A}_{j_2}, \mathcal{A}_{j_3}\}, \tag{7.129}$$

as they are distinct and intersect in $\mathcal{X}$.

Dualizing this argument, i.e., applying it to $\overline{\mathsf{\Pi}}(\mathbf{A})$, we arrive at three distinct points representing the plane,

$$\mathcal{A}: \quad \{\mathcal{X}_{j_1}, \mathcal{X}_{j_2}, \mathcal{X}_{j_3}\}, \tag{7.130}$$

depending on the index j of the largest element in $\mathbf{A}$ and adapting the mapping $j \to (j_1, j_2, j_3)$.

7.4.1.3 Selecting Independent Rows in $\mathsf{\Gamma}(\mathbf{L})$ and $\overline{\mathsf{\Gamma}}(\mathbf{L})$

Constraints for the plane–line incidence involve the Plücker matrices $\mathsf{\Gamma}(\mathbf{L})$ and $\overline{\mathsf{\Gamma}}(\mathbf{L})$. They have two degrees of freedom, which can be used to select two distinct points or two distinct planes representing the line $\mathcal{L}$. For example, the four dependent constraints in

$$\mathcal{L} \in \mathcal{A}: \quad \mathsf{\Gamma}(\mathbf{L})\mathbf{A} = \left[\begin{array}{ccc|c} 0 & L_6 & -L_5 & -L_1 \\ -L_6 & 0 & L_4 & -L_2 \\ L_5 & -L_4 & 0 & -L_3 \\ \hline L_1 & L_2 & L_3 & 0 \end{array}\right] \mathbf{A} = \mathbf{0}_{4\times 1} \tag{7.131}$$

can be reduced to two linearly independent constraints, leading to the reduced Plücker matrix

$$\mathsf{\Gamma}^{(s)}(\mathbf{L}) := \mathsf{\Gamma}^{(j)}(\mathbf{L}) = \begin{bmatrix} \boldsymbol{e}_{j_1}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{j_2}^{[4]\mathsf{T}} \end{bmatrix} \mathsf{\Gamma}(\mathbf{L}) = \begin{bmatrix} \boldsymbol{X}_{j_1}^{\mathsf{T}} \\ \boldsymbol{X}_{j_2}^{\mathsf{T}} \end{bmatrix}, \tag{7.132}$$

with

$$\begin{aligned} &j \to (j_1, j_2): \\ &\{1 \to (1,4),\ 2 \to (2,4),\ 3 \to (3,4),\ 4 \to (2,3),\ 5 \to (1,3),\ 6 \to (1,2)\}, \end{aligned} \tag{7.133}$$

depending on the index j where the element L_j has largest absolute value in $\mathbf{L}$. We then obtain two distinct 3D points representing the 3D line

$$\mathcal{L}: \quad \{\mathcal{X}_{j_1}, \mathcal{X}_{j_2}\}. \tag{7.134}$$

Analogously, from the dual Plücker matrix $\overline{\mathsf{\Gamma}}(\mathbf{L})$ of a line, we would select two distinct planes through $\mathcal{L}$,

$$\mathcal{L}: \quad \{\mathcal{A}_{j_1}, \mathcal{A}_{j_2}\}. \tag{7.135}$$

As mentioned, the selection principle works for both real and ideal geometric elements. It can be shown that the selection leads to the most stable representation in the sense that numerical uncertainty in the representing elements leads to the least numerical uncertainty of the represented element. In the special case of canonical entities, we observe: The canonical entities of 2D and 3D coordinate systems are represented by other canonical

elements. For instance, the plane at infinity is represented by the three points at infinity in the direction of the three coordinate axes.

As an example for using this selection, take the identity constraint for two lines $\mathcal{L} \equiv \mathcal{M}$: $\mathsf{\Gamma}(\mathbf{L})\overline{\mathsf{\Gamma}}(\mathbf{M}) = \mathsf{0}_{4\times4}$ (cf. (7.73), p. 306), which contains 16 linearly dependent constraints. As 3D lines have four degrees of freedom, only four constraints are necessary for checking their identity. In order to select four independent constraints out of the 16 constraints, we select the indeces l and m for the largest elements in $\mathcal{L}$ and $\mathcal{M}$ and the corresponding rows in $\mathsf{\Gamma}(\mathbf{L})$ and $\overline{\mathsf{\Gamma}}(\mathbf{M})$, respectively, and obtain the four constraints

$$\mathsf{\Gamma}^{(s)}(\mathbf{L})\,\overline{\mathsf{\Gamma}}^{(s)}(\mathbf{M}) = \begin{bmatrix} \boldsymbol{e}_{l_1}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{l_2}^{[4]\mathsf{T}} \end{bmatrix} \mathsf{\Gamma}(\mathbf{l})\,\overline{\mathsf{\Gamma}}(\mathbf{M}) \left[\boldsymbol{e}_{m_1}^{[4]},\ \boldsymbol{e}_{m_2}^{[4]}\right] = \begin{bmatrix} \mathbf{A}_{l_1}^{\mathsf{T}} \\ \mathbf{A}_{l_2}^{\mathsf{T}} \end{bmatrix} [\mathbf{X}_{m_1}\ \ \mathbf{X}_{m_2}] = \mathsf{0}_{2\times2} \tag{7.136}$$

which express nothing more than the fact that the two points representing $\mathcal{L}$ need to pass through the two planes representing $\mathcal{M}$.

7.4.2 Affinity and Similarity

7.4.2.1 Minimal Solution for 2D Affinity

We start with determining the six parameters of a 2D affinity, cf. Table 6.1, p. 254. We assume a set of corresponding points is available. When employing Euclideanly normalized coordinates for the points this allows us to use the relation

$$\mathbf{x}_i'^e = \mathsf{A}\mathbf{x}_i^e\,, \tag{7.137}$$

as no scaling is necessary. Each point induces two constraints, so we need three corresponding points. If three lines $l_i, i = 1, 2, 3$ are given, with no pair of them parallel, we can determine the three intersection points and apply the same scheme.

We exploit the special structure of the affinity and write it as

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{7.138}$$

with the six parameters $[a, b, c, d, e, f]$. Obviously, the determination of $[a, b, c]$ does not depend on the y_i'-coordinates and, similarly, the parameters $[d, e, f]$ do not depend on the x_i' coordinates. Thus, we can compress the determination of the six parameters into the system

$$\begin{bmatrix} x_1' & y_1' \\ x_2' & y_2' \\ x_3' & y_3' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \tag{7.139}$$

and directly solve for the six parameters.

7.4.2.2 Minimal Solutions for 2D and 3D Similarity Transformations

As a 2D similarity has four degrees of freedom, we need two point correspondences. Two line correspondences are not sufficient, as the mapping scale cannot be determined. A 3D similarity requires at least three points for the determination of the seven parameters, which already induces two additional constraints, namely an angle and a distance ratio of two sides of the spatial triangle, defined by the given correspondences.

Therefore, there is only the minimal, nonredundant solution for the 2D similarity, which is left as an exercise.

Exercise 7.9

7.4.3 General Homographies in 2D and 3D

7.4.3.1 2D Homography

A 2D homography $\mathcal{H}(\mathsf{H})$ has eight degrees of freedom, as only the ratios of the nine elements are relevant. We assume a set of corresponding entities, points or lines, is available. This allows us to use selected constraints, cf. Sect. 7.4.1.1, p. 317

$$\boldsymbol{g}_i(\mathbf{x}_i, \mathbf{x}'_i, \mathsf{H}) = \mathsf{S}^{(s)}(\mathbf{x}'_i)\mathsf{H}\mathbf{x}_i = -\mathsf{S}^{(s)}(\mathbf{l}_i)\mathsf{H}^\mathsf{T}\mathbf{l}'_i = (\mathbf{x}_i^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_i))\mathrm{vec}\mathsf{H} = \mathbf{0}\,. \tag{7.140}$$

As each point or line pair induces two constraints, we need four points or four lines, or a mixture of both. We only discuss the two cases of four points and four lines. For the other cases cf. Hartley and Zisserman (2000, Sect. 4.1.4).

We write the four point constraints as

$$\begin{bmatrix} \mathbf{x}_1^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_1) \\ \mathbf{x}_2^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_2) \\ \mathbf{x}_3^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_3) \\ \mathbf{x}_4^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_4) \end{bmatrix} \mathrm{vec}\mathsf{H} = \mathbf{0}_{8\times 1}\,. \tag{7.141}$$

This is a homogeneous equation system with eight equations for the nine parameters in H, written row by row.

If we assume that no point is at infinity, and all points have been conditioned (cf. Sect. 6.9, p. 286) and normalized, then the third coordinate of a point is always $\neq 0$ and is the largest coordinate, thus $j = 3$ in (7.140), and so we can work with Euclideanly normalized coordinates:

$$\underbrace{\begin{bmatrix} 0 & -x_1 & x_1y'_1 & 0 & -y_1 & y_1y'_1 & 0 & -1 & y'_1 \\ x_1 & 0 & -x_1x'_1 & y_1 & 0 & -y_1x'_1 & 1 & 0 & -x'_1 \\ 0 & -x_2 & x_2y'_2 & 0 & -y_2 & y_2y'_2 & 0 & -1 & y'_2 \\ x_2 & 0 & -x_2x'_2 & y_2 & 0 & -y_2x'_2 & 1 & 0 & -x'_2 \\ 0 & -x_3 & x_3y'_3 & 0 & -y_3 & y_3y'_3 & 0 & -1 & y'_3 \\ x_3 & 0 & -x_3x'_3 & y_3 & 0 & -y_3x'_3 & 1 & 0 & -x'_3 \\ 0 & -x_4 & x_4y'_4 & 0 & -y_4 & y_4y'_4 & 0 & -1 & y'_4 \\ x_4 & 0 & -x_4x'_4 & y_4 & 0 & -y_4x'_4 & 1 & 0 & -x'_4 \end{bmatrix}}_{\boldsymbol{B}} \begin{bmatrix} h_{11} \\ h_{21} \\ h_{31} \\ h_{12} \\ h_{22} \\ h_{32} \\ h_{13} \\ h_{23} \\ h_{33} \end{bmatrix} = \mathbf{0}\,. \tag{7.142}$$

Thus, the unknown vector of the transformation parameters is the right null space of the 8×9 matrix $\boldsymbol{B}$:

minimal solution for the 2D homography

$$\mathrm{vec}\mathsf{H} = \mathsf{null}(\boldsymbol{B})\,, \tag{7.143}$$

cf. Sect. A.11, p. 777. In case any three points are collinear, the rank of the matrix $\boldsymbol{B}$ drops below 8 and no unique solution is available.

In case one or two points are at infinity,[5] the matrix $\boldsymbol{B}$ has to be set up following (7.141), without assuming the third coordinate of the points to be the largest one.

We will discuss the representation of the uncertainty of a homography in Sect. 10.2.3.3, p. 384, the uncertainty of a homography derived from uncertain points in Sect. 10.3.1.3, p. 387, and the rigorous estimation of a homography from more than four points in Sect. 10.6.3, p. 424.

Both 2D elations and conjugate rotations can be determined from two corresponding points.

Exercise 7.11
Exercise 7.12

[5] Three points at infinity would not allow a solution as they are collinear.

7.4.3.2 3D Homography

The procedure can easily be transferred to determine homographies in 3D for points, lines, or planes. We need five 3D points or five planes to determine the 15 parameters of a spatial homography, as each pair of geometric entities results in three linearly independent constraints. Two corresponding lines lead to four independent constraints, so we need four corresponding lines; but then we already have one redundant constraint. Situations with redundant constraints will be discussed when estimating homographies, cf. Sect. 10.6.3, p. 424.

7.5 Exercises

Basics

1. (2) Derive an expression for the angle $\alpha_{x,y,z}$ between the lines $y \wedge x$ and $y \wedge z$ which lies in the range $[0, 2\pi)$.
2. (1) Give an explicit expression for three 3D points to be collinear, show that the scalar constraint $h(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ is linear in all homogeneous coordinates $\mathbf{x}_i$ and give the Jacobian of the residual of the constraint with respect to the homogeneous coordinates of the three points. Interprete the result.
3. (1) Give a simple expression for the Jacobians $\partial D/\partial \mathbf{X}$ and $\partial D/\partial \mathbf{Y}$, where $D = |\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}|$. Check the result using the origin and the unit points on the three axes.
4. (3) Assume a vehicle V_1 moves arbitrarily in a plane and its position $\boldsymbol{x}_i$ and orientation α_i are known. It observes the directions to a second vehicle V_2, which is assumed to move on a straight line at constant speed. Show that observations $(\boldsymbol{x}_i, \alpha_i), i = 1, ..., 4$, at four different times are needed to derive the path of the second vehicle, i.e., its position, direction and speed at a time t_i. Is there any configuration where the problem cannot be solved?
   Hint: Model the situation in space-time with XY representing the plane and Z representing the time.
5. (2) Show that shifting a line $\mathbf{L}$ by $\boldsymbol{T}$ into $\mathbf{L}'$ yields $\mathbf{L}' = [\boldsymbol{L}_h, \boldsymbol{L}_0 - \boldsymbol{T} \times \boldsymbol{L}_h]$.
6. (3) Give explicit expressions for the following planes together with the Jacobians with respect to the given entities. Assume the given entities are in general position.
   a. (1) The plane $\mathcal{A}$ passes through three points $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$.
   b. (2) The plane $\mathcal{A}$ passes through a point $\mathcal{X}$ and is parallel to a plane $\mathcal{B}$.
   c. (2) The plane $\mathcal{A}$ passes through a point $\mathcal{X}$ and is orthogonal to a line $\mathcal{L}$.
   d. (2) The plane $\mathcal{A}$ passes through a line $\mathcal{L}$ and is parallel to a line $\mathcal{M}$.
   e. (2) The plane $\mathcal{A}$ passes through point $\mathcal{X}$ and $\mathcal{Y}$ and is parallel to a line $\mathcal{L}$.
   f. (2) The plane $\mathcal{A}$ passes through point $\mathcal{X}$ and is parallel to two lines $\mathcal{L}$ and $\mathcal{M}$.
7. (3) Give explicit expressions for the following 3D lines together with the Jacobians w.r.t. the given entities, which are assumed to be in general position.
   a. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$ and is parallel to two planes $\mathcal{A}$ and $\mathcal{B}$.
   b. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$, is orthogonal to a line $\mathcal{M}$, and is parallel to a plane $\mathcal{A}$.
   c. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$ and is orthogonal to two lines $\mathcal{M}$ and $\mathcal{N}$.
   d. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$ and passes through two lines $\mathcal{M}$ and $\mathcal{N}$.
   e. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$, passes through a line $\mathcal{M}$, and is orthogonal to a line $\mathcal{N}$.
   f. (2) The line $\mathcal{L}$ passes through a point $\mathcal{X}$, passes through a line $\mathcal{M}$ and is orthogonal to a plane $\mathcal{A}$.

g. (2) The line $\mathcal{L}$ passes through two lines $\mathcal{L}$ and $\mathcal{M}$, and is orthogonal to these lines.
h. (2) The line $\mathcal{L}$ lies in a plane $\mathcal{A}$ and passes through two lines $\mathcal{M}$ and $\mathcal{N}$.
i. (2) The line $\mathcal{L}$ lies in a plane $\mathcal{A}$, passes through a line $\mathcal{M}$, and is orthogonal to a line $\mathcal{N}$.

8. (3) Give explicit expressions for the following 3D points and give their Jacobians w.r.t. the given entities, which are assumed to be in general position.

   a. (2) The point $\mathcal{X}$ lies on three planes $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$.
   b. (2) The point $\mathcal{X}$ lies in the plane $\mathcal{A}$ and is closest to the point $\mathcal{Y}$; thus, the point $\mathcal{X}$ is the foot point of $\mathcal{Y}$ on $\mathcal{A}$.
   c. (2) The foot point of the origin on a 3D line is the point $\mathcal{X}$ which lies on the line $\mathcal{L}$ and is closest to the origin. Prove

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{L}_h \times \boldsymbol{L}_0 \\ |\boldsymbol{L}_h| \end{bmatrix} \tag{7.144}$$

   (see Fig. 5.16, p. 218).
   d. (2) The point $\mathcal{X}$ lies on the line $\mathcal{L}$ and is closest to the point $\mathcal{Y}$.
   e. (2) The point $\mathcal{X}$ which has shortest distances from a point $\mathcal{Y}$ and a plane $\mathcal{A}$.
   f. (2) The point $\mathcal{X}$ which has shortest distances from a point $\mathcal{Y}$ and a line $\mathcal{L}$.
   g. (2) The point $\mathcal{X}$ is closest to two lines $\mathcal{L}$ and $\mathcal{M}$ (cf. Exercise 12, p. 618).

9. (1) Give an explicit expression for determining the four parameters of a similarity transformation from two corresponding points using

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} . \tag{7.145}$$

10. (2) Represent points $\mathcal{X}_i$ in the plane with complex numbers $z_i = x_i + jy_i$, with $j^2 = -1$. Show that the mapping $z' = mz + t$ with complex numbers is a similarity. Determine the complex parameters m and t of a similarity from two point pairs, (z_i, z_i'). Relate the representation with complex numbers to the representation in the previous exercise.
11. (1) Determine the five parameters of a planar elation from two point pairs, $(\mathcal{X}_i, \mathcal{X}_i')$, $i = 1, 2$.
12. (2) Determine the four parameters in R and $\mathsf{K} = \mathrm{Diag}([c, c, 1])$ of the conjugate rotation in (6.126), p. 282 from two point pairs, $(\mathcal{X}_i, \mathcal{X}_i')$, $i = 1, 2$.
    Hint: First determine c by requiring the angles $\angle(\mathbf{x}_i, \mathbf{x}_i')$, $i = 1, 2$, to be equal. This leads to a third-degree polynomial in c^2 (cf. Brown et al., 2007; Jin, 2008; Kúkelová and Pajdla, 2007; Kúkelová et al., 2010).
13. (1) Prove (7.34), p. 299. *Hint:* Express the normals of the two lines with the direction vector and apply the trigonometric rules for the difference of two angles.

Computer Experiments

14. (3) Evaluate the efficiency of the constructions in Table 7.1, p. 293 w.r.t. the number of operations. Code the expressions as a function of the individual vector elements and compare the number of multiplications for these expressions with those with matrix and vector multiplications thereby avoiding the multiplications involving zeros. How large is the expected increase in speed? Verify this speed advantage empirically by measuring CPU times.
15. (2) Given a regular conic $\mathcal{C}$ and a point $\mathcal{X}(\boldsymbol{x})$, write a computer program `y = footpoint_on_regular_conic(C,x)` to find the foot point, i.e., the point $\mathcal{Y}(\boldsymbol{y})$ on the conic which is closest to $\mathcal{X}$. *Hint:* Use a parametric form of the conic.

a. Translate and rotate the point and the conic such that the conic has its centre at the origin and the semi-axis corresponding to the largest eigenvalue is the x-axis.
b. Represent the conic in parametric form as

$$\boldsymbol{y}(\phi) = \begin{bmatrix} y_1(\phi) \\ y_2(\phi) \end{bmatrix} = \begin{bmatrix} a\cos\phi \\ b\sin\phi \end{bmatrix}, \qquad \text{or} \qquad \boldsymbol{y}(\phi) = \begin{bmatrix} y_1\phi \\ y_2\phi \end{bmatrix} = \begin{bmatrix} a\cosh\phi \\ b\sinh\phi \end{bmatrix}, \quad (7.146)$$

depending on whether the conic is an ellipse or a hyperbola.
c. Use the direction vector $\boldsymbol{v}$ of the tangent at $\boldsymbol{y}$,

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -a\sin\phi \\ b\cos\phi \end{bmatrix}, \qquad \text{or} \qquad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} a\sinh\phi \\ b\cosh\phi \end{bmatrix}. \quad (7.147)$$

d. Using $\phi_0 = \operatorname{atan2}(x_2, x_1)$ as initial value, solve the equation

$$\boldsymbol{v}^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) = 0 \quad (7.148)$$

for ϕ numerically (in MATLAB use `fzero`).
e. Check the program for points χ in all four quadrants and on all four axes.
f. Show that the solution always converges to the correct value. What happens if χ is the origin?

Chapter 8
Rotations

8.1 Rotations in 3D 325
8.2 Concatenation of Rotations 337
8.3 Relations Between the Representations for Rotations 338
8.4 Rotations from Corresponding Vector Pairs 339
8.5 Exercises 340

This chapter discusses rotations in 3D as special transformations of points. Rotations play a central role in modelling natural phenomena. They are often part of rigid body motions and as such are shape preserving transformations. Rotations deserve special attention, as often the constraint of rigid body motion can be used to advantage. The number of parameters is significantly smaller than with the more general transformations and more efficient and more stable solutions of estimation problems are possible.

Since the structure of rotations is very rich there exist various representations, where each of them is useful for certain tasks. Besides the trivial representation with the rotation matrix we discuss several minimal representations with three parameters. The representation of a rotation matrix based on the exponential form of a skew symmetric matrix is the starting point for optimally estimating rotations and for representing uncertain rotations, which in Chap. 10, p. 359 will be generalized to the other transformations. Since all minimal representations have singularities in the sense that the estimation of the rotation parameters may fail at or close to certain rotations, we also discuss quaternions, which can be used for a homogeneous representation of rotations. The chapter ends with closed form solutions for determining rotations from pairs of directions.

We especially discuss rotations in 3D and assume the reader is familiar with 2D rotations. We discuss various representations for rotations and their particular pros and cons.

A remark on notation: As this section is primarily devoted to 3D rotations, we simplify notation and use *small* boldface letters to denote 3-vectors.

8.1 Rotations in 3D

8.1.1 Exponential Form of the Rotation Matrix 326
8.1.2 Elements of the Rotation Matrix for Representation 327
8.1.3 Rotations with Euler Angles 328
8.1.4 Rotation with Rotation Axis and Rotation Angle 331
8.1.5 Rotations with Quaternions 332
8.1.6 Differential Rotations 336

Rotations are linear transformations $\mathcal{R}$ of the n-dimensional space on itself,

$$\mathcal{R} : \mathbb{R}^n \to \mathbb{R}^n \qquad \boldsymbol{x}' = \mathsf{R}\boldsymbol{x}\,. \tag{8.1}$$

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_8

The rotation matrices build the group of special, orthogonal n-dimensional linear mappings $SO(n)$: special, because $|R| = +1$; orthogonal, because $R^\mathsf{T} = R^{-1}$ or $R^\mathsf{T}R = RR^\mathsf{T} = I_n$. While the group $SO(2)$ of 2D rotations of the plane is commutative, this does not hold for any other group $SO(n)$ with $n > 2$. The concatenation rules for general homographies, discussed in Sect. 6.3.2, p. 261, specifically hold for rotations. Rotations are length preserving. According to Euler's rotation theorem (cf. e.g., Kanatani, 1990, p. 202), in 3D there exists a rotation axis.

There are various representations for rotations in 3D. We discuss the following:

1. Representation with the *matrix exponential* of a skew symmetric matrix. It is very useful for modelling motions using differential equations. The main properties of skew symmetric matrices are collected in App. A.4, p. 770.
2. Direct representation with the elements of a 3×3 matrix constrained by the orthonormal relationships. This representation is useful in many estimation problems if no error analysis is necessary, e.g., when deriving approximate values.
3. Representation with *Euler angles*. It is necessary when modelling sensor orientation (e.g., of a camera) in instruments, in vehicles, in aircraft, or in robotics, or when visualizing motions. The representation with angles shows singularities in certain configurations in the process of parameter estimation. Concatenation can only be performed via the corresponding rotation matrix.
4. Representation with *axis and angle*. Rotations in 3D map the 2D-sphere S^2 onto itself. The representation with axis and angle is useful when modelling motions using differential equations, especially when modelling small rotations. Estimation of large rotations is cumbersome. Again, concatenation can only be performed via the corresponding rotation matrix.
5. Representation with *quaternions*. It consists at most of quadratic parameter terms and is the only representation which shows no singularities in the complete range of 3D rotations. Concatenation can be performed directly on the parameter level. Two common representations, in particular the one by Rodriguez and Cayley, are special cases of the quaternion representation; however, they can only represent rotations not equal to 180°.

We will discuss the relations between the different representations, how to derive the parameters from a given rotation matrix, how to determine rotations from given point pairs, and how to represent differential rotations, which are required for rigorous estimation procedures.

8.1.1 Exponential Form of the Rotation Matrix

Rotation matrices can be expressed as matrix exponentials of skew symmetric 3×3 matrices. Let the vector $\boldsymbol{\theta}$ parametrize the rotation; then with the skew matrix $S_\theta = S(\boldsymbol{\theta})$, we obtain the corresponding rotation matrix,

$$R(\boldsymbol{\theta}) = \mathrm{e}^{S(\boldsymbol{\theta})} . \tag{8.2}$$

This can directly be proven using the definition of the matrix exponential:

$$R(\boldsymbol{\theta}) = I_3 + S_\theta + \frac{1}{2!}S_\theta^2 + \frac{1}{3!}S_\theta^3 + \frac{1}{4!}S_\theta^4 + \dots . \tag{8.3}$$

The vector $\boldsymbol{\theta}$ is parallel to the rotation axis $\boldsymbol{\theta} = R(\boldsymbol{\theta})\,\boldsymbol{\theta}$, as all products with $\boldsymbol{\theta}$ are zero except for the first one.

Remark: The exponential representation is valid in all dimensions: Since an $N \times N$-skew symmetric matrix has $\binom{N}{2}$ nondiagonal independent entries, rotation matrices likewise has $\binom{N}{2}$ degrees of freedom.

Therefore, a 2×2-rotation matrix depends on one parameter, a 3×3-rotation matrix depends on three independent parameters. ◇

We now seek an explicit expression for the rotation matrix. Collecting the even and the odd terms and using the series for $\sin(\theta)$ and $\cos(\theta)$ we can show Exercise 8.7

$$R(\boldsymbol{\theta}) = I_3 + \frac{\sin\theta}{\theta} S_\theta + \frac{1-\cos\theta}{\theta^2} S_\theta^2 \tag{8.4}$$

with $\theta = |\boldsymbol{\theta}|$. This equation was first given by Rodriguez (1840). We will later derive this representation in a purely geometrical way when discussing the axis-angle representation, and we will show that the rotation angle θ is $|\boldsymbol{\theta}|$.

We can invert (8.2),

$$S_\theta = \ln R(\boldsymbol{\theta}) \,, \tag{8.5}$$

and in this way determine the rotation axis and the rotation angle. Of course, we can only derive rotation angles in the principal range $(-\pi, \pi)$. Note that the eigenvalues of any rotation matrix R are

$$\lambda_1 = 1 \qquad \lambda_{2,3} = \mathrm{e}^{\pm i\theta} = \cos\theta \pm i \sin\theta \,. \tag{8.6}$$

They result from (A.36), p. 771 and (8.2). The eigenvalue $\lambda_1 = 1$ leads to $\boldsymbol{r} = R\boldsymbol{r}$, where $\boldsymbol{r}$ is the direction of the rotation axis. Exercise 8.8

8.1.2 Elements of the Rotation Matrix for Representation

We now consider the representation of a rotation by the rotation matrix itself with its elements, columns, and rows:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3] = \begin{bmatrix} \boldsymbol{r}_1^\mathsf{T} \\ \boldsymbol{r}_2^\mathsf{T} \\ \boldsymbol{r}_3^\mathsf{T} \end{bmatrix} . \tag{8.7}$$

The representation is obviously redundant since a rotation only has three degrees of freedom. For symmetry reasons the relationship $R^\mathsf{T} R = I_3$ contains six constraints, e.g., with its columns $\boldsymbol{c}_i$,

$$|\boldsymbol{c}_1|^2 = 1 \,, \qquad |\boldsymbol{c}_2|^2 = 1 \,, \qquad |\boldsymbol{c}_3|^2 = 1 \,, \tag{8.8}$$

$$\boldsymbol{c}_1^\mathsf{T} \ \boldsymbol{c}_2 = 0 \,, \qquad \boldsymbol{c}_2^\mathsf{T} \ \boldsymbol{c}_3 = 0 \,, \qquad \boldsymbol{c}_3^\mathsf{T} \ \boldsymbol{c}_1 = 0 \,. \tag{8.9}$$

Alternatively, there are similar constraints on its rows $\boldsymbol{r}_i$. Its elements, rows, and columns can easily be interpreted. We show this for the task of rotating a coordinate system.

1. The column vectors $\boldsymbol{c}_i$ of the rotation matrix are the images of the basic vectors $\boldsymbol{e}_i \doteq \boldsymbol{e}_i^{[3]}$ of the coordinate system to be rotated, as

$$\boldsymbol{c}_i = R\boldsymbol{e}_i \,. \tag{8.10}$$

2. The row vectors $\boldsymbol{r}_i$ of the rotation matrix are the pre-images of the basic vectors, since $\boldsymbol{e}_i = R\boldsymbol{r}_i$; hence

$$\boldsymbol{r}_i = R^\mathsf{T} \boldsymbol{e}_i \,. \tag{8.11}$$

3. The individual elements of the matrix can be interpreted as the cosines of the angles between the basic unit vectors before and after the rotation, $\boldsymbol{e}_i$ and $\boldsymbol{c}_j$, respectively,

$$R = [r_{ij}] = \begin{bmatrix} \boldsymbol{e}_1^\mathsf{T} \\ \boldsymbol{e}_2^\mathsf{T} \\ \boldsymbol{e}_3^\mathsf{T} \end{bmatrix} [\boldsymbol{c}_1 \ \boldsymbol{c}_2 \ \boldsymbol{c}_3] = \begin{bmatrix} \boldsymbol{e}_1 \cdot \boldsymbol{c}_1 & \boldsymbol{e}_1 \cdot \boldsymbol{c}_2 & \boldsymbol{e}_1 \cdot \boldsymbol{c}_3 \\ \boldsymbol{e}_2 \cdot \boldsymbol{c}_1 & \boldsymbol{e}_2 \cdot \boldsymbol{c}_2 & \boldsymbol{e}_2 \cdot \boldsymbol{c}_3 \\ \boldsymbol{e}_3 \cdot \boldsymbol{c}_1 & \boldsymbol{e}_3 \cdot \boldsymbol{c}_2 & \boldsymbol{e}_3 \cdot \boldsymbol{c}_3 \end{bmatrix} . \tag{8.12}$$

Sometimes this representation is called *direction cosine matrix* (cf. Klumpp, 1976).

8.1.3 Rotations with Euler Angles

Representing a rotation in $\mathbb{R}^3$ is easiest using rotation angles. We need three angles for a complete rotation in 3D, one within each of the coordinate planes, which in 3D is identical to rotating around the coordinate axes. An arbitrary rotation may be realized as a concatenation of three elementary rotations. We again assume that rotations describe motions of a point, an object, or a frame in a fixed reference frame.

8.1.3.1 Elementary Rotations

We denote rotations of an object around an axis $\boldsymbol{r}$ by an angle θ as rotation matrix $R_r(\theta)$. Hence, we obtain the three elementary rotations (see Fig. 8.1):

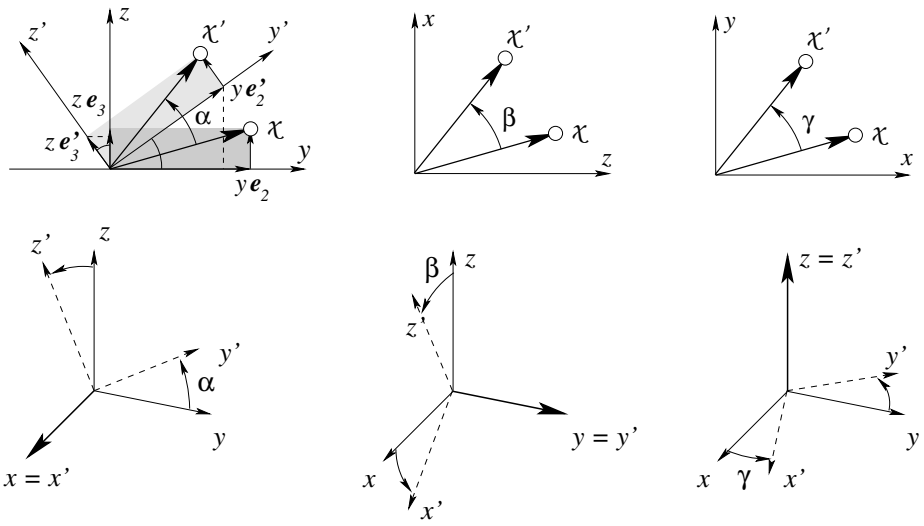


Fig. 8.1 Elementary rotations in the three coordinate planes. **Top row:** rotations of point χ seen along the rotation axes. **Left:** rotation around the x-axis. Seen along the x-axis the angle α appears between $\boldsymbol{x}$ and $\boldsymbol{x}'$, between the old y-axis and the new y' axis $\boldsymbol{e}_2$ and $\boldsymbol{e}_2'$, respectively, and between the old z-axis and the new z'-axis $\boldsymbol{e}_3$ and $\boldsymbol{e}_3'$. **Middle:** rotation around the y-axis. **Right:** rotation around the z-axis. The third axis always points towards the reader. All rotations follow the right-hand rule. **Second row:** oblique view of the three basic rotations applied to the coordinate axes

1. Rotation with α around the 1-axis (x-axis). The x-coordinate remains unchanged. Seen along the x-axis the point χ is rotated towards the z-axis by the angle α leading to χ'. The angle α also exists between the vector pairs $(y\boldsymbol{e}_2, y\boldsymbol{e}_2')$ and $(z\boldsymbol{e}_3, z\boldsymbol{e}_3')$. Thus we have

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y\cos\alpha - z\sin\alpha \\ y\sin\alpha + z\cos\alpha \end{bmatrix} \quad \text{or} \quad R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}. \tag{8.13}$$

This can be seen from Fig. 8.1, left: The coordinates of point χ are $\boldsymbol{x} = x\boldsymbol{e}_1 + y\boldsymbol{e}_2 + z\boldsymbol{e}_3$. After rotation, we obtain $\boldsymbol{x}' = x\boldsymbol{e}_1' + y\boldsymbol{e}_2' + z\boldsymbol{e}_3'$, with $\boldsymbol{e}_1' = \boldsymbol{e}_1$ and

$$\boldsymbol{e}_2' = \begin{bmatrix} 1 \\ \cos\alpha \\ \sin\alpha \end{bmatrix} \qquad \boldsymbol{e}_3' = \begin{bmatrix} 1 \\ -\sin\alpha \\ \cos\alpha \end{bmatrix}. \tag{8.14}$$

2. Rotation with β around the 2-axis (y-axis): The y-axis is kept. Seen along the y-axis we turn the point towards the x-axis by the angle β, see Fig. 8.1, centre,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x\cos\beta + z\sin\beta \\ y \\ -x\sin\beta + z\cos\beta \end{bmatrix} \quad \text{or} \quad R_2(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}. \tag{8.15}$$

   Observe, the sign of $\sin\beta$ is negative in the left lower part, as for small rotations the z-coordinate of a point in the first quadrant becomes smaller.
3. Rotation with γ around the 3-axis (z-axis). We keep the z-axis. Seen along the z-axis we turn the point towards the y-axis by the angle γ, see Fig. 8.1, right,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x\cos\gamma - y\sin\gamma \\ x\sin\gamma + y\cos\gamma \\ z \end{bmatrix} \quad \text{or} \quad R_3(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8.16}$$

8.1.3.2 Concatenated Elementary Rotations

We now apply the concatenation rules, cf. Sect. 6.3.2.2, p. 262, to obtain general rotations from the elementary rotations $R_1(\alpha)$, $R_2(\beta)$, and $R_3(\gamma)$, shown in Fig. 8.1. We obtain the four cases:

A Rotation of the object with its frame: $\boldsymbol{x}' = R_A\boldsymbol{x}$. The second and third rotations are rotations around the axes of the fixed reference system. Multiplication with the elementary rotation matrices from the left yields:

$$R_A(\alpha,\beta,\gamma) = R_3(\gamma)R_2(\beta)R_1(\alpha) \tag{8.17}$$

or, explicitly,

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{8.18}$$

B Rotation of the object with its frame: $\boldsymbol{x}' = R_B\boldsymbol{x}$. The second and third rotations have to be done around the rotated axes of the object frame. Multiplication of the elementary rotation matrices from the right yields

$$R_B(\alpha,\beta,\gamma) = R_1(\alpha)R_2(\beta)R_3(\gamma) \tag{8.19}$$

or, explicitly,

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\beta & \sin\beta \\ \cos\gamma\sin\beta\sin\alpha + \sin\gamma\cos\alpha & -\sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & -\cos\beta\sin\alpha \\ -\cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha & \sin\gamma\sin\beta\cos\alpha + \cos\gamma\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{8.20}$$

C The object is fixed. The reference system is rotated and the object is described in the rotated reference system: ${}^2\boldsymbol{x} = R_C\,{}^1\boldsymbol{x}$. The second and third rotations are around the original fixed axes of the reference frame. Multiplication with the transposed elementary rotation matrices from the right yields

$$R_C(\alpha,\beta,\gamma) = R_1^{\mathsf{T}}(\alpha)R_2^{\mathsf{T}}(\beta)R_3^{\mathsf{T}}(\gamma) = R_A^{\mathsf{T}}. \tag{8.21}$$

This is just the inverse, thus the transposed rotation matrix of case A.

D The object is fixed. The reference frame is rotating and the object is described in the rotated reference frame: ${}^2\boldsymbol{x} = \boldsymbol{R}_D\ {}^1\boldsymbol{x}$. The second and third rotations are around the rotated axes. Multiplication of the transposed elementary rotation matrices from the left yields

$$\boldsymbol{R}_D(\alpha,\beta,\gamma) = \boldsymbol{R}_3^\mathsf{T}(\gamma)\boldsymbol{R}_2^\mathsf{T}(\beta)\boldsymbol{R}_1^\mathsf{T}(\alpha) = \boldsymbol{R}_B^\mathsf{T}. \tag{8.22}$$

This is the inverse, thus transposed rotation matrix of case B.

As can be observed in all cases A to D, rotation matrices with three angles always have one element which is a single trigonometric term, while the corresponding column and row contain products of two trigonometric terms. This makes it simple to determine rotation angles from a rotation matrix. The other elements are sums of products of trigonometric terms.

8.1.3.3 Determination of Rotation Angles from a Rotation Matrix

Given a rotation matrix, the rotation angles can be derived. This presumes:

1. The sequence and type of the elementary rotations are known.
2. We do not have a singular case, where determination is not possible.

In all cases, the rotation matrix has one term which depends only on one of the three angles. The corresponding row and column may be used to determine the remaining two angles. For two numerical reasons, it is of advantage to use the arctan function with two variables, cf. Sect. 7.1.3, p. 298: (1) The angle will be determined correctly in the interval $[0, 2\pi)$ or $[-\pi, \pi)$, and (2) the precision of the angle is higher in contrast to the arccos function, which for small angles yields inaccurate results.

For example, consider the rotation matrix in (8.18). Generally, we obtain

$$\alpha = \operatorname{atan2}(R_{32}, R_{33}), \tag{8.23}$$

$$\beta = \operatorname{atan2}(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}), \tag{8.24}$$

$$\gamma = \operatorname{atan2}(R_{21}, R_{11}). \tag{8.25}$$

Two remarks are necessary:

1. This calculation fails if $\cos\beta = 0$ or $\beta = \pm 90^o$. These are singular cases since γ and α are not definite. Then the rotation matrix takes the form

$$\begin{bmatrix} 0 & -\sin\gamma\cos\alpha + \cos\gamma\sin\alpha & \sin\gamma\sin\alpha + \cos\gamma\cos\alpha \\ 0 & \sin\gamma\sin\alpha + \cos\gamma\cos\alpha & -\cos\gamma\sin\alpha + \sin\gamma\cos\alpha \\ -1 & 0 & 0 \end{bmatrix} \tag{8.26}$$

or

$$\begin{bmatrix} 0 & -\sin(\gamma-\alpha) & \cos(\gamma-\alpha) \\ 0 & \cos(\gamma-\alpha) & \sin(\gamma-\alpha) \\ -1 & 0 & 0 \end{bmatrix} \tag{8.27}$$

singularity if $\beta = \pi/2$

and only depends on $\gamma - \alpha$. Thus β can be determined, but neither of the two other angles, only their difference. To visualize this situation: The first rotation $\boldsymbol{R}_1(\alpha)$ and the third rotation $\boldsymbol{R}_3(\gamma)$ are around the same object centred axis, whereas the rotation around the z-axis is opposite to the rotation around the x-axis. Any representation with three angles shows this type of singularity. This is why we later discuss representations with four parameters which are free from this problem.

2. We may wonder why the sign of $\cos\beta$ does not influence the determination of α. Actually it does. But the resulting rotation matrix is invariant to a change of the sign of $\cos\beta$. The reason is an ambiguity in the solution since

$$R_3(\gamma)R_2(\beta)R_1(\alpha) = R_3(\gamma+\pi)R_2(\pi-\beta)R_1(\alpha+\pi)\,. \tag{8.28}$$

Therefore the sign of $\cos\beta$ can be chosen to be positive – the reason we used the positive square root in (8.24) – leading to one of the two solutions. The user may choose one of the two solutions, depending on his knowledge about the range of the angles.

two solutions: (α,β,γ) *and* $(\alpha+\pi,\pi-\beta,\gamma+\pi)$

8.1.4 Rotation with Rotation Axis and Rotation Angle

8.1.4.1 Rotation Matrix from a Given Rotation Axis and Rotation Angle

If the rotation axis is represented by a normalized vector $\boldsymbol{r} = [r_1, r_2, r_3]^\mathsf{T}$ with $|\boldsymbol{r}| = 1$ and a rotation angle θ, the rotation matrix is given by

$$R_{r,\theta} = I_3 + \sin\theta\, S_r + (1-\cos\theta) S_r^2\,, \tag{8.29}$$

with the skew symmetric matrix $S_r = S(\boldsymbol{r})$. If both axis and angle are inverted, the rotation matrix does not change: $R(\boldsymbol{r},\theta) = R(-\boldsymbol{r},-\theta)$. Observe, this is identical to the exponential of the skew symmetric matrix $S(\theta\boldsymbol{r})$ in (8.4), p. 327.

Proof: The rotation matrix (8.29) can also be written as, cf. (A.43):

$$R_{r,\theta} = \cos\theta\, I_3 + (1-\cos\theta)\, D_r + \sin\theta\, S_r\,. \tag{8.30}$$

We show (1) $\boldsymbol{r} = R_{r,\,\theta}\boldsymbol{r}$, (2) $R_{r,\,\theta}$ is a rotation matrix and (3) the rotation angle is θ. (1) and (2) can be easily verified. We only prove (3).

An arbitrary point $Q(\boldsymbol{q})$ (Fig. 8.2) having distance $|\boldsymbol{o}|$ from the rotation axis may be written as $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{o}$, where $\boldsymbol{o} \perp \boldsymbol{r}$ and $\boldsymbol{p}$ are the components orthogonal and parallel to the rotation axis. We now decompose

$$\boldsymbol{q}' = R\boldsymbol{q} = \cos(\theta)\,(\boldsymbol{p}+\boldsymbol{o}) + (1-\cos\theta)\,\boldsymbol{r}\boldsymbol{r}^\mathsf{T}(\boldsymbol{p}+\boldsymbol{o}) + \sin(\theta)\,\boldsymbol{r}\times(\boldsymbol{p}+\boldsymbol{o})$$

and get:

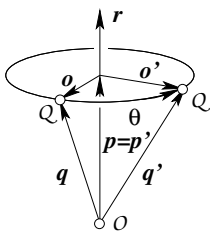


Fig. 8.2 Rotation of $\boldsymbol{q}$ around the axis $\boldsymbol{r}$ with the angle θ yields $\boldsymbol{q}'$

$$\begin{aligned}\boldsymbol{q}' &= \cos(\theta)\,(\lambda\boldsymbol{r}+\boldsymbol{o}) + \lambda(1-\cos\theta)\boldsymbol{r} + \sin(\theta)\,S_r\boldsymbol{o}\\ &= \underbrace{\lambda\boldsymbol{r}}_{\boldsymbol{p}'} + \underbrace{(\cos(\theta)\,\boldsymbol{o} + \sin(\theta)\,(\boldsymbol{r}\times\boldsymbol{o}))}_{\boldsymbol{o}'}\end{aligned}$$

since $\boldsymbol{r}\times\boldsymbol{p} = \boldsymbol{0}$, $\boldsymbol{r}^\mathsf{T}\boldsymbol{o} = 0$, $\boldsymbol{r}^\mathsf{T}\boldsymbol{p} = \boldsymbol{r}^\mathsf{T}\lambda\boldsymbol{r} = \lambda$. Hence, the rotated vector $\boldsymbol{q}'$ has the component $\boldsymbol{p}' = \boldsymbol{p} \,||\, \boldsymbol{r}$ parallel to $\boldsymbol{r}$ and the component $\boldsymbol{o}' = \sin\theta\,(\boldsymbol{r}\times\boldsymbol{o}) + \cos\theta\,\boldsymbol{o}$ perpendicular to $\boldsymbol{r}$. As $\boldsymbol{o}\cdot\boldsymbol{o}' = \cos\theta\,\boldsymbol{o}\cdot\boldsymbol{o}$ and because of the length preservation property $|\boldsymbol{o}| = |\boldsymbol{o}'|$, we obtain $\cos(\boldsymbol{o},\boldsymbol{o}') = \cos\theta$, and thus a rotation around $\boldsymbol{r}$ with the angle θ. ◇

8.1.4.2 Rotation Axis and Rotation Angle from a Given Rotation Matrix

Given a rotation matrix $R = (r_{ij})$, we may determine the rotation angle and the rotation axis in the following way:

1. Rotation angle. The angle can be determined from the trace,

$$\mathrm{tr}R = r_{11} + r_{22} + r_{33} = 1 + 2\cos\theta\,, \tag{8.31}$$

and the length of the vector $\boldsymbol{a}$ from the skew symmetric part,

$$S_a = 2\sin\theta\, S_r = R - R^{\mathsf{T}}\,, \tag{8.32}$$

of the rotation matrix

$$\boldsymbol{a} = -\begin{pmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{pmatrix} = 2\,\boldsymbol{r}\,\sin\theta\,. \tag{8.33}$$

This yields

$$\theta = \mathrm{atan2}(|\boldsymbol{a}|, \mathrm{tr}R - 1), \tag{8.34}$$

as $|\boldsymbol{a}| = 2\sin\theta$ and $\mathrm{tr}R - 1 = 2\cos\theta$.

2. Rotation axis. Here we need to distinguish between three cases:

   a. If $\theta = 0$, we have a null rotation, i.e., $R = I_3$, and the axis cannot be determined.
   b. If $\theta = \pi = 180°$, then $\sin\theta = 0$ and the rotation matrix is symmetric and has the form $R = -I_3 + 2D_r$. Therefore, we may determine the rotation axis from one of the three normalized columns of

$$2D_r = R + I_3 = 2\boldsymbol{r}\,\boldsymbol{r}^{\mathsf{T}}\,,$$

   since D_r has rank 1 and all columns are proportional to $\boldsymbol{r}$. For numerical reasons we choose the column with largest absolute value. The sign is irrelevant, as we rotate by 180°.
   c. In all other cases, $0 < \theta < \pi$ holds, and therefore $|\boldsymbol{a}| > 0$. We may then determine the rotation axis from $\boldsymbol{a}$ by normalization,

$$\boldsymbol{r} = \frac{\boldsymbol{a}}{|\boldsymbol{a}|}\,. \tag{8.35}$$

   Observe, (8.33) does not allow us to derive the sign of the rotation axis uniquely, since $-\boldsymbol{r}$ and $-\theta$ would yield the same $\boldsymbol{a}$. We fixed the sign of $\sin\theta \geq 0$, and took the sign of the rotation axes from the vector $\boldsymbol{a}$.

Of course the tests $\theta = 0$ and $\theta = \pi$ if 2a and 2b need to be replaced by $|\theta| < t_\theta$ and $|\theta - \pi| < t_\theta$, where the tolerance t_θ depends on the numerical accuracy of the computation.

Example 8.1.34: ***Rotation matrices.*** The matrix

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{8.36}$$

leads to $\boldsymbol{a}^{\mathsf{T}} = [-1, -1, -1]$ and therefore to $\theta = \mathrm{atan2}\,(\sqrt{3}, -1) = +120°$ and thus $\boldsymbol{r}^{\mathsf{T}} = -\sqrt{3}/3\ [1, 1, 1]$. The rotation matrix $\mathrm{Diag}([1, -1, -1])$ leads to $|\boldsymbol{a}| = 0$ and $R + I_3 = \mathrm{Diag}([2, 0, 0])$. Therefore it rotates around the x-axis by 180°. ◇

8.1.5 Rotations with Quaternions

The representation by an axis and an angle is visually intuitive, but it requires trigonometric functions similarly to the representation by Euler angles. If we want to represent the rotation matrix without trigonometric functions and do not require a direct interpretation

of the parameters then a representation with polynomials only up to second degree can be chosen. The rotation matrix then depends on four parameters which are collected in a 4-vector, called a *quaternion*. Since scaling a quaternion with a scalar $\neq 0$ does not change the rotation, quaternions are homogeneous 4-vectors when used for representing rotations.

The representation of rotations by quaternions is the only one with four parameters which is unique, except for the scaling, and does not show singularities during estimation of parameters.[1] This results from the fact that the set of normalized quaternions, i.e., the 3-sphere $S^3 \in \mathbb{R}^4$, has no border, similarly to the set of normalized vectors on the circle in 2D or on the sphere in 3D (cf. Stuelpnagel, 1964).

8.1.5.1 Quaternions

Quaternions $\mathbf{q}$ build an algebra comparable to that of complex numbers. They are written in small upright boldface letters, as they are homogeneous 4-vectors. We represent quaternions in two ways:

1. Representation as 4-vector:
$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} . \tag{8.37}$$
As with homogeneous coordinates, we distinguish between two parts: the scalar part $q = q_0$ and the vector part $\boldsymbol{q} = [q_1, q_2, q_3]^\mathsf{T}$. If we treat a quaternion as a vector, the first element, the scalar part, is denoted by q_0; if we treat the quaternion as an algebraic entity, the scalar part is denoted by q. This representation allows us to embed quaternions into the framework of linear algebra, especially when representing their uncertainty using the covariance matrix of the vector $\mathbf{q}$.
2. Representation as ordered pair
$$\mathbf{q} = (q, \boldsymbol{q}), \tag{8.38}$$
which directly provides a link to vector algebra, as we will see immediately.

We thus interpret a quaternion as a 4-vector in linear algebra or as a basic element of an algebra of its own.[2]

Quaternions $\mathbf{q}$ and $\mathbf{r}$ are added by elementwise addition:

$$\mathbf{p} = \mathbf{q} + \mathbf{r} \qquad \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} q_0 + r_0 \\ q_1 + r_1 \\ q_2 + r_2 \\ q_3 + r_3 \end{bmatrix} \qquad (p, \boldsymbol{p}) = (q + r, \boldsymbol{q} + \boldsymbol{r}) . \tag{8.39}$$

The multiplication
$$\mathbf{p} = \mathbf{q}\mathbf{r} \tag{8.40}$$
of two quaternions $\mathbf{q}$ and $\mathbf{r}$ is defined in the following manner, using the partitioning into a scalar and a vector part:

quaternion multiplication

$$(p, \boldsymbol{p}) = (qr - \boldsymbol{q}\cdot\boldsymbol{r}, r\boldsymbol{q} + q\boldsymbol{r} + \boldsymbol{q} \times \boldsymbol{r}) . \tag{8.41}$$

If $\mathbf{q} = (0, \boldsymbol{q})$ and $\mathbf{r} = (0, \boldsymbol{r})$, i.e., if the scalar part is zero, we obtain

quaternion fuse scalar and vector product

$$(p, \boldsymbol{p}) = (-\boldsymbol{q}\cdot\boldsymbol{r}, \boldsymbol{q} \times \boldsymbol{r}) . \tag{8.42}$$

[1] The *Cayley–Klein-parameters*, which can also be used for representing a rotation with four parameters, are directly related to the quaternions.

[2] Quaternions also can be represented as hyper-complex numbers with three distinct imaginary parts, $\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3$, with multiplication rules $i^2 = j^2 = k^2 = ijk = -1$, found by W. R. Hamilton, (cf. Lam, 2002).

Hence, the multiplication of quaternions integrates the scalar product and the cross product, which was the basic motivation for William R. Hamilton (1805-1865) to invent the quaternions.

If we use quaternions as vectors, we obtain the product $\mathbf{p} = \mathbf{qr}$ or

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} q_0 r_0 - q_1 r_1 - q_2 r_2 - q_3 r_3 \\ q_1 r_0 + q_0 r_1 - q_3 r_2 + q_2 r_3 \\ q_2 r_0 + q_3 r_1 + q_0 r_2 - q_1 r_3 \\ q_3 r_0 - q_2 r_1 + q_1 r_2 + q_0 r_3 \end{bmatrix}, \tag{8.43}$$

which is bilinear in the two 4-vectors. We may thus write the quaternion multiplication as a matrix vector product,

$$\mathbf{p} = \mathsf{M}_q \mathbf{r} = \overline{\mathsf{M}}_r \mathbf{q}\,, \tag{8.44}$$

with the 4×4 matrices M_q and $\overline{\mathsf{M}}_r$,

$$\mathsf{M}_q = \begin{bmatrix} q_0 & -\boldsymbol{q}^\mathsf{T} \\ \boldsymbol{q} & q_0 \mathsf{I}_3 + \mathsf{S}(\boldsymbol{q}) \end{bmatrix} = \left[\begin{array}{c|ccc} q_0 & -q_1 & -q_2 & -q_3 \\ \hline q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{array}\right] \tag{8.45}$$

and

$$\overline{\mathsf{M}}_r = \begin{bmatrix} r_0 & -\boldsymbol{r}^\mathsf{T} \\ \boldsymbol{r} & r_0 \mathsf{I}_3 - \mathsf{S}(\boldsymbol{r}) \end{bmatrix} = \left[\begin{array}{c|ccc} r_0 & -r_1 & -r_2 & -r_3 \\ \hline r_1 & r_0 & r_3 & -r_2 \\ r_2 & -r_3 & r_0 & r_1 \\ r_3 & r_2 & -r_1 & r_0 \end{array}\right], \tag{8.46}$$

depending on the 4-vectors $\mathbf{q}$ and $\mathbf{r}$, respectively. These are at the same time the Jacobians of the product w.r.t. the two factors needed for variance propagation.

Exercise 8.25

Quaternion multiplication is not commutative, due to the integrated cross product. Howell and Lafon (1975) offer a fast algorithm for quaternion multiplication, which only needs eight normal multiplications, one division by 2, and 27 additions.

inverse quaternion

The inverse element of $\mathbf{q}$ w.r.t. multiplication is

$$\mathbf{q}^{-1} = \mathbf{q}^* / |\mathbf{q}|^2\,. \tag{8.47}$$

Here we have the *conjugate* quaternion (analogously to the conjugate complex number),

$$\mathbf{q}^* = \begin{bmatrix} q \\ -\boldsymbol{q} \end{bmatrix} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix}, \tag{8.48}$$

and the quadratic norm

$$|\mathbf{q}|^2 = q^2 + \boldsymbol{q}\cdot\boldsymbol{q} = q_0^2 + q_1^2 + q_2^2 + q_3^2\,. \tag{8.49}$$

Then we have multiplication matrices

$$\mathsf{M}_{q^{-1}} = \mathsf{M}_q^{-1} \qquad \text{and} \qquad \mathsf{M}_{q^*} = \mathsf{M}_q^\mathsf{T}. \tag{8.50}$$

This can easily be verified.

unit quaternion

Unit quaternions $\mathbf{e}$ have norm $|\mathbf{e}| = 1$. Then

$$\mathsf{M}_{e^{-1}} = \mathsf{M}_e^{-1} = \mathsf{M}_e^\mathsf{T} \tag{8.51}$$

is a rotation matrix in $SO(4)$, which means that unit quaternions play the same role in $\mathbb{R}^4$ as direction vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$. Moreover, rotations in $\mathbb{R}^2$ may also be represented by normalized direction vectors $\boldsymbol{e} = [a, b]^\mathsf{T}$ with $|\boldsymbol{e}| = 1$:

$$R(\boldsymbol{e}) = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} .$$

8.1.5.2 Rotation Matrices Based on Quaternions

Rotations can be represented by quaternions. If we multiply a quaternion $\mathbf{p} = [p, \boldsymbol{p}^\mathsf{T}]^\mathsf{T}$ with an arbitrary quaternion $\mathbf{q} \neq \mathbf{0}$ and its inverse $\mathbf{q}^{-1}$,

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1} , \tag{8.52}$$

the vector part of $\mathbf{p}$ is rotated. We obtain

$$p' = p \quad \text{and} \quad \boldsymbol{p}' = \frac{1}{|\mathbf{q}|^2} \left((q^2 - \boldsymbol{q}^\mathsf{T}\boldsymbol{q})\, I_3 + 2\, D_q + 2\, q\, S_q \right) \boldsymbol{p} \tag{8.53}$$

where the dyad $D_q := D(\boldsymbol{q})$ and the skew matrix $S_q := S(\boldsymbol{q})$ depend only on the vector part of the quaternion. The matrix

$$R_Q = \frac{1}{|\mathbf{q}|^2} \left((q^2 - \boldsymbol{q}^\mathsf{T}\boldsymbol{q})\, I_3 + 2\, D_q + 2\, q\, S_q \right) \tag{8.54}$$

in (8.53) is a rotation matrix. Thus, if we want to rotate a point with inhomogeneous coordinates, $\boldsymbol{x} = [x_1, x_2, x_3]^\mathsf{T}$, we form the quaternion $\mathbf{p} = (0, \boldsymbol{x})^\mathsf{T}$ and apply (8.52).

Explicitly, we may write (8.54) as

$$R_Q = \frac{1}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\,(q_1 q_2 - q_0 q_3) & 2\,(q_1 q_3 + q_0 q_2) \\ 2\,(q_2 q_1 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\,(q_2 q_3 - q_0 q_1) \\ 2\,(q_3 q_1 - q_0 q_2) & 2\,(q_3 q_2 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} . \tag{8.55}$$

All elements are rational and purely quadratic in the coefficients q_i. We now use a unit quaternion,

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \boldsymbol{r}\, \sin\frac{\theta}{2} \end{bmatrix} = \cos\frac{\theta}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \sin\frac{\theta}{2} \begin{bmatrix} 0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} , \tag{8.56}$$

where $\boldsymbol{r}$ is a unit vector. Then we obtain from (8.54) the axis-angle representation for a rotation matrix (8.29). From (8.53), therefore, after some simple arrangements, we get the matrix: Exercise 8.11

$$R_{\overline{Q}} = I_3 + 2(q^2\, I_3 + q\, S_q + S_q^2) = \begin{bmatrix} 1 - 2\,(q_2^2 + q_3^2) & 2\,(q_1 q_2 - q_0 q_3) & 2\,(q_1 q_3 + q_0 q_2) \\ 2\,(q_2 q_1 + q_0 q_3) & 1 - 2\,(q_1^2 + q_3^2) & 2\,(q_2 q_3 - q_0 q_1) \\ 2\,(q_3 q_1 - q_0 q_2) & 2\,(q_3 q_2 + q_0 q_1) & 1 - 2\,(q_1^2 + q_2^2) \end{bmatrix} . \tag{8.57}$$

Example 8.1.35: ***Rotation with quaternions.*** (1) The rotation matrix (8.36) results from $\mathbf{q} = \left[-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]^\mathsf{T}$. (2) The rotation matrix $\mathrm{Diag}([1, -1, -1])$ results from $\mathbf{q} = [0, 1, 0, 0]^\mathsf{T}$. ◇

8.1.5.3 Rodriguez Representation

In aerial photogrammetry, normalized quaternions have frequently been used yielding a 3-parameter representation following Rodriguez (1840). For this representation, we have the quaternion (cf. Mikhail et al., 2001)

$$\mathbf{q} = \left[1, \frac{a}{2}, \frac{b}{2}, \frac{c}{2}\right]^\mathsf{T} = \left[1, \frac{1}{2}\boldsymbol{m}^\mathsf{T}\right]^\mathsf{T}, \tag{8.58}$$

where the scalar part is normalized to 1, similarly to the Euclidean normalization of homogeneous vectors. With the parameter vector $\boldsymbol{m} = [a, b, c]^\mathsf{T}$, we may write the Rodriguez matrix as

$$R_R(\boldsymbol{m}) = \frac{1}{4 + |\boldsymbol{m}|^2}\left((4 - \boldsymbol{m}^\mathsf{T}\boldsymbol{m})\, I_3 + 2D_m + 4S_m\right) \tag{8.59}$$

(cf. (8.54)) or, explicitly, as in (8.60),

$$R_R(a, b, c) = \frac{1}{4 + a^2 + b^2 + c^2}\begin{bmatrix} 4 + a^2 - b^2 - c^2 & 2ab - 4c & 2ac + 4b \\ 2ab + 4c & 4 - a^2 + b^2 - c^2 & 2bc - 4a \\ 2ac - 4b & 2bc + 4a & 4 - a^2 - b^2 + c^2 \end{bmatrix}. \tag{8.60}$$

It follows from (8.56) that a rotation with quaternion (8.58) is equivalent to the rotation with the quaternion

$$\mathbf{q} = \left(1, \boldsymbol{r}\tan\frac{\theta}{2}\right). \tag{8.61}$$

It is obvious that this quaternion cannot represent rotations of 180°. Given $\boldsymbol{m}$, we can easily derive the rotation axis from $\boldsymbol{r} = \frac{\boldsymbol{m}}{|\boldsymbol{m}|}$ and the rotation angle from $\theta = 2\arctan(\frac{1}{2}|\boldsymbol{m}|)$. For small angles we have $\theta \approx |\boldsymbol{m}|$ and $\boldsymbol{m} \approx [\alpha, \beta, \gamma]^\mathsf{T}$, see below.

8.1.5.4 Cayley Representation

There is a close relation between the Rodriguez representation and a rational representation with a skew symmetric matrix: we represent the rotation with the vector $\boldsymbol{u} = (\frac{a}{2}, \frac{b}{2}, \frac{c}{2}) = \frac{1}{2}\boldsymbol{m}$. Using (8.58), we can show $I_3 + S_u = R_R(2\boldsymbol{u})\,(I_3 - S_u)$ or $I_3 + S_u = (I_3 - S_u)\,R_R(2\boldsymbol{u})$. Therefore, we have the following *Cayley representation*, $R_C(\boldsymbol{u}) = R_R(2\boldsymbol{u})$, of a rotation matrix, proposed by A. Cayley:

$$R_C(\boldsymbol{u}) = (I_3 + S_u)(I_3 - S_u)^{-1} = (I_3 - S_u)^{-1}(I_3 + S_u)\,; \tag{8.62}$$

(cf. Fallat and Tsatsomeros, 2002). This representation is not suited for angles equal to or close to 180° either. We have the inverse relations

$$S_u = (I_3 + R_C(\boldsymbol{u}))(I_3 - R_C(\boldsymbol{u}))^{-1} = (I_3 - R_C(\boldsymbol{u}))(I_3 + R_C(\boldsymbol{u}))^{-1}\,. \tag{8.63}$$

Generally, the transformation $B = (I + A)(I - A)^{-1}$ of a matrix A following (8.62) is called the Cayley transformation of the matrix A. For skew symmetric matrices it yields rotation matrices.

The Cayley representation for rotations (8.62) is valid in all space dimensions.

8.1.6 Differential Rotations

Small rotations occur when estimating rotations by iterations or when modelling rotational motions.

When estimating rotations, we usually start with some approximate rotation $R^a = R(\boldsymbol{\theta}^a)$, say, depending on an approximate*rotation vector* $\boldsymbol{\theta}^a$, and – following the idea of multiplicative concatenation of transformations – represent the unknown rotation $R(\boldsymbol{\theta})$ by the product of the approximate rotation with a differential rotation $R(\mathrm{d}\boldsymbol{\theta})$,

$$R(\boldsymbol{\theta}) = R(\mathrm{d}\boldsymbol{\theta})R(\boldsymbol{\theta}^a)\,, \tag{8.64}$$

with the differential rotation[3]

$$R(\mathrm{d}\boldsymbol{\theta}) = \begin{bmatrix} 1 & -\mathrm{d}\theta_3 & \mathrm{d}\theta_2 \\ \mathrm{d}\theta_3 & 1 & -\mathrm{d}\theta_1 \\ -\mathrm{d}\theta_2 & \mathrm{d}\theta_1 & 1 \end{bmatrix} = I_3 + S(\mathrm{d}\boldsymbol{\theta})\,. \tag{8.65}$$

When modelling rotational motions, we may express rotational velocity by $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$, by $R(\boldsymbol{\omega})$, or explicitly by $R(\boldsymbol{\omega}) = I_3 + S(\boldsymbol{\omega})$, leading to $\boldsymbol{x} + \dot{\boldsymbol{x}}\Delta t = (I_3 + S(\dot{\boldsymbol{\theta}}\Delta t))\boldsymbol{x}$. This yields the classical differential equation for a rotational motion,

$$\dot{\boldsymbol{x}} = S(\dot{\boldsymbol{\theta}})\boldsymbol{x} = \dot{\boldsymbol{\theta}} \times \boldsymbol{x} = \boldsymbol{\omega} \times \boldsymbol{x}\,. \tag{8.66}$$

We now compare the differential rotation matrices for the various representations. We obtain the following identities for the *differential rotation vector*:

$$\mathrm{d}\boldsymbol{\theta} \overset{(8.4)}{=} \begin{bmatrix} \mathrm{d}\theta_1 \\ \mathrm{d}\theta_2 \\ \mathrm{d}\theta_3 \end{bmatrix} \overset{(8.12)}{=} \begin{bmatrix} \mathrm{d}r_{32} \\ \mathrm{d}r_{13} \\ \mathrm{d}r_{21} \end{bmatrix} \overset{(8.18)}{=} \begin{bmatrix} \mathrm{d}\alpha \\ \mathrm{d}\beta \\ \mathrm{d}\gamma \end{bmatrix} \overset{(8.29)}{=} \mathrm{d}\theta \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{8.67}$$

$$\overset{(8.60)}{=} \begin{bmatrix} \mathrm{d}a \\ \mathrm{d}b \\ \mathrm{d}c \end{bmatrix} \overset{(8.54)}{=} \frac{2}{q_0} \begin{bmatrix} \mathrm{d}q_1 \\ \mathrm{d}q_2 \\ \mathrm{d}q_3 \end{bmatrix} \overset{(8.57),|q|=1}{=} 2 \begin{bmatrix} \mathrm{d}q_1 \\ \mathrm{d}q_2 \\ \mathrm{d}q_3 \end{bmatrix} \overset{(8.62)}{=} 2 \begin{bmatrix} \mathrm{d}u_1 \\ \mathrm{d}u_2 \\ \mathrm{d}u_3 \end{bmatrix} \tag{8.68}$$

The first five differential vectors contain the differential rotation angles as elements. The last three differential vectors, related to the quaternion representation, contain the halved differential angles as elements.

Using differential rotations, we observe an evident relation between the skew symmetric matrix of a finite rotation vector $\boldsymbol{\theta}$ and the exponential form of the rotation:

$$R(\boldsymbol{\theta}) = \mathrm{e}^{S(\theta)} = \lim_{n\to\infty} \left(I_3 + \frac{S(\boldsymbol{\theta})}{n}\right)^n = \lim_{n\to\infty} \prod_{i=1}^{n} R\left(\frac{\boldsymbol{\theta}}{n}\right) \tag{8.69}$$

where we use the classical definition of the exponential function $\mathrm{e}^x = \lim_{n\to\infty}(1 + x/n)^n$, as for large n we have $I_3 + S(\boldsymbol{\theta})/n \approx R(\boldsymbol{\theta}/n)$.

8.2 Concatenation of Rotations

Due to the group structure of rotations, concatenation of two rotations R' and R'' leads to a new rotation, e.g., $R = R''R'$, so the concatenation of rotations using the matrix representations is straightforward.

However, there is no simple way of concatenating rotations on the level of rotation parameters when working with Euler angles; there exists no simple expression for the angles (α, β, γ) of the rotation $R(\alpha, \beta, \gamma) = R(\alpha'', \beta'', \gamma'')R(\alpha', \beta', \gamma')$ if the angles $(\alpha', \beta', \gamma')$ and $(\alpha'', \beta'', \gamma'')$ are given.

In contrast to Euler angles, all other representations have a simple concatenation rule with rotation parameters, which are all derived from the concatenation rule for quaternions.

Concatenation with Quaternions. The concatenation of two rotations using quaternions uses the quaternion product: Let the first rotation be $\mathbf{p}' = \mathbf{q}'\mathbf{p}\mathbf{q}'^{-1}$, then the second leads to $\mathbf{p}'' = \mathbf{q}''\mathbf{p}'\mathbf{q}''^{-1} = \mathbf{q}''\mathbf{q}'\mathbf{p}\mathbf{q}'^{-1}\mathbf{q}''^{-1} = (\mathbf{q}''\mathbf{q}')\,\mathbf{p}\,(\mathbf{q}''\mathbf{q}')^{-1}$. Hence, we directly have

$$R(\mathbf{q}) = R(\mathbf{q}'')\,R(\mathbf{q}')\,, \quad \text{and} \quad \mathbf{q} = \mathbf{q}''\mathbf{q}'\,. \tag{8.70}$$

[3] We also could have set $R(\boldsymbol{\theta}) = R(\boldsymbol{\theta}^a)R(\mathrm{d}\boldsymbol{\theta})$. This would only change the meaning of the differential rotation vector $\mathrm{d}\boldsymbol{\theta}$, not the general properties of the multiplicative scheme.

The product of the quaternions representing a rotation is the quaternion of the concatenated rotation. The concatenation rules for transformations transfer to quaternions.

Concatenation of the Rodriguez and the Cayley Parameters. Since the Rodriguez representation and the representation with skew symmetric matrices are special cases of the quaternion representation, they also allow us to directly concatenate the parameters.

Given two sets of parameters, $\boldsymbol{m}' = [a', b', c']^\mathsf{T}$ and $\boldsymbol{m}'' = [a'', b'', c'']^\mathsf{T}$, of the Rodriguez representation for rotations, we obtain with (8.70) the parameters $\boldsymbol{m} = [a, b, c]^\mathsf{T}$ of the concatenated rotation $R(\mathbf{m}) = R(\mathbf{m}'')\, R(\mathbf{m}')$,

$$\boldsymbol{m} = \frac{4(\boldsymbol{m}' + \boldsymbol{m}'') + 2\boldsymbol{m}'' \times \boldsymbol{m}'}{4 - \boldsymbol{m}' \cdot \boldsymbol{m}''} \,. \tag{8.71}$$

Rotations which are given with axis and angle can be concatenated with relationship (8.71) as well, since $[a, b, c]^\mathsf{T} = 2\boldsymbol{r} \tan(\alpha/2)$.

Analogously, we obtain the concatenation rule with the parameters of the Cayley representation with skew symmetric matrices:

$$\boldsymbol{u} = \frac{\boldsymbol{u}' + \boldsymbol{u}'' + \boldsymbol{u}'' \times \boldsymbol{u}'}{1 - \boldsymbol{u}' \cdot \boldsymbol{u}''} \,. \tag{8.72}$$

8.3 Relations Between the Representations for Rotations

We discussed seven different representations of rotations:

- matrix exponential of a skew symmetric matrix in Sect. 8.1.1, p. 326,
- the elements of a constrained 3×3 matrix R in Sect. 8.1.2, p. 327,
- Eulerian angles in Sect. 8.1.3, p. 328,
- axis and angle, $R_{r,\theta}$, in Sect. 8.1.4, p. 331,
- quaternions, $R_Q(\boldsymbol{q})$, in Sect. 8.1.5.2, p. 335,
- Rodriguez parameters, $R_R(\boldsymbol{m})$, in Sect. 8.1.5.3, p. 335,
- Cayley parameters, $R_C(\boldsymbol{u})$, in Sect. 8.1.5.4, p. 336.

The exponential and axis-angle representations use the same parameters, namely the rotation vector $\boldsymbol{\theta} = \theta\boldsymbol{r}$. The direct link to the Eulerian angles is only possible for differential angles. Therefore, Table 8.1 collects the relations between only four of the representations. In addition, we also collect relations to differential Eulerian angles. For small angles, the

Table 8.1 Relations between representations for 3D rotations. For differential angles the rotation vector is $\mathrm{d}\boldsymbol{\theta}$ and identical to the Rodriguez vector $\mathrm{d}\boldsymbol{m}$

	diff. angle $\mathrm{d}\boldsymbol{\theta} = \begin{bmatrix} \mathrm{d}\theta_1 \\ \mathrm{d}\theta_2 \\ \mathrm{d}\theta_3 \end{bmatrix}$	axis + angle $(\boldsymbol{r}, \theta)$	quaternion $\mathbf{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}$	Rodriguez matrix $\boldsymbol{m}$	Cayley matrix $\boldsymbol{u}$
Eq. for R		(8.29)	(8.54)	(8.60)	(8.62)
$(\boldsymbol{r}, \theta) =$	$\boldsymbol{r} = \mathsf{N}(\mathrm{d}\boldsymbol{\theta})$ $\theta = \lvert\mathrm{d}\boldsymbol{\theta}\rvert$	$(\boldsymbol{r}, \theta)$	$\boldsymbol{r} = \mathsf{N}(\boldsymbol{q})$, $\theta = 2\,\mathrm{atan2}(\lvert\boldsymbol{q}\rvert, q_0)$	$\boldsymbol{r} = \mathsf{N}(\boldsymbol{m})$, $\theta = 2\,\mathrm{atan}(\lvert\boldsymbol{m}\rvert/2)$	$\boldsymbol{r} = \mathsf{N}(\boldsymbol{u})$, $\theta = 2\,\mathrm{atan}(\lvert\boldsymbol{u}\rvert)$
$\mathbf{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} =$	$\begin{bmatrix} 1 \\ \frac{1}{2}\mathrm{d}\boldsymbol{\theta} \end{bmatrix}$	$\begin{bmatrix} \cos(\theta/2) \\ \boldsymbol{r}\sin(\theta/2) \end{bmatrix}$	$\mathbf{q}$	$\begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{m} \end{bmatrix}$	$\begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}$
$\boldsymbol{m} =$	$\mathrm{d}\boldsymbol{\theta}$	$2\,\boldsymbol{r}\tan(\theta/2)$	$2\,\boldsymbol{q}/q_0$	$\boldsymbol{m}$	$2\,\boldsymbol{u}$
$\boldsymbol{u} =$	$\frac{1}{2}\mathrm{d}\boldsymbol{\theta}$	$\boldsymbol{r}\tan(\theta/2)$	$\boldsymbol{q}/q_0$	$\frac{1}{2}\boldsymbol{m}$	$\boldsymbol{u}$

rotation vector $\mathrm{d}\boldsymbol{\theta}$ with first order approximation is proportional to the parameter vectors

of the other representations. As the vector $\boldsymbol{r}$ of the rotation axis is of unit length, we can derive it from the other representations by normalization. The two representations with four parameters, the axis-angle representation and the quaternion representation, are unique up to the sign. The nonredundant representations are restricted to rotations without 180°.

8.4 Rotations from Corresponding Vector Pairs

8.4.1 Rotation from Three Pairs of Orthonormal Vectors 339
8.4.2 Rotation from One, Two, and Three Pairs of Arbitrary Vectors . . . 339
8.4.3 Approximation of a Matrix by a Rotation Matrix 340

Given a set $\{(\boldsymbol{a}', \boldsymbol{a}''), (\boldsymbol{b}', \boldsymbol{b}''), ...\}$ of corresponding vector pairs which are related by an unknown rotation, e.g., $\boldsymbol{a}'' = R\boldsymbol{a}'$, the rotation matrix can be derived directly.

As each pair $(\boldsymbol{a}', \boldsymbol{a}'')$ contains two constraints for the rotation matrix, we need at least two pairs for its determination. This leaves us with one redundant constraint, namely the angle between the vectors needs to be invariant.

We first discuss the direct solutions for some cases, without exploiting the redundancy. In all cases, we assume the given vectors are not linearly dependent, i.e., not coplanar.

The least squares solution for arbitrarily many vector pairs is discussed in the context of the similarity transformation, cf. Sect. 10.5.4.3, p. 408.

8.4.1 Rotation from Three Pairs of Orthonormal Vectors

Given the three pairs $\{(\boldsymbol{e}'_1, \boldsymbol{e}''_1), (\boldsymbol{e}'_2, \boldsymbol{e}''_2), (\boldsymbol{e}'_3 \boldsymbol{e}''_3)\}$ of corresponding orthonormal vectors, the rotation matrix follows from

$$R = R'' R'^{\mathsf{T}} = [\boldsymbol{e}''_1, \boldsymbol{e}''_2, \boldsymbol{e}''_3] \begin{bmatrix} \boldsymbol{e}_1{}'^{\mathsf{T}} \\ \boldsymbol{e}_2{}'^{\mathsf{T}} \\ \boldsymbol{e}_3{}'^{\mathsf{T}} \end{bmatrix} = \boldsymbol{e}''_1 \boldsymbol{e}_1{}'^{\mathsf{T}} + \boldsymbol{e}''_2 \boldsymbol{e}_2{}'^{\mathsf{T}} + \boldsymbol{e}''_3 \boldsymbol{e}_3{}'^{\mathsf{T}}. \tag{8.73}$$

This can easily be proven using the relation $\boldsymbol{e}_i{}'^{\mathsf{T}} \boldsymbol{e}'_j = \delta_{ij}$; we immediately obtain $\boldsymbol{e}''_i = R\boldsymbol{e}'_i$: The two matrices $R' = [\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3]$ and $R'' = [\boldsymbol{e}''_1, \boldsymbol{e}''_2, \boldsymbol{e}''_3]$ are rotation matrices which rotate the basic vectors $\boldsymbol{e}_i$ of the underlying object coordinate system, $\boldsymbol{e}'_i = R'\boldsymbol{e}_i$ and $\boldsymbol{e}''_i = R''\boldsymbol{e}_i$.

8.4.2 Rotation from One, Two, and Three Pairs of Arbitrary Vectors

Three Pairs. Now, given three pairs of noncoplanar vectors, $\{(\boldsymbol{a}', \boldsymbol{a}''), (\boldsymbol{b}', \boldsymbol{b}''), (\boldsymbol{c}', \boldsymbol{c}'')\}$ which are – as before – mutually related by an unknown rotation, e.g., $\boldsymbol{a}'' = R\boldsymbol{a}'$, the rotation matrix is obtained from

$$R = \boldsymbol{a}'' \tilde{\boldsymbol{a}}'^{\mathsf{T}} + \boldsymbol{b}'' \tilde{\boldsymbol{b}}'^{\mathsf{T}} + \boldsymbol{c}'' \tilde{\boldsymbol{c}}'^{\mathsf{T}} \tag{8.74}$$

using the vectors

$$\tilde{\boldsymbol{a}}' = \frac{\boldsymbol{b}' \times \boldsymbol{c}'}{|\boldsymbol{a}'\ \boldsymbol{b}'\ \boldsymbol{c}'|}, \quad \tilde{\boldsymbol{b}}' = \frac{\boldsymbol{c}' \times \boldsymbol{a}'}{|\boldsymbol{a}'\ \boldsymbol{b}'\ \boldsymbol{c}'|}, \quad \tilde{\boldsymbol{c}}' = \frac{\boldsymbol{a}' \times \boldsymbol{b}'}{|\boldsymbol{a}'\ \boldsymbol{b}'\ \boldsymbol{c}'|} \tag{8.75}$$

(cf. Kanatani, 1990, pp. 138–140). This relation can be derived by solving $[\boldsymbol{a}'', \boldsymbol{b}'', \boldsymbol{c}''] = R[\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}']$ for R using the determinant $|\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}'|$, the cofactor matrix $[\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}']^\circ = [\boldsymbol{b}' \times \boldsymbol{c}', \boldsymbol{c}' \times \boldsymbol{a}', \boldsymbol{a}' \times \boldsymbol{b}']$, and the relation $A^{-1} = A^\circ\mathsf{T}/|A|$. It easily can be proven, using $(\boldsymbol{b}' \times \boldsymbol{c}')^\mathsf{T}\boldsymbol{a}' = |\boldsymbol{a}'\ \boldsymbol{b}'\ \boldsymbol{c}'|$, $(\boldsymbol{b}' \times \boldsymbol{c}')^\mathsf{T}\boldsymbol{b}' = 0$, etc.

If both triplets $(\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}')$ and $(\boldsymbol{a}'', \boldsymbol{b}'', \boldsymbol{c}'')$ of vectors are right- or are left-handed, the resulting rotation matrix has determinant $|R| = +1$.

The given vectors may have emerged from an observation or estimation process. Then the resulting matrix is not a rotation matrix, but only close to it: the columns and rows are not normalized and not mutually perpendicular. Determining the best fitting rotation matrix $\widehat{R}$ for a given matrix Q is discussed below.

Two Pairs. Next, only two pairs, $(\boldsymbol{a}', \boldsymbol{b}')$ and $(\boldsymbol{a}'', \boldsymbol{b}'')$, are given. Completing the configuration to three pairs allows us to use the construction of the previous case. Therefore, we obtain the rotation matrix (8.74) with the vectors $\tilde{\boldsymbol{a}}'$, $\tilde{\boldsymbol{b}}'$ and $\tilde{\boldsymbol{c}}'$ from (8.75), where the third vectors

$$\boldsymbol{c}' = \boldsymbol{a}' \times \boldsymbol{b}' \qquad \boldsymbol{c}'' = \boldsymbol{a}'' \times \boldsymbol{b}' ,$$

by construction, are not coplanar to the first two given ones, respectively. Again, if the vectors result from an observation or estimation process, the matrix R only approximates a rotation matrix.

Minimal Rotation from One Pair of Vectors. The last case considers only two given vectors, $\boldsymbol{a} := \boldsymbol{a}'$ and $\boldsymbol{b} := \boldsymbol{a}''$, where the angle between the two is not 0 or 180°. The rotation from $\boldsymbol{a}$ to $\boldsymbol{b}$ is not unique. The minimal rotation in the plane spanned by $\boldsymbol{a}$ and $\boldsymbol{b}$ is obtained from (Weber, 2003b),

$$R_{ab} = I + 2\tilde{\boldsymbol{b}}\tilde{\boldsymbol{a}}^\mathsf{T} - \frac{1}{1 + \tilde{\boldsymbol{a}}\cdot\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{a}} + \tilde{\boldsymbol{b}})(\tilde{\boldsymbol{a}} + \tilde{\boldsymbol{b}})^\mathsf{T} , \quad \text{with} \quad \tilde{\boldsymbol{a}} = \mathsf{N}(\boldsymbol{a}) \qquad \tilde{\boldsymbol{b}} = \mathsf{N}(\boldsymbol{b}) . \tag{8.76}$$

The expression is valid for vectors of arbitrary dimension. In $\mathbb{R}^3$, we have $R_{ab}\ (\boldsymbol{a} \times \boldsymbol{b}) = \boldsymbol{a} \times \boldsymbol{b}$.

8.4.3 Approximation of a Matrix by a Rotation Matrix

Given a nonorthonormal matrix Q, we seek the best fitting rotation matrix $\widehat{R}$ for Q (Arun et al., 1987). For an arbitrary 3×3 matrix Q with its singular value decomposition, $Q = USV^\mathsf{T}$, the rotation matrix

$$\widehat{R} = UV^\mathsf{T} \tag{8.77}$$

minimizes the Frobenius norm $||\widehat{R} - Q||_F = \sum_{ij}(r_{ij} - q_{ij})^2$. The result is plausible: The two matrices U and V are rotation matrices; the matrix S is a real diagonal matrix with nonnegative entries. Substituting the unit matrix for S preserves the rotational components of Q. The proof is given in Arun et al. (1987).

8.5 Exercises

Basics

1. (2) Generate a rotation matrix R with $r = \mathsf{N}([1, 2, 3]^\mathsf{T})$ and $\theta = 240°$. Then determine the rotation vector and the angle using (8.5) and (8.6), e.g., with the MATLAB function `logm`. Explain the result.

2. (2) Define a coordinate system S_o of your office room, a coordinate system S_d of your office desk and their relative translation. Use (8.12) to determine the rotation matrix ${}^{d}R_o$ from the office to the desk coordinate system. Check the result by expressing the coordinates of the lamp $\mathcal{L}$ of your room in both coordinate systems, and transforming the coordinates from the office into the desk coordinate system.
3. (1) Relate the rotation vector $\boldsymbol{\theta}$ of the exponential representation to the vector $\boldsymbol{u}$ of the Cayley representation.
4. (3) *Spherical linear interpolation* (SLERP): Given two unit quaternions $\mathbf{p}$ and $\mathbf{q}$ enclosing an angle ϕ, show that the quaternion

$$\mathbf{r}(t) = \mathbf{p}\frac{\sin\left((1-t)\phi\right)}{\sin\phi} + \mathbf{q}\frac{\sin(t\phi)}{\sin\phi} \qquad 0 \leq t \leq 1 \tag{8.78}$$

   interpolates between the given quaternions, where the interpolation fulfils the constraints:

   a. The vector $\mathbf{r}(t)$ is a unit quaternion.
   b. The vector $\mathbf{r}(t)$ lies on the plane spanned by $\mathbf{p}$ and $\mathbf{q}$.
   c. For $t = 0$ and $t = 1$, the quaternion $\mathbf{r}(t)$ is identical to the two given quaternions.
   d. The angle between $\mathbf{r}(t)$ and $\mathbf{p}$ is $t\phi$.
   e. Equation (8.78) holds for arbitrary dimensions.

5. (1) Given a rotation vector $\boldsymbol{\theta} = [1, -2, 3]^{\mathsf{T}}$ and the corresponding rotation matrix $R(\boldsymbol{\theta}) = \mathrm{e}^{S_\theta}$. Determine the vector $\boldsymbol{u}$ (cf. Sect. 8.1.5.4, p. 336), leading to the same rotation matrix and verify this using (8.62) and (8.63).
6. (3) Given N rotation matrices $R_n, n = 1, ..., N$, a) discuss procedures for determining an average rotation matrix and b) show the rotation parameters are affected when using (a) Euler angles, (b) skew matrices, (c) quaternions, and (d) rotation vectors,

Proofs

7. (3) Prove (8.4), p. 327.
8. (2) Prove (8.6), p. 327, cf. App. A.13, p. 781.
9. (2) Prove that R_A and R_B are given by (8.18), p. 329 and by (8.20), p. 329.
10. (2) Prove (8.54), p. 335 from (8.52), p. 335.
11. (2) Derive (8.29), p. 331 from (8.54), p. 335 using the unit quaternion (8.56), p. 335, and the trigonometric relations $\cos 2\alpha = \cos^2\alpha - \sin^2\alpha$ and $\sin 2\alpha = 2\cos\alpha\sin\alpha$.
12. (2) Prove (A.33), p. 771.
13. (1) Prove (8.30) from (8.29).
14. (2) Prove that for the rotation matrix (8.29) the following two properties are valid: (a) $\boldsymbol{r} = R_{r,\theta}\,\boldsymbol{r}$, (b) $R_{r,\theta}$ is a rotation matrix.
15. (3) Show that the rotation axis $\boldsymbol{r}$ of a rotation matrix R can be determined from the null space of $I_3 - R$. What advantages and disadvantages does this method have compared to the one given in Sect. 8.1.4.2, p. 331?
16. (2) Show the matrices M_q in (8.45) and $\overline{\mathsf{M}}_r$ in (8.46) to be orthogonal.

Applications

17. (3) A rollercoaster at a funfair is positioned at [280,150,7] m in the coordinate system of the fair. The driving direction has a direction angle of 30° counted from the x-axis to the y-axis. The slope of the trail is +20°. The coaster is rolled left by 30°. The coordinate system of the coaster is defined the following way: The driving direction is the x-axis, the y-axis points to the left, and the z-axis points upwards. There is a

church close to the funfair. The tip of its tower has the coordinates [405,110,30] m, again in the coordinate system of the fair.

Question: In which octant (right, left, up/down, ahead/back) is the tip of the tower referring to the coordinate system of the coaster? *Hint:* Determine the coordinates of the tip of the tower in the coordinate system of the coaster. How can you check the transformation matrix?

18. (3) At a funfair we have a wing carousel and a Ferris wheel, see the Fig. 8.3 with xy- and xz-plots. The centre of the carousel $\mathcal{C}$ has the coordinates [35,40,15] m. The chains are fixed at a horizontal wheel with radius $R_A = 5$ m, i.e., $\overline{HC} = r$. The length l of the chains is 7 m. The Ferris wheel's centre is at [110,30,40] m. The radius r_B to the cabins is 35 m. The Ferris wheel has an angle of $\beta_B = 45°$ to the coordinate system of the fair.

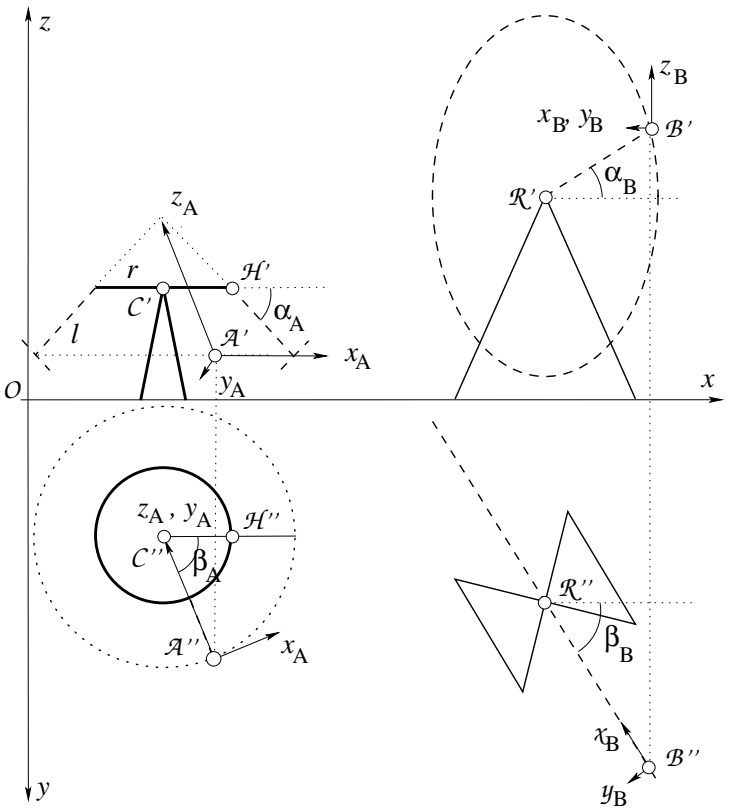


Fig. 8.3 Person $\mathcal{A}$ in a swing carousel and person $\mathcal{B}$ in a Ferris wheel. **Top:**: xz-projection. **Bottom:** xy-projection (not in scale)

At a certain time two persons $\mathcal{A}$ and $\mathcal{B}$ are in the carousel and in the wheel, respectively. The current position of $\mathcal{A}$ can be described by the two angles $\alpha_A = 50°$ and $\beta_A = 65°$. The position of $\mathcal{B}$ by the angle $\alpha_B = 30°$. The local coordinate system $S_A = (x_A, y_A, z_A)$ of $\mathcal{A}$ has the viewing direction as x-axis; the z-axis points along the chains above the person. The coordinate system $S_B = (x_B, y_B, z_B)$ of $\mathcal{B}$ is independent of the angle β_B, with the x-axis, that is the viewing direction of the person, and the z-axis pointing upward.

The task is to determine the direction in which the two persons see each other.

a. Determine the matrices for the displacements $\mathcal{M}_A$ and $\mathcal{M}_B$ of the reference coordinate system into the coordinate systems of the two persons.
b. Determine the coordinates of each of the persons in the coordinate system of the other person, respectively.

Chapter 9
Oriented Projective Geometry

9.1 Oriented Entities and Constructions 344
9.2 Transformation of Oriented Entities 355
9.3 Exercises 358

Classical projective geometry in general does not distinguish between the two opposite directions of a line or the two sides of a plane. Oriented projective geometry provides a framework that accounts for situations where it is very useful to take the orientation of entities into account, for instance, see Fig. 9.1:

- Lines may be given an orientation from the sequence of two points defining the line. As a consequence the signed intersection point of two consecutive edges in a directed polygon tells us whether the polyline makes a right or a left turn.
- Line segments in an image may obtain an orientation depending on the direction of the gradient vector of the image function.
- Planes may inherit an orientation from the sequence of three points defining the plane or when the plane is the boundary of a polyhedron, e.g., with the normal pointing outwards, guaranteeing consistency of all bounding faces.
- Conics and quadrics partition 2D and 3D space, respectively, into two or even more regions which we might want to handle differently.

Fig. 9.1 Examples for oriented elements. **(a)** triangle with oriented sides, **(b)** triangle with orientation opposite to (a), **(c)** oriented edge segments in an image: the area left of the edge segments is brighter than the area right of the edge segment, **(d)** building with normals of bounding planes showing outwards; in case of a convex polyhedron, knowing the lighting direction, we can infer whether a plane is lit or in shadow, **(e)** ellipse with interior and exterior region

We are then able to solve the following tasks, for example:

- Deciding on which side of an oriented 2D line a point lies.
- Deciding whether two oriented lines are parallel or antiparallel, i.e., parallel with opposite direction.
- Deciding whether a point is in front of or behind a camera.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_9

Stolfi's oriented projective geometry (Stolfi, 1991; Vinicius et al., 2001), which contains concepts defining the sign of geometric entities and their relations, is suitable for solving these tasks. It will be outlined in this chapter.

sign and orientation of entities

We first define the sign and the orientation of basic geometric entities, discriminating between their internal and external directions. Constructed entities and the spatial configurations will inherit their orientation from the given entities. This will lead us to the notion of *chirality* or handedness of spatial configurations of oriented entities. For example, a point in 2D may sit either right or left of a directed line, a property which can be derived from the sign of the functions of the given entities. Finally, we will analyse under which conditions transformations preserve or change orientation of geometric entities and preserve or change the chirality of geometric configurations.

handedness, chirality of configurations

Affine transformations preserve orientation or chirality. This is in contrast to general collineations, which are indefinite in this respect, leading to the notion of quasi-affine collineations which only operate on a part of the projective space and thus preserve orientation.

9.1 Oriented Entities and Constructions

9.1.1 Geometric Entities 344
9.1.2 Chiral Geometric Configurations 348
9.1.3 Orientation of Geometric Constructions 352
9.1.4 Signed Distances 354

9.1.1 Geometric Entities

9.1.1.1 The Oriented Point and Its Antipode

antipode of a point

Oriented projective geometry starts with distinguishing between the points $\mathcal{X}(\mathbf{x})$ and its *antipode* $\neg\mathcal{X}(-\mathbf{x})$. Thus the sign of the last component of a homogeneous coordinate vector becomes relevant. Consequently, the identity relation for two points $\mathcal{X}$ and $\mathcal{Y}$ reads

$$\mathcal{X}(\mathbf{x}) \equiv \mathcal{Y}(\mathbf{y}) \quad \Leftrightarrow \quad \exists\, \lambda > 0, \quad \mathbf{x} = \lambda \mathbf{y}\,. \tag{9.1}$$

The only distinction is the restriction on the sign of the scaling factor; it must be positive for two points to be identical. Finite antipodal points $\mathcal{X}$ and $\neg\mathcal{X}$ in $\mathbb{R}^n$ refer to the same position. We also have

$$\mathcal{X} = \neg(\neg\mathcal{X})\,. \tag{9.2}$$

sign of a point

We can distinguish between points based on the sign of the homogeneous part x_h of their homogeneous coordinate vector $\mathbf{x} = [\boldsymbol{x}_0^\mathsf{T}, x_h]^\mathsf{T}$.

Definition 9.1.20: ***Sign of a point.*** The sign of a point $\mathcal{Y}(\mathbf{y}) \in \mathbb{P}^n$ is the sign of the homogeneous part of its homogeneous coordinates. We write this

$$\operatorname{sign}(\mathcal{Y}) = \operatorname{sign}(y_{n+1})\,. \tag{9.3}$$

A point $\mathcal{Y}$ with $\operatorname{sign}(\mathcal{Y}) = 1$ is called a *positive point*, and its antipode is then a *negative point*: $\operatorname{sign}(\neg\mathcal{Y}) = -1$. Consequently, points at infinity with $y_{n+1} = 0$ have sign 0. ◇

Observe, points in Euclidean normalization have $x_{n+1} = 1$, i.e., they have positive sign.

9.1.1.2 The Oriented Projective Space

In analogy to the projective space, we can define an oriented projective space.

The projective space $\mathbb{P}^n$ consists of all lines through the origin of $\mathbb{R}^{n+1}$, each line representing a point in $\mathbb{P}^n$, as discussed for $n = 2$ in Sect. 5.1.2.3, p. 199. We now distinguish between lines through the origin of $\mathbb{R}^{n+1}$ having the opposite orientation. The oriented projective space $\mathbb{T}^n$ consists of all *directed* or *oriented* lines through the origin O of $\mathbb{R}^{n+1}$. In analogy to the projective space, we define:

Definition 9.1.21: *Oriented projective space.* The oriented projective space $\mathbb{T}^n$ consists of all $n+1$-dimensional points $\mathcal{X}$ with homogeneous coordinates $\mathbf{x} \in \mathbb{R}^{n+1} \setminus \mathbf{0}$, *oriented projective space*

$$\mathcal{X}(\mathbf{x}) \in \mathbb{T}^n(\mathbb{R}) : \quad \mathbf{x} \in \mathbb{R}^{n+1} \setminus \mathbf{0}\,, \tag{9.4}$$

with the equivalence relation

$$\mathcal{X}(\mathbf{x}) \equiv y(\mathbf{y}) \quad \Leftrightarrow \quad \mathbf{x} = \lambda \mathbf{y}, \text{ for some } \lambda > 0\,. \tag{9.5}$$

◇

In 2D, the oriented projective plane $\mathbb{T}^2$ consists of two real planes and the line at infinity (see Fig. 9.2). This can be expressed as

$$\mathbb{T}^n = \underbrace{\{\mathbf{x}|x_{n+1} > 0\}}_{\text{front range}} \cup \underbrace{\{\mathbf{x}|x_{n+1} < 0\}}_{\text{back range}} \cup \underbrace{\{\mathbf{x}|x_{n+1} = 0\}}_{\text{at infinity}}\,. \tag{9.6}$$

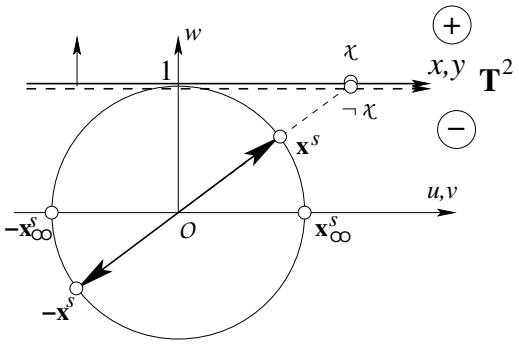


Fig. 9.2 Oriented points in $\mathbb{T}^2$. The point with inhomogeneous coordinates (x, y) is represented twice with homogeneous coordinates, once with $\mathcal{X}(x, y, 1)$ and once as its antipodal point with $\neg\mathcal{X}(-x, -y, -1)$, both also in spherical normalization as $\mathbf{x}^s = \mathsf{N}([x, y, 1]^\mathsf{T}$ and $-\mathbf{x}^s = \mathsf{N}([-x, -y, -1]^\mathsf{T})$. The xy-plane is assumed to be oriented upwards; thus, the points $\mathcal{X}$ with positive component x_{n+1} lie on the front range and $\neg\mathcal{X}$ lie on the back range. The points at infinity, say $\mathcal{X}_\infty$ with coordinates $\mathbf{x}_\infty = \mathsf{N}([u, v, 0]^\mathsf{T})$ and its antipodal point with coordinates $-\mathbf{x}_\infty$, are treated as distinct points. Thus, the 2-sphere S^2 also represents the oriented projective plane, but with antipodal points taken as distinct points, unlike in normal projective geometry

Specifically, oriented 2D points $\mathcal{X}(\mathbf{x})$ with $x_{n+1} > 0$ are said to be at the front range of $\mathbb{T}^n$, whereas points $\mathcal{X}(\mathbf{x})$ with $x_{n+1} < 0$ are said to be at the back range of the plane.

Points $\mathcal{X}$ with $x_{n+1} = 0$, thus $\mathbf{x} = [u, v, 0]^\mathsf{T}$, represent directions, where the direction $\mathcal{X}$ and its antipode $\neg\mathcal{X}$ with coordinates $[-u, -v, 0]^\mathsf{T}$ (the opposite direction) are distinct. Obviously, this is intuitive and useful, e.g., for representing directions during navigation, and constitutes a clear advantage of oriented projective geometry over classical projective geometry, where opposite directions are unified.

We similarly can define oriented 3D points and their antipodes. As an example, we then can distinguish between the zenith point with oriented homogeneous coordinates

$\mathbf{Z} = [0, 0, 1, 0]^\mathsf{T}$ and the nadir point with coordinates $\mathbf{N} = [0, 0, -1, 0]^\mathsf{T}$, where one is the antipode of the other, $\mathbf{Z} = \neg\mathbf{N}$.

9.1.1.3 Oriented 2D Lines

Oriented 2D lines ℓ not at infinity have an *internal direction* $\boldsymbol{d}_l$ and an *external direction* $\boldsymbol{n}_l$, namely the normal $\boldsymbol{n}_l = \boldsymbol{l}_h$ of the line. We already introduced these properties in Sect. 7.1.1.2, p. 294.

Example 9.1.36: ***The y-axis.*** The y-axis has homogeneous coordinates $\mathbf{l} = [-1, 0, 0]^\mathsf{T} = -\boldsymbol{e}_1^{[3]}$. Its internal direction is $\boldsymbol{d}_l = [0, 1]^\mathsf{T}$ and its normal points in the negative x-direction. Thus we have the relation

$$\boldsymbol{d}_l = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = R_{-90}\, \boldsymbol{l}_h = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} .$$

The point at infinity v_l in the direction of the y-axis therefore has homogeneous coordinates $[\boldsymbol{d}_l^\mathsf{T}, 0]^\mathsf{T} = [0, 1, 0]^\mathsf{T}$ pointing upward, whereas its antipode is $\neg v_l([0, -1, 0]^\mathsf{T}$, pointing downward. ◇

dual oriented projective plane

The oriented or directed line $\ell(\mathbf{l})$ has the antipode line $\neg\ell(-\mathbf{l})$ with opposite direction. The *dual oriented projective plane* $\mathbb{T}^{*2}$ consists of all oriented lines, similar to the dual projective plane, which contains all undirected lines.

Lines ℓ not at infinity partition the plane into a positive and a negative region. The positive region $\mathcal{R}_l^+$ contains all nonnegative points with $\mathbf{x}^\mathsf{T}\mathbf{l} > 0$. We can thus partition the oriented projective plane into the following three regions:

$$\begin{aligned}
\mathbb{T}^2 = \quad & \mathcal{R}_l^+ \cup \mathcal{R}_l^- \cup \mathcal{R}_l^0 && (9.7)\\
= \quad & \left\{\mathcal{X}(\mathbf{x}) \,|\, \text{sign}(\mathcal{X}) \geq 0,\ \mathbf{x}^\mathsf{T}\mathbf{l} > 0 \ \text{ or } \ \text{sign}(\mathcal{X}) \leq 0,\ \mathbf{x}^\mathsf{T}\mathbf{l} < 0\right\} && (9.8)\\
& \cup \left\{\mathcal{X}(\mathbf{x}) \,|\, \text{sign}(\mathcal{X}) \geq 0,\ \mathbf{x}^\mathsf{T}\mathbf{l} < 0 \ \text{ or } \ \text{sign}(\mathcal{X}) \leq 0,\ \mathbf{x}^\mathsf{T}\mathbf{l} > 0\right\} && (9.9)\\
& \cup \left\{\mathcal{X}(\mathbf{x}) \,|\, \mathbf{x}^\mathsf{T}\mathbf{l} = 0\right\} . && (9.10)
\end{aligned}$$

The region $\mathcal{R}_l^+$ is the region left of the directed line, i.e., on the positive side of ℓ. The region $\mathcal{R}_l^-$ is the region on the right side of the directed line, see Fig. 9.3.

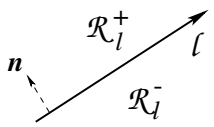


Fig. 9.3 The oriented 2D line partitions the oriented projective plane $\mathbb{T}^2$ in a positive and a negative region $\mathcal{R}_l^+$ and $\mathcal{R}_l^-$, respectively

Finally, $\mathcal{R}_l^0 = \ell$ is the set of all points on the line, be they positive or negative, and including the two vanishing points of the directed line, one pointing in the direction of the line, the other in the opposite direction.

Mapping $\mathbb{T}^2$ to $\mathbb{P}^2$ using stereographic projection

We now give a complete visualization of the oriented projective plane. For this we exploit the stereographic projection, which we used to show the canonical elements of the coordinate system (see Fig. 5.27 left, p. 244). There, we saw that all points $\mathcal{X}([x, y, 1]^\mathsf{T})$ of the real plane are mapped into the interior of the unit circle, whereas the points at infinity are mapped onto the unit circle. It can now be easily proven that negative points with coordinates $[x, y, -1]^\mathsf{T}$ are mapped outside of the unit circle. Figure 9.4 shows the visualization of a gridded square in the first quadrant for the case where all its points are positive and for the where that all points are negative. In this way the oriented projective plane $\mathbb{T}^2$ is mapped to the projective plane $\mathbb{P}^2$.

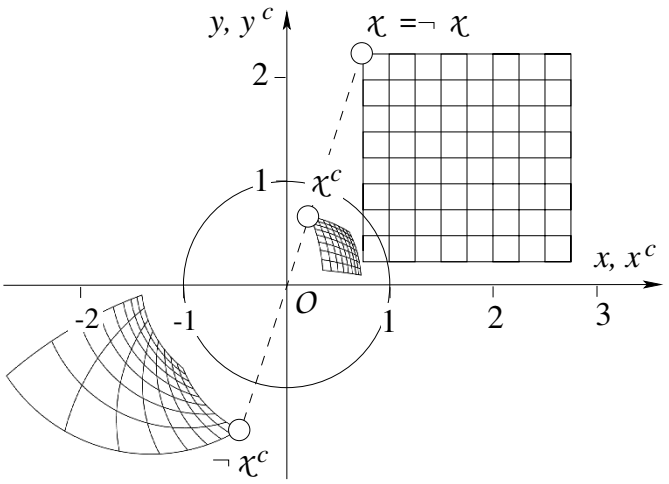


Fig. 9.4 Visualization of the oriented projective plane $\mathbb{T}^2$ in $\mathbb{P}^2$. We use the stereographic projection $\sigma : \mathbb{T}^2 \to \mathbb{P}^2$ (see Fig. 5.26, p. 243). The line at infinity of the projective plane $\mathbb{T}^2$ is mapped to the unit circle in $\mathbb{P}^2$. The front range of the projective plane is mapped into the interior, the back range into the exterior of the unit circle. The gridded square in the upper right with positive points $\mathcal{X}$ in $\mathbb{T}^2$ is mapped into the bent grid in $\mathbb{P}^2$ within the unit circle, e.g., $\mathcal{X}^c$. If the points were negative ($\neg\mathcal{X}$), they were mapped into the outside of the unit circle in $\mathbb{P}^2$, here in the lower left bent grid, e.g., $\neg\mathcal{X}^c$ (see Fig. 5.27, p. 244, Exercise 11, p. 245). This visualization can be transferred to 3D as in Fig. 5.27, right

9.1.1.4 Oriented Planes

Oriented planes are distinguished the same way. The plane $\mathcal{A}(\mathbf{A})$ has the antipodal plane $\neg\mathcal{A}(-\mathbf{A})$, see Fig. 9.5. The *dual oriented projective space* $\mathbb{T}^{*3}$ consists of all oriented planes.

internal and external direction of a plane

As with straight lines, finite planes have an *internal direction* and an *external direction*. In $\mathbb{R}^3$, the *external direction* is the homogeneous part $\boldsymbol{A}_h$ of the homogeneous vector $\mathbf{A}$, thus the normal of the plane $\boldsymbol{N} = \boldsymbol{A}_h$.

The *internal direction* can be visualized by a circular path on the plane related to the external direction by the right-hand rule, see Fig. 9.5. Again, as with lines, finite planes also separate the space into a positive and a negative region, the normal of the plane pointing towards the positive region.

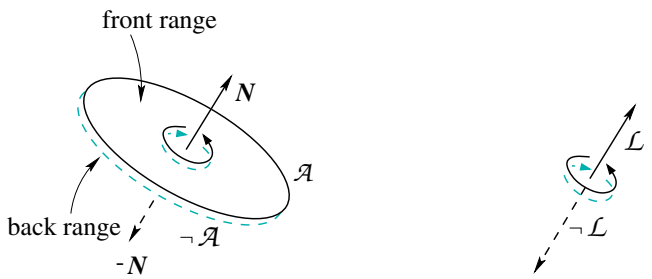


Fig. 9.5 Oriented plane and 3D line. **Left:** Oriented plane $\mathcal{A}$ with internal and external direction. The antipodal plane $\neg\mathcal{A}$ has the opposite direction. The front range of the plane is seen from the side of the normal $\boldsymbol{N} = \boldsymbol{A}_h$ of $\mathcal{A}$, the back range from the other side. **Right:** oriented 3D line $\mathcal{L}$ with internal and external direction. The antipodal line has the opposite direction

9.1.1.5 Oriented 3D Lines

The 3D line $\mathcal{L}(\mathbf{L})$ has the antipodal line $\neg\mathcal{L}(-\mathbf{L})$, see Fig. 9.5. Both parameter vectors exist in the oriented projective space $\mathbb{T}^5$.

internal and external direction of a 3D line

Finite 3D lines $\mathcal{L}(\mathbf{L})$ also have an *internal* and an *external* direction. The *internal direction* $\boldsymbol{D}_L$ of a 3D line in $\mathbb{R}^3$ is the homogeneous part $\boldsymbol{D}_L = \boldsymbol{L}_h$ of $\mathbf{L}$. In $\mathbb{T}^3$, it is the point at infinity $[\boldsymbol{L}_h^\mathsf{T}, 0]^\mathsf{T}$ of $\mathcal{L}$. The *external direction* of a 3D line can be represented by a circular path in a plane perpendicular to $\mathcal{L}$, with the direction related to the internal direction by the right-hand rule, see Fig. 9.5. The dual of an oriented 3D line is also oriented.

9.1.1.6 Direction of Lines and Planes Not at Infinity

We primarily refer to the internal direction of a line and the external direction of a plane, which are related directly to the homogeneous coordinate vectors. Therefore, we use the following definition for the directions of lines and planes.

Definition 9.1.22: ***Directions of lines and planes.*** The directions of lines and planes not at infinity are related to the homogeneous parts of their homogeneous coordinates. They may alternatively be represented as directions or as homogeneous vectors of the points at infinity:

$$\mathcal{d}_l : \boldsymbol{d}_l = R_{-90}\boldsymbol{l}_h \qquad \mathcal{d}_l : \mathbf{d}_{l\infty} = \begin{bmatrix} \boldsymbol{d}_l \\ 0 \end{bmatrix}, \tag{9.11}$$

$$\mathcal{N}_A : \boldsymbol{N}_A = \boldsymbol{A}_h \qquad \mathcal{N}_A : \mathbf{N}_{A\infty} = \begin{bmatrix} \boldsymbol{A}_h \\ 0 \end{bmatrix}, \tag{9.12}$$

$$\mathcal{D}_L : \boldsymbol{D}_L = \boldsymbol{L}_h \qquad \mathcal{D}_L : \mathbf{D}_{L\infty} = \begin{bmatrix} \boldsymbol{L}_h \\ 0 \end{bmatrix}. \tag{9.13}$$

◇

9.1.1.7 Orientation of Conics and Quadrics

Regular conics partition the plane into two regions, which can be characterized by the sign of the expression $C(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x}$. This can be easily seen when viewing the equation of the conic $C(\mathbf{x}) = 0$ as a level set of the function $C(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x}$, which separates areas, and observing where $C(\mathbf{x})$ is positive and where it is negative. For example the outside of the unit circle $\mathsf{C} = \mathrm{Diag}([1, 1, -1])$ is positive, whereas the outside of the unit circle $\mathsf{C} = \mathrm{Diag}([-1, -1, 1])$ is negative.

9.1.2 Chiral Geometric Configurations

We now investigate configurations of multiple directed geometric entities and characterize their topology. We discuss configurations which cannot be transformed into themselves by a rubber sheet transform except after being mirrored at a 2D line or a plane. Such configurations are *handed* and are called *chiral*, a term frequently used in physics and chemistry.

handedness, chirality of configurations

For example, take a point and a directed 2D line. The point either is on its left, i.e., on the positive side, or on the right, i.e., the negative side of the line. Mirroring the configuration at a line changes this spatial relation.

We collect these relations and characterize them with their orientation or handedness. In all cases, we assume that the given points have a positive sign.

9.1.2.1 Configurations in 2D

Sequence of Three 2D Points. The sequence (u, v, w) of three points in general position generates a chiral configuration, see Fig. 9.6, left. The chirality, denoted by chir$(\cdot)$,

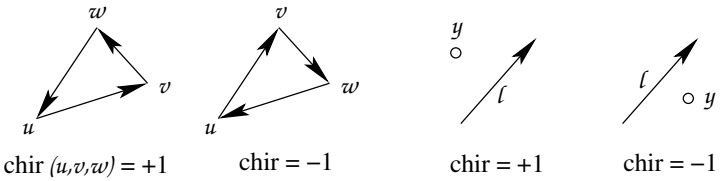


Fig. 9.6 Chirality of three positive 2D points and of a point line configuration with positive point

of this configuration is the sign of the area $A = \det[\mathbf{u}, \mathbf{v}, \mathbf{w}]/2$ of this triangle, see Fig. 5.17, p. 222,

$$\text{chir}(u, v, w) = \text{sign}\left(\det[\mathbf{u}, \mathbf{v}, \mathbf{w}]\right) , \tag{9.14}$$

again assuming the points are positive. If the three points all are negative, the chirality has the opposite sign than with positive points. The chirality of sequences of points with different orientations is not defined.

The handedness of this configuration changes with the sign of the permutation of the three points.

2D Line and Point. As just discussed, a 2D line and a point not sitting on the line are chiral, see Fig. 9.6, right. Their chirality is

$$\text{chir}(\ell, y) = \text{sign}(< \mathbf{l}, \mathbf{y} >) = \begin{cases} +1, & \text{if point } y \text{ is on left side of } \ell , \\ -1, & \text{if point } y \text{ is on right side of } \ell . \end{cases} \tag{9.15}$$

Barycentric Coordinates and Point in Triangle. With the results of the last few subsections, we can easily identify the relation between a point and a triangle, especially whether the point is inside the triangle. For this purpose, we introduce Barycentric coordinates of a point w.r.t. a triangle.

Definition 9.1.23: ***Barycentric coordinates.*** The Barycentric coordinates $\mathbf{a}$ of a point t w.r.t. a triangle (u_1, u_2, u_3) are given by

$$\mathbf{a} = [\mathbf{u}_1^e, \mathbf{u}_2^e, \mathbf{u}_3^e]^{-1} \mathbf{t}^e , \tag{9.16}$$

assuming all homogeneous coordinates are Euclideanly normalized. ◇

The coordinates $\mathbf{t}$ can be written as

$$\mathbf{t} = \sum_{i=1}^{3} a_i \mathbf{u}_i^e . \tag{9.17}$$

Barycentric coordinates

The Barycentric coordinates fulfil $a_1 + a_2 + a_3 = 1$, as can easily be seen from (9.16). If the first two of the Barycentric coordinates are zero, the point t is identical to the point of the triangle with the third index. If one of the Barycentric coordinates is zero, the point t lies on the line joining the points with the other two indices. The sign of the Barycentric coordinates can be used to characterize the position of t w.r.t. the triangle.

Proposition 9.1.5: ***Point inside a triangle.*** A point t lies inside the triangle (u_1, u_2, u_3) if its Barycentric coordinates are all positive, thus

$$\text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3) = 1 , \tag{9.18}$$

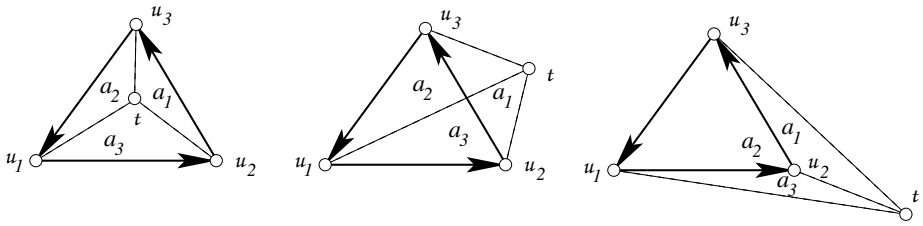


Fig. 9.7 Relation of a point and a triangle. **Left:** If the point t is in the interior of the triangle, the Barycentric coordinates $[a_1, a_2, a_3]^\mathsf{T}$ are all positive. **Middle and Right:** If the point is outside the triangle, one or two of the three Barycentric coordinates are negative. The point is on the right, thus on the negative side of the corresponding lines. **Middle:** The point t is on the negative side of (u_2u_3), thus $a_1 < 0$. **Right:** The point is right of the line (u_2u_3) and the line (u_1u_2), thus $a_1 < 0$ and $a_3 < 0$

see Fig. 9.7. A value a_i represents the signed ratio A_i/A, where A_i is the area of the triangle with the point u_i exchanged for t and A is the area of the triangle (u_1, u_2, u_3).

Proof: The values a_i are the ratio of double the signed areas A_i of the triangles (t, u_2, u_3), (u_1, t, u_3) and (u_1, u_2, t) to double the total area A. This follows from the equation system (9.16) by solving it, e.g., with Cramer's rule. For example, for $i = 1$ we obtain

$$a_1 = \frac{|\mathbf{t}^e, \mathbf{u}_2^e, \mathbf{u}_3^e|}{|\mathbf{u}_1^e, \mathbf{u}_2^e, \mathbf{u}_3^e|} = \frac{2A_1}{2A} . \tag{9.19}$$

◇

9.1.2.2 Configurations in 3D

Sequence of Four 3D Points. A sequence of four positive 3D points in general position generates a chiral configuration. The chirality of this configuration is the sign of the volume

$$V = \frac{1}{6}|\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}| \tag{9.20}$$

of the tetrahedron

$$\text{chir}(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}) = \text{sign}\left(\det[\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}]\right) . \tag{9.21}$$

The handedness of the configuration changes with the sign of the permutation of the four points. The chirality is the same if all points are negative. Again, the chirality of a sequence of four points with different signs is not defined.

The chirality is positive if the last point is on the positive side of the plane through the other three points, as can be seen from the configuration in Fig. 9.8, left, with the determinant of the four points

$$\text{chir}(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}) = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 2 . \tag{9.22}$$

Plane and 3D Point. A plane $\mathcal{A}$ and a 3D point $\mathcal{T}$ not sitting on the plane form a chiral configuration. Its chirality is

$$\text{chir}(\mathcal{A}, \mathcal{T}) = \text{sign}(< \mathbf{A}, \mathbf{T} >) = \begin{cases} +1, & \text{if point } \mathcal{T} \text{ is on front range of } \mathcal{A} , \\ -1, & \text{if point } \mathcal{T} \text{ is on back range of } \mathcal{A} . \end{cases} \tag{9.23}$$

It does not change when exchanging $\mathcal{T}$ and $\mathcal{A}$ in (9.23).

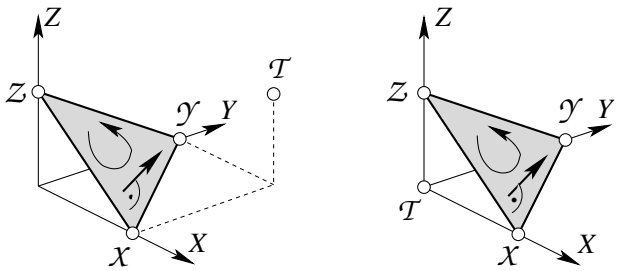


Fig. 9.8 Oriented configuration of four positive points $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$, and $\mathcal{T}$. **Left**: positive chirality, the point $\mathcal{T}$ is on the positive side of the plane $(\mathcal{XYZ})$. **Right**: negative chirality, the point $\mathcal{T}$ is on the negative side of the plane $(\mathcal{XYZ})$

Two 3D Lines. Two 3D lines which do not intersect build a chiral configuration. We may reach the second line from the first one by a right- or left-hand screw motion. The chirality of this configuration is

$$\text{chir}(\mathcal{L}, \mathcal{M}) = \text{sign}(< \mathbf{L}, \mathbf{M} >_D) = \begin{cases} +1, & \text{if line } \mathcal{M} \text{ is reached from } \mathcal{L} \text{ by a left screw}, \\ -1, & \text{if line } \mathcal{M} \text{ is reached from } \mathcal{L} \text{ by a right screw}, \end{cases} \tag{9.24}$$

see Fig. 9.9.

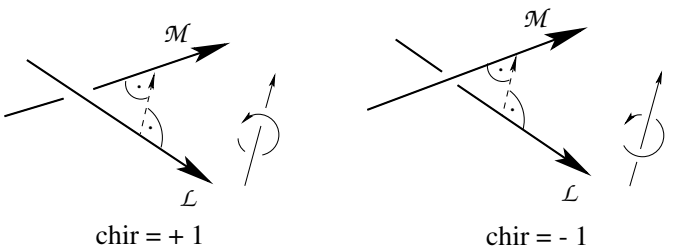


Fig. 9.9 Chirality of two 3D lines: the chirality is positive if we have a left-hand screw when moving the first into the second line. **Left:** $\text{chir}(\mathcal{L}, \mathcal{M}) = +1$. **Right:** $\text{chir}(\mathcal{L}, \mathcal{M}) = -1$

3D Point and Tetrahedron. The test whether a point $\mathcal{T}$ is inside a tetrahedron $(\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4)$ can be done with Barycentric coordinates $\mathbf{W}$ in 3D, cf. Sect. 9.1.2.1, p. 349 and Fig. 9.10, left. They are defined using the Euclideanly normalized homogeneous coordinates of the points by

$$\mathbf{W} = [\mathbf{U}_1^e, \mathbf{U}_2^e, \mathbf{U}_3^e, \mathbf{U}_4^e]^{-1} \mathbf{T}^e \,. \tag{9.25}$$

The coordinates W_i also add to 1 and represent the ratio of the volumes V_i of the tetrahedra with $\mathcal{U}_i$ replaced by $\mathcal{T}$, to the volume V of the given tetrahedron, e.g.,

$$W_1 = \frac{|\mathbf{T}^e, \mathbf{U}_2^e, \mathbf{U}_3^e, \mathbf{U}_4^e|}{|\mathbf{U}_1^e, \mathbf{U}_2^e, \mathbf{U}_3^e, \mathbf{U}_4^e|} = \frac{6V_1}{6V} \,. \tag{9.26}$$

Proposition 9.1.6: ***Point inside a tetrahedron.*** A point $\mathcal{T}$ is inside a tetrahedron $(\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4)$ if its barycentric coordinates are all positive.

The proof is similar to the one for the point in triangle test, cf. (9.18).

3D Line and 3D Triangle. Given a spatial triangle $(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ and a 3D line $\mathcal{L}$, see Fig. 9.10, right, a line $\mathcal{L}$ passes through the face of the triangle if the signs of the three chiralities $\text{chir}(\mathcal{L}, \mathcal{L}_{ij})$ of the line $\mathcal{L}$ w.r.t. the directed sides $\mathcal{L}_{ij} = \mathcal{X}_i \wedge \mathcal{X}_j$ of the triangle

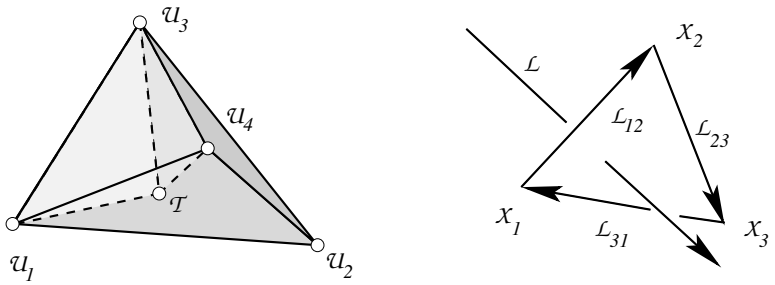


Fig. 9.10 **Left**: Mutual relation between a point $\mathcal{T}$ and a tetrahedron. As an example, the sixfold volume of the tetrahedron $(\mathcal{U}_1, \mathcal{T}, \mathcal{U}_3, \mathcal{U}_4)$ is V_2: the Barycentric coordinate W_2 is the ratio of this volume V_2 to the volume V of the complete tetrahedron. **Right**: Mutual relation between a 3D line $\mathcal{L}$ and a spatial triangle. Line $\mathcal{L}$ approaches the interior of the triangle from its positive side, why all chiralities of $\mathcal{L}$ with the directed lines $\mathcal{L}_{ij} = \mathcal{X}_i \wedge \mathcal{X}_j$ are positive

are identical,

$$\text{chir}(\mathcal{L}, \mathcal{L}_{12}) = \text{chir}(\mathcal{L}, \mathcal{L}_{23}) = \text{chir}(\mathcal{L}, \mathcal{L}_{31}) \,. \tag{9.27}$$

The signs are positive if the line approaches the interior of the triangle from the positive side.

From these chiral configurations we can derive characteristics of more complex configurations, which can then be characterized by more than one sign. These signs can be used to exclude alternatives during object recognition.

9.1.3 Orientation of Geometric Constructions

When we join or intersect geometric entities, the resulting entities have a unique orientation or sign provided they are not at infinity. We begin with the join and intersection where the given points are positive, which is the normal case when starting from inhomogeneous coordinates.

9.1.3.1 Orientation of Constructions in 2D

Join of Two Points and Line Segments. The line $\ell = \mathcal{X} \wedge \mathcal{Y}$ joining the two points $\mathcal{X}(\mathbf{x})$ and $\mathcal{Y}(\mathbf{y})$ of an ordered pair $(\mathcal{X}, \mathcal{Y})$ has the direction $\boldsymbol{d}(\ell) = \boldsymbol{d}_l = (x_h \boldsymbol{y}_0 - y_h \boldsymbol{x}_0)$. The direction depends on the position and the sign of the points.

direction of line joining two points

The *line* joining two points with the *same* sign has the direction given by the order of the two points (see Fig. 9.11):

$$\boldsymbol{d}(\ell) = \boldsymbol{d}_l = \boldsymbol{y} - \boldsymbol{x} \qquad \mathbf{d}(\ell) = \begin{bmatrix} \boldsymbol{y} - \boldsymbol{x} \\ 0 \end{bmatrix} . \tag{9.28}$$

Observe, for the direction, we have

$$\boldsymbol{d}(\mathcal{X} \wedge \mathcal{Y}) = -\boldsymbol{d}(\mathcal{Y} \wedge \mathcal{X}) \,. \tag{9.29}$$

direction of line segment

The *line segment* $\mathcal{S} = (\mathcal{X}\,\mathcal{Y})$ between two points of the *same* sign contains all points in between the given points:

$$\mathcal{S} = (\mathcal{X}\,\mathcal{Y}) : \qquad \mathcal{S} = \{\mathcal{Z} | \mathbf{z} = (1-\alpha)\mathbf{x} + \alpha\mathbf{y}, \alpha \in [0,1]\} \,. \tag{9.30}$$

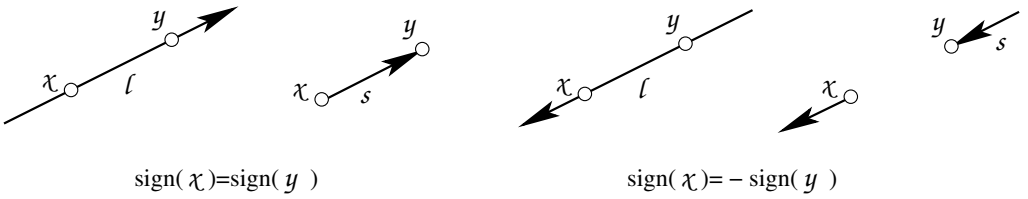


Fig. 9.11 Constructions of oriented 2D lines $\ell = \mathcal{X} \wedge y$ and 2D segments $s = (\mathcal{X}\, y)$. **Left:** oriented line and segment from two points with the same sign. **Right:** oriented line and segment from two points with different signs

If the signs of the two points are different, the line joining the two points has the opposite direction, thus $\boldsymbol{d}(\mathcal{X} \wedge y) = -\boldsymbol{d}(\neg\mathcal{X} \wedge y)$. The line segment $s = (\neg\mathcal{X}\, y)$ obviously contains a point at infinity; thus, it joins the two points after passing the line at infinity.

Analogously, the join of two positive 3D points $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$ has the direction $\boldsymbol{D}(\mathcal{L}) = \boldsymbol{D}_L = \boldsymbol{Y} - \boldsymbol{X}$. We also have $\mathbf{D}(\mathcal{X} \wedge \mathcal{Y}) = -\mathbf{D}(\mathcal{Y} \wedge \mathcal{X})$ for the direction, see Fig. 9.13. The construction of 3D line segments follows the same rules as in 2D.

Intersection of Two Lines. The intersection $\mathcal{X} = \ell \cap m$ of two directed lines leads to an oriented point $\mathcal{X}$, see Fig. 9.12. If the intersection point is finite, the sign is *positive* if the *shortest turn* from ℓ to m at $\mathcal{X}$ is a *left turn*. In the case of a correspondent *right turn*, the sign of the point is *negative*; if the lines are *parallel*, the sign is 0:

sign of intersection point of two 2D lines

$$\text{sign}(\ell \cap m) = \begin{cases} +1, & \text{if a turn from } \ell \text{ to } m \text{ at } \mathcal{X} \text{ is a left turn}, \\ -1, & \text{if a turn from } \ell \text{ to } m \text{ at } \mathcal{X} \text{ is a right turn}, \\ 0, & \text{if lines are parallel}. \end{cases} \tag{9.31}$$

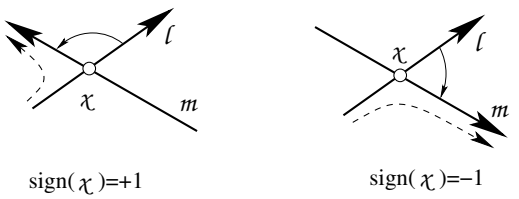


Fig. 9.12 Constructions of oriented 2D points. **Left**: the shortest path from the first to the second line via the intersection point is a left turn, the intersection point is positive. **Right**: the shortest turn to the right leads to a negative intersection point

9.1.3.2 Orientation of Constructions in 3D

Intersection of Two Planes. Two oriented planes $\mathcal{A}$ and $\mathcal{B}$ intersect in an oriented 3D line $\mathcal{L}$. Its external direction is the rotation direction from the first to the second plane, i.e., the rotation direction of their normal vectors. The three vectors $(\boldsymbol{A}_h, \boldsymbol{B}_h, \boldsymbol{L}_h)$ form a right-handed system. Observe, the intersection of two planes is anticommutative: $\mathcal{A} \cap \mathcal{B} = -\mathcal{B} \cap \mathcal{A}$.

Join of a 3D Line and a Positive Point. The plane $\mathcal{A} = \mathcal{L} \wedge \mathcal{X}$ has a definite orientation: it is directed such that the point $\mathcal{X}$ is on the positive side of $\mathcal{A}$ when looking at the positive side of it. The join of a 3D line and a point is commutative: $\mathcal{L} \wedge \mathcal{X} = \mathcal{X} \wedge \mathcal{L}$.

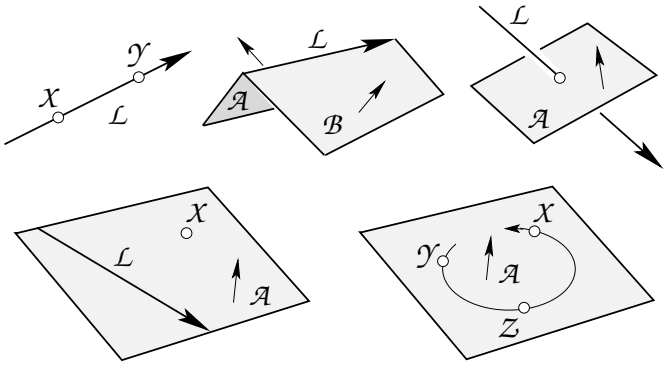


Fig. 9.13 Constructions of oriented 3D elements

Intersection of a 3D Line and a Plane. The intersection point $\mathcal{X} = \mathcal{L} \cap \mathcal{A}$ has a definite sign,

$$\operatorname{sign}(\mathcal{L} \cap \mathcal{A}) = \begin{cases} +1, & \text{if line } \mathcal{L} \text{ approaches plane } \mathcal{A} \text{ from the positive side,} \\ -1, & \text{if line } \mathcal{L} \text{ approaches plane } \mathcal{A} \text{ from the negative side,} \end{cases} \tag{9.32}$$

referring in both cases to the direction of the line $\mathcal{L}$. The intersection of a 3D line with a plane is commutative.

Plane Through Three Points. The plane $\mathcal{A}$ through three points is given by

$$\mathcal{A} = \mathcal{Y} \wedge \mathcal{Z} \wedge \mathcal{X} = \mathcal{L}(\mathcal{Y}, \mathcal{Z}) \wedge \mathcal{X} \,. \tag{9.33}$$

Its orientation is given by the right-hand rule. The exterior orientation of the plane is given by the orientation of the three points: the chirality of the triangle $(\mathcal{YZX})$ is positive when seen from the positive side of the plane. The orientation of the plane changes with the sign of the three points. It also changes with the sign of the permutation of the three points.

9.1.4 Signed Distances

Generally, distances are nonnegative values. However, in special cases, e.g., when providing the distance of a 2D point from a 2D line, it is useful to also know the relative position of the two elements, e.g., whether the point is on the positive or negative side of the 2D line. This information is therefore encoded in the sign of the distance, so that it indicates on which side of the line the point is, including the special case that the sign may be 0 if the point lies on the line.

In the sections on distances (7.1.3, p. 297 and 7.2.3, p. 308), we already gave the equations for signed distances. Here, we collect the interpretations of the signs of these distances. Care has to be taken if points are involved, as the signs of the points may change the signs of the distance.

- The sign of the distance $d_{xy} = |x_h \boldsymbol{y}_0 - y_h \boldsymbol{x}_0|/(x_h y_h)$ depends on the signs of the points. It is positive if both points have the same sign; otherwise, the sign of the distance is negative. In this case the line segment $s(\mathcal{xy})$ passes the line at infinity, see Fig. 9.11, p. 353. The same is true of the distance d_{XY} of 3D points.
- If a 2D point $\mathcal{x}$ is positive, the distance $d_{xl} = \langle \mathbf{x}, \mathbf{l} \rangle / (|x_h \boldsymbol{l}_h|)$ from a 2D line $\mathcal{l}$ is positive if the point is at the left of the line; otherwise, it is negative. The argument transfers to the distance d_{XA} of a point from a plane. The sign of the distance changes with the sign of the point. Observe, the Euclidean parameters d and S of the Hessian form

of 2D lines and planes are the negative values of the corresponding distances d_{xl} and d_{XA}.

- The sign of the distance $d_{LM} = \langle \mathbf{L}, \mathbf{M} \rangle_D / |\boldsymbol{L}_h \times \boldsymbol{M}_h|$ between two 3D lines is the chirality of the configuration, which is positive if the screw motion between the lines is a left screw, and negative otherwise.

9.2 Transformation of Oriented Entities

9.2.1 Projective Transformations 355
9.2.2 Affine and Quasi-affine Mappings 357

We transfer the concept of oriented projective geometry to transformations. We treat two transformations $\mathcal{G}$ and $\mathcal{H}$ as identical if their transformation matrices differ only by a *positive* scale factor. Thus, similarly to points, we have

$$\mathcal{G}(\mathsf{G}) \equiv \mathcal{H}(\mathsf{H}) \quad \Leftrightarrow \quad \exists \lambda > 0 \quad \mathsf{G} = \lambda \mathsf{H} \,. \tag{9.34}$$

The reason is evident: multiplying a homogeneous coordinate vector with $-\mathsf{I}_{n+1}$ changes its sign.

Transformations of geometric entities now may or may not influence their sign or direction and may or may not influence the chirality of chiral configurations. We therefore want to identify transformations which totally preserve or reverse orientation and distinguish them from those which preserve, reverse, or destroy orientation and chirality only in parts of their domain.

We will see that projective transformations generally do not preserve orientation and chirality. They depend on the actual transformation, on the actual local configuration, and on the actual position. On the other hand, in the hierarchy of collineations, affinities turn out to be the most general transformations, with a very clear characteristic with respect to chirality transformations. Only if we restrict the domain of a general collineation can we arrive at what is called quasi-affine transformations, which behave predictably when transforming oriented entities (cf. Hartley and Zisserman, 2000, Sect. 21.1).

9.2.1 Projective Transformations

In this section we discuss the situation based on an example and draw the relevant conclusions in the next section.

Assume the homography (see Fig. 9.14)

$$\mathbf{x}' = \mathsf{H}\mathbf{x} \qquad \text{with} \quad \mathsf{H} = \mathsf{H}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \mathsf{I}_3 - \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} [-1\ 0\ 1] \,, \tag{9.35}$$

or, explicitly,

$$x' = \frac{1}{x} \qquad y' = \frac{y}{x}. \tag{9.36}$$

It is a perspectivity (cf. Sect. 6.5, p. 277) with the projection centre $\mathbf{z} = [-1, 0, 1]^{\mathsf{T}}$, the fixed line $\mathbf{l}^f = [-1, 0, 1^{\mathsf{T}}]$, and $\det(\mathsf{H}) = -1$, as well as $\mathsf{H}^\circ = \mathsf{H}^{\circ\mathsf{T}} = -\mathsf{H}$.

We first discuss the *sign change* of points. Consider the pre-image ℓ_∞^- of the line at infinity, i.e., the line which is mapped to the line at infinity, $\ell_\infty = [0, 0, 1]^{\mathsf{T}}$. It is given by

$$\mathbf{l}_\infty^- = (\mathsf{H}^\circ)^{-1}\mathbf{l}_\infty = -\mathsf{H}\begin{bmatrix}0\\0\\1\end{bmatrix} = -\mathbf{h}_3 = -\boldsymbol{e}_1, \tag{9.37}$$

where $\mathbf{h}_3$ is the third row of H. The line $\mathbf{l}_\infty^- = -\mathbf{h}_3$ then partitions the plane into two regions $\mathcal{R}_l^+$ and $\mathcal{R}_l^-$ as discussed above, p. 346. In the example, this pre-image is the line $\ell_\infty^- = -[1,0,0]^\mathsf{T}$, thus the y-axis. The region $\mathcal{R}^+$ is the left half plane containing all points with negative x-coordinates and $\mathcal{R}^-$ is the right half plane, both excluding the y-axis. Thus, as the sign of the third coordinate of $\mathbf{x}'$ defines its sign, which is identical to the first coordinate of $\mathbf{x}$, points in $\mathcal{R}_l^+$ are mapped to negative points and points in $\mathcal{R}_l^-$ to positive points.

As the sign change depends on the position of the point in the plane, there is no simple characterization of a general projective mapping w.r.t. the sign of transformed points based on the transformation matrix alone.

As a consequence, a line segment of two positive points which lie on different sides of the line $\mathbf{l}_\infty^-$, say $s = (\mathcal{X}z)$ in Fig. 9.14, will intersect this line. The intersection point, here the point p, will be mapped to infinity. Hence, the line segment is mapped to a line segment $s' = (\mathcal{X}'z')$ bounded by two points with different signs, thus not mapped to a line segment with only points not in infinity (cf. the text after (9.30), p. 352). Actually, the two points $\mathcal{X}$ and y are fixed points in nonoriented projective geometry as they are on the fixed line. In oriented projective geometry, the direction of the joining line changes: $\ell' = \neg\ell$, as $z' = \neg z$.

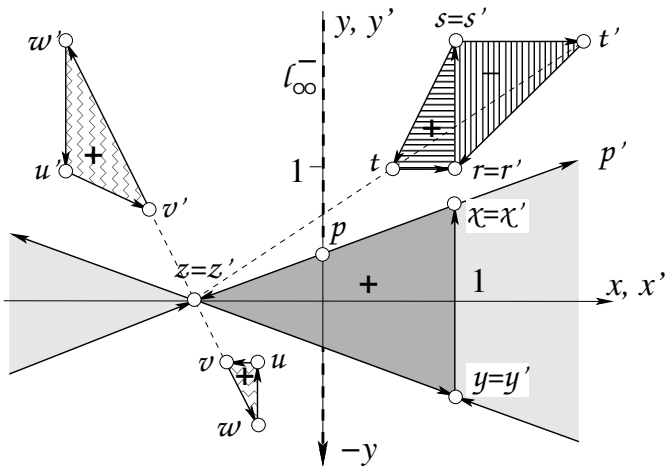


Fig. 9.14 Orientation and chirality of the homography (9.35). *Orientation:* The line segment $(\mathcal{X}z)$ intersecting the y-axis, thus the pre-image ℓ_∞^- of the line at infinity, contains the point p. It is mapped to the point $p' \in \ell_\infty$ at infinity. Thus the path from $\mathcal{X}$ to z via p maps to the infinite path from $\mathcal{X}'$ to z' via p'. The corners of the triangle $(\mathcal{X}yz)$ are mapped to themselves, but the connecting line segments $(\mathcal{X}z)$ and (zy) are not preserved and their original orientation is destroyed. *Chirality:* Triangles completely on one side of the pre-image of the line at infinity, here left or right of the y-axis, such as the triangles (uvw) and (rst), are mapped to triangles with a definite chirality: The positive chirality of the triangle (uvw) is preserved by the mapping, whereas the positive chirality of the triangle (rst) is reversed

Now let us analyse the *chirality* of configurations mapped by this homography. Assume a sequence of points is on the positive side of the pre-image ℓ_∞^- of the line at infinity, like (u, v, w) in Fig. 9.14. Then it is mapped to a sequence of three positive points. Obviously, this holds for all triangles with points on the positive side of the pre-image of the line at infinity. If the triangle completely is on the right side of ℓ_∞^-, the chirality is reversed.

In contrast, assume one of the three points is on the positive, the other two on the negative side, as in the triangle $(\mathcal{X}, y, z)$ in Fig. 9.14. Then the chirality of the transformed triangle is not defined.

General projective transformations which are not affinities have no uniform property w.r.t. oriented projective geometry, unless we restrict the mapping to some subregion.

9.2.2 Affine and Quasi-affine Mappings

In order to find orientation preserving transformations, we note two consequences of the discussion above:

1. Special projectivities, namely affinities, can be characterized.
2. A projectivity can not be characterized, unless we restrict the projection to a subregion.

We therefore define projective mappings which preserve orientation and chirality. These are the affine and quasi-affine mappings (cf. Hartley and Zisserman, 2000, Sect. 21.1).

Affine transformations have the general form

$$\mathsf{A} = \begin{bmatrix} M & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & s \end{bmatrix} , \tag{9.38}$$

with an arbitrary 3×3 matrix M and a translation vector $\boldsymbol{t}$, if $s \neq 0$. Then we have the following two characteristics:

1. The sign of points does not change if $s > 0$. Thus the sign of the affinity of an affine mapping $\mathcal{A}(\mathsf{A})$ is

$$\operatorname{sign}(\mathcal{A}) = \operatorname{sign}(s) = \begin{cases} +1, & \text{affinity } \mathcal{A} \text{ preserves the sign of points}, \\ -1, & \text{affinity } \mathcal{A} \text{ reverses the sign of points}. \end{cases} \tag{9.39}$$

   We usually assume that affine transformations are represented with $s = 1$, i.e., having sign $+1$.
2. The chirality of chiral configurations changes if the determinant of the affinity transformation is negative. For example, consider four 3D points with their homogeneous coordinates collected in the 4×4 matrix X; then the sign of this configuration is $|X|$ and changes with $|\mathsf{A}|$ as $|\mathsf{A}X| = |\mathsf{A}|\,|X|$. Thus the chirality $\operatorname{chir}(\mathcal{A})$ of an affine mapping $\mathcal{A}(\mathsf{A})$ follows from

$$\operatorname{chir}(\mathcal{A}) = \operatorname{sign}(|M|)\operatorname{sign}(\mathcal{A}) = \operatorname{sign}(|\mathsf{A}|) = \begin{cases} +1, & \text{affinity } \mathcal{H} \text{ is chirality preserving}, \\ -1, & \text{affinity } \mathcal{H} \text{ is chirality reversing}. \end{cases} \tag{9.40}$$

The findings are collected in the following theorem.

Theorem 9.2.6: ***Orientation and chirality characterization of affinities.*** Affinities can be characterized w.r.t. the preservation of orientation of points and the chirality of configurations. Their orientation is given by (9.39) and (9.40).

If the domain of a projective transformation is restricted to one side of the pre-image of the hyperplane at infinity, the mapping also can be characterized. Such a restricted mapping is called quasi-affine.

Definition 9.2.24: ***Quasi-affine projective mapping.*** A collineation $\mathbb{P}^n \to \mathbb{P}^n : \mathbf{x} \to \mathbf{x}'$ is called quasi-affine with respect to a domain $\mathcal{D}$ if for all sets of points $\mathcal{S} = \{\mathbf{x}_1, ..., \mathbf{x}_{n+1}\}$, we have $|\mathbf{x}'_1, ..., \mathbf{x}'_{n+1}| = k|\mathbf{x}_1, ..., \mathbf{x}_{n+1}|$, where the factor k has the same sign for all point sets $\mathcal{S} \in \mathcal{D}$. ◇

This obviously holds for all affine mappings. In the example of the last section it is true for all domains $\mathcal{D}$ either completely left or right of the y-axis. The consequence is the following theorem:

Theorem 9.2.7: ***Quasi-affine collineations.*** A regular collineation $\mathcal{H}(\mathsf{H}) : \mathbb{P}^n \to \mathbb{P}^n$ which is not an affinity and whose domain is restricted to the one side of the hyperplane $\mathcal{A} : \mathbf{A} = \mathsf{H}^\mathsf{T}\boldsymbol{e}_{n+1}$ is quasi-affine. The chirality $\operatorname{chir}(\mathcal{H}_{\text{quasiaffine}})$ of a quasi-

affine projective mapping restricted to one side of the hyperplane $\mathcal{A} : \mathbf{A} = \mathsf{H}^{\mathsf{T}}\boldsymbol{e}_{n+1}$ is $\text{chir}(\mathcal{H}_{\text{quasiaffine}}) = |\det \mathsf{H}|$ if the domain is restricted to the positive side of $\mathcal{A}$; otherwise, it is $\text{chir}(\mathcal{H}_{\text{quasiaffine}}) = -|\det \mathsf{H}|$.

For example, restricting the domain of the mapping with H from (9.35) to the left half plane $x < 0$ yields a quasi-affine mapping with positive chirality, whereas restricting the domain to the right half plane $x > 0$ also is a quasi-affine mapping, but with negative chirality.

We will exploit these properties of collineations when analysing the geometry of mappings with single and multiple cameras, which for physical reasons only map points in front of the camera.

9.3 Exercises

1. (2) Show that the normal of an oriented line joining two points in 2D is left of the line.
2. (2) Under what conditions is the intersection point of two 2D lines positively oriented?
3. (2) Devise a test for checking whether two 2D lines are antiparallel.
4. (2) A 2D line derived from two oriented points has a direction vector in one of the four quadrants. In which quadrant is the direction of $\mathit{l} = \mathit{x} \wedge \mathit{y}$ as a function of the signs $\text{sign}(\mathit{x})$ and $\text{sign}(\mathit{y})$. *Hint:* Use the first two columns of $\mathsf{S}(\mathbf{l})$.
5. (1) Determine the directions of the lines $\mathcal{L}_i, i = 1, ..., 4$, which are the four columns of $\mathsf{\Pi}(\mathbf{X})$ in Fig. 7.7, left, p. 314. What do you assume?
6. (2) Determine the directions of the planes $\mathcal{A}_i, i = 1, ..., 6$, which are the six rows of $\mathsf{\Pi}(\mathbf{X})$ in Fig. 7.7, centre and right, p. 314. What do you assume?
7. (3) Determine the directions of the four planes $\mathcal{A}_i, i = 1, ..., 4$, representing the columns of $\mathsf{\Gamma}(\mathbf{L})$ (see Fig. 7.8, p. 314) of a directed line $\mathcal{L}$ and interpret them.
8. (3) Determine the signs of the four points $\mathcal{X}_i, i = 1, ..., 4$, representing the columns of $\bar{\mathsf{\Gamma}}(\mathbf{L})$ (cf. (7.8), p. 314) of a directed line $\mathcal{L}$ and interpret them.

Chapter 10
Reasoning with Uncertain Geometric Entities

10.1 Motivation 360
10.2 Representing Uncertain Geometric Elements 364
10.3 Propagation of the Uncertainty of Homogeneous Entities 386
10.4 Evaluating Statistically Uncertain Relations 393
10.5 Closed Form Solutions for Estimating Geometric Entities 395
10.6 Iterative Solutions for Maximum Likelihood Estimation 414
10.7 Exercises 432

Geometric entities in many cases are uncertain in a statistical sense: they deviate to a certain extent from some ideal value. This may be due to the randomness of the observation process, e.g., when identifying points or edges extracted by some image analysis procedure, or due to the lack of knowledge when specifying some geometric constraint, e.g., the perpendicularity between two lines, or even only due to rounding errors resulting from finite machine precision.

In the following, we first discuss the representation of uncertain homogeneous entities and the properties of different normalizations. We especially introduce a representation of the uncertainty which is minimal, thus does not contain singular covariance matrices.

The construction of uncertain homogeneous vectors and matrices relies on the classical techniques of variance propagation and will force us to reconsider the problem of equivalence of now uncertain homogeneous entities. Checking geometric relations will be seen as performing statistical tests.

Finally, we develop methods for the estimation of geometric elements and transformation parameters. We will discuss both closed form solutions which are either suboptimal or only optimal under restricted preconditions, and maximum likelihood estimates. They are generally applicable, statistically optimal, and at the same time use a minimal representation for the uncertainty and the estimated parameters.

Integrating uncertainty into projective geometry goes back at least to Kanatani (1991, 1996), who used Euclideanly normalized entities, Collins (1993), who worked with the Bingham distribution for representing uncertain homogeneous vectors, and Criminisi (1997), who presented various methods for deriving 3D information from single images together with its uncertainty. Chap. 5.2 in Hartley and Zisserman (2000) explicitly addresses the representation of homogeneous entities with covariance matrices. Introductory papers on uncertain reasoning in the plane and on minimal representations of uncertain entities are Meidow et al. (2009) and Förstner (2010b). Whereas using second-order statistics for describing uncertainty of geometric entities is well-established in the area of photogrammetry, rigorous statistical evaluation is taken as the golden standard also in computer vision, see the key paper by Triggs et al. (2000).

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_10

10.1 Motivation

When assuming geometric entities to be uncertain we are faced with at least the following three problems, in the order of the chapters in Part I:[1]

1. *constructing* uncertain geometric entities,
2. *testing* geometric relations, p. 362, and
3. *estimating* geometric entities, p. 362.

We look at the problem of handling uncertain parameters again, but now from the perspective of uncertain geometric reasoning, accepting some overlap between this section and the key ideas of Part I.

uncertain constructions

Construction. Constructing uncertain geometric elements and transformations refers to situations where the number of given constraints is equal to the degrees of freedom, as compiled in Chap. 7, p. 291. Thus, there are no redundant, possibly contradicting, observations or constraints, as when deriving the intersection point of a plane and a line, or when deriving the eight parameters of a general 2D homography from four corresponding points.

Although checking the resulting parameters, say of the intersection point or the homography parameters, is not possible, we may derive the uncertainty of these parameters if we know the uncertainty of the given entities and constraints. The derived uncertainty is theoretical, but can be used to evaluate the configuration, e.g., to identify whether the given four point pairs are in a general configuration, or whether they form a critical configuration, which may be indicated by a very large or even infinite uncertainty of the derived parameters. The derived uncertainty may be introduced in the next step of geometric reasoning.

For this reason, we will generally describe an uncertain geometric entity by its parameters, together with the uncertainty of the parameters. We will use a statistical view and represent the uncertainty by a probability distribution or by parameters describing this distribution.

For example, consider the situation in Fig. 10.1. Given are the four uncertain points

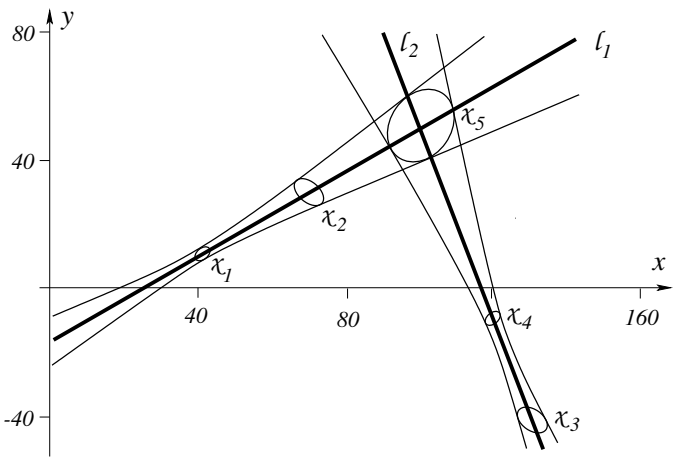


Fig. 10.1 Uncertain point x_5 derived as intersection of two uncertain lines ℓ_1 and ℓ_2 which themselves are derived from uncertain points x_1 to x_4: Variance propagation is straightforward in this case and can be transferred to other processes of generating geometric entities and transformations

[1] This motivation can also be used before a lecture on the basics of statistics, omitting some equations, and provided the audience is to some extent familiar with homogeneous coordinates of 2D points and lines.

represented by the coordinates and their covariance matrices,

$$\mathcal{X}_1 : \left\{ \begin{bmatrix} 40 \\ 10 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} \right\} \qquad \mathcal{X}_2 : \left\{ \begin{bmatrix} 70 \\ 30 \end{bmatrix}, \begin{bmatrix} 16 & -8 \\ -8 & 16 \end{bmatrix} \right\}$$

and

$$\mathcal{X}_3 : \left\{ \begin{bmatrix} 130 \\ -40 \end{bmatrix}, \begin{bmatrix} 16 & -8 \\ -8 & 16 \end{bmatrix} \right\} \qquad \mathcal{X}_4 : \left\{ \begin{bmatrix} 120 \\ -10 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} \right\}.$$

The resultant uncertain point $\mathcal{X}_5$, which is the intersection of the two joining lines $\mathcal{l}_1$ and $\mathcal{l}_2$, is

$$\mathcal{X}_5 : \left\{ \begin{bmatrix} 100 \\ 50 \end{bmatrix}, \begin{bmatrix} 81.88 & 15.51 \\ 15.51 & 128.83 \end{bmatrix} \right\}.$$

The variance propagation on which this result is based can be applied to all constructions discussed in the previous sections.

representation of uncertainty

We represent an uncertain coordinate $\underline{x}$ as a pair

$$\{\mu_x, \sigma_x^2\}. \tag{10.1}$$

The coordinate μ_x is the mean value and σ_x^2 is the variance of the stochastic variable $\underline{x}$ (designated by an underscore) describing the stochastic nature of the experiment. Since an observation can be interpreted as an estimate for the mean, we can take the coordinate x, derived via an image analysis algorithm, as an estimate $\widehat{\mu}_x$ for the mean μ_x, the standard deviation $\sigma_{\widehat{\mu}_x}$ of this estimate representing the expected variation when repeating the experiment. Thus, we could also write the result of the experiment as $\{\widehat{\mu}_x, \sigma^2_{\widehat{\mu}_x}\}$; however, we will stick to the less complex notation in (10.1).

If the relative precision, say σ_x/μ_x, of a distance is high enough, i.e., better than 1%, propagation of uncertainty can be approximated by simple variance propagation and can exploit the multilinearity of the constructions to explicitly state the required Jacobians. Then, neglecting higher-order terms has a practically tolerable influence on the propagated variances and covariances (but see the discussion in Sects. 2.7.6 to 2.7.7, p. 46 ff.). If the uncertain random vector $\{\boldsymbol{\mu}_x, \Sigma_{xx}\}$ is transformed by the vector-valued smooth function $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, we obtain an uncertain vector $\{\boldsymbol{\mu}_y, \Sigma_{yy}\}$ with mean and variance

propagation of uncertainty

$$\boldsymbol{\mu}_y = \boldsymbol{f}(\boldsymbol{\mu}_x) \qquad \Sigma_{yy} = J_{yx} \Sigma_{xx} J_{yx}^{\mathsf{T}} \quad \text{with} \quad J_{yx} = \left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \right) \Big|_{x=\mu_x}. \tag{10.2}$$

When actually applying these relations, we evaluate all expressions at $\boldsymbol{\mu}_x = \boldsymbol{x}$, leading to $\boldsymbol{\mu}_y = \boldsymbol{y}$, assuming the observed or derived values are sufficiently good estimates of the mean.

This type of variance propagation can be seen as propagating the metric for measuring distances between two entities, the inverse covariance matrix can be interpreted as the metric for measuring distances between two vectors.

This can be seen from the following: The *Mahalanobis distance* between two statistically independent points $p(\boldsymbol{p})$ and $q(\boldsymbol{q})$ with covariance matrices Σ_{pp} and Σ_{qq} is given by, cf. (3.32), p. 69:

Mahalanobis distance

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{(\boldsymbol{q} - \boldsymbol{p})^{\mathsf{T}} (\Sigma_{pp} + \Sigma_{qq})^{-1} (\boldsymbol{q} - \boldsymbol{p})} = \sqrt{\boldsymbol{d}^{\mathsf{T}} \, \Sigma_{dd}^{-1} \, \boldsymbol{d}}, \tag{10.3}$$

with the coordinate difference $\boldsymbol{d} = \boldsymbol{q} - \boldsymbol{p}$. The Mahalanobis distance has unit [1], as it takes the uncertainty of the two points into account. Linearly transforming both points, leading to, say, $\boldsymbol{d}' = A\boldsymbol{d}$ with a regular matrix A, leaves the Mahalanobis distance invariant if the variance propagation is applied to the covariance matrices of $\mathbf{p}$ and $\boldsymbol{q}$. We know the Mahalanobis distance both as an optimization function (4.35), p. 84, and as a test statistic (3.32), p. 69.

As the matrix $W_{dd} = \Sigma_{dd}^{-1}$ weighs individual components of $\boldsymbol{d}$ and can be directly related to the metric tensor in tensor calculus, cf. Kanatani (1993), variance propagation actually transfers the metric through a linear mapping (A in our example). Therefore, applying statistical variance propagation following (10.2) can be interpreted as "just" propagating metrics for properly measuring distances. For the special case that the Jacobian $J_{yx} = \partial \boldsymbol{y}/\partial \boldsymbol{x}$ is regular, we have the weight propagation, derived from (10.2),

$$W_{yy} = J_{yx}^{-\mathsf{T}}\, W_{xx}\, J_{yx}^{-1}\,. \tag{10.4}$$

testing uncertain geometric relations

Testing. Examples for geometric relations to be tested are the incidence of a plane $\mathcal{A}$ and a point $\mathcal{X}$ by $\mathbf{A}^\mathsf{T}\mathbf{X} = 0$, and constraints between geometric entities in different reference frames, e.g., when checking the correspondence of two points $\mathcal{x}$ and $\mathcal{x}'$ related by a given, possibly uncertain, homography $\mathcal{H}$ by $\mathsf{S}(\mathbf{x}')\mathsf{H}\mathbf{x} = \mathbf{0}$.

We demonstrate the relevance of rigorous statistical testing with a simple example. Take the situation in the following Fig. 10.2, where we want to check whether the uncertain points $\mathcal{x}_i$ lie on the uncertain line $\mathcal{l}$. Obviously, taking a simple geometrically motivated

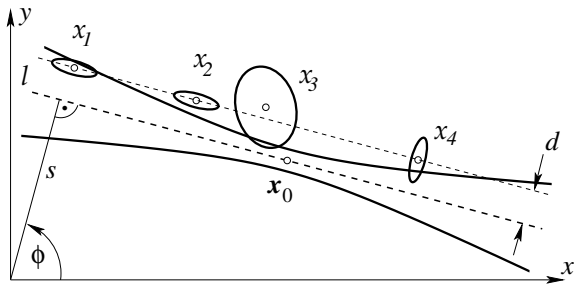


Fig. 10.2 For testing, it is necessary to rigorously take the uncertainty into account. The figure illustrates the problem when testing a point–line incidence. Shown is a line $\mathcal{l}$ with its uncertainty regions, which is a hyperbola, and its centre point $\boldsymbol{x}_0$ and four points $\mathcal{x}_i$ with their standard ellipse. The simple distance d of a point from the line is not an appropriate criterion for testing. The uncertainty of a point may be visualized by a standard ellipse, that of a line by a standard hyperbola. As can be seen, the decision whether the point lies on the line depends on (1) the precision of the point (compare $\mathcal{x}_2$, $\mathcal{x}_3$ and $\mathcal{x}_4$), (2) on the position of the point along the line (compare points $\mathcal{x}_1$ and $\mathcal{x}_2$), and (3) on the precision of the line. The situation is even more complex in 3D. However, the problem becomes simple if we perform a statistically rigorous test based on the Mahalanobis distance, as only a single significance level needs to be specified for all possible tests

threshold on the distance d of $\mathcal{x}_i$ from $\mathcal{l}$ – see the dotted lines – is not sufficient.

All constraints may be written in the form $\boldsymbol{f}(\mathbb{E}(\underline{\boldsymbol{p}}|H_0)) = \mathbf{0}$: the vector-valued function $\boldsymbol{f}$ should hold for the expected values $\mathbb{E}(\underline{\boldsymbol{p}})$ of the parameters $\underline{\boldsymbol{p}}$ if we assume a given null hypothesis H_0 is true and tested with respect to some alternative hypothesis H_a (which generally asserts H_0 does not hold). The actual parameters $\boldsymbol{p}$ will lead to a discrepancy, $\boldsymbol{f}(\boldsymbol{p}) = \boldsymbol{d} \neq \mathbf{0}$. For simplicity, we write the hypothesis as

$$\boldsymbol{f}(\boldsymbol{p}) = \boldsymbol{d} \stackrel{!}{=} \mathbf{0}\,, \tag{10.5}$$

indicating the discrepancy $\boldsymbol{d}$ should be $\mathbf{0}$ under the null hypothesis.

estimation of geometric entities

Estimation. Estimating geometric elements is required in the case of redundant information, as when fitting a line through a set of more than two points or determining a 2D homography from more than four corresponding points. Due to the uncertainty of the elements involved, the set of necessary constraints, e.g., that all points lie on the line, is inconsistent. Allowing for corrections or residuals, the problem can be regularized by

minimizing some optimization function. This optimization function may be motivated algebraically or statistically, depending on the goal. We will discuss both, since algebraic optimization often allows direct solutions without requiring approximate values for the parameters. On the other hand, statistically motivated optimization procedures allow us to provide the covariance matrix of the estimated parameters or functions of these parameters for further testing, i.e., for evaluating the result.

algebraic and statistical view on optimization

For example, let 3D points on two roofs be measured by a laser scanner. We assume the two roofs have the same slope and the data are geo-referenced, a ground plan of the building is given, and therefore the direction of the ridge of the roof is assumed to be known. This means we face the situation in Fig. 10.3. The gable point z, which is

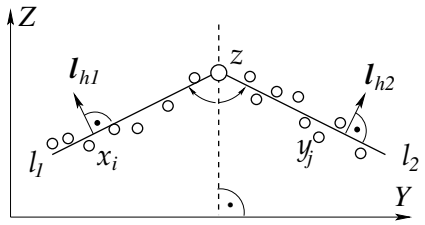


Fig. 10.3 Example for estimation with homogeneous entities. Estimation of the gable z of a symmetric roof from 3D points x_i and y_j, assuming the main direction of the roof is given, and lies in the X-direction

the intersection of the two symmetric slope lines ℓ_1 and ℓ_2, can be determined in a joint estimation scheme using the following constraints. First, the points x_i and y_j need to be on the two corresponding lines $\ell_1([a_1, b_1, c_1]^\mathsf{T})$ and $\ell_2([a_2, b_2, c_2]^\mathsf{T})$,

$$x_i \in \ell_1 : \quad \widehat{\mathbf{x}}_i^\mathsf{T}\, \widehat{\mathbf{l}}_1 = 0,\ |\widehat{\mathbf{x}}_i| = 1 \quad i = 1, ..., I \tag{10.6}$$

$$y_i \in \ell_2 : \quad \widehat{\mathbf{y}}_j^\mathsf{T}\, \widehat{\mathbf{l}}_2 = 0,\ |\widehat{\mathbf{y}}_j| = 1 \quad j = 1, ..., J\,. \tag{10.7}$$

Second, the gable point z should be the intersection point of ℓ_1 and ℓ_2, and the normals $\mathbf{l}_{h1} = [a_1, b_1]^\mathsf{T}$ and $\mathbf{l}_{h2} = [a_2, b_2]^\mathsf{T}$ of the two lines should be symmetric, thus $a_1 = -a_2$ and $b_1 = b_2$,

$$z = \ell_1 \cap \ell_2 : \quad \widehat{\mathbf{z}} \times (\widehat{\mathbf{l}}_1 \times \widehat{\mathbf{l}}_2) = \mathbf{0} \qquad |\widehat{\mathbf{z}}| = 1 \tag{10.8}$$

$$\text{symmetry:} \quad \widehat{a}_1\widehat{b}_2 + \widehat{a}_2\widehat{b}_1 = 0 \qquad |\widehat{\mathbf{l}}_1| = |\widehat{\mathbf{l}}_2| = 1\,. \tag{10.9}$$

This is a general estimation problem for multiple homogeneous entities with constraints between the observed values and the unknown parameters, and constraints on the observed quantities and on the parameters. Though there is no one-step algebraic solution to this problem, approximate values can easily be determined and used for a rigorous maximum likelihood estimation of all quantities.

estimating multiple geometric entities

Observe, we here assumed the correspondence problem to be solved, i.e., the points x_i belong to the left and the points y_j belong to the right side of the roof, and there is no constraint for guaranteeing the intersection point z separates the points x_i and y_j.

Generally. all unknown entities, observations and parameters should be spherically normalized in order to allow for entities at infinity. Due to the imposed length constraint, the covariance matrix of a spherical normalized entity is singular. This is plausible as a normalized entity is an element of a lower dimensional, generally curved space which is a *manifold*, [2] e.g., a sphere for homogeneous point coordinate vectors. In order to cope with this problem, we perform each iteration of the estimation in the tangent space and go back

[2] At each point of an n-dimensional manifold there is a neighbourhood which is homeomorphic to the Euclidean space of dimension n, i.e., there exists a continuous function between the manifold and the Euclidean space such that its inverse also is continuous. Loosely speaking, each neighbourhood of a point of the manifold is a slightly deformed Euclidean space.

to the curved manifold. Take for example the task of estimating the mean of three spherically normalized homogeneous vectors, see Fig. 10.4. The observations $\mathbf{l}_i, i = 1, 2, 3$, are

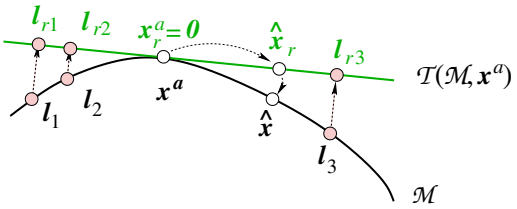


Fig. 10.4 Principle of estimation in the tangent space $\mathcal{T}(\mathcal{M}, \boldsymbol{x}^a)$ of a curved manifold $\mathcal{M}$ at an approximate point $\boldsymbol{x}^a$ using reduced coordinates. Normalizations generally lead to algebraic entities which are points on a curved manifold, e.g., normalizing homogeneous 3-vectors leads to vectors which are points on a sphere, which is a two-dimensional manifold. Whereas the projection from the manifold to the tangent space is linear, the back projection generally is not

mapped onto the tangent space $\mathcal{T}(\mathcal{M}, \mathbf{x}^a)$ of the curved manifold $\mathcal{M}$ at some approximate value $\mathbf{x}^a$ for the mean $\mathbf{x}$. This leads to reduced coordinates $\boldsymbol{l}_{ri}$, reduced as their dimension is reduced by the projection. The reduced coordinates are averaged leading to $\widehat{\mathbf{x}}_r$ in the tangent space of the manifold. Finally, this point is transferred back to the manifold, leading to the an improved estimate $\widehat{\boldsymbol{x}}$. This is the motivation to discuss the representation of uncertain geometric entities and transformations more in detail, and especially to discuss minimal parametrizations.

In the following, we first discuss the representation of uncertain homogeneous entities and the properties of different normalizations. We especially introduce a representation of uncertainty which is minimal, i.e., does not contain singular covariance matrices. The construction of uncertain homogeneous vectors and matrices relies on the classical techniques of variance propagation and will force us to reconsider the problem of equivalence of now uncertain homogeneous entities. Checking geometric relations will lead to statistical tests which due to the nonlinearity of the relations need to be generalized to the situation where the null-hypothesis clearly is not fulfilled. Finally, we develop methods for the estimation of geometric elements and transformation parameters. There, we will discuss both closed form solutions, which are either suboptimal or only optimal under restricted preconditions, and maximum likelihood estimates, which are generally applicable and at the same time use a minimal representation for the uncertainty and the estimated parameters.

10.2 Representing Uncertain Geometric Elements

10.2.1 Using Uncertain Homogeneous Coordinates 365
10.2.2 Uncertain Homogeneous Coordinate Vectors 366
10.2.3 Uncertain Transformations 381

To describe uncertain geometric entities we will introduce several representations which are required within different contexts of geometric reasoning. Whereas Euclidean representations of uncertain geometric entities appear as observations or as final outputs at the beginning or the end of the reasoning chain, uncertain homogeneous entities are the key for simplifying geometric reasoning and for handling elements at infinity. We first discuss the general problem of representing geometric entities in projective spaces and then develop individual representations for each of the geometric entities.

10.2.1 Using Uncertain Homogeneous Coordinates

In Sect. 5.1, we defined the projective space $\mathbb{P}^n$ as the equivalence class of all vectors $\lambda\mathbf{x}$, $\lambda \neq 0$, see (5.33), p. 207. We need to find a suitable equivalence relation for uncertain representatives of projective entities: after Euclidean normalization it should lead to the same uncertainty as that if we had performed all calculations with a Euclidean representation.

It seems difficult to add uncertainty to such an equivalence class, since the scale ambiguity needs to be taken into account in a proper way. However, concrete calculations are always performed with specific representatives, namely after selecting specific values for the scale λ. Hence, we exploit the equivalence relation

$$\mathbf{x} \equiv \mathbf{y} \quad \Leftrightarrow \quad \mathbf{x} = \lambda\mathbf{y} \quad \text{for some} \quad \lambda \neq 0 \tag{10.10}$$

either when normalizing a homogeneous vector, i.e., by choosing a very specific λ, or when testing two vectors w.r.t. equivalence. Then adding uncertainty to representatives, say to $\mathbf{x} \in \mathbb{R}^{n+1}$, is simple, e.g., by specifying the probability density function $p(\mathbf{x}) = p(x_1, ..., x_{n+1})$, or, in the case of a Gaussian distribution, by specifying the $(n+1)\times(n+1)$ covariance matrix $\Sigma_{\mathbf{xx}}$. Uncertainty propagation during construction of new entities from given ones may then be applied to the representatives, leading to the probability density function of the derived entity. The situation is sketched in Fig. 10.5, p. 365, which shows the necessary steps:

1. Transition from Euclidean parameters to homogeneous coordinates.
   This transition is trivial in the case of 2D or 3D points. For the other elements it is easy.

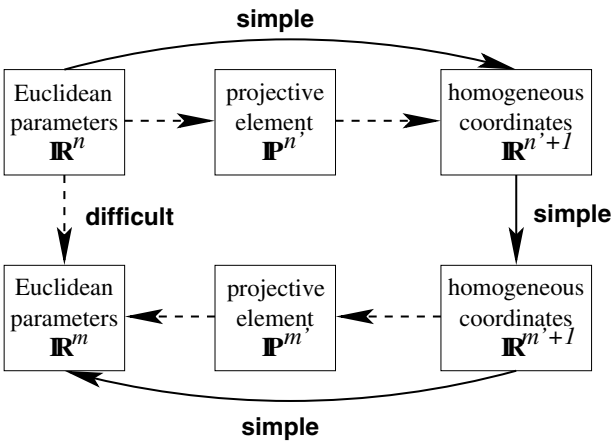


Fig. 10.5 Ways of geometric reasoning with uncertain homogeneous coordinates. Classical geometric reasoning with uncertain Euclidean entities (left column: $\mathbb{R}^n \mapsto \mathbb{R}^m$) may be difficult. Reasoning using homogeneous coordinates (right column) is comparatively simple, as shown in this section. Homogeneous coordinate vectors in $\mathbb{R}^{n'+1}$ or in $\mathbb{R}^{m'+1}$ are representatives for projective elements (middle column, $\mathbb{P}^{n'}$ or $\mathbb{P}^{m'}$) and may carry their uncertainty. The dimensions n' and m' depend on the type of geometric entity. The transitions from a Euclidean representation in $\mathbb{R}^n$ to a homogeneous one in $\mathbb{R}^{n'+1}$, the reasoning with homogeneous coordinates $\mathbb{R}^{n'+1} \mapsto \mathbb{R}^{m'+1}$, and the transition back to a Euclidean representation $\mathbb{R}^{m'+1} \mapsto \mathbb{R}^m$ require only a few types of nonlinear operations. Computational reasoning uses homogeneous vectors in some Euclidean space, which are interpreted as elements in some projective space by the user

If the geometric elements are uncertain, the transition is rigorous for points; the uncertainty of the Euclidean coordinates transfers to the corresponding homogeneous coordinates.

A rigorous transition for the other uncertain elements with arbitrary distribution is difficult in general as nonlinear operations are involved, e.g., trigonometric functions of the angle ϕ of the Hessian normal form of a straight line.
If we restrict the discussion to the first two moments of the distribution, i.e., the covariance matrix, and the relative accuracies are high, an approximation of the covariance matrix of the homogeneous coordinates can be determined easily by variance propagation.

2. Construction of new geometric entities, see Fig. 10.5, right column.
Due to the multilinearity of most constructions, this transition is easy. Again, the transition for uncertain entities with arbitrary distribution is difficult in general, as at least products of homogeneous coordinates are involved. If the relative accuracy is high and if only second moments are considered, variance propagation leads to good approximations of the covariance matrices of constructed elements. The analysis of the bias of products is relevant here, see Sect. 2.7.6, p. 44.
3. Transition from homogeneous coordinates to Euclidean parameters.
Primarily, this is a normalization, possibly followed by ratios or trigonometric functions. Again, in general the transition is difficult, due to the inherent nonlinearities, yet it may be simplified by restricting uncertainty propagation to second moments, small variances and reasonably small relative precision. This is shown in Sect. 2.7.7, p. 46. However, there are pitfalls when characterizing ratios of random variables, see Sect. 2.7.7, p. 46.

10.2.2 Uncertain Homogeneous Coordinate Vectors

We now derive uncertain homogeneous vectors for all entities from Euclidean parametrizations if their uncertainty is given.

10.2.2.1 Uncertain 2D Points

uncertain 2D point

Uncertain 2D Points in the Euclidean Plane. Uncertain 2D points with coordinates $\underline{\boldsymbol{x}} = [\underline{x}, \underline{y}]^\mathsf{T}$ in the Euclidean plane are given by their mean $\boldsymbol{\mu}_x$ and their covariance matrix $\boldsymbol{\Sigma}_{xx}$,

$$\{\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}\} = \left\{ \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \right\}, \tag{10.11}$$

where the covariance of $\underline{x}$ and $\underline{y}$ depends on the correlation coefficient ρ_{xy} via $\sigma_{xy} = \rho_{xy}\sigma_x\sigma_y$. As discussed in Sect. 2.4.4, p. 29, an uncertain point can be visualized by its *standard ellipse*:

$$(\boldsymbol{x} - \boldsymbol{\mu}_x)^\mathsf{T} \boldsymbol{\Sigma}_{xx}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_x) = 1\,. \tag{10.12}$$

The standard ellipse visualizes important properties of the uncertainty, see Fig. 2.6, right, p. 32.

It is often desirable to characterize the uncertainty by one single number. Two measures are common:

1. The square root of the trace of the covariance matrix,

$$\sigma_H = \sqrt{\mathrm{tr}\boldsymbol{\Sigma}_{xx}} = \sqrt{\lambda_1 + \lambda_2}\,. \tag{10.13}$$

Helmert point error

It is visualized by the semi-diagonal of the bounding box of the standard ellipse. It thus gives the maximum directional uncertainty of the point, reached if $\rho = \pm 1$. It also is called the Helmert point error.
The square root of the arithmetic mean of the eigenvalues, which is $\sigma_H/\sqrt{2}$, is also sometimes given.

2. The geometric mean of the standard deviations,

$$\sigma_G = \sqrt{\sigma_x \sigma_y} = \sqrt[4]{|\Sigma_{xx}|} = \sqrt[4]{\lambda_1 \lambda_2}\,, \tag{10.14}$$

   which is identical to the fourth root of the determinant. The geometric mean gives the radius of the confidence circle with the same area as the confidence ellipse.

Both measures have the same unit as the standard deviations of the coordinates.

Uncertain Point Representing a Line Element. Sometimes we want to represent a point with infinite uncertainty in one special direction, say $\boldsymbol{d}$. This may be also a simplified model of a line (or an edge) element in an image, where we assume the position to be uncertain across the line by some degree but totally uncertain along the line, and has a fixed direction. This is a special case of an uncertain line, see Sect. 10.2.2.3, p. 373. We will use this representation for deriving a closed form solution for the intersection point of multiple lines, see Sect. 10.5.3.1, p. 401. Then it is of advantage to represent the uncertainty using the inverse covariance matrix or the *precision matrix* $\mathsf{W}_{xx} = \Sigma_{xx}^{-1}$.

precision matrix

A 2D or 3D *edge element* can therefore be represented as a point with a special weight matrix. It is a point $\boldsymbol{x} \in \mathbb{R}^n$ with zero weight in a given fixed direction, say $\boldsymbol{d}$, and isotropic uncertainty, say with weight w, in the other directions perpendicular to $\boldsymbol{d}$, see Fig. 10.6. Then its *singular precision matrix*, see Sect. 2.4.4.3, p. 33, is given by

line element as uncertain point

$$\mathsf{W}_{xx} = w \left(\mathsf{I}_n - \frac{\boldsymbol{d}\boldsymbol{d}^\mathsf{T}}{\boldsymbol{d}^\mathsf{T}\boldsymbol{d}} \right) , \tag{10.15}$$

where the matrix after the factor w is a projection matrix having $n-1$ eigenvalues 1 and one eigenvalue 0, which therefore is idempotent. The corresponding standard ellipse is

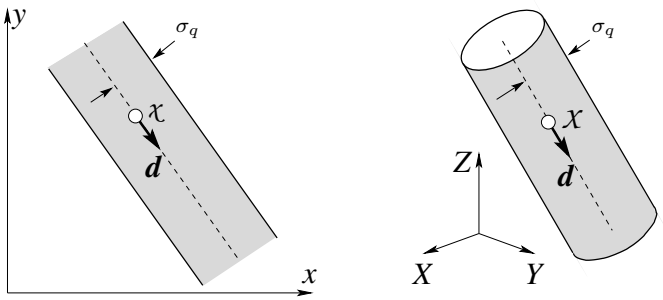


Fig. 10.6 Line element: Uncertain point with zero precision, i.e., infinite uncertainty in one direction. **Left:** 2D point. **Right:** 3D point

degenerated to an infinitely long pair of parallel lines or, in 3D, a cylinder with axis in the direction $\boldsymbol{d}$ and with radius $\sigma_q = 1/\sqrt{w}$. This model may obviously be used to represent an infinitely long line with uncertainty only across the line and no directional uncertainty.

Uncertain Euclideanly Normalized Homogeneous Coordinates. For 2D points, we immediately obtain the transition from uncertain Euclidean coordinates $\{\boldsymbol{\mu}_x, \Sigma_{xx}\}$ to uncertain Euclideanly normalized homogeneous coordinates $\{\boldsymbol{\mu}_{\mathbf{x}^e}, \Sigma_{\mathbf{x}^e\mathbf{x}^e}\}$:

$$\{\boldsymbol{\mu}_x, \Sigma_{xx}\} \quad \rightarrow \quad \{\boldsymbol{\mu}_{\mathbf{x}^e}, \Sigma_{\mathbf{x}^e\mathbf{x}^e}\} = \left\{ \begin{bmatrix} \boldsymbol{\mu}_x \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \Sigma_{xx} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \right\} . \tag{10.16}$$

We obviously assumed the homogeneous parameter 1 of $\boldsymbol{\mu}_{\mathbf{x}^e}$ to be nonstochastic, or with variance 0.

The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}^e\mathbf{x}^e}$ has rank 2, as we would expect, since a 2D point has two degrees of freedom. The null space is

$$\mathsf{null}\,(\boldsymbol{\Sigma}_{\mathbf{x}^e\mathbf{x}^e}) = \boldsymbol{e}_3\,, \tag{10.17}$$

null space of covariance matrix

indicating the 3D standard ellipsoid, see Fig. 10.7, is flat in the w-direction. The null space results from the constraint $g(\mathbf{x}) = \boldsymbol{e}_3^\mathsf{T}\mathbf{x} - 1 = 0$, or, linearized, $g(\widehat{\mathbf{x}}^a) + \boldsymbol{g}^\mathsf{T}\widehat{\Delta\mathbf{x}} = 0$, with the Jacobian $\boldsymbol{g} = \partial g/\partial\mathbf{x} = \boldsymbol{e}_3$. This can be shown using estimation model C, see Sect. 4.8.1, p. 162, taking the unconstrained homogeneous vector as an observation, imposing the constraint with $\boldsymbol{B} = \boldsymbol{e}_3$, and deriving the covariance matrix of the fitted observation $\widehat{\boldsymbol{l}}$ given in Table 4.10, p. 172.

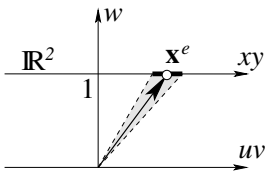


Fig. 10.7 Standard ellipse of an uncertain 2D point $\boldsymbol{x} \in \mathbb{R}^2$, represented as uncertain Euclideanly normalized homogeneous 3-vector $\mathbf{x}^e$. The flatness of the 3D-ellipsoid indicates the covariance matrix is singular. Its null space points in the direction of $\boldsymbol{e}_3$, i.e., of the w-axis. Joining the flat ellipsoid with the origin yields an elliptic cone: it represents the uncertainty of the direction of the homogeneous vector $\mathbf{x}$

Uncertain Spherically Normalized Coordinates. The domain of spherically normalized random vectors is the unit sphere S^2. There are several ways to generalize the Gaussian distribution of the two-dimensional plane to the two-dimensional sphere (see Mardia and Jupp, 1999). Here, we will use the projected Gaussian distribution resulting from spherical normalization of a three-dimensional random vector with nonzero mean. In addition, we assume the directional uncertainty to be small enough that the normalized vectors from the sphere in a first approximation can be approximated by points on the tangent plane at the projected mean vector $\mathsf{N}(\boldsymbol{\mu}_\mathbf{x})$. In the following we represent uncertain entities by their mean and their covariance matrix, not requiring the entities to have a Gaussian distribution.

distribution on the sphere

Thus, we start from some uncertain homogeneous coordinates $\{\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}}\}$, which in the most simple case will be Euclideanly normalized coordinates, and normalize them spherically, with the Jacobian

spherical normalization of uncertain 2D point

Exercise 10.24

$$J_s(\mathbf{x}) := J_{\mathbf{x}^s\mathbf{x}}(\mathbf{x}) = \frac{\partial\mathbf{x}^s}{\partial\mathbf{x}} = \frac{1}{|\mathbf{x}|}\left(I_3 - \frac{\mathbf{x}\mathbf{x}^\mathsf{T}}{\mathbf{x}^\mathsf{T}\mathbf{x}}\right) \tag{10.18}$$

to be evaluated at the mean value $\boldsymbol{\mu}_\mathbf{x}$:

$$\{\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}}\} \quad \rightarrow \quad \{\boldsymbol{\mu}_{\mathbf{x}^s}, \boldsymbol{\Sigma}_{\mathbf{x}^s\mathbf{x}^s}\} = \left\{\frac{\boldsymbol{\mu}_\mathbf{x}}{|\boldsymbol{\mu}_\mathbf{x}|}, J_{\mathbf{x}^s\mathbf{x}}(\boldsymbol{\mu}_\mathbf{x})\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}}J_{\mathbf{x}^s\mathbf{x}}^\mathsf{T}(\boldsymbol{\mu}_\mathbf{x})\right\}; \tag{10.19}$$

see Fig. 10.8. The covariance matrix has null space $\mathsf{null}(\boldsymbol{\Sigma}_{\mathbf{x}^s\mathbf{x}^s}) = \mathbf{x}^s$. Therefore, the standard ellipsoid of the 3-vector $\mathbf{x}^s$ is flat and lies in the tangent plane at the unit sphere at $\boldsymbol{\mu}_{\mathbf{x}^s}$. The null space results from linearizing the nonlinear constraint $g(\mathbf{x}^s) = 1/2(\mathbf{x}^{s\mathsf{T}}\mathbf{x}^s - 1) = 0$ with the Jacobian $\boldsymbol{g} = \partial g/\partial\mathbf{x}^s = \mathbf{x}^s$.

Remark: We illustrate the effect of spherical normalization on the uncertainty structure of a field of image points. When dealing with images taken with a camera this can be interpreted as the transition from uncertain image coordinates to uncertain image ray directions. If we assume the inhomogeneous coordinates $\boldsymbol{x}$ to have homogeneous and isotropic uncertainty, the standard ellipses in the real plane

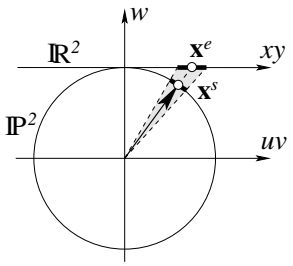


Fig. 10.8 Standard ellipse of an uncertain 2D point, represented as uncertain spherically normalized homogeneous 3-vector $\mathbf{x}^s \in \mathbb{P}^2$. Its covariance matrix is singular with null space pointing in the direction of $\mathbf{x}^s$

are circles, with some radius σ_x indicating the positional uncertainty. This is a reasonable model for the stochastic properties of coordinates of image points. When normalizing the homogeneous vectors $\mathbf{x}$ spherically, we obtain the uncertainty of direction vectors. This yields a rotationally symmetric uncertainty field on the unit sphere, where the standard ellipses decrease and become more narrow with increasing distance from the principal direction $[0, 0, 1]$, see Fig. 10.9. This does not appear a reasonable stochastical model for wide angle cameras. Below we will discuss the situation where the directional uncertainty is homogeneous and isotropic.

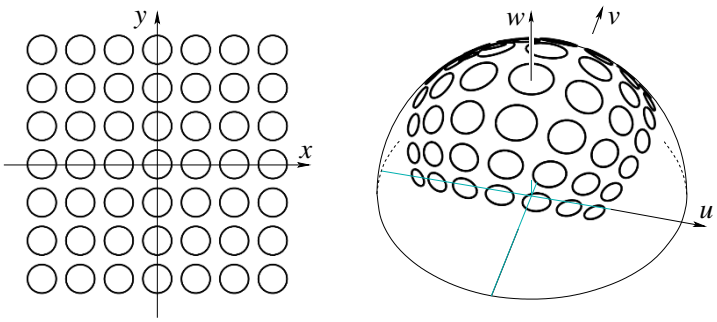


Fig. 10.9 Uncertainty fields: The homogeneous and isotropic uncertainty field in the real plane of an image is mapped to an inhomogeneous anisotropic uncertainty field on the unit sphere

◇

Representation of the Standard Ellipse. The standard ellipse is a conic, where its centre is the mean of the uncertain point. Given the mean $\boldsymbol{\mu}_x$ and the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{xx}}$ of the homogeneous coordinates of an uncertain point, in contrast to the Euclidean representation (10.12), p. 366, the standard ellipse is represented by *standard ellipse*

$$\mathbf{x}^\mathsf{T} \left(\boldsymbol{\Sigma}_{\mathrm{xx}} - \boldsymbol{\mu}_\mathrm{x}\boldsymbol{\mu}_\mathrm{x}^\mathsf{T}\right)^\circ \mathbf{x} = 0\,, \tag{10.20}$$

(see Ochoa and Belongie (2006) and Meidow et al. (2009)), where the $^\circ$ indicates the cofactor matrix, see (A.19), p. 769. This representation may be useful for plotting the standard ellipse.

Reduced Coordinates for Minimal Representation of Uncertain Homogeneous Coordinate Vectors. Singular covariance matrices cause problems during estimation or when determining the Mahalanobis distance of two entities, as they require the inverse covariance matrix. We therefore develop a representation of a spherically normalized homogeneous vector $\mathbf{x}^s$ with a regular covariance matrix. This can be achieved by representing the uncertainty in the two-dimensional tangent space of $\mathbf{x}^s$ (see Åström, 1998). We choose a basis for the two-dimensional tangent space at $\boldsymbol{\mu}_{\mathrm{x}^s}$, see Fig. 10.10. It is the

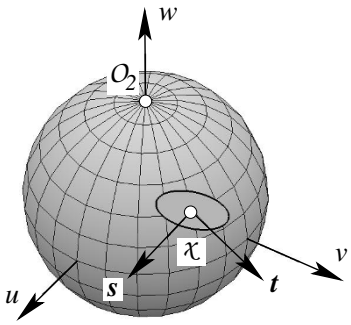


Fig. 10.10 Reduced coordinates for representing an uncertain point $\mathcal{X}(\underline{\mathbf{x}}^s)$ on the unit sphere S^2, which represents the projective plane $\mathbb{P}^2$. A point with mean $\boldsymbol{\mu}_{\mathbf{x}^s}$, which is uncertain on the unit sphere, is projected into the tangent plane at the mean. Its uncertainty in the tangent space, which is the null space of $\boldsymbol{\mu}_x^\mathsf{T}$ and spanned by two basis vectors, say $\boldsymbol{s}$ and $\boldsymbol{t}$, has only two degrees of freedom and leads to a regular 2×2 covariance matrix (the ellipse shown in the figure) of the 2-vector $\boldsymbol{x}_r$ of the reduced coordinates in the tangent plane

null space of the vector $\boldsymbol{\mu}_\mathbf{x}^\mathsf{T}$,

$$[\boldsymbol{s}, \boldsymbol{t}] = \mathsf{null}(\boldsymbol{\mu}_\mathbf{x}^\mathsf{T}) \,. \tag{10.21}$$

This representation of the null space is not unique, as any 2D rotation R_2 leads to a valid null space, $[\boldsymbol{s}, \boldsymbol{t}]\, R_2$.[3] For the moment, we assume the basis $[\boldsymbol{s}, \boldsymbol{t}]$ is fixed. We also use only $\mathbf{x}$ as an argument, as the null space is invariant to scaling of $\mathbf{x}$ with a positive factor.

We now define a random 2-vector $\underline{\boldsymbol{x}}_r$ in the tangent space at $\boldsymbol{\mu}_{\mathbf{x}^s}$,

$$\underline{\boldsymbol{x}}_r \sim \mathcal{M}(\mathbf{0}, \Sigma_{x_r x_r}) \,. \tag{10.22}$$

reduced coordinates

The 2-vector $\underline{\boldsymbol{x}}_r$ has mean $\mathbf{0}$ and covariance matrix $\Sigma_{x_r x_r}$. These coordinates are called *reduced coordinates* in the following. They have been proposed by Förstner (2010a, 2012) and are equivalent to the *local coordinates* in Absil et al. (2008, Sect. 4.1.3), though not used there for optimization.

The uncertain spherically normalized point vector can now be represented using the vector $\mathbf{x}^t$,

$$\underline{\mathbf{x}}^t = \boldsymbol{\mu}_{\mathbf{x}^s} + [\boldsymbol{s}, \boldsymbol{t}]\underline{\boldsymbol{x}}_r = \boldsymbol{\mu}_{\mathbf{x}^s} + \boldsymbol{s}\underline{x}_{r1} + \boldsymbol{t}\underline{x}_{r2} \,, \tag{10.23}$$

in the tangent space with subsequent spherical normalization,

$$\underline{\mathbf{x}}^s(\underline{\boldsymbol{x}}_r) = \mathsf{N}(\boldsymbol{\mu}_{\mathbf{x}^s} + [\boldsymbol{s}, \boldsymbol{t}]\underline{\boldsymbol{x}}_r) \,. \tag{10.24}$$

Jacobian $J_r(\cdot)$ of reducing homogeneous coordinates

We therefore have the 3×2 Jacobian of a spherically homogeneous 3-vector $\mathbf{x}^s$ with respect to the 2-vector of the reduced coordinates $\boldsymbol{x}_r$,

$$J_r(\boldsymbol{\mu}_x) \doteq \left. \frac{\partial \mathbf{x}^s}{\partial \boldsymbol{x}_r} \right|_{x=\mu_x} = \mathsf{null}\left(\boldsymbol{\mu}_x^\mathsf{T}\right) . \tag{10.25}$$

We explicitly have

$$\mathrm{d}\mathbf{x}^s = J_r(\boldsymbol{\mu}_x)\, \mathrm{d}\mathbf{x}_r \,, \tag{10.26}$$

a relation which we will use regularly when estimating homogeneous vectors, and which will be generalized for homogeneous vectors obeying more than one constraint.

Remark: Observe, the Jacobian $J_r(\boldsymbol{\mu}_x)$ spans the same space as the skew symmetric matrix $\mathsf{S}(\boldsymbol{\mu}_x)$. Thus the null space of $\boldsymbol{\mu}_x^\mathsf{T}$ could also be represented by selecting two independent columns of $\mathsf{S}(\boldsymbol{\mu}_x)$, thus

[3] The null space is the space orthogonal to the argument. Here we refer to the null space as an orthonormal matrix with the columns spanning the null space. This representation is not unique, and depends on the linear algebra package used; see Sect. A.11, p. 777.

by the matrix $\mathsf{S}^{(s)}$, see Sect. 7.4.1, p. 317. Since $\boldsymbol{\mu}_x^\mathsf{T}$ is the Jacobian of the constraint $g = \boldsymbol{\mu}_x^\mathsf{T}\boldsymbol{\mu} - 1$, the method with the null space generalizes more easily, especially to Plücker coordinates of 3D lines. ◇

The Jacobian J_r has the following properties:

$$J_r^\mathsf{T} J_r = I_2 \qquad J_r J_r^\mathsf{T} = I_3 - \mathbf{x}^s \mathbf{x}^{s\mathsf{T}} . \tag{10.27}$$

In the following, we will name all Jacobians of a homogeneous entity with respect to their reduced coordinates $J_r(\cdot)$, where the argument influences the definition.

We obtain the covariance matrices of $\underline{\mathbf{x}}^s$ as a function of the covariance matrix of the reduced coordinates $\underline{\boldsymbol{x}}_r$ from (10.24) (Hartley and Zisserman, 2000, Eq. (5.9)),

$$\Sigma_{\mathrm{x}^s\mathrm{x}^s} = J_r(\boldsymbol{\mu}_\mathrm{x})\, \Sigma_{x_r x_r}\, J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x}) . \tag{10.28}$$

Given spherically normalized coordinates, multiplying (10.23) with $J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})$ leads to an explicit expression for the reduced coordinates,

$$\underline{\boldsymbol{x}}_r = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})\, \underline{\mathbf{x}}^t \approx J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})\, \underline{\mathbf{x}}^s ; \tag{10.29}$$

the approximation is valid up to first-order terms of the Taylor expansion. This yields the inverse relation to (10.28), namely

$$\Sigma_{x_r x_r} = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})\, \Sigma_{x^s x^s}\, J_r(\boldsymbol{\mu}_\mathrm{x}) , \tag{10.30}$$

which establishes a one-to-one relation between the covariance matrices of the spherically normalized coordinates and the reduced coordinates. When using reduced coordinates, we represent an uncertain 2D point by its spherically normalized homogeneous coordinates and the corresponding covariance matrix of the reduced coordinates,

$$\mathcal{X} : \{\mathbf{x}^s, \Sigma_{x_r x_r}\} . \tag{10.31}$$

We will use reduced coordinates regularly, as they allow easy testing and estimation.

For the determination of the null space see A.11, p. 777.

Transforming Uncertain Homogeneous to Euclidean Vectors. Let an uncertain point be given by its homogeneous coordinates $\boldsymbol{\mu}_\mathrm{x} = [\boldsymbol{\mu}_{\mathrm{x}_0}^\mathsf{T}, \mu_{\mathrm{x}_h}]^\mathsf{T}$ and the associated covariance matrix Σ_{xx}; then the uncertain Euclidean point is determined by

$$\{\boldsymbol{\mu}_x, \Sigma_{xx}\} = \left\{ \frac{\boldsymbol{\mu}_{\mathrm{x}_0}}{\mu_{\mathrm{x}_h}}, J_{x\mathrm{x}}^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x}) \Sigma_{\mathrm{xx}} J_{x\mathrm{x}}(\boldsymbol{\mu}_\mathrm{x}) \right\} , \tag{10.32}$$

with the Jacobian

$$J_{x\mathrm{x}}^\mathsf{T}(\mathbf{x}) = \frac{\partial \boldsymbol{x}}{\partial \mathbf{x}} = \frac{1}{x_h^2} [x_h I_2 \mid -\boldsymbol{x}_0] \tag{10.33}$$

to be evaluated at the mean $\boldsymbol{\mu}_\mathrm{x}$.

Remark: Similarly to above, see Fig. 10.9, we illustrate the effect of Euclidean normalization on the uncertainty structure of a field of image points. We now assume that the normalized direction vectors $\mathbf{x}^s$ have homogeneous and isotropic uncertainty, see Sect. 4.6.2.3, p. 121, the standard ellipses on the unit sphere are circles with some radius σ_α indicating the directional uncertainty. This is a reasonable model for the stochastic properties of image rays of omnidirectional cameras. When normalizing these vectors Euclideanly, we obtain a radially symmetric uncertainty field, where the standard ellipses increase and become more elongated with increasing distance from the origin, see 10.11. For cameras with large viewing angles, the uncertainty of the directions cannot be reliably represented using Euclideanly normalized coordinates. ◇

Synopsis of Representations for an Uncertain 2D Point. The following Table 10.1 collects the representations of an uncertain 2D point derived in the previous sections. Given are a sample value, possibly constraints it has to fulfil, its covariance matrix,

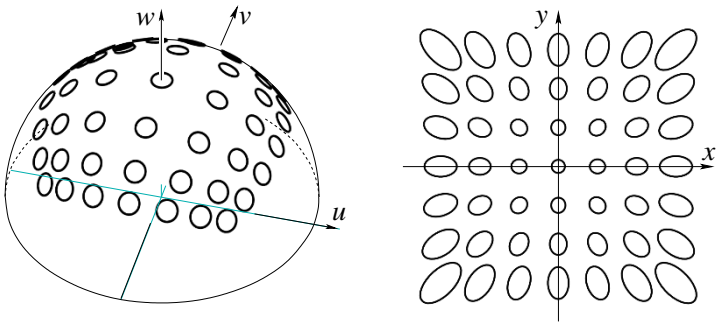


Fig. 10.11 Uncertainty fields: The homogeneous and isotropic uncertainty field on the unit sphere is mapped to an inhomogeneous anisotropic uncertainty field on a real image plane

and possibly its null space. These representations all have their particular roles within geometric reasoning with uncertain entities.

Table 10.1 Representations of an uncertain 2D point

name	sample value	constraint	cov. matrix	null space
Euclidean	$\boldsymbol{x} \in \mathbb{R}^2$	–	Σ_{xx}	$\emptyset$
homogeneous	$\mathbf{x} \in \mathbb{P}^2$	–	$\Sigma_{\mathbf{x}\mathbf{x}}$	–
- Euclideanly normalized	$\mathbf{x}^e \in \mathbb{P}^2$	$x_3 = 1$	$\Sigma_{\mathbf{x}^e\mathbf{x}^e}$	$\boldsymbol{e}_3$
- spherically normalized	$\mathbf{x}^s \in \mathbb{P}^2$	$\lvert\mathbf{x}^s\rvert = 1$	$\Sigma_{\mathbf{x}^s\mathbf{x}^s}$	$\mathbf{x}^s$
- minimal representation	$\mathbf{x}^s \in \mathbb{P}^2$	$\lvert\mathbf{x}^s\rvert = 1$	$\Sigma_{x_r x_r}$	$\emptyset$

- The Euclidean representation $\mathcal{N}(\boldsymbol{x}, \Sigma_{xx})$ is needed for modelling the observation process or for presenting the result of a reasoning or estimation process. No points at infinity can be represented.
- The homogeneous representation $\mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{x}\mathbf{x}})$ with no constraints on the sample vector appears as a result of constructions. The covariance matrix may have full rank if no constraints are imposed during reasoning or estimation. It is in no way unique, due to free scaling and due to the freedom in choosing the uncertainty of the scaling, see the discussion below.
- The Euclideanly normalized homogeneous representation $\mathcal{N}(\mathbf{x}^e, \Sigma_{\mathbf{x}^e\mathbf{x}^e})$ is used as an interface between the Euclidean representation and the other homogeneous representations. It is unique, but cannot represent elements at infinity.
- The spherically normalized homogeneous representation $\mathcal{N}(\mathbf{x}^s, \Sigma_{\mathbf{x}^s\mathbf{x}^s})$ is the only one which allows us to represent elements at infinity and is unique (up to the sign of the homogeneous vector).
- The minimal representation $\mathcal{N}(\mathbf{x}^s, \Sigma_{x_r x_r})$ integrates the spherically normalized homogeneous representation for the sample and the full rank representation of the Euclidean representation. Here it is assumed that the null space is determined algebraically as a function of $\mathbf{x}^s$. As the covariance matrix $\Sigma_{x_r x_r}$ generally has full rank, this representation will be used for testing and estimation.

We will find these representations for all geometric entities, including transformations.

10.2.2.2 Uncertain 3D Points

Uncertain 3D points are represented with their uncertain Euclidean coordinates $\boldsymbol{X}$ and the corresponding 3×3 covariance matrix,

$$\mathcal{X}: \quad \{\boldsymbol{\mu}_X, \Sigma_{XX}\}. \tag{10.34}$$

Uncertain Euclideanly normalized homogeneous coordinates can be rigorously derived:

$$\{\boldsymbol{\mu}_X, \Sigma_{XX}\} \quad \rightarrow \quad \{\boldsymbol{\mu}_{\mathrm{X}^e}, \Sigma_{\mathrm{X}^e\mathrm{X}^e}\} = \left\{ \begin{bmatrix} \boldsymbol{\mu}_X \\ 1 \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \right\}. \tag{10.35}$$

Again, the covariance matrix is singular with rank 3, corresponding to the number of degrees of freedom of a 3D point, and the null space is $\boldsymbol{e}_4^{[4]}$.

In the case of a small directional uncertainty of an uncertain homogeneous vector $\underline{\mathbf{X}} \sim M(\boldsymbol{\mu}_\mathrm{X}, \Sigma_{\mathrm{XX}})$, the uncertain spherically normalized homogeneous vector approximately is distributed according to

$$\underline{\mathbf{X}}^s \sim \mathcal{M}\left(\frac{\boldsymbol{\mu}_\mathrm{X}}{|\boldsymbol{\mu}_\mathrm{X}|}, J_{\mathrm{X}^s\mathrm{X}}(\boldsymbol{\mu}_\mathrm{X})\, \Sigma_{\mathrm{XX}}\, J_{\mathrm{X}^s\mathrm{X}}^\mathsf{T}(\boldsymbol{\mu}_\mathrm{X}) \right), \tag{10.36}$$

with the 4×4 Jacobian

$$J_{\mathrm{X}^s\mathrm{X}}(\mathbf{X}) = \frac{\partial \mathbf{X}^s}{\partial \mathbf{X}} = \frac{1}{|\mathbf{X}|}\left(I_4 - \frac{\mathbf{X}\mathbf{X}^\mathsf{T}}{\mathbf{X}^\mathsf{T}\mathbf{X}} \right) \tag{10.37}$$

evaluated at $\mathbf{X} = \boldsymbol{\mu}_\mathrm{X}$.

The reduced homogeneous coordinates $\underline{\boldsymbol{X}}_r \in \mathbb{R}^3$ of an uncertain 3D point $\mathcal{X}(\underline{\mathbf{X}}^s)$ are given by

$$\underline{\boldsymbol{X}}_r \sim M(\mathbf{0}, \Sigma_{X_rX_r}), \tag{10.38}$$

with the covariance matrix resulting from

$$\Sigma_{X_rX_r} = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{X})\, \Sigma_{\mathrm{X}^s\mathrm{X}^s}\, J_r(\boldsymbol{\mu}_\mathrm{X}). \tag{10.39}$$

The 4×3 Jacobian $J_r(\mathbf{X})$ is

$$J_r(\mathbf{X}) = \frac{\partial \mathbf{X}}{\partial \boldsymbol{X}_r} = \mathsf{null}(\mathbf{X}^\mathsf{T}). \tag{10.40}$$

The three columns of the orthonormal matrix $J_r(\boldsymbol{\mu}_{\mathrm{X}^s})$ span the three-dimensional tangent space at the three-dimensional unit sphere S^3 representing the projective space $\mathbb{P}^3$. For given $\boldsymbol{\mu}_{\mathrm{X}^s}$ and $\Sigma_{X_rX_r}$, this allows us to derive the covariance matrix of the spherically normalized vector $\underline{\mathbf{X}}^s$,

$$\Sigma_{\mathrm{X}^s\mathrm{X}^s} = J_r(\boldsymbol{\mu}_\mathrm{X})\, \Sigma_{X_rX_r} J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{X}). \tag{10.41}$$

Finally, deriving the Euclidean coordinates of an uncertain 3D point from homogeneous ones is achieved with

$$\{\boldsymbol{\mu}_X, \Sigma_{XX}\} = \left\{ \frac{\boldsymbol{\mu}_{\mathrm{X}_0}}{\mu_{\mathrm{X}_h}}, J^\mathsf{T}(\boldsymbol{\mu}_\mathrm{X})\, \Sigma_{\mathrm{XX}}\, J(\boldsymbol{\mu}_\mathrm{X}) \right\}, \tag{10.42}$$

with the Jacobian J, the matrix of derivatives of the inhomogeneous $\boldsymbol{X}$ to the homogeneous vector $\mathbf{X}$,

$$J \doteq J_{X\mathrm{X}}^\mathsf{T}(\mathbf{X}) = \frac{\partial \boldsymbol{X}}{\partial \mathbf{X}} = \frac{1}{X_h^2}\left[X_h I_3 \mid -\boldsymbol{X}_0 \right], \tag{10.43}$$

to be evaluated at the mean $\boldsymbol{\mu}_\mathrm{X}$.

The discussion of the different representations of an uncertain 2D point can directly be transferred to the representation of an uncertain 3D point.

10.2.2.3 Uncertain 2D Line and Plane

Euclidean Representation of an Uncertain 2D Line. The uncertainty of a 2D line can be investigated with any line representation. We will discuss the Hessian representation

and what is called the *centroid form*, a representation which makes the uncertainty of the line explicitly visible.

We start with the Hessian parameters $\boldsymbol{\mu}_h = [\mu_\phi, \mu_d]^\mathsf{T}$, see Sect. 5.1.2.1, p. 197. The uncertain line is then represented by

$$\mathcal{l}: \quad \{\boldsymbol{\mu}_h, \Sigma_{hh}\} = \left\{ \begin{bmatrix} \mu_\phi \\ \mu_d \end{bmatrix}, \begin{bmatrix} \sigma_\phi^2 & \rho_{\phi d}\sigma_\phi\sigma_d \\ \rho_{\phi d}\sigma_\phi\sigma_d & \sigma_d^2 \end{bmatrix} \right\}. \tag{10.44}$$

with the covariance $\sigma_{\phi d}$ of ϕ and d depending on the correlation coefficient $\rho_{\phi d}$, which in general is not 0.

We again represent the uncertainty by a standard ellipse, now in the space of the two line parameters ϕ and d. Each of the points of this ellipse represents a line in the xy-plane. The envelope of all these lines can be shown to be a hyperbola, see Fig. 10.12 and Peternell and Pottmann (2001, Theorem 3). All lines whose parameters lie within the standard ellipse lie within the area bounded by the two parts of the *standard hyperbola*.

Exercise 10.6

standard hyperbola

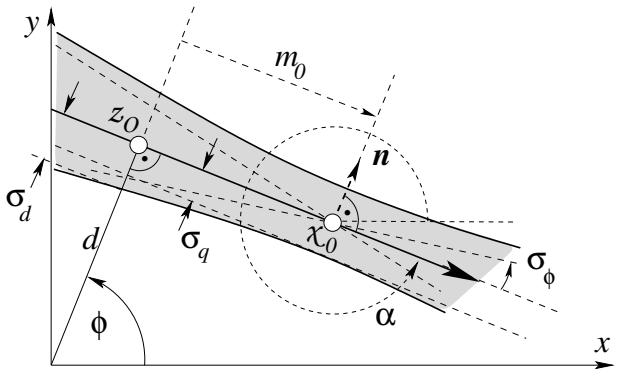


Fig. 10.12 Uncertain 2D line and its representation in the real plane. Centroid χ_0, direction α of the line, direction ϕ of the normal $\boldsymbol{n}$, $\phi = \alpha + \pi/2$, distance d to the origin, foot point z_O of the origin, distance m_0 of the centroid from the foot point, standard deviations $\sigma_\alpha = \sigma_\phi$, σ_d, and σ_q: of the direction α, the distance d and the position of the centroid across the line, respectively. The standard deviation σ_ϕ is visualized as the angle between the (mean) line and one of the asymptotic lines of the hyperbola

The figure visualizes important characteristics of the uncertain 2D line.

- The standard deviation of a point on the line measured across the line is bounded by the hyperbola. For an arbitrary point $\boldsymbol{x}$, we obtain the uncertainty of its distance to the line from $\underline{d}_x = x\cos\underline{\phi} + y\sin\underline{\phi} - \underline{d}$. Its variance is $\sigma_{d_x}^2 = (-x\sin\phi + y\cos\phi)^2\sigma_\phi^2 - 2(-x\sin\phi + y\cos\phi)\sigma_{\phi d} + \sigma_d^2$.
  If the angular uncertainty is not zero, there exists a unique point on the line, the centroid χ_0, for which this uncertainty is lowest, namely σ_q. With the distance $m = x\sin\phi - y\cos\phi$ of the point along the line, counted from the foot point z_O, we obtain the m for which the uncertainty $\sigma_{d_x}(m)$ is smallest, namely

$$m_0 = -\frac{\sigma_{\phi d}}{\sigma_\phi^2}, \tag{10.45}$$

  with the minimum variance across the line

$$\sigma_q^2 = \sigma_d^2 - \frac{\sigma_{\phi d}^2}{\sigma_\phi^2}. \tag{10.46}$$

  The point χ_0 has the coordinates

$$\boldsymbol{x}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} m_0 \\ d \end{bmatrix} \tag{10.47}$$

with the direction $\alpha = \phi - \pi/2$ of the line. We therefore have the *centroid representation* of an uncertain line,

centroid representation of 2D line

$$\ell : \quad \{\boldsymbol{x}_0, \alpha; \sigma_q, \sigma_\alpha\} \,. \tag{10.48}$$

In this representation the two uncertain elements, the position across the line and its direction, are statistically independent, as if the coordinate system is centred, the foot point $\boldsymbol{z}_0$ is at the origin and $m_0 = 0$ in (10.45).

- It is not reasonable to characterize the uncertainty of a line with one single value, as we did for the 2D point, since the direction ϕ and the distance d have different units. Also, in practice the standard deviation σ_ϕ will usually be much smaller numerically than the standard deviation σ_d. They only become comparable when the data are conditioned, e.g., when a straight line in an image is represented in units of the image diameter.

Transition to Uncertain Homogeneous Line Parameters. We start with the transition from the Hessian normal form $\ell(\boldsymbol{\mu}_h, \Sigma_{hh})$ in (10.44) to *Euclideanly normalized* homogeneous coordinates of an *uncertain line*. As the transition to homogeneous coordinates is nonlinear, we can only derive their covariance matrix approximately. By variance propagation, we obtain the first-order approximation

$$\{\boldsymbol{\mu}_h, \Sigma_{hh}\} \quad \to \quad \{\boldsymbol{\mu}_{\mathrm{l}^e}, \Sigma_{\mathrm{l}^e\mathrm{l}^e}\}\,, \tag{10.49}$$

with

$$\boldsymbol{\mu}_{l^e} = \begin{bmatrix} \cos\mu_\phi \\ \sin\mu_\phi \\ -\mu_d \end{bmatrix}, \qquad \Sigma_{\mathrm{l}^e\mathrm{l}^e} = J_{\mathrm{l}^e h}(\boldsymbol{\mu}_h)\Sigma_{hh}J_{\mathrm{l}^e h}(\boldsymbol{\mu}_h)^\mathsf{T} \quad \text{with} \quad J_{\mathrm{l}^e h}(\boldsymbol{h}) = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & -1 \end{bmatrix}, \tag{10.50}$$

where the Jacobian is to be evaluated at the mean vector. The covariance matrix is singular with rank 2 since the Jacobian has rank 2, again corresponding to the two degrees of freedom of a 2D line. Its null space is

$$\mathsf{null}\,(\Sigma_{\mathrm{l}^e\mathrm{l}^e}) = \begin{bmatrix} \cos\mu_\phi \\ \sin\mu_\phi \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{\mathrm{l}_h} \\ 0 \end{bmatrix}. \tag{10.51}$$

As the line vector is Euclideanly normalized, it lies on a straight circular unit cylinder with the c-axis as the cylinder axis, see Fig. 10.13. The flat standard ellipsoid lies on the tangent plane to that cylinder, centred at $\boldsymbol{\mu}_{\mathrm{l}^e}$. Connecting the points of this ellipse with the origin results in an elliptic cone. This cone visually represents the directional uncertainty of the homogeneous vector $\underline{\mathrm{l}}^e$.

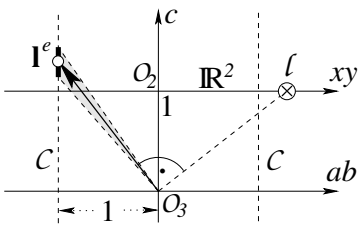


Fig. 10.13 Uncertain 2D line with Euclideanly normalized homogeneous coordinates $\mathrm{l}^e = [\cos\phi, \sin\phi, -d]^\mathsf{T} = [a, b, c]^\mathsf{T}$. The line ℓ lies on the xy-plane $\mathbb{R}^2$. It is perpendicular to the drawing plane and directed away from the reader, indicated by the crossed circle at ℓ. The vector l^e is the normal on the plane through ℓ and O_3. The point l^e lies on the unit cylinder $\mathcal{C}(a^2 + b^2 = 1)$ resulting from the Euclidean normalization. The uncertainty of the line ℓ across the line is represented by the uncertainty of $c = -d$ along the cylinder. The uncertainty of the direction of the line is represented by the uncertainty of $[a, b] = [\cos\phi, \sin\phi]$ in the tangent plane at the cylinder

Spherically Normalized 2D Line Coordinates. Spherically normalized lines can be achieved similarly to (10.19) by

$$\underline{\mathbf{l}}^s \sim \mathcal{M}(\boldsymbol{\mu}_{l^s}, \boldsymbol{\Sigma}_{l^s l^s}) = \mathcal{M}\left(\frac{\boldsymbol{\mu}_l}{|\boldsymbol{\mu}_l|}, J_{l^s l}(\boldsymbol{\mu}_l)\, \boldsymbol{\Sigma}_{ll}\, J_{l^s l}^{\mathsf{T}}(\boldsymbol{\mu}_l)\right), \tag{10.52}$$

with the Jacobian

$$J_{l^s l}(\mathbf{l}) = \frac{1}{|\mathbf{l}|}\left(I_3 - \frac{\mathbf{l}\mathbf{l}^{\mathsf{T}}}{\mathbf{l}^{\mathsf{T}}\mathbf{l}}\right) \tag{10.53}$$

evaluated at the mean $\boldsymbol{\mu}_{l^s}$. The resulting vector $\mathbf{l}^s$ lies on the unit sphere representing the dual projective plane, see Fig. 10.14. The covariance matrix $\boldsymbol{\Sigma}_{l^s l^s}$ is singular with

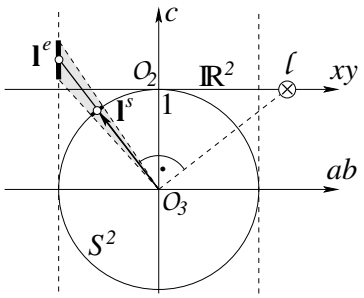


Fig. 10.14 Uncertain 2D line with spherically normalized homogeneous coordinates $\mathbf{l}^s$. As in Fig. 10.13 the line lies in the plane $\mathbb{R}^2$ and points into the drawing plane. The homogeneous coordinate vector $\mathbf{l}^s$ is the unit normal on the plane through ℓ and O_3. It lies on the sphere

rank 2 and null space $\mathbf{l}^s$, indicated in Fig. 10.14 by the flat standard ellipse lying on the tangent plane in $\mathbf{l}^s$ at S^2. Again, the derivation of the parameters of the distribution is approximate, omitting higher terms.

Representation of the Standard Hyperbola. The standard ellipse is a conic with its centre, which is the mean of the centroid $\mathbf{x}_0$. Given the mean $\boldsymbol{\mu}_l$ and the covariance matrix $\boldsymbol{\Sigma}_{ll}$ of the homogeneous coordinates of an uncertain line, the standard hyperbola is represented by

$$\mathbf{x}^{\mathsf{T}}\left(\boldsymbol{\Sigma}_{ll} - \boldsymbol{\mu}_l \boldsymbol{\mu}_l^{\mathsf{T}}\right)\mathbf{x} = 0 \tag{10.54}$$

(see Meidow et al., 2009). Thus the term in brackets is a conic matrix, here of a hyperbola.

Reduced Homogeneous Coordinates of an Uncertain 2D Line. The reduced homogeneous coordinates $\underline{\boldsymbol{l}}_r$ of the uncertain 2D line parameters $\underline{\mathbf{l}}^s$ are defined in the tangent space of S^2 at $\boldsymbol{\mu}_{l^s}$ in full equivalence to 2D points, see Sect. 10.2.2.1. Thus, an uncertain 2D line is represented as

$$\underline{\mathbf{l}}^s(\underline{\boldsymbol{l}}_r) = \mathsf{N}(\boldsymbol{\mu}_{l^s} + J_r(\boldsymbol{\mu}_{l^s})\underline{\boldsymbol{l}}_r)\,. \tag{10.55}$$

This allows us to derive the covariance matrix of $\underline{\mathbf{l}}^s$ for given $\boldsymbol{\mu}_{l_r}$ and $\boldsymbol{\Sigma}_{l_r l_r}$,

$$\boldsymbol{\Sigma}_{l^s l^s} = J_r(\boldsymbol{\mu}_{l^s})\, \boldsymbol{\Sigma}_{l_r l_r}\, J_r^{\mathsf{T}}(\boldsymbol{\mu}_{l^s})\,, \tag{10.56}$$

with the Jacobian $J_r(\mathbf{l}) = \mathsf{null}(\mathbf{l}^{\mathsf{T}})$ now evaluated at $\boldsymbol{\mu}_{l^s}$. The inverse relation therefore is

$$\boldsymbol{\Sigma}_{l_r l_r} = J_r^{\mathsf{T}}(\boldsymbol{\mu}_{l^s})\, \boldsymbol{\Sigma}_{l^s l^s}\, J_r(\boldsymbol{\mu}_{l^s})\,. \tag{10.57}$$

Transition to Uncertain Hessian Parameters. If the uncertain line is given by uncertain homogeneous parameters, thus by $(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_{ll})$, we determine the uncertain Hessian parameters $\underline{\boldsymbol{h}} = [\underline{\phi}, \underline{d}]^{\mathsf{T}}$, Sect. 10.2.2.3, from

$$\{\boldsymbol{\mu}_l, \Sigma_{ll}\} \quad \rightarrow \quad \{\boldsymbol{\mu}_h, \Sigma_{hh}\} = \left\{ \begin{bmatrix} \operatorname{atan2}(b,a) \\ -\dfrac{c}{\sqrt{a^2+b^2}} \end{bmatrix}, J_{hl}(\boldsymbol{\mu}_l)\Sigma_{ll}J_{hl}^{\mathsf{T}}(\boldsymbol{\mu}_l) \right\}, \tag{10.58}$$

with the Jacobian

$$J_{hl}(\mathbf{l}) = \frac{\partial \mathbf{h}}{\partial \mathbf{l}} = \frac{1}{s^3}\begin{bmatrix} -bs & as & 0 \\ ac & bc & -s^2 \end{bmatrix} \qquad s = \sqrt{a^2+b^2} \tag{10.59}$$

evaluated at the mean value $\boldsymbol{\mu}_l$ of $\mathbf{l} = [a, b, c]^{\mathsf{T}}$.

Synopsis of Representations for Uncertain 2D lines. The following Table 10.2 collects the representations of an uncertain 2D line. Given are a sample value, possibly constraints it has to fulfil, its covariance matrix, and possibly its null space. These representations all have their role within geometric reasoning with uncertain entities.

Table 10.2 Representations of an uncertain 2D line

name	sample value	constraint	cov. matrix	null space
Hessian	$\boldsymbol{h} \in \mathbb{R}^2$	–	Σ_{hh}	$\emptyset$
centroid	$\boldsymbol{x}_0 \in \mathbb{R}^2$, $\alpha \in [0, 2\pi)$	–	σ_q, σ_α	$\emptyset$
homogeneous	$\mathbf{l} \in \mathbb{P}^2$	–	Σ_{ll}	–
- Euclideanly normalization	$\mathbf{l}^e \in \mathbb{P}^2$	$\lvert\boldsymbol{l}_h\rvert = 1$	$\Sigma_{l^e l^e}$	$[\boldsymbol{l}_h^{\mathsf{T}}, 0]^{\mathsf{T}}$
- spherically normalization	$\mathbf{l}^s \in \mathbb{P}^2$	$\lvert\mathbf{l}^s\rvert = 1$	$\Sigma_{l^s l^s}$	$\mathbf{l}^s$
- minimal representation	$\mathbf{l}^s \in \mathbb{P}^2$	$\lvert\mathbf{l}^s\rvert = 1$	$\Sigma_{l_r l_r}$	$\emptyset$

- The *Hessian normal form* can represent all uncertain lines not at infinity. The representation is unique; thus, we can distinguish between the two different orientations. The numerical stability of the representation depends on the correlation $\rho_{\phi d}$, thus on the distance of the centroid from the foot point in relation to the distance of the line from the origin. Therefore, conditioning following Sect. 6.9, p. 286 is recommended.
- The *centroid form* can also also represent all lines not at infinity. In contrast to the Hessian normal form, it is numerically stable, which is due to the zero correlation between the direction α and the position q across the line. The centroid form is the one which naturally arises during image processing when extracting straight line segments.
- The properties of the homogeneous line representations are the same as for 2D points, see Sect. 10.2.2.3, p. 377.

10.2.2.4 Uncertain Plane

The uncertainty of the plane can be visualized by a standard hyperboloid similarly to visualizing the uncertainty of a 2D line by the standard hyperbola in Fig. 10.12, p. 374. The hyperboloid is the envelope of all planes on the 3D standard ellipsoid of the three plane parameters, say the three coordinates of the foot point of the origin on the plane. It is an elliptical hyperboloid of two sheets, see Fig. 10.15. There exists a point $\mathcal{X}_0$ on the plane where the uncertainty perpendicular to the plane is minimal. This is the centre of the hyperboloid. The uncertainty of the orientation of the plane is modelled by the uncertainty of two angles around two mutually perpendicular axes, $\mathcal{L}_1$ and $\mathcal{L}_2$, passing through $\mathcal{X}_0$. This angular uncertainty will generally be different for the two axes.

For representing an uncertain plane, we start from the centroid form, as it naturally results from a best fitting plane through a set of given points. The centroid form of a plane is best given by

Exercise 10.28

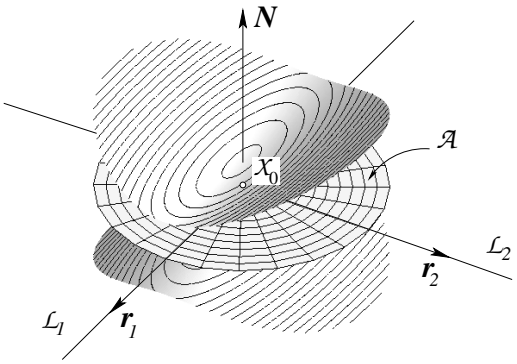


Fig. 10.15 Uncertain plane. The mean plane is shown as a circular disc in the middle, containing the centre point $\mathcal{X}_0$ of the standard hyperboloid of two sheets indicating the uncertainty perpendicular to the plane. At the centre point $\mathcal{X}_0$, the uncertainty across the plane is smallest. The mutually perpendicular 3D lines $\mathcal{L}_1$ and $\mathcal{L}_2$ through $\mathcal{X}_0$ are the axes of maximal and minimal rotational uncertainty of the plane. Isolines of uncertainty perpendicular to the plane are ellipses with their large semi-axis in the direction of $\mathcal{L}_1$

centroid representation of plane

- The centroid $\boldsymbol{X}_0$.
- The local coordinate system at $\boldsymbol{X}_0$ represented by a rotation matrix

$$R = [\boldsymbol{r}_1 \mid \boldsymbol{r}_2 \mid \boldsymbol{N}] , \tag{10.60}$$

  where a point on the plane is given as a function of local plane coordinates $[x_1, x_2]^\mathsf{T}$,

$$\boldsymbol{X} = \boldsymbol{X}_0 + x_1\boldsymbol{r}_1 + x_2\boldsymbol{r}_2 . \tag{10.61}$$

  The plane has normal $\boldsymbol{N}$. The directions $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ give the two axes $\mathcal{L}_1$ and $\mathcal{L}_2$ with maximum and minimum angular uncertainties.
- The maximum and minimum standard deviations, σ_α and σ_β, of the normal. The maximum standard deviation, σ_α, belongs to the rotation around $\boldsymbol{r}_1$, which is the uncertainty of the direction $\boldsymbol{r}_2$ in the direction towards the normal. The minimum uncertainty is the rotation around $\boldsymbol{r}_2$, i.e., the uncertainty of the direction $\boldsymbol{r}_1$ towards the normal.
- The standard deviation σ_q of the position of $\boldsymbol{X}_0$ perpendicular to the plane. At the same time this is the minimum standard deviation perpendicular to the plane of a point on the plane.

Thus the centroid representation of a plane is given by

$$\mathcal{A}: \quad \{\boldsymbol{X}_0, R; \sigma_\alpha^2, \sigma_\beta^2, \sigma_q^2\} \qquad \sigma_\alpha \geq \sigma_\beta , \tag{10.62}$$

i.e., nine parameters specify an uncertain plane: three for the position of the centre $\mathcal{X}_0$, three for the rotation matrix R in some adequate representation, and three for the standard deviations, where we assume $\sigma_\alpha \geq \sigma_\beta$. Again, the three random variables α, β, and q are stochastically independent.

We now derive the covariance matrix $\Sigma_{\mathrm{A}^e\mathrm{A}^e}$ of the Euclideanly normalized plane parameters $\mathbf{A}^e = [\boldsymbol{N}^\mathsf{T}, -S]^\mathsf{T}$. This is relevant if individual planes are determined from a point cloud, thus obtained in centroid representation, and in a second step are tested for identity or used for constructing 3D lines or 3D points. For this, we refer to the local coordinate system in the centroid. The plane coordinates ${}^c\mathbf{A}^e$ centred at $\boldsymbol{X}_0$ are then given by ${}^c\mathbf{A}^e = \boldsymbol{e}_3^{[4]} = [0, 0, 1, 0]^\mathsf{T}$, as the normal points towards the local ${}^cZ$ axis. Its covariance matrix is

$${}^c\Sigma_{\mathrm{A}^e\mathrm{A}^e} = \mathrm{Diag}([\sigma_\alpha^2, \sigma_\beta^2, 0, \sigma_q^2]) . \tag{10.63}$$

The parameters $\mathbf{A}^e$ of $\mathcal{A}$ can be determined by moving ${}^c\mathbf{A}^e$ into $\mathbf{A}^e$ with

$$\mathsf{M}_c = \begin{bmatrix} R & \boldsymbol{X}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \tag{10.64}$$

yielding

$$\mathbf{A}^e = \mathsf{M}_c^{-\mathsf{T}}\, {}^c\mathbf{A}^e \tag{10.65}$$

and its covariance matrix Exercise 10.14

$$\Sigma_{\mathrm{A}^e\mathrm{A}^e} = \mathsf{M}_c^{-\mathsf{T}}\, {}^c\Sigma_{\mathrm{A}^e\mathrm{A}^e}\, \mathsf{M}_c^{-1}. \tag{10.66}$$

The null space is

$$\mathsf{null}(\Sigma_{\mathrm{A}^e\mathrm{A}^e}) = \begin{bmatrix} \boldsymbol{A}_h \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{N} \\ 0 \end{bmatrix}. \tag{10.67}$$

Euclidean and spherical normalization of homogeneous plane coordinates is similar to that of homogeneous 2D line coordinates. The covariance matrix $\Sigma_{A_rA_r}$ of the reduced coordinates $\underline{\boldsymbol{A}}_r$ is obtained using the Jacobian $J_r(\boldsymbol{\mu}_\mathrm{A}) = \mathsf{null}(\boldsymbol{\mu}_\mathrm{A}^\mathsf{T})$, now evaluated at the mean vector $\boldsymbol{\mu}_\mathrm{A}$ of the plane coordinates $\mathbf{A}$ and following (10.57). Equation (10.56) can be used to derive the covariance matrix of the spherically normalized plane coordinates from the covariance matrix of the reduced coordinates.

10.2.2.5 Uncertain 3D Line

Uncertainty of a 3D line. The uncertainty of a 3D line, see Sect. 5.4.1, p. 216ff., can be represented by a confidence region indicating the confidence regions of all points along the line, see Fig. 10.16. It is *not* a quadric, as can be seen easily from this counter

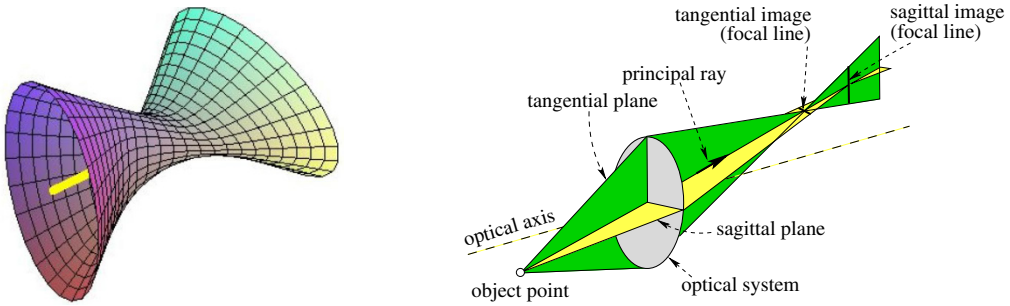


Fig. 10.16 Uncertain 3D line and astigmatism of an optical system. **Left:** Uncertain 3D line. **Right:** Light ray distribution of astigmatism, following Stewart (1996, Fig. 4.12, p. 58), see http://llis.nasa.gov/llis_lib/images/1009152main_0718-7.jpg

example: Let the 3D line $\mathcal{L}$ be given by two points $\mathcal{X}$ and $\mathcal{Y}$. Let the uncertainty of these two points be large across the joining line in directions which are different for each of the two points. Then the standard ellipses of these two points are thin, i.e., are elongated in these directions. This situation cannot be realized by a hyperboloid of one sheet, since if one cross section is a thin ellipse, all other cross sections also need to be thin, but in the same direction. We find this pattern in astigmatism, where in a first approximation, two differently oriented focal lines at different distances occur.

Euclidean and Spherical Normalization of an Uncertain 3D Line. The procedure for generating the 6×6 covariance matrix for the Euclideanly normalized Plücker coordinates $\mathbf{L}^e$ of a 3D line, as defined by, say, a 3D point and a 3D direction, uses estimation model C for constraints between the observations only (see Sect. 4.8.2.6, p. 170), indicated in the discussion on the null space of $\Sigma_{\mathrm{x}^e\mathrm{x}^e}$ after (10.17), p. 368. Since a 3D line has four degrees of freedom, we expect the covariance to have rank 4. We start from some line vector $\mathbf{L}$ fulfilling the Plücker constraint and having some covariance matrix Σ_{LL}, which need not have the correct rank.

Euclidean normalization yields

$$\underline{\mathbf{L}}^e = \frac{\underline{\mathbf{L}}}{|\boldsymbol{L}_h|}, \tag{10.68}$$

where we (in a first step) take the denominator as a fixed value. This leads to a scaling of the covariance matrix by the factor $1/|\boldsymbol{L}_h|^2$, which we will take into account later. For deriving the final covariance matrix of $\underline{\mathbf{L}}^e$ we need to enforce the normalization constraint. But we also need to apply the Plücker constraint in order to obtain a valid covariance matrix of rank 4. The two constraints are

$$\boldsymbol{g}(\mathbf{L}^e) = \begin{bmatrix} \frac{1}{2}(\boldsymbol{L}_h^{e\mathsf{T}}\boldsymbol{L}_h^e - 1) \\ \boldsymbol{L}_h^{e\mathsf{T}}\boldsymbol{L}_0^e \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{10.69}$$

Linearization yields the Jacobian

$$B^e = \begin{bmatrix} \boldsymbol{L}_h^e & \boldsymbol{L}_0^e \\ \mathbf{0} & \boldsymbol{L}_h^e \end{bmatrix}. \tag{10.70}$$

Using (4.467), p. 170, (4.59), p. 87, performing the estimation with $\Sigma_{ll} = I_6$, and taking the factor $1/|\boldsymbol{L}_h|^2$ into account yields the covariance matrix of the Euclideanly normalized 3D line,

$$\Sigma_{\mathrm{L}^e\mathrm{L}^e} = J_{\mathrm{L}^e L}\Sigma_{\mathrm{LL}}J_{\mathrm{L}^e L}^{\mathsf{T}} \quad \text{with} \quad J_{\mathrm{L}^e L} = \frac{1}{|\boldsymbol{L}_h|}(I_6 - B^e B^{e\mathsf{T}}). \tag{10.71}$$

This covariance matrix thus has null space

$$\mathsf{null}(\Sigma_{\mathrm{L}^e\mathrm{L}^e}) = B^e. \tag{10.72}$$

Spherical normalization of an uncertain line vector $\{\mathbf{L}, \Sigma_{\mathrm{LL}}\}$ with the normalized line vector $\mathbf{L}^s = \mathbf{L}/|\mathbf{L}|$ also leads to a 6×6 covariance matrix $\Sigma_{\mathrm{L}^s\mathrm{L}^s}$ which has rank 4. With $\overline{\mathbf{L}}^s = [\boldsymbol{L}_0^{s\mathsf{T}}, \boldsymbol{L}_h^{s\mathsf{T}}]^{\mathsf{T}}$, it has null space

$$\mathsf{null}\,(\Sigma_{\mathrm{L}^s\mathrm{L}^s}) = B^s = [\mathbf{L}^s \;\; \overline{\mathbf{L}}^s] = \begin{bmatrix} \boldsymbol{L}_h^s & \boldsymbol{L}_0^s \\ \boldsymbol{L}_0^s & \boldsymbol{L}_h^s \end{bmatrix}, \tag{10.73}$$

since now the normalization condition is $\frac{1}{2}(\mathbf{L}^{\mathsf{T}}\mathbf{L} - 1) = 0$. Thus we have the rank 4 covariance matrix

$$\Sigma_{\mathrm{L}^s\mathrm{L}^s} = J_{\mathrm{L}^s\mathrm{L}}\Sigma_{\mathrm{LL}}J_{\mathrm{L}^s\mathrm{L}}^{\mathsf{T}} \quad \text{with} \quad J_{\mathrm{L}^s\mathrm{L}} = \frac{1}{|\mathbf{L}|}(I_6 - B^s B^{s\mathsf{T}}). \tag{10.74}$$

Reduced Coordinates of a 3D Line. The transfer of the minimal representation already shown for points to 3D lines requires some care. The tangent space is four-dimensional, as two constraints on the six parameters of the Plücker coordinates have to be fulfilled. The tangent space is perpendicular to $\mathbf{L}$ and its dual is $\overline{\mathbf{L}}$, as $\mathbf{L}^{\mathsf{T}}\mathbf{L} - 1 = 0$ and $\overline{\mathbf{L}}^{\mathsf{T}}\mathbf{L} = 0$ hold. Therefore, the tangent space is given by the four columns of the 6×4 matrix

$$J_r(\mathbf{L}) = \frac{\partial \mathbf{L}^s}{\partial \boldsymbol{L}_r} = \mathsf{null}(B^{s\mathsf{T}}) = \mathsf{null}\left(\begin{bmatrix} \mathbf{L}^{\mathsf{T}} \\ \overline{\mathbf{L}}^{\mathsf{T}} \end{bmatrix}\right), \tag{10.75}$$

with orthonormal $J_r(\mathbf{L})$. We again define the reduced coordinates $\boldsymbol{L}_r$ of a 3D line $\mathcal{L}$ in this four dimensional space. With random perturbations of $\underline{\boldsymbol{L}}_r$ we have the general 6-vector

$$\underline{\mathbf{L}}^t(\boldsymbol{\mu}_{\mathrm{L}}, \underline{\boldsymbol{L}}_r) = \boldsymbol{\mu}_{\mathrm{L}} + J_r(\boldsymbol{\mu}_{\mathrm{L}})\,\underline{\boldsymbol{L}}_r \tag{10.76}$$

in the tangent space (index t), depending on the mean vector of the uncertain 3D line and the random 4-vector of the reduced coordinates $\underline{\boldsymbol{L}}_r$.

In order to arrive at a random 6-vector which is both spherically normalized and fulfils the Plücker constraint for finite random perturbations, we need to normalize

$\mathbf{L}^t = [\boldsymbol{L}_h^{t\mathsf{T}}, \boldsymbol{L}_0^{t\mathsf{T}}]^\mathsf{T}$ accordingly. For this we observe that the two 3-vectors $\boldsymbol{L}_h^t$ and $\boldsymbol{L}_0^t$ generally are not orthogonal, see Fig. 10.17. Following the idea of Bartoli and Sturm (2005), we rotate these vectors in their common plane such that they become orthogonal and the directional correction to both components is the same. We apply linear interpolation

enforcing the Plücker constraint

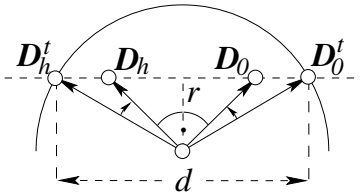


Fig. 10.17 Enforcing the Plücker constraint on a 6-vector $\mathbf{L}^t = [\boldsymbol{L}_h^{t\mathsf{T}}, \boldsymbol{L}_0^{t\mathsf{T}}]^\mathsf{T}$. The vectors $\boldsymbol{D}_0^t$ and $\boldsymbol{D}_h^t$ are the normalized components $\boldsymbol{L}_0^t$ and $\boldsymbol{L}_h^t$ of the line parameters. They should be perpendicular. Starting from approximate vectors $\boldsymbol{D}_0^t$ and $\boldsymbol{D}_h^t$ of the vector $\mathbf{L}^t$ in the tangent space (index t), we can easily enforce the perpendicularity in a symmetric manner. Scaling the vectors $\boldsymbol{D}_0$ and $\boldsymbol{D}_h$ with $|\boldsymbol{L}_0|$ and $|\boldsymbol{L}_h|$ yields a valid Plücker vector

of the directions $\boldsymbol{D}_h^t = \mathsf{N}(\mathbf{L}_h^t)$ and $\boldsymbol{D}_0^t = \mathsf{N}(\mathbf{L}_0^t)$, which guarantees that we only correct within the common plane. With the distance $d = |\boldsymbol{D}_h^t - \boldsymbol{D}_0^t|$ and the shortest distance $r = \sqrt{1 - d^2/4}$ of the origin to the line joining $\boldsymbol{D}_h^t$ and $\boldsymbol{D}_0^t$, we have the perpendicular directions $\boldsymbol{D}_h$ and $\boldsymbol{D}_0$,

$$\boldsymbol{D}_{h,0} = (1/2 \pm r/d)\, \boldsymbol{D}_h^t + (1/2 \mp r/d)\, \boldsymbol{D}_0^t \,. \tag{10.77}$$

The 6-vector

$$\mathbf{M} = \begin{bmatrix} |\boldsymbol{L}_h^t|\, \boldsymbol{D}_h \\ |\boldsymbol{L}_0^t|\, \boldsymbol{D}_0 \end{bmatrix} \tag{10.78}$$

now fulfils the Plücker constraint but needs to be spherically normalized. This finally leads to the normalized stochastic 3D line coordinates

$$\underline{\mathbf{L}} = \mathsf{N}(\underline{\mathbf{L}}^t(\boldsymbol{\mu}_\mathrm{L}, \underline{\boldsymbol{L}}_r)) \doteq \frac{\underline{\mathbf{M}}}{|\underline{\mathbf{M}}|} \,, \tag{10.79}$$

which guarantee that $\underline{\mathbf{L}}$ fulfils the Plücker constraint. Observe, we overload the normalization operator $\mathsf{N}(\cdot)$ for 6-vectors, leading to a normalized Plücker vector. Using (10.76), the inverse relation to (10.79), up to a first-order approximation, we obtain

$$\underline{\boldsymbol{L}}_r = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{L})\, \underline{\mathbf{L}} \,, \tag{10.80}$$

since $J_r(\boldsymbol{\mu}_\mathrm{L})$ is an orthonormal matrix. The relation between the covariance matrices of $\underline{\mathbf{L}}$ and $\underline{\boldsymbol{L}}_r$ therefore are

$$\boldsymbol{\Sigma}_{\mathrm{LL}} = J_r(\boldsymbol{\mu}_\mathrm{L})\, \boldsymbol{\Sigma}_{L_r L_r}\, J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{L}) \,, \qquad \boldsymbol{\Sigma}_{L_r L_r} = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{L})\, \boldsymbol{\Sigma}_{\mathrm{LL}}\, J_r(\boldsymbol{\mu}_\mathrm{L}) \,, \tag{10.81}$$

in full analogy to the other geometric entities.

10.2.3 Uncertain Transformations

Transformations in our context are represented by matrices, say by A, or by their vectors, say $\boldsymbol{a} = \mathrm{vec}A$. The number of independent parameters is usually lower than the number of matrix entries, as the matrix shows certain regularities, such as orthonormality or homogeneity. Therefore, the covariance matrix $\boldsymbol{\Sigma}_{aa}$ of the vector $\underline{\boldsymbol{a}}$ of the stochastic matrix $\underline{A}$ will be rank deficient, and an estimation of the entries needs to take constraints between the elements of $\underline{A}$ into consideration in order to enforce the properties of the transformation. Transformations A represented with a minimal set of parameters, say $\boldsymbol{b}$, are free from

such constraints; however, minimal representations generally do not cover the complete space of the transformations, e.g., the skew matrix representation for a rotation cannot realize rotations by 180°.

In order to circumvent this problem, we exploit the fact that regular transformations generally form a multiplicative group (see Eade, 2014). Therefore an uncertain transformation $\underline{A}$ can be represented by the mean transformation $\mathbb{E}(\underline{A})$ multiplied by a small regular random transformation $A(\underline{\Delta \boldsymbol{b}})$, which depends on a *minimal* set of small random parameters $\underline{\Delta \boldsymbol{b}}$. This small random transformation $A(\underline{\Delta \boldsymbol{b}})$ thus is close to the unit transformation, and up to a first-order approximation can be written as $I + B(\underline{\Delta \boldsymbol{b}})$, where the structure of the matrix $B(\underline{\Delta \boldsymbol{b}})$ depends on the type of the transformation group. Thus, generally we have[4]

$$\underline{A} = A(\underline{\Delta \boldsymbol{b}})\, \mathbb{E}(\underline{A}) \approx (I + B(\underline{\Delta \boldsymbol{b}}))\, \mathbb{E}(\underline{A})\,. \tag{10.82}$$

Instead of an additive correction of the original transformation, derived from a Taylor approximation, we hence establish a multiplicative correction rule. The group property of the transformation guarantees that the concatenation of the two transformations yields a transformation of the same type. Secondly, we apply the Taylor expansion only for small random transformations.

This is motivated by the fact that we always can write the regular matrix $A(\Delta \boldsymbol{b})$ of the small transformation as an exponential map, see (8.2), p. 326, and App. A.13, p. 781,

$$A(\Delta \boldsymbol{b}) = \exp(B(\Delta \boldsymbol{b})) = I + B(\Delta \boldsymbol{b}) + \frac{1}{2} B^2(\Delta \boldsymbol{b}) + \ldots\,. \tag{10.83}$$

For small $B(\Delta \boldsymbol{b})$, we may use the Taylor expansion and the uncertain transformation in a form which is linear in the minimal set of parameters. The constraints on transformation A transfer to constraints on the matrix B.

We can represent the uncertainty of $\underline{A}$ by the *regular* covariance matrix $\Sigma_{\Delta b \Delta b}$ of the minimal set of parameters $\underline{\Delta \boldsymbol{b}}$. Furthermore, since the matrix $B(\underline{\Delta \boldsymbol{b}})$ linearly depends on $\underline{\Delta \boldsymbol{b}}$, we can use the linearized expression within an estimation scheme. Thus, the parameters $\Delta \boldsymbol{b}$ for representing the uncertainty of a transformation correspond to the reduced coordinates for representing the uncertainty of homogeneous vectors. Observe, the multiplicative linearization in (10.82) means we can represent the complete set of transformations without encountering any singularities, as has been shown for rotations in Sect. 8.1.3.3, p. 330. In the following, we will elaborate on this type of representation for rotations, motions, similarities, and homographies.

10.2.3.1 Uncertain Rotations

As shown in Chap. 8, rotations, in contrast to the other geometric elements and transformations discussed so far, show a large variety of representations, each of which is relevant in a certain context. We discuss the most important representations for uncertain rotations, namely using the exponential map and quaternions.

Exercise 10.18

Uncertain Rotation Matrices. An uncertain rotation matrix $\underline{R}$ can be represented by the 9-vector $\mathrm{vec}\underline{R}$ with the means of the elements and its 9×9 covariance matrix. As a rotation matrix has only 3 degrees of freedom, this covariance matrix, if it is derived from a minimal representation, will have rank three and a six-dimensional null space caused by the six constraints a matrix has to fulfil in order to be a rotation matrix. To avoid this overrepresentation, it is adequate to use the multiplicative partitioning from Sect. 10.2.3. Therefore we represent the uncertain rotation as the product of the mean rotation $\mathbb{E}(\underline{R})$ and an uncertain small rotation $R(\underline{\Delta \boldsymbol{r}})$,

[4] A multiplicative correction from the right, thus $\underline{A} = \mathbb{E}(\underline{A})\, A(\underline{\Delta \boldsymbol{b}})$ would be equally possible. This would change the interpretation of the parameters $\Delta \boldsymbol{b}$, since they would refer to the coordinate system before the transformation.

$$\underline{R} = R(\underline{\Delta \boldsymbol{r}})\mathbb{E}(\underline{R}) = \exp(S(\underline{\Delta \boldsymbol{r}}))\mathbb{E}(\underline{R}) \approx (I_3 + S(\underline{\Delta \boldsymbol{r}}))\mathbb{E}(\underline{R})\,. \tag{10.84}$$

For the small rotation vector $\Delta \boldsymbol{r}$ we can use any three-parameter set, including Eulerian angles, but the matrix $R(\underline{\Delta \boldsymbol{r}})$ is most easily determined using the Rodriguez representation (8.59), p. 336. The random small rotation vector $\underline{\Delta \boldsymbol{r}}$ has zero mean; it does not change the mean rotation, and carries all uncertainty:

$$\underline{\Delta \boldsymbol{r}} \sim \mathcal{M}(\mathbf{0}, \Sigma_{\Delta r \Delta r})\,. \tag{10.85}$$

The degrees of freedom of this representation are minimal; thus, the covariance matrix $\Sigma_{\Delta r \Delta r}$ in general will be regular. It may also capture correlations between the elements Δr_i.

Altogether, we have the minimal representation of uncertain rotations,

$$\underline{\mathcal{R}}: \quad \{\mathbb{E}(\underline{R}), \Sigma_{\Delta r \Delta r}\}\,. \tag{10.86}$$

Uncertain Quaternions. Quaternions, when representing rotations, are homogeneous vectors. Though they can be used unnormalized, usually unit quaternions are preferred. This corresponds to a spherical normalization of the 4-vector, so that we have the general form of an uncertain unit quaternion,

$$\underline{\mathbf{q}} \sim \mathcal{M}(\boldsymbol{\mu}_\mathrm{q}, \Sigma_\mathrm{qq}) \qquad |\underline{\mathbf{q}}| = 1\,, \tag{10.87}$$

with the covariance matrix having rank 3 and null space $\boldsymbol{\mu}_\mathrm{q}$, in full equivalence to spherically normalized homogeneous vectors of 3D points, see Sect. 10.2.2.2, p. 372.

As an alternative, we can also represent the uncertain quaternion $\underline{\mathbf{q}}$ with reduced coordinates, which is useful for estimation. An uncertain quaternion is the product of the mean quaternion $\boldsymbol{\mu}_\mathrm{q}$ and an uncertain small quaternion $\underline{\Delta \mathbf{q}}$,

$$\underline{\mathbf{q}} = \underline{\Delta \mathbf{q}}\, \mathbb{E}(\underline{\mathbf{q}}) = \overline{\mathsf{M}}_{\mu_\mathrm{q}} \underline{\Delta \mathbf{q}}\,, \tag{10.88}$$

with the 4×4 matrix $\overline{\mathsf{M}}$ from (8.44), p. 334. This allows an easy derivation of the covariance matrix of $\underline{\mathbf{q}}$ if the 3×3 covariance matrix $\Sigma_{\Delta r \Delta r}$ with the small rotation vector $\Delta \boldsymbol{r}$ is given, e.g., from some estimation procedure,

$$\Delta \mathbf{q} = \mathsf{N}\left(\begin{bmatrix} 1 \\ \frac{1}{2} \Delta \boldsymbol{r} \end{bmatrix} \right)\,. \tag{10.89}$$

Thus, we obtain the 4×4 covariance matrix

$$\Sigma_\mathrm{qq} = \frac{1}{4} \overline{\mathsf{M}}_{\mu_\mathrm{q}} \begin{bmatrix} 0 & \mathbf{0}^\mathsf{T} \\ \mathbf{0} & \Sigma_{\Delta r \Delta r} \end{bmatrix} \overline{\mathsf{M}}_{\mu_\mathrm{q}}^\mathsf{T}\,. \tag{10.90}$$

Due to $\boldsymbol{\mu}_\mathrm{q} = \overline{\mathsf{M}}_{\mu_\mathrm{q}} \boldsymbol{e}_1^{[4]}$ and $\overline{\mathsf{M}}_\mathrm{q}^\mathsf{T} = \overline{\mathsf{M}}_\mathrm{q}^{-1}$ for unit quaternions, the covariance matrix Σ_qq has null space $\boldsymbol{\mu}_\mathrm{q}$. We thus have the minimal representation of uncertain rotations with quaternions,

$$\underline{\mathcal{R}}: \quad \{\mathbb{E}(\underline{\mathbf{q}}), \Sigma_{\Delta r \Delta r}\}\,, \tag{10.91}$$

just replacing the representation for the mean rotation in (10.86), p. 383.

10.2.3.2 Uncertain Motions and Similarities

The groups of motions and of similarities follow the same scheme as for rotations. We only discuss uncertain similarities, as motions are a special case if we set the scale to one. The representation presented here is comparable to the one used by Pennec and Thirion (1997), see also Pennec (2006).

An uncertain similarity is represented as[5]

$$\underline{\mathsf{M}} = \mathsf{M}(\underline{\Delta \boldsymbol{p}})\mathbb{E}(\underline{\mathsf{M}}) = \exp(A(\underline{\Delta \boldsymbol{p}}))\ \mathbb{E}(\underline{\mathsf{M}}) \approx (I + A(\underline{\Delta \boldsymbol{p}}))\mathbb{E}(\underline{\mathsf{M}})\,, \tag{10.92}$$

starting from the mean similarity $\mathbb{E}(\underline{\mathsf{M}})$ and applying a random "small similarity" $\mathsf{M}(\underline{\Delta \boldsymbol{p}})$ close to the unit matrix. This small similarity depends on the seven parameters, cf. (6.36), p. 255

$$\Delta \boldsymbol{p} = \begin{bmatrix} \Delta \boldsymbol{r} \\ \Delta \boldsymbol{T} \\ \Delta \lambda \end{bmatrix}\,, \tag{10.93}$$

and reads

$$\mathsf{M}(\Delta \boldsymbol{p}) = \mathsf{M}(\Delta \boldsymbol{T}, \Delta \boldsymbol{r}, \Delta \lambda) = \begin{bmatrix} \mathrm{e}^{\Delta \lambda} R(\Delta \boldsymbol{r}) & \Delta \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}\,. \tag{10.94}$$

Observe, we use the factor $\mathrm{e}^{\Delta \lambda}$ in the small similarity, which guarantees that the sign of the scale of $\boldsymbol{p}$ does not change. For small $\Delta \lambda$, we also could have used $1 + \Delta \lambda$ as the correcting factor. As a result, we represent an uncertain similarity by

$$\underline{\mathcal{M}}: \quad \{\mathbb{E}(\underline{\mathsf{M}}), \Sigma_{\Delta p \Delta p}\}\,, \tag{10.95}$$

i.e., by its mean and the covariance matrix of the minimal parameter set $\Delta \boldsymbol{p}$. The small similarity can be approximated at $\widehat{\Delta \boldsymbol{p}}^a = \mathbf{0}$ or $\mathsf{M}(\widehat{\Delta \boldsymbol{p}}^a) = I_4$ by

$$\mathsf{M}(\Delta \boldsymbol{p}) \approx \begin{bmatrix} (1 + \Delta \lambda)(I_3 + S(\Delta \boldsymbol{r})) & \Delta \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \approx I_4 + \mathsf{A}(\Delta \boldsymbol{p}) \tag{10.96}$$

with

$$\mathsf{A}(\Delta \boldsymbol{p}) = \begin{bmatrix} \Delta \lambda I_3 + S(\Delta \boldsymbol{r}) & \Delta \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} = \begin{bmatrix} \Delta \lambda & -\Delta r_3 & +\Delta r_2 & \Delta t_1 \\ +\Delta r_3 & \Delta \lambda & -\Delta r_1 & \Delta t_2 \\ -\Delta r_2 & +\Delta r_1 & \Delta \lambda & \Delta t_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}\,. \tag{10.97}$$

The representation for uncertain motions is achieved by setting $\lambda = 1$ and $\Delta \lambda = 0$. Observe, $I_4 + A(\Delta \boldsymbol{p})$ is linear in the parameters and used for estimation, but is only an approximation to a similarity transformation, since the upper left 3×3 matrix is not a scaled rotation matrix.

10.2.3.3 Uncertain Homographies

Minimal Representation of an Uncertain Homography. Homographies are represented by a homogeneous matrix H. In 2D, the 3×3 matrix H should depend on eight parameters.

Again, we represent the uncertain homography $\underline{\mathsf{H}}$ with the mean transformation $\mathbb{E}(\underline{\mathsf{H}})$ and realize a multiplicative deviation using a small random homography $\mathsf{H}(\underline{\Delta \boldsymbol{p}})$ close to an identity transform (see Begelfor and Werman, 2005). The resulting formula is fully equivalent to (10.84),

$$\underline{\mathsf{H}} = \mathsf{H}(\underline{\Delta \boldsymbol{p}})\ \mathbb{E}(\underline{\mathsf{H}}) = \exp(K(\underline{\Delta \boldsymbol{p}}))\ \mathbb{E}(\underline{\mathsf{H}}) \approx (I_3 + K(\underline{\Delta \boldsymbol{p}}))\ \mathbb{E}(\underline{\mathsf{H}})\,. \tag{10.98}$$

The linearization of the small homography uses a 3×3 matrix K depending on a minimal set $\underline{\Delta \boldsymbol{p}}$ of eight parameters. We now exploit the homography H and specify the matrix K.

As a homography is unique up to scale, we need to fix the scale in some reasonable manner. This is often done by fixing one of the elements of H, e.g., $h_{33} = 1$, or by fixing the scale of H via the Frobenius norm, thus $||\mathsf{H}||_2 = 1$. The first choice does not work if

[5] Using the letter M instead of the more intuitive letter S to avoid confusion with the skew symmetric matrix.

the chosen element is close to 0, a case which cannot be excluded in general. The second choice is frequently applied in closed form solutions, as the constraint is quadratic in the elements of H. In iterative estimation schemes, this constraint is more cumbersome than the one given below.

Here it is useful to normalize the homographies by their determinant, i.e., to require

$$|\mathsf{H}(\boldsymbol{p})| = 1 \qquad \text{and} \qquad |\mathsf{H}(\Delta\boldsymbol{p})| = 1\,. \tag{10.99}$$

For a mapping from an n-dimensional projective space to another, these matrices are elements of the special linear group $\mathrm{SL}(n+1)$ containing all matrices with determinant 1.

The determinant constraint (10.99) on H is now equivalent to the following trace constraint on the matrix K (see (A.144), p. 781),

$$|\mathsf{H}| = 1 \quad \Leftrightarrow \quad \mathrm{tr}(\mathsf{K}) = 0\,, \tag{10.100}$$

since $|\exp(\mathsf{A})| = \exp(\mathrm{tr}(\mathsf{A}))$ for any arbitrary matrix A. Thus, the differential of a homography $\Delta\mathsf{H}$ at the approximate point I_3 turns out to be a traceless 3×3 matrix $\mathsf{K}(\Delta\boldsymbol{p})$ if we want to preserve the determinant of the update. The trace constraint can easily be realized by the representation

$$\mathsf{K}(\Delta\boldsymbol{p}) = \begin{bmatrix} \Delta p_1 & \Delta p_4 & \Delta p_7 \\ \Delta p_2 & \Delta p_5 & \Delta p_8 \\ \Delta p_3 & \Delta p_6 & -\Delta p_1 - \Delta p_5 \end{bmatrix}, \qquad \mathrm{tr}\mathsf{K}(\Delta\boldsymbol{p}) = 0, \tag{10.101}$$

which linearly depends on eight parameters and guarantees that K is traceless, so that we can again represent an uncertain homography by

$$\mathcal{H}: \qquad \{\mathbb{E}(\underline{\mathsf{H}}), \Sigma_{\Delta p \Delta p}\} \tag{10.102}$$

with a regular 8×8 covariance matrix $\Sigma_{\Delta p \Delta p}$.

Relation to the Covariance Matrix of vec$\mathsf{H}$. The homography matrix H is often directly determined, and it is desirable to characterize its uncertainty by the covariance matrix of its elements,

$$\mathcal{H}: \{\mathbb{E}(\underline{\mathsf{H}}), \Sigma_{\mathrm{hh}}\}\,, \tag{10.103}$$

with the vector $\mathbf{h} = \mathrm{vec}\mathsf{H}$. The relation to $\Sigma_{\Delta p \Delta p}$ can be established using (10.98). We have

$$\Delta\mathbf{h} = \Delta\mathrm{vec}\mathsf{H} = \mathrm{vec}\left(\mathsf{K}(\Delta\boldsymbol{p})\mathbb{E}(\underline{\mathsf{H}})\right) = (\mathbb{E}(\underline{\mathsf{H}}^\mathsf{T}) \otimes \mathsf{I}_3)\mathrm{vec}(\mathsf{K}(\Delta\boldsymbol{p}))\,. \tag{10.104}$$

The vector $\boldsymbol{k}(\Delta\boldsymbol{p}) := \mathrm{vec}(\mathsf{K}(\Delta\boldsymbol{p}))$ can be written as

$$\boldsymbol{k}(\Delta\boldsymbol{p}) = \begin{bmatrix} \Delta\boldsymbol{p} \\ -\Delta p_1 - \Delta p_5 \end{bmatrix} = \underbrace{\mathsf{J}_{k,\Delta p}}_{9\times 8}\,\Delta\boldsymbol{p}\,, \tag{10.105}$$

with the Jacobian

$$\mathsf{J}_{k,\Delta p} = \left[\begin{array}{c} \mathsf{I}_8 \\ \hline -1 \mid 0 \mid 0 \mid 0 \mid -1 \mid 0 \mid 0 \mid 0 \end{array}\right]. \tag{10.106}$$

This yields the Jacobian of $\mathbf{h}$ w.r.t. the parameters $\Delta\boldsymbol{p}$ in (10.104),

$$\mathsf{J}_{\mathrm{h},\Delta p} = (\mathbb{E}(\underline{\mathsf{H}}^\mathsf{T}) \otimes \mathsf{I}_3)\;\mathsf{J}_{k,\Delta p}\,, \tag{10.107}$$

which is useful for deriving the covariance matrix of $\mathbf{h}$ by $\Sigma_{\mathrm{hh}} = \mathsf{J}_{\mathrm{h},\Delta p}\Sigma_{\Delta p \Delta p}\mathsf{J}_{\mathrm{h},\Delta p}^\mathsf{T}$.

The inverse relation uses the determinant constraint (10.100). If $\{\mathsf{H}, \Sigma_{\mathrm{hh}}\}$ is given, with arbitrary determinant and a covariance matrix (not necessarily of the correct rank), we first need to scale H and then enforce the rank constraint. Assuming a 3×3 matrix H, scaling is performed by

$$\underline{\mathsf{M}} = \frac{\underline{\mathsf{H}}}{\underline{f}} \quad \text{with} \quad \underline{f} = (\text{abs}(|\underline{\mathsf{H}}|))^{1/3} \,. \tag{10.108}$$

It can be shown (see App. A.15, p. 783) that the covariance matrix of the scaled vector $\mathbf{m} = \text{vec}\mathsf{M}$ is

$$\Sigma_{\text{mm}} = J_{\text{mh}} \Sigma_{\text{hh}} J_{\text{mh}}^{\mathsf{T}} \,, \tag{10.109}$$

with the Jacobian

$$J_{\text{mh}} = \frac{1}{f} \left(I_9 - \frac{1}{3} \mathbf{h} \mathbf{i}^{\mathsf{T}} \right) \,, \tag{10.110}$$

with the vector

$$\mathbf{i} = \text{vec}(\mathsf{H}^{-\mathsf{T}}) \,. \tag{10.111}$$

Using $\Delta\mathsf{H} = \mathsf{K}(\Delta\boldsymbol{p})\mathbb{E}(\underline{\mathsf{H}})$ from (10.104), p. 385 leads to $\Delta\mathsf{H}(\mathbb{E}(\mathsf{H}))^{-1} = \mathsf{K}(\Delta\boldsymbol{p})$. Vectorization and selection of the first eight elements from $\text{vec}\mathsf{K}$ yields

$$\Delta\boldsymbol{p} = [I_8 \mid \mathbf{0}]((\mathbb{E}(\mathsf{H}))^{-\mathsf{T}} \otimes I_3)\Delta\mathbf{h} \,, \tag{10.112}$$

or, finally, eliminating the factor f,

$$\Delta\boldsymbol{p} = J_{\Delta p,\text{h}} \Delta\mathbf{h} = [I_8 \mid \mathbf{0}]((\mathbb{E}(\mathsf{H}))^{-\mathsf{T}} \otimes I_3) \left(I_3 - \frac{1}{3} \mathbf{h} \mathbf{i}^{\mathsf{T}} \right) \Delta\mathbf{h} \,. \tag{10.113}$$

The Jacobian $J_{\Delta p,\text{h}}$ is invariant to the scaling of H. It is the basis for the final covariance matrix $\Sigma_{\Delta p \Delta p} = J_{\Delta p,\text{h}} \Sigma_{\text{hh}} J_{\Delta p,\text{h}}^{\mathsf{T}}$, which is regular in general. We can check $J_{\Delta p,\text{h}} J_{\text{h},\Delta p} = I_8$.

Other regular transformations, e.g., affinities or scaled rotations, can be represented the same way.

10.3 Propagation of the Uncertainty of Homogeneous Entities

10.3.1 Uncertain Geometric Elements and Transformations 386
10.3.2 Equivalence of Uncertain Homogeneous Vectors 390

In most cases, uncertain points, lines, planes, and transformations result from processing uncertain observed entities, either by direct construction or by some estimation process. Therefore, propagation of uncertainty is an indispensable task. This especially holds for algorithms such as RANSAC (see Sect. 4.7.7, p. 153), where samples quickly need to be evaluated statistically. Uncertainty propagation can easily be applied to all the constructions discussed in the previous sections. We demonstrate the method using the determination of a transformation from a minimal set of corresponding points as an example.

10.3.1 Uncertain Geometric Elements and Transformations

10.3.1.1 Uncertainty of Multilinear Forms

Most constructions appear as bilinear forms. They can be written in general as

$$\mathbf{c} = \mathbf{c}(\mathbf{a}, \mathbf{b}) = \mathsf{U}(\mathbf{a})\mathbf{b} = \mathsf{V}(\mathbf{b})\mathbf{a} \tag{10.114}$$

with matrices $\mathsf{U}(\mathbf{a})$ and $\mathsf{V}(\mathbf{b})$ depending on the given entities, represented as vectors $\mathbf{a}$ and $\mathbf{b}$. At the same time these matrices are the Jacobians of the vector $\mathbf{c}$ w.r.t. the vectors $\mathbf{a}$ and $\mathbf{b}$,

$$\mathsf{U} = \frac{\partial \mathbf{c}}{\partial \mathbf{b}}, \qquad \mathsf{V} = \frac{\partial \mathbf{c}}{\partial \mathbf{a}}, \tag{10.115}$$

which we need for variance propagation,

variance propagation for bilinear forms

$$\Sigma_{cc} = [\mathsf{V}(\boldsymbol{\mu}_a), \mathsf{U}(\boldsymbol{\mu}_b)] \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \begin{bmatrix} \mathsf{V}^\mathsf{T}(\boldsymbol{\mu}_a) \\ \mathsf{U}^\mathsf{T}(\boldsymbol{\mu}_b) \end{bmatrix}. \tag{10.116}$$

An example has been given in Sect. 5.1.2.7, p. 205, for the task of constructing a 2D line as the join of two 2D points.

Observe, the Jacobians need to be evaluated at the mean. Furthermore, if the variances are large, using only the first-order Taylor expansion may lead to biased result, see Sect. 2.7.6, p. 44. For trilinear forms, e.g., for determining the plane parameters $\mathbf{A}$ from the three 3D point coordinates, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, via $\mathbf{A} = \overline{\Gamma}^\mathsf{T}(\mathsf{T}(\mathbf{X})\mathbf{Y})\mathbf{Z}$, we can find analogous expressions which take possible correlations into account (see (7.49), p. 302). Exercise 10.7

The simple rule for variance propagation hides the fact that the resulting covariance matrices are not rank deficient, which we would expect. We will see below that this does not affect the use of these regular covariance matrices.

This method for uncertainty propagation when constructing geometric entities or (after vectorization) geometric transformations is a strong motivation to use homogeneous vectors and matrices to represent elements of projective geometry. We apply it in three basic contexts:

1. constructing geometric entities; an example has been given in Sect. 5.1.2.7, p. 205 for the task of constructing a 2D line as the join of two 2D points;
2. transforming geometric entities, where both the given entity and the transformation may be uncertain;
3. generating transformations from a minimal set of geometric entities.

10.3.1.2 Uncertainty of Mapped Points

We first give the uncertainty of the result of a mapping. Let the mapping be $\mathbf{x}' = \mathsf{H}\mathbf{x}$, and both the given point $\mathbf{x}$ and the mapping H be uncertain with covariance matrices Σ_{xx} and Σ_{hh}. Then the mapping can be written in two forms, see (7.112), p. 315:

$$\mathbf{x}' = \mathsf{H}\mathbf{x} = (\mathbf{x}^\mathsf{T} \otimes I_3)\mathbf{h}, \tag{10.117}$$

with the vector $\mathbf{h} = \mathrm{vec}(\mathsf{H})$ containing the elements of H columnwise. If the mapping and the point to be transferred are mutually independent, then the covariance matrix of the transferred point is

$$\Sigma_{x'x'} = \mathsf{H}\Sigma_{xx}\mathsf{H}^\mathsf{T} + (\mathbf{x}^\mathsf{T} \otimes I_3)\Sigma_{hh}(\mathbf{x} \otimes I_3). \tag{10.118}$$

The uncertain Euclidean coordinates result from (10.32), p. 371.

The next section will discuss how to determine the uncertainty of a transformation from a minimal set of points and then give an example which illustrates the possibility of predicting the uncertainty of transferred points.

10.3.1.3 Uncertainty of a Homography from a Set of Corresponding Points

Variance propagation can be used to derive the covariance matrix of a transformation determined from a set of corresponding points. We already discussed the determination of a homography from a minimal set of four corresponding point pairs in Sect. 7.4.3.1, p. 321. Here we discuss how to determine the covariance matrix of the derived 2D homography H and give a numerically stable algorithm, which is also valid for more than four points. Since the problem is nonlinear we need to linearize. We take the given coordinates and

the derived homography as the linearization point. This is exact if we only have four points; otherwise, the coefficients are slightly perturbed, which causes second-order effects in the covariance matrix as long as no outliers are present, which we do not assume here. Moreover, we do not have an explicit expression for the homography; this is why we apply implicit variance propagation, see Sect. 2.7.5, p. 43.

The model reads $\mathbf{x}'_i = \mathsf{H}\mathbf{x}_i$, $i = 1, 2, 3, 4$. This is equivalent to the constraints $\mathbf{x}'_i \times \mathsf{H}\mathbf{x}_i = \mathbf{0}$, $i = 1, 2, 3, 4$, which (pointwise) are algebraically dependent. We now may either select a set of two independent constraints or, equivalently, use reduced coordinates, leading to

$$\mathsf{S}^{(s)}(\mathbf{x}'_i)\mathsf{H}\mathbf{x}_i = \mathbf{0} \quad \text{or} \quad J_r^{\mathsf{T}}(\mathbf{x}'_i)\mathsf{H}\mathbf{x}_i = \mathbf{0}\,; \tag{10.119}$$

see the remark on the relation between $\mathsf{S}(\mathbf{x})$ and $\mathsf{null}(\mathbf{x}^{\mathsf{T}})$ below (10.26), p. 370. We will continue with the selected constraints from the first set.

Using the vector $\mathbf{h} = \mathrm{vec}(\mathsf{H})$, collecting the parameters columnwise, we arrive at the constraints (7.140), p. 321,

$$\boldsymbol{g}_i(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{h}) = \mathsf{S}^{(s)}(\mathbf{x}'_i)\mathsf{H}\,\mathbf{x}_i = -\mathsf{S}^{(s)}(\mathsf{H}\mathbf{x}_i)\,\mathbf{x}'_i = \mathsf{S}^{(s)}(\mathbf{x}'_i)(\mathbf{x}_i^{\mathsf{T}} \otimes I_3)\,\mathbf{h} = \mathbf{0}\,. \tag{10.120}$$

In order to determine the relation of the three vectors $\mathbf{h}, \mathbf{x}$ and $\mathbf{x}'$ in the presence of small perturbations, we analyse the total differential $\Delta\boldsymbol{g}_i = \mathbf{0}$ of $\boldsymbol{g}_i = \mathbf{0}$ with respect to the three vectors

$$\Delta\boldsymbol{g}_i = A_i\Delta\mathbf{h} + B_i\Delta\mathbf{x}_i + C_i\Delta\mathbf{x}'_i = \mathbf{0}\,, \tag{10.121}$$

with the three Jacobians

$$\underset{2\times 9}{A_i} = \mathbf{x}_i^{\mathsf{T}} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_i) \qquad \underset{2\times 3}{B_i} = \mathsf{S}^{(s)}(\mathbf{x}'_i)\mathsf{H} \qquad \underset{2\times 3}{C_i} = -\mathsf{S}^{(s)}(\mathsf{H}\mathbf{x}_i)\,. \tag{10.122}$$

For the complete coordinate vectors we have

$$A\Delta\mathbf{h} + B\Delta\mathbf{x} + C\Delta\mathbf{x}' = \mathbf{0}\,, \tag{10.123}$$

with the three matrices

$$\underset{8\times 9}{A} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} \qquad \underset{8\times 12}{B} = \mathrm{Diag}(\{B_i\}) \qquad \underset{8\times 12}{C} = \mathrm{Diag}(\{C_i\})\,. \tag{10.124}$$

Observe, matrix A is used to determine the parameter vector $\mathbf{h}$ from $|A\mathbf{h}| \to \min$ in an algebraic minimization scheme. Therefore, changes $\Delta\mathbf{h}$ which fulfil $A\Delta\mathbf{h} = \mathbf{0}$ can only lead to a change of the length of $\mathbf{h}$, not to a change in the transformation. So, all changes in $\mathbf{x}$ or $\mathbf{x}'$ that lead to changes of $\mathbf{h}$ that are *not* parallel to $\mathbf{h}$ should be captured in the covariance matrix of $\mathbf{h}$.

We hence interpret the changes of the vectors $\mathbf{h}$, $\mathbf{x}$, and $\mathbf{x}'$ as random perturbations, and treat them as stochastic variables $\underline{\Delta\mathbf{h}}$, $\underline{\Delta\mathbf{x}}$, and $\underline{\Delta\mathbf{x}'}$ with zero mean and covariance matrices Σ_{xx}, $\Sigma_{\mathrm{x'x'}}$, and Σ_{hh}, respectively.

We now solve for $\Delta\mathbf{h}$,

$$\underline{\Delta\mathbf{h}} = -A^{+}(B\underline{\Delta\mathbf{x}} + C\underline{\Delta\mathbf{x}'})\,, \tag{10.125}$$

with the pseudo-inverse $A^{+} = A^{\mathsf{T}}(AA^{\mathsf{T}})^{-1}$ of A valid for four points. It guarantees that $\mathbf{h}^{\mathsf{T}}\Delta\mathbf{h} = 0$, since $\mathbf{h}^{\mathsf{T}}A^{+} = \mathbf{h}^{\mathsf{T}}A^{\mathsf{T}}(AA^{\mathsf{T}})^{-1} = \mathbf{0}^{\mathsf{T}}$; i.e., only perturbations $\Delta\mathbf{h}$ orthogonal to $\mathbf{h}$ are present in $\underline{\Delta\mathbf{h}}$.

covariance matrix of homography

If we can assume $\underline{\mathbf{x}}$ to be independent of $\underline{\mathbf{x}'}$, the covariance matrix of the derived vector $\underline{\mathbf{h}}$ can be derived by variance propagation

$$\Sigma_{\mathrm{hh}} = A^{+}(B\Sigma_{\mathrm{xx}}B^{\mathsf{T}} + C\Sigma_{\mathrm{x'x'}}C^{\mathsf{T}})A^{+\mathsf{T}}. \tag{10.126}$$

Observe, we did not normalize $\mathbf{h}$ in any way; this can be done in a second step.

The complete procedure for determining the homography from point pairs and the uncertainty of the transformation parameters is given in the following algorithm. The minimal solution is achieved for $I = 4$ point pairs. The algorithm also encodes the conditioning of the coordinates together with their covariance matrices in lines (3), (6), and (7), see Sect. 6.9, p. 286. The resulting homography in line (10) together with its covariance matrix in line (14) refers to the conditioned coordinates, which is why unconditioning of this matrix together with its covariance matrix is necessary, see lines (15) to (17), see (2.139), p. 43.

Remark: If the algorithm is used for more than four points, the pseudo-inverse $\mathbf{A}^+$ of $\mathbf{A}$ needs to be replaced by the pseudo inverse of a rank 8 approximation of $\mathbf{A}$, which is why in line 12 the diagonal matrix $\mathbf{\Lambda}^+$ is the pseudo-inverse of $\mathbf{\Lambda}$ with only the eight largest singular values $\lambda_i \neq 0$, see the discussion after (4.521), p. 181. ◇

Algorithm 7: Algebraic solution for estimating 2D homography from $I \geq 4$ point pairs.
$[\mathsf{H}, \Sigma_{\mathrm{hh}}]$= `homography_from_point_pairs`$(\{\mathbf{x}, \Sigma_{\mathrm{xx}}, \mathbf{x}', \Sigma_{\mathrm{x'x'}}\}_i)$
Input: uncertain point pairs $\{\mathbf{x}_i, \Sigma_{\mathrm{x}_i\mathrm{x}_i}, \mathbf{x}'_i, \Sigma_{\mathrm{x}'_i\mathrm{x}'_i}\}, i = 1, ..., I$.
Assumptions: all points are uncorrelated, coordinates of a point may be correlated.
Output: algebraically estimated homography H with covariance matrix Σ_{hh}.

1 Conditioning matrices T_1 and T_2, (6.137), p. 287;
2 **for** *all points* **do**
3 | Conditioned coordinates: $\mathbf{x}_i := \mathsf{T}_1\mathbf{x}_i$, $\mathbf{x}'_i := \mathsf{T}_2\mathbf{x}'_i$;
4 | Coefficient matrices: $\mathbf{A}_i^{\mathsf{T}} = [\mathbf{x}_i^{\mathsf{T}} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_i)]$;
5 **end**
6 Condition covariance matrix Σ_{xx}: $\Sigma_{\mathrm{xx}} := (\mathbf{I}_4 \otimes \mathsf{T}_1)\Sigma_{\mathrm{xx}}(\mathbf{I}_4 \otimes \mathsf{T}_1^{\mathsf{T}})$;
7 Condition covariance matrix $\Sigma_{\mathrm{x'x'}}$: $\Sigma_{\mathrm{x'x'}} := (\mathbf{I}_4 \otimes \mathsf{T}_2)\Sigma_{\mathrm{x'x'}}(\mathbf{I}_4 \otimes \mathsf{T}_2^{\mathsf{T}})$;
8 Singular value decomposition: $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathsf{T}} = \mathbf{A} = [\mathbf{A}_i^{\mathsf{T}}]$;
9 Parameters: $\mathbf{h}$ singular vector $\boldsymbol{v}$ to smallest singular value;
10 Transformation: transfer of vector $\mathbf{h}$ into 3×3 matrix H to;
11 Jacobians: $\mathbf{B} = \mathrm{Diag}(\{\mathsf{S}^{(s)}(\mathbf{x}'_i)\mathsf{H}\})$, $\mathbf{C} = -\mathrm{Diag}(\{\mathsf{S}^{(s)}(\mathsf{H}\mathbf{x}_i)\})$;
12 Pseudo inverse: $\mathbf{A}^+ = \mathbf{V}\mathbf{\Lambda}^+\mathbf{U}^{\mathsf{T}}$, $\mathbf{\Lambda}^+$ only has 8 nonzero diagonal entries;
13 Jacobians: $\mathbf{J}_1 = \mathbf{A}^+\mathbf{B}$, $\mathbf{J}_2 = \mathbf{A}^+\mathbf{C}$;
14 Covariance matrix: $\Sigma_{\mathrm{hh}} = \mathbf{J}_1\Sigma_{\mathrm{xx}}\mathbf{J}_1^{\mathsf{T}} + \mathbf{J}_2\Sigma_{\mathrm{x'x'}}\mathbf{J}_2^{\mathsf{T}}$;
15 Uncondition transformation matrix: $\mathsf{H} := \mathsf{T}_2^{-1}\mathsf{H}\mathsf{T}_1$;
16 Transformation matrix for $\mathbf{h}$: $\mathsf{T} = \mathsf{T}_1^{\mathsf{T}} \otimes \mathsf{T}_2^{-1}$;
17 Uncondition covariance matrix: $\Sigma_{\mathrm{hh}} := \mathsf{T}\Sigma_{\mathrm{hh}}\mathsf{T}^{\mathsf{T}}$.

The following example demonstrates the interpolation and extrapolation effects of transferring points from one image to another using a homography derived from a set of four points.

Example 10.3.37: ***Point transfer between images of a planar object.*** Let us assume we have two perspective images of a planar facade taken with a straight line preserving camera. We can determine the homography $\mathcal{H} : \mathcal{X} \to \mathcal{X}'$ between the images by identifying four points $\mathcal{X}_i, i = 1, 2, 3, 4$, in one image for which we know the coordinates $\mathbf{x}'_i$ in the other image. Knowing this mapping, we can transfer any other point $\mathcal{X}$ from one image to the other, leading to image coordinates $\mathbf{x}'$, and, using the methods described, we can give the uncertainty $\Sigma_{x'x'}$ of the transferred point. This is possible due to the uncertainty Σ_{hh} of the measurement of the four points used for determining the homography matrix H and the uncertainty Σ_{xx} of the point to be transferred, see Fig. 10.18.

The 30 points in the upper left image are transferred to another image using four selected points indicated by a quadrangle: five different selections of quadruples are shown. The larger the quadrangle of points for determining the homography, the higher the accuracy of the transferred points indicated by the standard ellipses (enlarged by a factor of 5). The figures demonstrate the advantage of interpolation vs. extrapolation, as points inside the used quadrangle are much more precise than those outside. We assume that the standard deviation of the measured points is 0.5 pixel. In the upper right image, the upper left point with coordinates [column=162, row=81] pixel then has a standard ellipse with the semi axes (17, 2.7) pixel, whereas the third point in the same row with coordinates [229, 85] pixel has semi-axes (3.7,

1.7) pixel. The smallest standard ellipses have major semi-axes of approximately 1 pixel, namely in the middle of the right figure in row 2. ◇

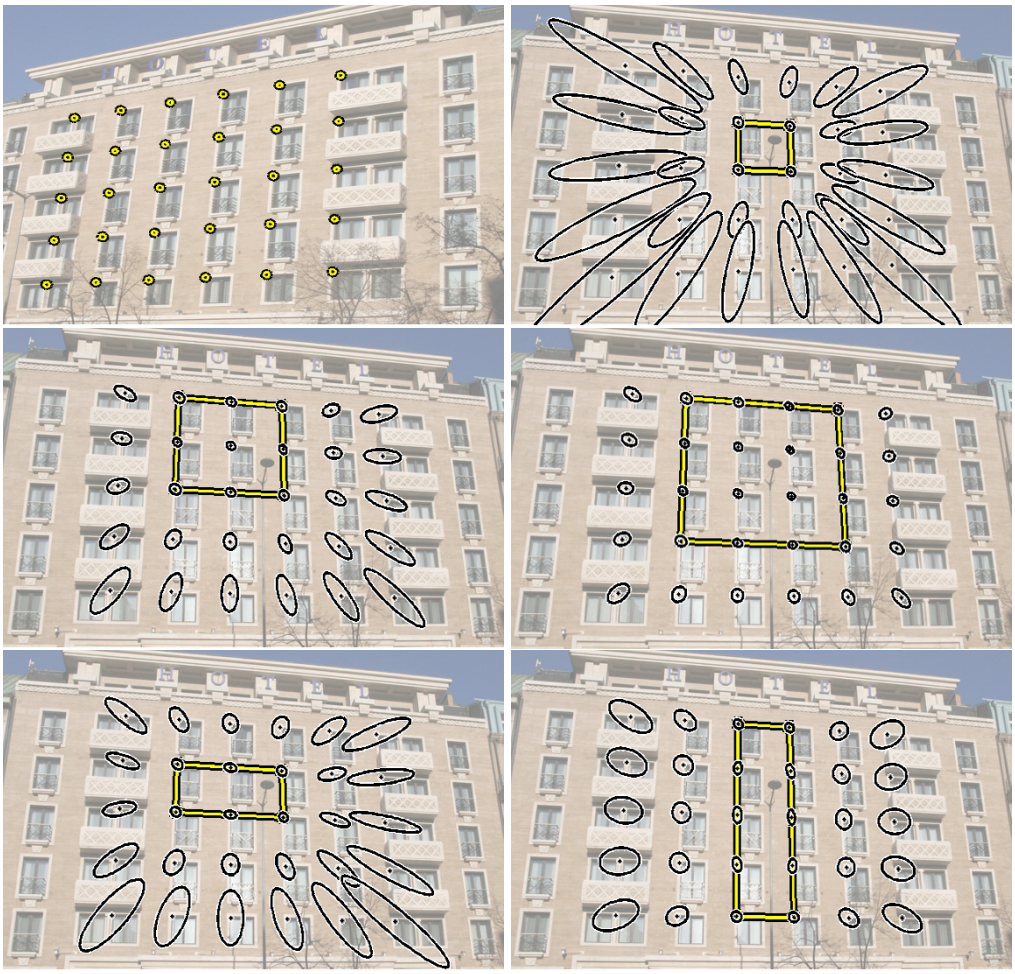


Fig. 10.18 Transferring points from one image into another with a homography for a planar object, see Example p. 389

10.3.2 Equivalence of Uncertain Homogeneous Vectors

So far, we have used any of the representations for uncertain geometric entities and assumed they are equivalent; however, we have not proven this yet. For this purpose, we generalize the equivalence relation between homogeneous vectors to one of uncertain homogeneous vectors. The essential part of a homogeneous vector is its direction, so we refer to spherically normalized vectors and use them to define the equivalence of uncertain homogeneous vectors.

Definition 10.3.25: ***Equivalent homogeneous stochastic vectors.*** Two stochastic homogeneous vectors $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ with probability density functions $p_{\mathrm{x}}(\mathbf{x})$ and $p_{\mathrm{y}}(\mathbf{y})$ are equivalent if the spherically normalized vectors

$$\underline{\mathbf{x}}^s = \frac{\underline{\mathbf{x}}}{|\underline{\mathbf{x}}|} \qquad \text{and} \qquad \underline{\mathbf{y}}^s = \frac{\underline{\mathbf{y}}}{|\underline{\mathbf{y}}|} \tag{10.127}$$

have the same probability density functions,

$$\mathcal{X}(\underline{\mathbf{x}}) \equiv y(\underline{\mathbf{y}}) \quad \Leftrightarrow \quad p_{\mathbf{x}^s}(\mathbf{x}^s) = p_{\mathbf{y}^s}(\mathbf{y}^s)\,. \tag{10.128}$$

◇

This corresponds to the equivalence relation for nonstochastic vectors,

$$\mathcal{X}(\mathbf{x}) \equiv y(\mathbf{y}) \quad \Leftrightarrow \quad \mathbf{x}^s = \mathbf{y}^s\,, \tag{10.129}$$

with identical scaling. It holds for all vectors, including elements at infinity.

If the homogeneous vectors are normally distributed and the relative accuracy is high enough, the equivalence relation only needs to refer to the first two moments.

Definition 10.3.26: ***Equivalent normally distributed homogeneous vectors.*** Two normally distributed stochastic homogeneous vectors $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ with the distributions $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{xx}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}}, \Sigma_{\mathbf{yy}})$ are equivalent if the spherically normalized vectors have the same mean and the same covariance matrix,

$$\mathcal{X}(\underline{\mathbf{x}}) \equiv y(\underline{\mathbf{y}}) \quad \Leftrightarrow \quad (\boldsymbol{\mu}_{\mathbf{x}^s}, \Sigma_{\mathbf{x}^s\mathbf{x}^s}) = (\boldsymbol{\mu}_{\mathbf{y}^s}, \Sigma_{\mathbf{y}^s\mathbf{y}^s})\,. \tag{10.130}$$

◇

This definition of equivalence does not correspond to the classical equivalence relation for nonstochastic homogeneous vectors $\mathbf{x} \cong \lambda\mathbf{x}$. Following variance propagation, it should generalize to $(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{xx}}) \cong (\lambda\boldsymbol{\mu}_{\mathbf{x}}, \lambda^2\Sigma_{\mathbf{xx}})$. This would not be general enough, however. We might even have a regular covariance matrix for homogeneous coordinates of a 2D point, which certainly cannot be made equivalent to a singular covariance matrix by scaling.

We want to show this by an example and by construction.

1. Let two points $\mathcal{X}_1$ and $\mathcal{X}_2$ be given,

$$\underline{\mathbf{x}}_1 = \begin{bmatrix} \underline{x}_1 \\ \underline{y}_1 \\ 1 \end{bmatrix} \qquad \underline{\mathbf{x}}_2 = \begin{bmatrix} \underline{x}_2 \\ \underline{y}_2 \\ 1 \end{bmatrix}\,, \tag{10.131}$$

with the identical covariance matrices

$$\Sigma_{\mathbf{x}_1\mathbf{x}_1} = \Sigma_{\mathbf{x}_2\mathbf{x}_2} = \sigma^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\,. \tag{10.132}$$

The joining line

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2 = \mathsf{S}(\mathbf{x}_1)\mathbf{x}_2 = -\mathsf{S}(\mathbf{x}_2)\mathbf{x}_1 = \begin{bmatrix} y_1 - y_2 \\ x_2 - x_1 \\ x_1 y_2 - x_2 y_1 \end{bmatrix} \tag{10.133}$$

(see (5.1.2.4), p. 201) has the covariance matrix

$$\Sigma_{\mathbf{ll}} = \sigma^2 \begin{bmatrix} 2 & 0 & -x_1 - x_2 \\ 0 & 2 & -y_1 - y_2 \\ -x_1 - x_2 & -y_1 - y_2 & x_1^2 + x_2^2 + y_1^2 + y_2^2 \end{bmatrix} \tag{10.134}$$

with the determinant

$$D = 2\sigma^6[(x_2 - x_1)^2 + (y_2 - y_1)^2]\,. \tag{10.135}$$

The determinant is only zero if the two points $\mathcal{X}_1$ and $\mathcal{X}_2$ are identical.
Generally, therefore, the covariance of the generated line is regular. This seemingly leads to a contradiction: Each line can be represented by its parameters, the direction

ϕ and the distance d of the Hessian normal form, which can be transferred to a homogeneous line vector $\mathbf{m}$ with a singular covariance matrix Σ_{mm} with rank 2. This matrix cannot be proportional to a regular covariance matrix Σ_{ll}.
Obviously, we can expect that covariance matrices may be regular when constructing geometric entities. However, if we would perform a spherical normalization of the resulting line parameters $\mathbf{l}$ with the regular covariance matrix from (10.134), we would obtain a normalized line with a singular covariance matrix, see Sect. (10.2.2.1), p. 368.

2. We can easily construct a regular covariance matrix for homogeneous vectors.
Let the uncertain point $\underline{\mathbf{x}} = \underline{\mathbf{x}}_1$ from (10.131) be given. Its covariance matrix (10.132) has rank 2. We generate the point $z(\mathbf{z})$ with the coordinates

$$\underline{\mathbf{z}} = \underline{\lambda}\,\underline{\mathbf{x}} \tag{10.136}$$

and with the *stochastic* factor

$$\underline{\lambda} \sim \mathcal{M}(\mu_\lambda, \sigma_\lambda^2) \tag{10.137}$$

and obtain the covariance matrix

$$\Sigma_{\mathrm{zz}} = \mu_\lambda^2 \Sigma_{\mathrm{xx}} + \sigma_\lambda^2 \boldsymbol{\mu}_{\mathrm{x}} \boldsymbol{\mu}_{\mathrm{x}}^{\mathsf{T}} . \tag{10.138}$$

It is clearly regular. The two uncertain points $\mathcal{X}$ and z are equivalent according to the definition (10.130).
Theoretically, we could also introduce correlations between $\underline{\mathbf{x}}$ and $\underline{\lambda}$ and thus generate a confidence ellipsoid of $\underline{\mathbf{z}}$ with nearly arbitrary axes. The spherical normalization of $\underline{\mathbf{z}}$ always leads to the same stochastic vector, $\underline{\mathbf{z}}^s$.

Figure 10.19 visualizes different instances of the uncertainty of homogeneous coordinates. Let the point $\mathcal{X}$ be represented by the line $\lambda\mathbf{x}$ through the origin O of the (uvw)-system. Its uncertainty is represented by a standard cone centred at O and mean direction $\lambda\boldsymbol{\mu}_{\mathrm{x}}$.

- The point $\mathcal{X}$, when given with its Euclidean coordinates $\underline{\mathbf{x}}^e$, has a confidence ellipsoid which is flat and is in the plane $w = 1$ like an elliptic pancake. The null space of the covariance matrix $\Sigma_{\mathrm{x}^e\mathrm{x}^e}$ is perpendicular to that plane, thus $\boldsymbol{e}_3$.

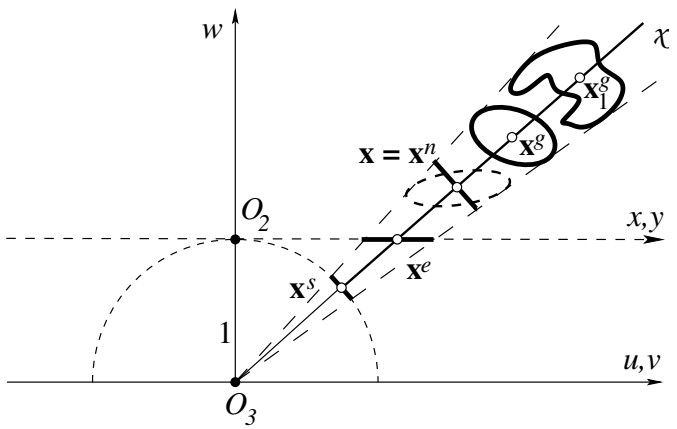


Fig. 10.19 Equivalence of stochastic homogeneous vectors for the point $\mathcal{X}$, visualized in $\mathbb{R}^3$. Starting with the vector $\mathbf{x}^e$ with its flat confidence ellipse lying in the plane $w = 1$, we obtain an elliptical confidence cone with its centre at the origin O_3. Its intersection with the unit sphere serves as the reference. All plotted confidence regions are equivalent. The quite irregular distribution at $\mathbf{x}_1^g$ is also equivalent to the others, if its projection onto the unit sphere leads to the same probability distribution, i.e., as long as the uncertainty of the direction $(O_3\mathcal{X})$ is identical

- After spherical normalization, we obtain the coordinate vector $\underline{\mathbf{x}}^s$. Its covariance matrix $\Sigma_{\mathrm{x}^s\mathrm{x}^s}$ is also singular, visualized by a flat confidence ellipsoid, and serves as reference

for the other representations of the uncertain point. It lies in the tangent plane at $\mathbf{x}^s$. The null space of $\Sigma_{\mathbf{x}^s\mathbf{x}^s}$ is $\mathbf{x}^s$.

- A general point $\underline{\mathbf{x}}^g$ (g standing for *general*) has a regular covariance matrix $\Sigma_{\mathbf{x}^g\mathbf{x}^g}$ which does not need to be aligned with the vector $\mathbf{x}^g$ in any manner. As long as its projection to the unit sphere leads to the uncertain vector $\underline{\mathbf{x}}^s$, it represents the same uncertain point.
  Obviously, there is no restriction concerning the regularity of the covariance matrices or w.r.t. their null space.
- We can take an uncertain point with homogeneous vector $\mathbf{x}$ and covariance matrix $\Sigma_{\mathbf{xx}}$, and keep its coordinates, $\mathbf{x}^n = \mathbf{x}$, while changing its covariance matrix such that it is proportional to that of the spherically normalized point. This can be achieved with the transformation

$$\mathbf{x}^n = J_n\mathbf{x} = \mathbf{x} \qquad \Sigma_{\mathbf{x}^n\mathbf{x}^n} = J_n\Sigma_{\mathbf{xx}}J_n^\mathsf{T} \quad \text{with} \quad J_n = \frac{\partial\mathbf{x}^n}{\partial\mathbf{x}} = I_3 - \frac{\mathbf{x}\mathbf{x}^\mathsf{T}}{\mathbf{x}^\mathsf{T}\mathbf{x}}\,, \tag{10.139}$$

  resulting from the constraint $|\underline{\mathbf{x}}^n| = |\mathbf{x}|$. This stochastic vector, up to the fixed scale $\lambda = 1/|\mathbf{x}|$, is identical to $\underline{\mathbf{x}}^s$:

$$\underline{\mathbf{x}}^s = \frac{\underline{\mathbf{x}}^n}{|\mathbf{x}|}\,. \tag{10.140}$$

  Remark: This change of the stochastic properties of a stochastic vector, without the change of its values, is a special case of a gauge transformation which is used to enforce a certain coordinate system, here the scale of the vector, see Sect. 4.5.1, p. 109. ◇
- Finally, the distribution of the homogeneous vector $\mathbf{x}_1^g$ may be far from symmetric, as long as the projection to the unit sphere leads to the same distribution of $\mathbf{x}^s$.

Summarizing, we can represent uncertain geometric entities, including transformations, as uncertain homogeneous vectors by adjoining a covariance matrix to them. First-order variance propagation can be applied with practically negligible bias. As only the direction of homogeneous vectors is of concern, the covariances need not be singular. Regular covariance matrices of homogeneous entities indicate that the scale of the homogeneous vectors is uncertain, which does not affect spatial reasoning with uncertain quantities.

10.4 Evaluating Statistically Uncertain Relations

Evaluation of the relation between two geometric entities can be reduced to statistical hypothesis testing. In this section, we provide the test statistic for the most relevant cases.

Testing can be based on the following scheme:

1. Choose an adequate constraint $\boldsymbol{f}(\mathbf{a}, \mathbf{b})$ between the two geometric entities $\boldsymbol{a}$ and $\boldsymbol{b}$. The constraint is assumed to be zero if the null hypothesis, and thus the geometric relation, holds. The constraint has R degrees of freedom, indicating the minimum number of functionally independent constraints. For given values $\mathbf{a}$ and $\mathbf{b}$, the constraint either is a scalar measure d or a vector-valued measure $\boldsymbol{d}$.
2. If the degrees of freedom R are larger than 1, distinguish between two cases:

   a. When testing identities, use the reduced coordinates to advantage.
   For example, if checking the identity of two 2D points $\boldsymbol{x}$ and $\boldsymbol{y}$, we can use the difference $\boldsymbol{d}$ of the reduced coordinates,

$$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})(\mathbf{y}^s - \mathbf{x}^s)\,, \tag{10.141}$$

   and enforce the reduction to be done with the same projection as

$$\boldsymbol{d} = \boldsymbol{y}_r - \boldsymbol{x}_r = J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})\,\mathbf{y}^s - J_r^\mathsf{T}(\boldsymbol{\mu}_\mathrm{x})\,\mathbf{x}^s\,, \qquad J_r(\boldsymbol{\mu}_\mathrm{x}) = \mathsf{null}(\boldsymbol{\mu}_\mathrm{x}^\mathsf{T})\,, \tag{10.142}$$

with $\boldsymbol{\mu}_{\mathrm{x}}$, which is the mean value of the point (in practice replaced by some estimate, e.g., $\boldsymbol{\mu}_{\mathrm{x}} := \mathbf{x}$). Observe, due to the projection with $J_r(\boldsymbol{\mu}_{\mathrm{x}})$ the signs of $\mathbf{x}^s$ and $\mathbf{y}^s$ need not be the same.

b. When checking the incidence of a 3D line with a 3D point or a plane (see Table 10.4, p. 395), select two independent constraints. For the incidence constraint $\mathcal{X} \in \mathcal{L}$ with $\mathbf{D} = \overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{X} \stackrel{!}{=} \mathbf{0}$ of a 3D line and a 3D point, instead of performing a selection we directly arrive at two independent constraints by projection onto the null space of $\mathsf{\Gamma}(\boldsymbol{\mu}_L)$. The vector to be tested is

$$\underset{2\times 1}{\boldsymbol{d}} = \mathsf{null}^{\mathsf{T}}(\mathsf{\Gamma}(\boldsymbol{\mu}_{\mathrm{L}}))\,\overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{X}\,. \tag{10.143}$$

Proof: Let the singular value decomposition of $\overline{\mathsf{\Gamma}}(\mathbf{L})$ be

$$\overline{\mathsf{\Gamma}}(\mathbf{L}) = UDV^{\mathsf{T}} = s^2(\boldsymbol{u}_1\boldsymbol{u}_2^{\mathsf{T}} - \boldsymbol{u}_2\boldsymbol{u}_1^{\mathsf{T}})\,, \tag{10.144}$$

with

$$U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{a}, \boldsymbol{b}]\,, \qquad V = [\boldsymbol{u}_2, -\boldsymbol{u}_1, \boldsymbol{c}, \boldsymbol{d}]\,, \qquad D = \mathrm{Diag}([s, s, 0, 0])\,, \tag{10.145}$$

with the 4-vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ representing two planes generating $\mathbf{L}$ (see Sect. 7.3.2.2, p. 313), and the other vectors $\boldsymbol{a}$, etc., which are irrelevant. Then we have the two obviously independent constraints,

$$\begin{bmatrix} \boldsymbol{u}_1^{\mathsf{T}} \\ \boldsymbol{u}_2^{\mathsf{T}} \end{bmatrix} \overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{X} = s^2 \begin{bmatrix} \boldsymbol{u}_2^{\mathsf{T}} \\ -\boldsymbol{u}_1^{\mathsf{T}} \end{bmatrix} \mathbf{X} \stackrel{!}{=} \mathbf{0}\,. \tag{10.146}$$

As the vectors $\boldsymbol{u}_i$ span the column space of $\overline{\mathsf{\Gamma}}(\mathbf{L})$ and (due to $\overline{\mathsf{\Gamma}}(\mathbf{L})\mathsf{\Gamma}(\mathbf{L}) = \mathsf{0}$, see (7.73), p. 306) also span the null space of $\mathsf{\Gamma}(\mathbf{L})$, we arrive at (10.143). ◇

Similarly, we obtain two independent constraints for the plane–line incidence $\mathcal{L} \in \mathcal{X}$,

$$\underset{2\times 1}{\boldsymbol{d}} = \mathsf{null}^{\mathsf{T}}(\overline{\mathsf{\Gamma}}(\boldsymbol{\mu}_{\mathrm{L}}))\,\mathsf{\Gamma}(\mathbf{L})\mathbf{A}. \tag{10.147}$$

In practice, it is necessary in both cases to use some estimate for the 3D line $\boldsymbol{\mu}_{\mathrm{L}}$, e.g., the line $\mathbf{L}$ itself. Care has to be taken that $\mathbf{L}$ fulfils the Plücker constraint, as otherwize the rank of the corresponding matrices $\mathsf{\Gamma}(\mathbf{L})$ or $\overline{\mathsf{\Gamma}}(\mathbf{L})$ is 4, and the null space is empty.

3. Determine the variance or covariance matrix of the measure $\underline{d}$ or $\underline{\boldsymbol{d}}$,

$$\mathbb{D}(\underline{d}) = \sigma_d^2 \qquad \text{or} \qquad \mathbb{D}(\underline{\boldsymbol{d}}) = \Sigma_{dd}, \tag{10.148}$$

respectively. For example, when testing the identity with (10.141), we obtain the covariance matrix

$$\Sigma_{dd} = J_r^{\mathsf{T}}(\boldsymbol{\mu}_{\mathrm{x}})(\Sigma_{\mathrm{x}^s\mathrm{x}^s} + \Sigma_{\mathrm{y}^s\mathrm{y}^s})J_r(\boldsymbol{\mu}_{\mathrm{x}})\,. \tag{10.149}$$

4. Determine the test statistic,

$$T_d = \frac{d^2}{\sigma_d^2} \qquad \text{or} \qquad T_d = \boldsymbol{d}^{\mathsf{T}}\Sigma_{dd}^{-1}\boldsymbol{d}\,, \tag{10.150}$$

respectively. It always can be interpreted as a square Mahalanobis distance of d or $\boldsymbol{d}$ from the null hypothesis.
5. Choose a significance level S, e.g., 99%.
6. Determine the critical value $\chi^2_{R,S}$.
7. The null hypothesis is rejected if $T_d > \chi^2_{R,S}$; otherwize, there is no reason to reject it.

Tables 10.3 and 10.4 collect the measures d or $\boldsymbol{d}$ for some relations in 2D and 3D. The matrices $\mathsf{P}_3 = [I_2, \mathbf{0}]$ ((7.9), p. 294) and $\mathsf{P}_4 = [I_3, \mathbf{0}]$ ((7.81), p. 307) are used to handle directions of 2D lines and planes, selecting the homogeneous part $\boldsymbol{l}_h = \mathsf{P}_3\mathbf{l}$ of a 2D line vector or the homogeneous part $\boldsymbol{A}_h = \mathsf{P}_3\mathbf{A}$ of a plane. The point at infinity of the line ℓ is given by $\mathbf{e}_3 \cap \mathbf{l} = \mathsf{S}_3\mathbf{l}$. The parallelity and orthogonality of planes can be handled similarly.

Exercise 10.8

Table 10.3 Linear constraints between two geometric entities in two dimensions, including points $\mathcal{X}$ and lines $\mathcal{l}$. The degrees of freedom R are identical to the number of independent constraints. The matrix $\mathsf{P}_3 = [\mathsf{I}_2, \mathbf{0}]$ selects the homogeneous part of a line, the matrix $\mathsf{G}_3 = \mathsf{P}_3^\mathsf{T}\mathsf{P}_3 = \mathrm{Diag}([1,1,0])$ is used to measure the angle between the normals. The matrix $\mathsf{S}_3 = \mathsf{S}(\boldsymbol{e}_3)$ applied to $\mathbf{l}$ yields the point at infinity of $\mathcal{l}$

	relation in 2D		bilinear constraint	R	Eq.
1	point $\mathcal{X}$, line $\mathcal{l}$	$\mathcal{X} \in \mathcal{l}$	$d = \mathbf{x}^\mathsf{T}\mathbf{l} \stackrel{!}{=} 0$	1	(7.21)
2	two points $\mathcal{X}$, $\mathcal{Y}$	$\mathcal{X} \equiv \mathcal{Y}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_x)(\mathbf{y}^s - \mathbf{x}^s) \stackrel{!}{=} \mathbf{0}$	2	(10.141)
3	two lines $\mathcal{l}$, $\mathcal{m}$	$\mathcal{l} \equiv \mathcal{m}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_1)(\mathbf{m}^s - \mathbf{l}^s) \stackrel{!}{=} \mathbf{0}$	2	(10.141)
4		$\mathcal{l} \perp \mathcal{m}$	$d = \mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m} \stackrel{!}{=} 0$	1	(7.28)
5		$\mathcal{l} \| \mathcal{m}$	$d = \mathbf{l}^\mathsf{T}\mathsf{S}_3\mathbf{m} \stackrel{!}{=} 0$	1	(7.27)
6		$\mathcal{l} \uparrow\uparrow \mathcal{m}$	dto. and $\mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m} > 0$	1	
7		$\mathcal{l} \uparrow\downarrow \mathcal{m}$	dto. and $\mathbf{l}^\mathsf{T}\mathsf{G}_3\mathbf{m} < 0$	1	

Table 10.4 Linear constraints between two geometric entities in three dimensions, including points $\mathcal{X}$, lines $\mathcal{L}$ and planes $\mathcal{A}$. The degrees of freedom R are identical to the number of independent constraints. The matrix $\mathsf{P}_6 = [\mathsf{I}_3, \mathsf{0}_{3\times 3}]$ selects the homogeneous part of a 3D line

	relation in 3D		bilinear constraint	R	Eq.
8	point $\mathcal{X}$, plane $\mathcal{A}$	$\mathcal{X} \in \mathcal{A}$	$d = \mathbf{X}^\mathsf{T}\mathbf{A} \stackrel{!}{=} 0$	1	(7.57)
9	point $\mathcal{X}$, line $\mathcal{L}$	$\mathcal{X} \in \mathcal{L}$	$\boldsymbol{d} = \mathsf{null}(\mathsf{\Gamma}(\boldsymbol{\mu}_L))\,\mathsf{\Gamma}^\mathsf{T}(\overline{\mathbf{L}})\mathbf{X} \stackrel{!}{=} \mathbf{0}$	2	(10.143)
10	line $\mathcal{L}$, plane $\mathcal{A}$	$\mathcal{L} \in \mathcal{A}$	$\boldsymbol{d} = \mathsf{null}(\mathsf{\Gamma}(\overline{\boldsymbol{\mu}_L}))\,\mathsf{\Gamma}^\mathsf{T}(\mathbf{L})\mathbf{A} \stackrel{!}{=} \mathbf{0}$	2	(10.143)
11	two points $\mathcal{X}$, $\mathcal{Y}$	$\mathcal{X} \equiv \mathcal{Y}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_X)\,(\mathbf{Y}^s - \mathbf{X}^s) \stackrel{!}{=} \mathbf{0}$	3	(10.141)
12	two planes $\mathcal{A}$, $\mathcal{B}$	$\mathcal{A} \equiv \mathcal{B}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_A)(\mathbf{B}^s - \mathbf{A}^s) \stackrel{!}{=} \mathbf{0}$	3	(10.141)
13	two lines $\mathcal{L}$, $\mathcal{M}$	$\mathcal{L} \equiv \mathcal{M}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}_L)(\mathbf{M}^s - \mathbf{L}^s) \stackrel{!}{=} \mathbf{0}$	4	(10.141)
14		$\mathcal{L} \cap \mathcal{M} \neq \emptyset$	$d = \mathbf{L}^\mathsf{T}\overline{\mathbf{M}} \stackrel{!}{=} 0$	1	(7.58)
15		$\mathcal{L} \| \mathcal{M}$	$\boldsymbol{d} = J_r^\mathsf{T}(\boldsymbol{\mu}(\mathsf{P}_6\mathbf{L}))((\mathsf{P}_6\mathbf{M})^s - (\mathsf{P}_6\mathbf{L})^s) \stackrel{!}{=} \mathbf{0}$	2	(10.141)
16		$\mathcal{L} \uparrow\uparrow \mathcal{M}$	dto. and $\mathbf{L}^\mathsf{T}\mathsf{P}_6^\mathsf{T}\mathsf{P}_6\mathbf{M} > 0$	2	
17		$\mathcal{L} \uparrow\downarrow \mathcal{M}$	dto. and $\mathbf{L}^\mathsf{T}\mathsf{P}_6^\mathsf{T}\mathsf{P}_6\mathbf{M} < 0$	2	

10.5 Closed Form Solutions for Estimating Geometric Entities and Transformations

10.5.1 Motivation 395
10.5.2 Directly Estimating 2D Lines and Planes 396
10.5.3 Directly Estimating 2D and 3D Points 401
10.5.4 Directly Estimating Transformations 406
10.5.5 Best Fitting 3D Line 411

10.5.1 Motivation

The constructions of geometric entities, including transformations, discussed in the last few sections can be seen as special cases of procedures for estimation, however, with zero redundancy. In this section, we cover those cases where more constraints are available than necessary and where a closed form solution is still possible. We have discussed functional models and methods for direct solutions in Sect. 4.9, p. 176.

Generally, there are no direct solutions for a statistically optimal estimation. However, there are two ways to relax the problem to arrive at a closed form solution:

- An algebraically favourable function is minimized.
  Though the solutions are generally not statistically optimal in any sense, the covariance matrix of the resulting estimate can be given explicitly.
- The statistical properties of the given entities are assumed to have a special structure.
  For a small but important set of problems, closed form statistically optimal solutions are known. They are mostly least squares solutions and imply the observations are normally distributed. In most cases, the uncertainty of the basic observations must be *isotropic*, i.e., invariant to rotations of the coordinate system, or even *homogeneous*, i.e., identical for all observations. The log-likelihood function then is an expression which is quadratic in the unknown parameters, possibly with a quadratic constraint on the parameters. The resulting least squares problem usually can be partitioned into smaller sub-problems which have classical closed form solutions, such as the weighted mean or the rotation matrix best approximating a given matrix.

In all these cases, we give algorithms for parameter estimation together with explicit expressions for the covariance matrix of the estimated parameters. We also characterize those cases where the resultant covariance matrix shows additional rank defects, as this indicates critical configurations, and close to these configurations the accuracy of the result drastically deteriorates.

In general we need three types of algorithms:

1. A *direct algebraic solution*, useful for finding approximate values, in case no better algorithm is available or in case speed is of importance.
2. A *least squares solution*, which in most cases can be interpreted as a maximum likelihood estimation with a simple covariance structure for the given observations. This solution can be used as a final result if the implicitly assumed stochastical model is acceptable.
3. An *iterative maximum likelihood estimation scheme* for a stochastical model which is more general than the least squares solution, which makes it relevant in practice and allows shortcuts, or is useful as a prototype for similar estimation problems.

In this section we only give direct algebraic and least squares solutions. Iterative solutions are provided in Sect. 10.6, p. 414.

10.5.2 Directly Estimating 2D Lines and Planes

10.5.2.1 Algebraically Best Fitting 2D Line and Plane

Let I 2D points $\mathcal{X}_i, i = 1, ..., I$, be given. The line $\mathcal{l}$ best fitting the constraints $\mathcal{X}_i \in \mathcal{l}$ is given by minimizing the length of

$$\mathbf{c}(\mathbf{l}) = \underbrace{\begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \dots \\ \mathbf{x}_i^\mathsf{T} \\ \dots \\ \mathbf{x}_I^\mathsf{T} \end{bmatrix}}_{\mathsf{A}} \mathbf{l}, \tag{10.151}$$

with the constraint $|\mathbf{l}| = 1$. Observe, we treat the aggregation of the given vectors $\mathbf{x}_i$ and also the residual $\mathbf{c}$ as homogeneous entities.

Though we give a statistically motivated best solution in the next section, we can easily see that the solution here is neither statistically motivated, as no covariance information is introduced, nor unique, as each of the rows of the matrix A is a homogeneous vector and can be multiplied with an arbitrary scalar. Of course, conditioning and normalization are

recommended, cf. Sects. 6.8, p. 285, and 6.9, p. 286, and will eliminate this arbitrariness, except for the degree of the conditioning and the type of the normalization.

But the setup of this solution easily allows us to introduce additional constraints on ℓ. For example, assume the line ℓ is to pass through a set of J other observed lines $m_j, j = 1, ..., J$, and is to be parallel to another line n, see Fig. 10.20. Then we introduce

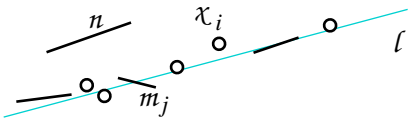


Fig. 10.20 Example for estimating a line with constraints of different modes: Fitting a line ℓ through points x_i and line segments m_j so that it is parallel to another line segment n can be achieved with a closed form solution

the constraints with the residual vectors,

$$\mathbf{c}_1(\mathbf{l}) = \underbrace{\begin{bmatrix} \mathsf{S}(\mathbf{m}_1) \\ \dots \\ \mathsf{S}(\mathbf{m}_j) \\ \dots \\ \mathsf{S}(\mathbf{m}_J) \end{bmatrix}}_{\mathsf{A}_1} \mathbf{l} \qquad \text{and} \qquad \mathbf{c}_2(\mathbf{l}) = \underbrace{\mathbf{n}^\mathsf{T}\mathsf{G}_3}_{\mathbf{a}_2^\mathsf{T}} \mathbf{l}, \tag{10.152}$$

by applying the parallelity constraint (7.27), p. 297. Minimizing $|\mathbf{c}(\mathbf{l})^\mathsf{T}, \mathbf{c}_1(\mathbf{l})^\mathsf{T}, \mathbf{c}_2(\mathbf{l})|$ under the constraint $|\mathbf{l}| = 1$ yields the algebraically best solution with the help of the singular value decomposition of the composed matrix $[\mathsf{A}^\mathsf{T}, \mathsf{A}_1^\mathsf{T}, \mathbf{a}_2]^\mathsf{T}$. Again, conditioning and normalization are necessary.

The solution setup directly transfers to the problem of finding the algebraically best fitting plane through I 3D points $\mathcal{X}_i, i = 1, ..., I$. Alternatively, or in addition, we could take observed 3D lines $\mathcal{L}_j$ to lie in the unknown plane, or enforce parallelity or orthogonality to a given line or plane, as all these constraints are linear in the plane parameters.

10.5.2.2 Statistically Best Fitting 2D Line and Plane

Given a set of I 2D points, a statistically best fitting line can be determined if the covariance matrices of the statistically independent points are multiples of a unit matrix, i.e., if the coordinates of each point are mutually statistically independent and have the same standard deviation, σ_i. Let the uncertain points be given by

$$x_i : \qquad \left\{\boldsymbol{x}_i, \sigma_i^2 I_2\right\} ; \tag{10.153}$$

then the parameters of the best fitting line minimize the weighted sum of squared distances of the points x_i from the line ℓ,

$$\Omega(\mathbf{l}) = \sum_{i=1}^{I} w_i \, d^2(\mathbf{l}, \boldsymbol{x}_i) \, . \tag{10.154}$$

In the following, we will use the weights w_i and the mean weight $\overline{w}$,

$$w_i = \frac{1}{\sigma_i^2} \qquad \overline{w} = \frac{\sum_{i=1}^{I} w_i}{I} \, . \tag{10.155}$$

We obtain the following intuitive result of the Maximum Likelihood estimate for the line parameters, see Fig. 10.21.[6]

1. The statistically best line $\widehat{\ell}$ passes through the *weighted centroid* of the given points $\mathcal{X}_i$ (see Fig. 10.12, p. 374),
$$\boldsymbol{x}_0 = \frac{\sum_{i=1}^{I} w_i \boldsymbol{x}_i}{I\overline{w}} \,. \tag{10.156}$$

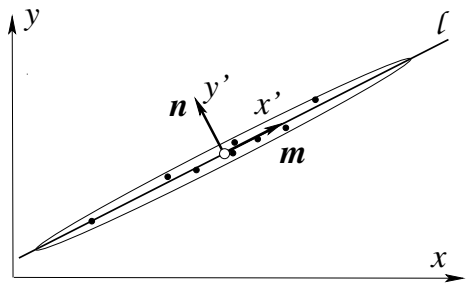


Fig. 10.21 The statistically best fitting line through eight points passes through the centroid and has the direction of the principal axis of the moment matrix of the points. Shown are eight points, their centroid, and the ellipse, representing the moment matrix, with their axes. Their lengths are identical to the square roots of the eigenvalues λ_i of the moment matrix of the point set

2. The *orientation* of the best line can be determined from the moment matrix of the 2D points,
$$M = \sum_{i=1}^{I} w_i (\boldsymbol{x}_i - \boldsymbol{x}_0)(\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T} = \left(\sum_{i=1}^{I} w_i \boldsymbol{x}_i \boldsymbol{x}_i^\mathsf{T} - I\overline{w}\boldsymbol{x}_0 \boldsymbol{x}_0^\mathsf{T} \right) . \tag{10.157}$$
With the eigenvector decomposition $M = \lambda_1 \boldsymbol{m}\boldsymbol{m}^\mathsf{T} + \lambda_2 \boldsymbol{n}\boldsymbol{n}^\mathsf{T}$, namely the eigenvectors $[\boldsymbol{m}, \boldsymbol{n}]$ belonging to the eigenvalues $[\lambda_1, \lambda_2], \lambda_1 \geq \lambda_2$, the direction of the line is $\boldsymbol{m}$.

best fitting 2D line

We finally obtain the estimated line as $\boldsymbol{n}^\mathsf{T}(\boldsymbol{x} - \boldsymbol{x}_0) = 0$, or, equivalently,
$$\widehat{\mathbf{l}} = \begin{bmatrix} n_1 \\ n_2 \\ -\boldsymbol{n}^\mathsf{T}\boldsymbol{x}_0 \end{bmatrix} . \tag{10.158}$$

The eigenvector $\boldsymbol{n}$, up to its sign, at the same time is the normal of the unknown line.

For numerical reasons, it is advisable to determine $\boldsymbol{n}$ from the SVD $X = USV^\mathsf{T}$ of the $I \times 2$ matrix $X = [\sqrt{w_i}(\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}]$: The estimated normal is the right singular vector $\boldsymbol{v}_i, i = \operatorname{argmin}_j(s_j)$, belonging to the smallest singular value s_i of X.

Proof: With the line parameters $\mathbf{l} = [\boldsymbol{l}_h^\mathsf{T}, l_0]^\mathsf{T}$, an unknown point $\mathcal{X}_0(\boldsymbol{x}_0)$ on the line, and the distance $d_i = \boldsymbol{l}_h^\mathsf{T}(\boldsymbol{x}_i - \boldsymbol{x}_0)$ of the point $\mathcal{X}_i$ from the line, the optimization function (10.154) can be written as
$$\Omega(\boldsymbol{l}_h, \boldsymbol{x}_0) = \sum_{i=1}^{I} w_i \boldsymbol{l}_h^\mathsf{T}(\boldsymbol{x}_i - \boldsymbol{x}_0)(\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}\boldsymbol{l}_h = \boldsymbol{l}_h^\mathsf{T} \underbrace{\sum_{i=1}^{I} w_i (\boldsymbol{x}_i - \boldsymbol{x}_0)(\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}}_{M} \boldsymbol{l}_h \,. \tag{10.159}$$

To determine $\boldsymbol{x}_0$, we differentiate $\Omega(\boldsymbol{l}_h, \boldsymbol{x}_0)$ with respect to $\boldsymbol{x}_0$:

[6] The ellipse shown in the figure is actually the standard ellipse of the empirical covariance or scatter matrix M of the point set magnified by a factor of 3. The covariance matrix of the point set is related to the moment matrix by $\widehat{\Sigma} = I/(I-1)\, M/\sum_i w_i$, cf. (4.356), p. 140.

$$\frac{1}{2}\frac{\partial \Omega}{\partial \boldsymbol{x}_0} = \boldsymbol{l}_h^\mathsf{T} \frac{\partial \sum_{i=1}^{I} w_i (\boldsymbol{x}_i - \boldsymbol{x}_0)(\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T}}{\partial \boldsymbol{x}_0} \boldsymbol{l}_h \tag{10.160}$$

$$= \boldsymbol{l}_h^\mathsf{T} \left(\sum_{i=1}^{I} -2 w_i (\boldsymbol{x}_i - \boldsymbol{x}_0)^\mathsf{T} \right) \boldsymbol{l}_h \tag{10.161}$$

$$= \boldsymbol{0}^\mathsf{T} . \tag{10.162}$$

This constraint certainly is fulfilled if we choose $\boldsymbol{x}_0$ to be the weighted centroid. As only the projection onto $\boldsymbol{l}_h$, i.e., the product of the two right terms, is relevant, any other point $\boldsymbol{x}_0 + t\boldsymbol{d}$ on the line, with the direction $\boldsymbol{d} \perp \boldsymbol{l}_h$ and some arbitrary parameter t, would also fulfil the constraint. Now, as $\boldsymbol{x}_0$ is known, we can determine $\boldsymbol{l}_h$ as the eigenvector belonging to the smallest eigenvalue of the moment matrix $\boldsymbol{M}$. ◇

An expression for the *covariance matrix* of the parameters can be derived by centring the data and rotating the coordinate system such that the direction of the line is zero. This is admissible, as the uncertainty of the points was assumed to be isotropic, i.e., their covariance matrix is invariant to translations and rotations. It allows us to use the eigenvalues of the moment matrix, which are invariant to translation and rotation.

In this configuration, see x', y' in Fig. 10.21, the line then has the classical form $\mathbb{E}(\underline{y}'_i) = k + mx'_i$ or $\mathbb{E}(\underline{y}'_i) = [1, x'_i][k, m]^\mathsf{T}$, where the first parameter $k = y'_0$ is the component of the centre across the line and the second parameter m is the tangent of the angle of the line to the x'-axis, which approximately equals the angle due to $\alpha \approx \tan \alpha$ for small angles.

The position q of the centre across the line and the direction α have variance, omitting the hat on these derived entities for simplicity

$$\sigma_q^2 = \sigma_{\widehat{k}}^2 = \frac{1}{N_{11}} \qquad \sigma_\alpha^2 = \sigma_{\widehat{m}}^2 = \frac{1}{N_{22}} \tag{10.163}$$

(cf. (4.49), p. 86) with the a priori variance factor $\sigma_0^2 = 1$ and the diagonal elements

$$N_{11} = \sum_{i=1}^{I} w_i = I\overline{w} \qquad N_{22} = \sum_{i=1}^{I} w_i x_i'^2 \tag{10.164}$$

of the normal equation system, cf. (4.39), p. 84. Observe, the precision of the direction α of the line and the direction ϕ of the normal are the same. The two estimated parameters $\widehat{k}$ and $\widehat{m}$ are statistically independent, as the off diagonal element $N_{12} = \sum_i w_i x'_i$ of the normal equation system is zero due to the rotation into the x-axis,

We now use the eigenvalues of the diagonal moment matrix

$$\lambda_1 = \sum_{i=1}^{I} w_i x_i'^2 \qquad \lambda_2 = \sum_{i=1}^{I} w_i y_i'^2 , \tag{10.165}$$

which are identical to those in the original coordinate system. Then the theoretical variances of the two parameters are

$$\sigma_q^2 = \frac{1}{I\overline{w}} \qquad \sigma_\alpha^2 = \frac{1}{\lambda_1} . \tag{10.166}$$

Hence, the directional precision only depends on the average weighted distance of the given points from the centroid, independent of their distribution along the line.

In the case we have enough data, say $I > 30$, and if no outliers are present, we can use the estimated variance factor

$$\widehat{\sigma}_0^2 = \frac{\sum_{i=1}^{I} w_i y_i'^2}{I-2} = \frac{\lambda_2}{I-2} . \tag{10.167}$$

Due to $N_{22} = \lambda_1$, the estimated variances of the estimated parameters,

$$\widehat{\sigma}_q^2 = \widehat{\sigma}_{\widehat{k}}^2 = \frac{\widehat{\sigma}_0^2}{N_{11}} \qquad \widehat{\sigma}_\alpha^2 = \sigma_{\widehat{m}}^2 = \frac{\widehat{\sigma}_0^2}{N_{22}} \tag{10.168}$$

uncertainty of best fitting 2D line

now read as

$$\widehat{\sigma}_q^2 = \widehat{\sigma}_{\hat{k}}^2 = \frac{1}{I-2}\,\frac{\lambda_2}{I\overline{w}} \qquad \widehat{\sigma}_\phi^2 = \widehat{\sigma}_\alpha^2 = \sigma_{\hat{m}}^2 = \frac{1}{I-2}\,\frac{\lambda_2}{\lambda_1}\,. \tag{10.169}$$

If the I points are equally spaced along the line with average distance Δs then we obtain

$$\sigma_q = \frac{\sigma_0}{\sqrt{I}} \qquad \sigma_\phi = \widehat{\sigma}_\alpha^2 = \frac{\sigma_0}{\Delta s}\sqrt{\frac{12}{I^3-I}}\,. \tag{10.170}$$

Moreover, the fitted first and the last points of the sequence are correlated. Their standard deviations σ_{q_j}, $j = 1, 2$, across the line and the correlation coefficient are

Exercise 10.13

$$\sigma_{q_1} = \sigma_{q_2} = 2\sigma_q \qquad \rho_{q_1 q_2} = -\frac{1}{2}\,. \tag{10.171}$$

The expression for the variance of the direction is an approximation, since we used $\alpha \approx \tan\alpha$, and only holds if the random noise is not too large, i.e., if the ratio λ_1/λ_2 of the two eigenvalues is large.

If this ratio is 1, the distribution of the point cloud is isotropic and the direction is completely uncertain, since the directions of the eigenvectors of the moment matrix are undetermined; its distribution is uniform in the range $[-\pi/2, +\pi/2]$, thus formally has a standard deviation of $\pi/\sqrt{12} \approx 0.907 \approx 52°$. Figure 10.22 exemplifies the relation between the standard deviation σ_α and the ratio $a/b = \sqrt{\lambda_1/\lambda_2}$ of the semi-axes of the ellipse representing the moments matrix (cf. Fig. 2.11, p. 56) for a set of $I = 20$ points. For ratios a/b above 2.5, the approximation $\sigma_\alpha \approx a/b/\sqrt{I-2}$ in (10.169) appears to be sufficiently good.

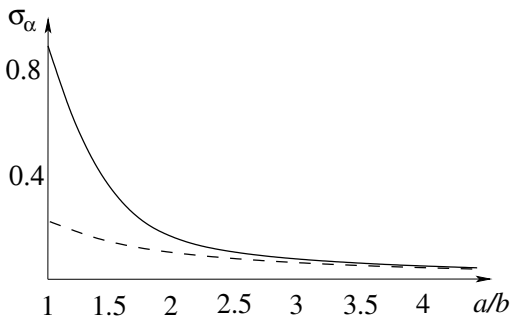


Fig. 10.22 Standard deviation σ_α of the direction of the semi axis of a 2D point cloud as a function of the ratio $a/b = \sqrt{\lambda_1/\lambda_2}$ of the semi axes (bold line) determined by simulation. Approximation $\sigma_\alpha = a/(b\sqrt{(I-2)})$ (dashed). The correct maximum standard deviation at $a/b = 1$ is $\pi/\sqrt{12} = 0.9 \approx 52°$ compared to the value $1/sqrt(20) \approx 13°$

The covariance matrix of the parameters of the line in Hessian normal form can now be derived using (10.45) to (10.47), p. 374. The covariance matrix of the homogeneous line parameters can be derived using (10.49, p. 375ff.)

Algorithm 8 summarizes the essential steps.

The procedure can be directly transferred to estimate the parameters of a plane from given 3D points in a statistically optimal manner if the 3D points are mutually uncorrelated and the coordinates $\boldsymbol{X}_i$ of each point have covariance matrix $\sigma_i^2 I_3$. The plane passes through the weighted centroid and its normal is the eigenvector belonging to the smallest eigenvalue of the empirical covariance or moment matrix M of the point set. The uncertainty of the plane can again be derived from the eigenvalues of M.

Exercise 10.28

Algorithm 8: Direct least squares estimation of a 2D line from mutually independent uncertain 2D points with isotropic accuracy
$[\boldsymbol{x}_0, \alpha; \sigma_q, \sigma_\alpha, \widehat{\sigma}_0^2, R]$=`direct_LS_2D_line_from_points`$(\{\boldsymbol{x}, \sigma\}_i)$
Input: list of 2D points $\{\boldsymbol{x}_i, \sigma_i\}, i = 1, ..., I \geq 2$.
Assumption: coordinates are conditioned.
Output: best fitting 2D line in centroid form $\ell(\boldsymbol{x}_0, \alpha; \sigma_q, \sigma_\alpha)$, estimated variance factor $\widehat{\sigma}_0^2$.

1 Redundancy $R = I - 2$;
2 **if** $R < 0$ **then stop**, not enough points;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$, mean weight $\bar{w}$;
4 Weighted centroid $\boldsymbol{x}_0$, (10.156);
5 Weighted moment matrix M, (10.157);
6 Eigenvector decomposition: $[V, \Lambda] = \mathtt{eig}(M)$;
7 Normal $\boldsymbol{n}$: eigenvector $\boldsymbol{v}_2$ belonging to smallest singular value λ_2;
8 Direction of normal $\alpha = \operatorname{atan2}(n_2, n_1) - \pi/2$;
9 Standard deviations: $\sigma_q = 1/\sqrt{\sum_i w_i}$ and $\sigma_\alpha = 1/\sqrt{\lambda_1}$;
10 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \lambda_2/R$ **else** $\widehat{\sigma}_0^2 = 1$.

10.5.3 Directly Estimating 2D and 3D Points

10.5.3.1 Algebraically Best Intersection Point of Lines and Planes

In 2D, the best fitting intersection point $\mathcal{X}$ can be determined via the technique shown in Sect. 10.5.2.1, p. 396, just by exchanging the roles of points and lines following the principle of duality.

In 3D, the *intersection point* $\mathcal{X}$ of I 3D lines $\mathcal{L}_i$ and J planes $\mathcal{A}_j$ needs to fulfil the constraints $\overline{\Gamma}(\tilde{\mathbf{L}}_i)\mathbf{X} = \mathbf{0}$ and $\tilde{\mathbf{A}}_j^\mathsf{T}\mathbf{X} = 0$, respectively. Algebraic minimization therefore leads to the estimated coordinates $\widehat{\mathbf{X}}$ as the right singular vector belonging to the smallest singular value of the $(4I + J) \times 4$ matrix

$$\begin{bmatrix} \{\overline{\Gamma}(\mathbf{L}_i)\} \\ \{\mathbf{A}_j^\mathsf{T}\} \end{bmatrix}, \tag{10.172}$$

assuming proper conditioning and normalization. Observe, the solution slightly depends on whether we use all constraints, as in (10.172), only selected algebraically independent constraints, taking $\overline{\Gamma}^{(s)}(\mathbf{L}_i)$, cf. (7.4.1.3), p. 319, or reduced constraints, taking $\mathsf{null}^\mathsf{T}(\Gamma(\mathbf{L}))\,\overline{\Gamma}(\mathbf{L})$, cf. (10.143), p. 394. The differences are statistically not significant.

Again, these algebraic solutions are useful for determining in closed form good approximate coordinates under a great variety of conditions; however, this comes at the expense of not achieving the statistically optimal estimate.

10.5.3.2 Least Squares Solution for Intersecting 2D and 3D Lines with Fixed Directions

A non-iterative optimal determination of an intersection point of I 2D lines ℓ_i is feasible if the lines show no directional, but only positional and normally distributed uncertainty across the line. So we can represent the lines by an arbitrary point $\boldsymbol{x}_i$ with positional uncertainty $\sigma_i^2 = 1/w_i$ only across the line, in a normalized direction $\boldsymbol{d}_i$, and with infinite uncertainty along the line (see Fig. 10.23):

$$\ell_i : \quad \{\mathbf{x}_i, \boldsymbol{d}_i, \sigma_i\} \quad \text{or} \quad \{\mathbf{x}_i, W_i\} \tag{10.173}$$

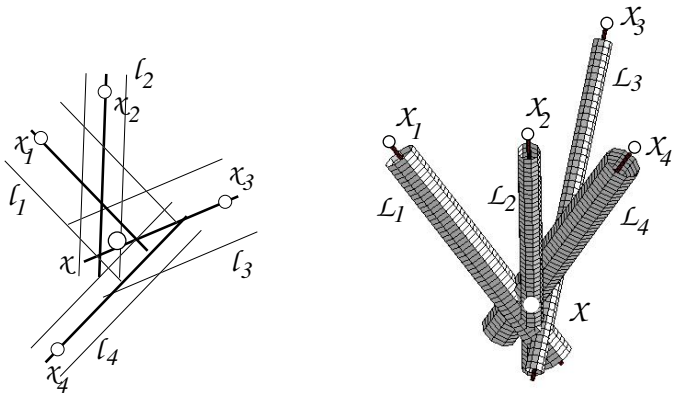


Fig. 10.23 Conditions for a closed form solution for a statistically optimal estimation of an intersection point. The directions of the lines have to be certain with possibly different positional accuracy. **Left:** 2D intersection point $\mathcal{X}$ of four lines $\ell_i(\boldsymbol{x}_i, \boldsymbol{d}_i)$ with parallel confidence bands. **Right:** 3D intersection point $\mathcal{X}$ from four lines $\mathcal{L}_i(\boldsymbol{X}_i, \boldsymbol{D}_i)$ with confidence cylinders

with

$$W_i = \frac{1}{\sigma_i^2}(I_2 - \boldsymbol{d}_i \boldsymbol{d}_i^\mathsf{T}) \,. \tag{10.174}$$

The intersection point $\mathcal{X}$ minimizes the weighted squared Euclidean distance

$$\widehat{\boldsymbol{x}} = \operatorname{argmin}_x \sum_{i=1}^{I} \left(\frac{d(\ell_i, \mathcal{X})}{\sigma_i} \right)^2 \tag{10.175}$$

with

$$\frac{d^2(\mathcal{X}, \ell_i)}{\sigma_i^2} = \frac{1}{\sigma_i^2} \left[|(\boldsymbol{x}_i - \boldsymbol{x})|^2 - |(\boldsymbol{x}_i - \boldsymbol{x})^\mathsf{T} \boldsymbol{d}_i|^2 \right] = (\boldsymbol{x}_i - \boldsymbol{x})^\mathsf{T} W_i (\boldsymbol{x}_i - \boldsymbol{x}) \,. \tag{10.176}$$

It is the weighted mean of the points $\boldsymbol{x}_i$ (cf. Antone and Teller, 2002, Eq. (18)),

$$\widehat{\boldsymbol{x}} = \left(\sum_{i=1}^{I} W_i \right)^{-1} \sum_{i=1}^{I} W_i \boldsymbol{x}_i \quad \text{with} \quad \widehat{\Sigma}_{\widehat{x}\widehat{x}} = \widehat{\sigma}_0^2 \left(\sum_{i=1}^{I} W_i \right)^{-1} , \tag{10.177}$$

and the empirical variance factor,

$$\widehat{\sigma}_0^2 = \frac{1}{I-2} \sum_{i=1}^{I} \left(\frac{d(\ell_i, \widehat{\mathcal{X}})}{\sigma_i} \right)^2 . \tag{10.178}$$

Example 10.5.38: ***Relation to the structure tensor and to junction estimation.*** Observe, if the lines are given by their normalized normal vectors $\boldsymbol{n}_i$, we have $I_2 - \boldsymbol{d}_i \boldsymbol{d}_i^\mathsf{T} = \boldsymbol{n}_i \boldsymbol{n}_i^\mathsf{T}$, and the solution is given by

$$\widehat{\boldsymbol{x}} = \left(\sum_{i=1}^{I} w_i \boldsymbol{n}_i \boldsymbol{n}_i^\mathsf{T} \right)^{-1} \sum_{i=1}^{I} w_i \boldsymbol{n}_i \boldsymbol{n}_i^\mathsf{T} \boldsymbol{x}_i \,. \tag{10.179}$$

This situation arises when locating junction points in a window of a digital image $g(\boldsymbol{x})$ as intersection points of all edge elements with position $\boldsymbol{x}_i$ and normal $\boldsymbol{n}_i = \nabla g(\boldsymbol{x}_i)$, at the same time weighting each edge element by its squared gradient $w_i = |\nabla g_i|^2$. The matrix in round brackets is what is called the *structure tensor* of that image window (cf. Förstner and Gülch, 1987), which is decisive for the precision of image matching. ◇

direct solution for 3D point from lines

Algorithm 9 summarizes the essential steps. The procedure for directly determining an intersection point in 2D can be transferred to the intersection of 3D lines (see Fig. 10.23), namely by assuming that their position is uncertain and their direction is certain

Algorithm 9: Direct least squares estimation of a 2D intersection point from lines with only positional uncertainty
$[\widehat{\boldsymbol{x}}, \widehat{\Sigma}_{\widehat{x}\widehat{x}}, \widehat{\sigma}_0^2, R]$ = `direct_LS_2D_point_from_lines`$(\{\boldsymbol{x}, \boldsymbol{d}, \sigma\}_i)$
Input: list of 2D lines in point–direction form $l_i\{\boldsymbol{x}_i, \boldsymbol{d}_i, \sigma_i\}, \boldsymbol{d}_i^\mathsf{T}\boldsymbol{d}_i = 1, i = 1, ..., I \geq 2$, $|\boldsymbol{d}_i| = 1$.
Assumption: the coordinates are conditioned.
Output: best fitting 2D intersection point $\chi(\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}})$, estimated variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = I - 2$;
2 **if** $R < 0$ **then stop**, not enough lines;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$;
4 Weight matrices $W_i = w_i(I_2 - \boldsymbol{d}_i\boldsymbol{d}_i^\mathsf{T}), i = 1, ..., I$;
5 Normal equation matrix $N = \sum_i W_i$;
6 Covariance matrix $\Sigma_{\widehat{x}\widehat{x}} = N^{-1}$;
7 Fitted point $\widehat{\boldsymbol{x}} = \Sigma_{\widehat{x}\widehat{x}} \sum_i W_i \boldsymbol{x}_i$;
8 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \sum_i w_i d(l_i, \widehat{\chi})/R$, see (10.176) **else** $\widehat{\sigma}_0^2 = 1$.

and representing them using an arbitrary reference point $\mathcal{X}_i \in \mathcal{L}_i$ and a fixed normalized direction $\boldsymbol{D}_i$ using the precision matrices $W_i = (I_3 - \boldsymbol{D}_i\boldsymbol{D}_i^\mathsf{T})/\sigma_i^2$ (cf. Sect. 10.2.2.1, p. 367 . This situation arises when determining the 3D coordinates of points for given image points and image orientations (cf. Sect. 13.4.1, p. 596). However, it shows large bias if the angles between the rays are small, cf. the comparison in Sect. 10.6.2.1, p. 419.

direct solution for 3D point from planes

The intersection of planes $\mathcal{A}_i$ with certain normals $\boldsymbol{N}_i$ but uncertain positions across the plane with standard deviations σ_{q_i} can be handled similarly, by using the representation with reference points $\mathcal{X}_i \in \mathcal{A}_i$ and their precision matrix

$$W_i = \boldsymbol{N}_i\boldsymbol{N}_i^\mathsf{T}/\sigma_{q_i}^2 \tag{10.180}$$

and (10.179).

10.5.3.3 Statistically Best Mean Direction

Directions $\boldsymbol{d}$ are represented as unit vectors, so they are elements of spherically normalized vectors in oriented projective geometry. Directions play a role in themselves, but in $\mathbb{R}^2$ or $\mathbb{R}^4$, they can also be used to represent rotations.

Given I normalized direction vectors $\boldsymbol{d}_i \in \mathbb{R}^n$ with isotropic directional uncertainty, their covariance matrices

$$\Sigma_{d_i d_i} = \sigma_i^2(I_n - \boldsymbol{d}_i\boldsymbol{d}_i^\mathsf{T}) \tag{10.181}$$

can be derived from (10.27) and (10.28), p. 371, with their isotropic directional uncertainty represented by the covariance matrix $\Sigma_{d_{r_i} d_{r_i}} = \sigma_i^2 I_{n-1}$ of the reduced homogeneous coordinates.

The standard ellipsoid thus is a flattened ball with normal parallel to $\boldsymbol{d}_i$. Minimizing the sum of the weighted squared distances, which depend on some distance s of two directions specified below

$$\widehat{\boldsymbol{d}} = \text{argmin}_d \sum_{i=1}^{I} \left(\frac{s(\boldsymbol{d}_i, \boldsymbol{d})}{\sigma_i} \right)^2 \tag{10.182}$$

leads to the optimal estimate for the direction

$$\widehat{\boldsymbol{d}} = N \left(\frac{\sum_{i=1}^{I} w_i \boldsymbol{d}_i}{\sum_{i=1}^{I} w_i} \right) \qquad w_i = \frac{1}{\sigma_i^2} . \tag{10.183}$$

The covariance matrix of the normalized vector $\widehat{\boldsymbol{d}}$ is given by

$$\widehat{\Sigma}_{\widehat{d}\widehat{d}} = \frac{\widehat{\sigma}_0^2}{\sum_{i=1}^{I} w_i} (I_n - \widehat{\boldsymbol{d}}\widehat{\boldsymbol{d}}^\mathsf{T}) , \tag{10.184}$$

with the estimated variance factor

$$\widehat{\sigma}_0^2 = \frac{1}{I-(n-1)} \sum_{i=1}^{I} w_i \, s^2(\boldsymbol{d}_i, \widehat{\boldsymbol{d}}) , \tag{10.185}$$

where we use the distance function

$$s^2(\boldsymbol{d}_i, \widehat{\boldsymbol{d}}) = |\boldsymbol{d}_i - \widehat{\boldsymbol{d}}|^2 . \tag{10.186}$$

Since we have approximation $s^2(\alpha_i) \approx \alpha_i^2$ for the distance function if the angles $\alpha_i = \angle(\boldsymbol{d}_i, \widehat{\boldsymbol{d}})$ are small, the sum of the weighted squares of the angles is also minimized.

Proof: The proof exploits the minimal parametrization of homogeneous vectors introduced in Sect. 10.2.2.1, p. 369. We interpret the directions $\boldsymbol{d}_i$ as spherically normalized homogeneous vectors and express the optimization problem with the minimal parametrization $\boldsymbol{d}_{ri}$ in the tangent space at some appropriate approximate value for the estimate. Specifically, we use $\widehat{\boldsymbol{d}}^a = \widehat{\boldsymbol{d}}$ as approximate value for the mean direction and show that this approximate value is the statistically optimal value.

We first reduce the observed directions

$$\boldsymbol{d}_{ri} = \boldsymbol{J}_r^\mathsf{T}(\widehat{\boldsymbol{d}}^a) \, \boldsymbol{d}_i . \tag{10.187}$$

Then we have $\widehat{\boldsymbol{d}}_r^a = \boldsymbol{J}_r^\mathsf{T}(\widehat{\boldsymbol{d}}^a)\widehat{\boldsymbol{d}}^a = \mathsf{null}(\widehat{\boldsymbol{d}}^{a\mathsf{T}})\widehat{\boldsymbol{d}}^a = \boldsymbol{0}$. Minimizing

$$\Omega = \sum_{i=1}^{I} w_i (\boldsymbol{d}_{ri} - \widehat{\boldsymbol{d}_r})^\mathsf{T} (\boldsymbol{d}_{ri} - \widehat{\boldsymbol{d}_r}) \tag{10.188}$$

leads to

$$\widehat{\boldsymbol{d}_r} = \frac{\sum_{i=1}^{I} w_i \boldsymbol{d}_{ri}}{\sum_{i=1}^{I} w_i} , \tag{10.189}$$

which is 0 due to (10.187) and (10.183), confirming $\widehat{\boldsymbol{d}}^a = \widehat{\boldsymbol{d}}$ is the optimal estimate. The weighted sum of squared residuals (10.188) is equal to the sum in (10.185) due to $\boldsymbol{d}_r = \boldsymbol{0}$ and $|\boldsymbol{d}_{ri}| \approx \tan(\angle(\boldsymbol{d}_i, \widehat{\boldsymbol{d}}))$. Therefore, the covariance matrix of the mean of the $\boldsymbol{d}_{ri}$ is $\boldsymbol{\Sigma}_{\widehat{d}_r\widehat{d}_r} = \sigma_0^2 / \sum_i w_i \, \boldsymbol{I}_2$, and hence, with (10.28), p. 371 we obtain (10.184). ◇

Algorithm 10 summarizes the procedure for the least squares estimation of the mean direction.

Algorithm 10: Direct least squares estimation of the mean of directions with isotropic directional uncertainty.
$[\widehat{\boldsymbol{d}}, \boldsymbol{\Sigma}_{\widehat{d}\widehat{d}}, \widehat{\sigma}_0^2, R]$=`direct_LS_mean_direction`$(\{\boldsymbol{d}_i,\sigma\}_i)$
Input: list of I directions $\in \mathbb{R}^n$: $\mathcal{d}_i\{\boldsymbol{d}_i, \sigma_i[\text{rad}]\}, i = 1, ..., I$, $I \geq n-1$, $|\boldsymbol{d}_i| = 1$.
Output: mean direction $\mathcal{d}(\widehat{\boldsymbol{d}}, \boldsymbol{\Sigma}_{\widehat{d}\widehat{d}})$, estimated variance factor $\widehat{\sigma}_0^2$.

1 Redundancy $R = I - (n-1)$;
2 **if** $R < 0$ **then stop**, not enough directions;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$;
4 Sum of weights $S_w = \sum_i w_i$;
5 Mean direction $\widehat{\boldsymbol{d}} = \mathsf{N}(\sum_i w_i \boldsymbol{d}_i)/S_w$;
6 Covariance matrix $\boldsymbol{\Sigma}_{\widehat{d}\widehat{d}} = (\boldsymbol{I}_n - \widehat{\boldsymbol{d}}\widehat{\boldsymbol{d}}^\mathsf{T})/S_w$;
7 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \sum_i w_i |\boldsymbol{d}_i - \widehat{\boldsymbol{d}}|^2 / R$ **else** $\widehat{\sigma}_0^2 = 1$.

10.5.3.4 Statistically Best Mean Axis

Axes $\boldsymbol{a} \in \mathbb{R}^n$ are homogeneous unit vectors, and, in contrast to directions, the vectors $\boldsymbol{a}$ and $-\boldsymbol{a}$ represent the same axis. Hence, they are elements in projective space $\mathbb{P}^{n-1}$. Axes play a role in themselves, but as quaternions in $\mathbb{P}^3$ they can also be used to represent rotations.

We assume I axes are given by their normalized vectors $\boldsymbol{a}_i \in R^n, i = 1, ..., I \geq n-1$, and their signs are not known. They are assumed to represent the same unknown axis $\boldsymbol{a}(\boldsymbol{a})$. Using the same approach as for directions, we minimize

$$\Omega(\boldsymbol{a}) = \sum_{i=1}^{I} w_i \sin^2 \alpha_i \,, \tag{10.190}$$

where $\alpha_i = \angle(\boldsymbol{a}_i, \boldsymbol{a})$ is the angle between $\boldsymbol{a}_i$ and $\boldsymbol{a}$. This is equivalent to maximizing

$$\overline{\Omega}(\boldsymbol{a}) = \boldsymbol{a}^\mathsf{T} \left(\sum_i w_i \boldsymbol{a}_i \boldsymbol{a}_i^\mathsf{T} \right) \boldsymbol{a} \,. \tag{10.191}$$

The optimal $\widehat{\boldsymbol{a}}$ therefore is given by the eigenvector corresponding to the largest eigenvalue of the weighted moment matrix

$$M = \sum_{i=1}^{I} w_i \boldsymbol{a}_i \boldsymbol{a}_i^\mathsf{T} \,. \tag{10.192}$$

Proof: Minimizing $\Omega(\boldsymbol{a})$ is identical to maximizing $\sum_i w_i \cos^2 \alpha_i$. But as $\cos^2 \alpha_i = \cos^2 \angle(\boldsymbol{a}_i, \boldsymbol{a}) = (\boldsymbol{a}_i^\mathsf{T} \boldsymbol{a})^2 = \boldsymbol{a}^\mathsf{T} \boldsymbol{a}_i \boldsymbol{a}_i^\mathsf{T} \boldsymbol{a}$, we need to maximize $\overline{\Omega}(\boldsymbol{a}) = \boldsymbol{a}^\mathsf{T} M \boldsymbol{a}$ with the matrix $M = \sum_{i=1}^{I} w_i \boldsymbol{a}_i \boldsymbol{a}_i^\mathsf{T}$. ◇

The covariance matrix has the same structure as the one for directions, except that the variance factor needs to be determined differently. The estimated variance factor is given by

$$\widehat{\sigma}_0^2 = \frac{\Omega}{I-(n-1)} = \frac{I\overline{w} - \overline{\Omega}(\widehat{\boldsymbol{a}})}{I-(n-1)} = \frac{I\overline{w} - \lambda_1}{I-(n-1)} \,, \tag{10.193}$$

where n is the dimension of the vectors $\boldsymbol{a}_i$ and λ_1 is the largest eigenvalue of M.

If the observed axes are aligned, i.e., they are directions, the two estimation procedures for a mean direction and a mean axis lead to the same result provided that the relative precision of the directions is sufficiently large, say the standard deviations of the directions are below 1/10 rad $\approx 6°$. Algorithm 11 summarizes the procedure for the direct least squares estimation of a mean axis.

Observe, Algorithm 11 can also be used for averaging directions. But it is slower than Algorithm 10.

Algorithm 11: Direct least squares estimation of the mean of axes with isotropic uncertainty.
$[\widehat{\boldsymbol{a}}, \Sigma_{\widehat{a}\widehat{a}}, \widehat{\sigma}_0^2, R]$=`direct_LS_mean_axis`$(\{\boldsymbol{a}, \sigma\} : i)$
Input: list of I observations of axes $\boldsymbol{a}_i\{\boldsymbol{a}_i, \sigma_i[\text{rad}]\} \in \mathbb{R}^n, i = 1, ..., I \geq n-1, |\boldsymbol{a}_i| = 1$.
Output: mean axis $\boldsymbol{a}(\widehat{\boldsymbol{a}}, \Sigma_{\widehat{a}\widehat{a}})$, estimated variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = I - (n-1)$;
2 **if** $R < 0$ **then stop**, not enough axes;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$;
4 Weighted moment matrix $M = \sum_i w_i \boldsymbol{a}_i \boldsymbol{a}_i^\mathsf{T}$;
5 Eigenvalue decomposition $M = R \boldsymbol{\Lambda} R^\mathsf{T}$, $R = [\boldsymbol{r}_1, \boldsymbol{r}_2]$, $\boldsymbol{\Lambda} = \text{Diag}([\lambda_1, \lambda_2])$, $\lambda_1 \geq \lambda_2$;
6 Mean axis $\widehat{\boldsymbol{a}} = \boldsymbol{r}_1$;
7 Covariance matrix $\Sigma_{\widehat{a}\widehat{a}} = (I_n - \widehat{\boldsymbol{a}}\widehat{\boldsymbol{a}}^\mathsf{T}) / \sum_i w_i$;
8 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = (\sum_i w_i - \lambda_1)/R$ **else** $\widehat{\sigma}_0^2 = 1$.

10.5.4 Directly Estimating Transformations

10.5.4.1 Direct Algebraically Optimal Estimation of Transformations

Planar and spatial homographies can be directly estimated from given correspondences for points, lines, and planes.

A *planar homography* $\mathcal{H}$ can be determined from I corresponding points $(\mathscr{x}_i, \mathscr{x}_i'), i = 1, ..., I$ and/or J corresponding lines $(\mathscr{l}_j, \mathscr{l}_j'), j = 1, ..., J$ based on the constraints

$$\mathsf{S}(\mathbf{x}_i')\mathsf{H}\mathbf{x}_i = \mathbf{0} \qquad \text{and/or} \qquad \mathsf{S}(\mathbf{l}_i)\mathsf{H}^\mathsf{T}\mathbf{l}_i' = \mathbf{0} \tag{10.194}$$

(cf. (7.23), p. 296), or, separating the parameters $\mathbf{h} = \mathrm{vec}\mathsf{H}$ for the homography,

$$(\mathbf{x}_i^\mathsf{T} \otimes \mathsf{S}(\mathbf{x}_i'))\,\mathbf{h} = 0 \qquad \text{and/or} \qquad (\mathsf{S}(\mathbf{l}_j) \otimes \mathbf{l}_j'^\mathsf{T})\,\mathbf{h} = 0\,. \tag{10.195}$$

Therefore, the optimal parameter vector results from the right singular vector belonging to the smallest singular value of the $(3I + 3J) \times 9$ matrix

$$A = \begin{bmatrix} \{\mathbf{x}_i^\mathsf{T} \otimes \mathsf{S}(\mathbf{x}_i')\} \\ \{\mathsf{S}(\mathbf{l}_i) \otimes \mathbf{l}_i'^\mathsf{T}\} \end{bmatrix}. \tag{10.196}$$

At least four point or line correspondences are necessary. Selecting linearly independent constraints is not required. Minimal mixed configurations (with four elements only) are a) one corresponding point and three corresponding lines, or b) three corresponding points and one corresponding line (cf. Hartley and Zisserman, 2000, Sect. 4.1.4), again indicating that the non-negativity of the redundancy, $R \geq 0$, is not a sufficient condition for the existence of a solution.

A *spatial homography* can be directly determined from I corresponding 3D points, $(\mathcal{X}_i, \mathcal{X}_i'), i = 1, ..., I$, from J corresponding planes, $(\mathcal{A}_j, \mathcal{A}_j'), j = 1, ..., J$, and/or from K corresponding 3D lines, $(\mathcal{L}_k, \mathcal{L}_k'), k = 1, ..., K$, using the constraints, cf. Sect. 7.2.2, p. 304

$$\mathsf{\Pi}(\mathbf{X}_i')\mathsf{H}\mathbf{X}_i = \mathbf{0} \qquad \overline{\mathsf{\Pi}}(\mathbf{A}_i)\mathsf{H}^\mathsf{T}\mathbf{A}_i' = \mathbf{0} \qquad \overline{\mathsf{\Gamma}}(\mathbf{L}_k')\mathsf{H}\mathsf{\Gamma}(\mathbf{L}_k) = \mathsf{0}\,. \tag{10.197}$$

Therefore, the parameters $\mathbf{h} = \mathrm{vec}\mathsf{H}$ can be determined as the right singular vector corresponding to the smallest singular value of the $(6I + 6J + 16K) \times 16$ matrix

$$A = \begin{bmatrix} \{\mathbf{X}_i^\mathsf{T} \otimes \mathsf{\Pi}(\mathbf{X}_i')\} \\ \{\overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{A}_i) \otimes \mathbf{A}_i'^\mathsf{T}\} \\ \{\mathsf{\Gamma}^\mathsf{T}(\mathbf{L}_k) \otimes \overline{\mathsf{\Gamma}}(\mathbf{L}_k')\} \end{bmatrix}. \tag{10.198}$$

At least five point or plane correspondences or two line correspondences are necessary. Sufficiency can be identified from the rank of the matrix A, determined algebraically. Selecting linearly independent constraints may be of numerical advantage, since it reduces the matrix to size $(3I+3J+4K) \times 16$, but this is not required. In both cases, conditioning and normalization are recommended.

10.5.4.2 Direct Least Squares Estimation of Rotations in 3D from Corresponding Directions

Let I corresponding normalized directions $(\mathbf{X}_i, \mathbf{X}_i'), i = 1, ..., I \geq 2$, be given. They are supposed to be related by a rotation $\tilde{\boldsymbol{X}}_i' = R\tilde{\boldsymbol{X}}_i$. The directions are assumed to have isotropic uncertainty with covariances $\sigma_i^2 I_3$ and $\sigma_i'^2 I_3$, respectively, and we accept an uncertainty of the length of the normalized direction to simplify the derivation. Then the best rotation is obtained by minimizing

$$\Omega(R) = \sum_{i=1}^{I} w_i |\boldsymbol{X}'_i - R\boldsymbol{X}_i|^2 \tag{10.199}$$

with the weights

$$w_i = \frac{1}{\sigma_i^2 + \sigma_i'^2} \,. \tag{10.200}$$

The optimal rotation results from

$$\widehat{R} = \text{argmax}_R \sum_{i=1}^{I} w_i \boldsymbol{X}_i{}'^{\mathsf{T}} R \boldsymbol{X}_i \,. \tag{10.201}$$

There are several solution procedures; one is based on an SVD, another is using the quaternion representation for rotations. They are equivalent (cf. Eggert et al., 1997).

The optimal rotation using SVD is (cf. Arun et al., 1987)

$$\widehat{R} = VU^{\mathsf{T}}, \tag{10.202}$$

where the matrices V and U result from the SVD of the asymmetric cross moment matrix

$$H = \sum_{i=1}^{I} w_i \boldsymbol{X}_i \boldsymbol{X}'^{\mathsf{T}}_i = UDV^{\mathsf{T}} \,. \tag{10.203}$$

The uncertainty of the rotation uses the representation

$$\underline{\widehat{R}} = R(\underline{\widehat{\Delta \boldsymbol{r}}})\, \mathbb{E}(\underline{R}) \,, \tag{10.204}$$

with the 3-vector

$$\underline{\widehat{\Delta \boldsymbol{r}}} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\widehat{\Delta r}\widehat{\Delta r}}) \,, \tag{10.205}$$

using the Rodriguez form (8.59) of the small rotation $R(\underline{\widehat{\Delta \boldsymbol{r}}})$. This 3-vector has covariance matrix, using the assumption $|\mathbf{X}_i| = |\mathbf{X}'_i| = 1$,

$$\begin{aligned} \Sigma_{\widehat{\Delta r}\widehat{\Delta r}} &= \left(\sum_{i=1}^{I} w_i (I_3 - \boldsymbol{X}'_i \boldsymbol{X}_i{}'^{\mathsf{T}}) \right)^{-1} \\ &= \begin{bmatrix} \sum_{i=1}^{I} w_i (Y_i'^2 + Z_i'^2) & \sum_{i=1}^{I} -w_i X'_i Y'_i & \sum_{i=1}^{I} -w_i X'_i Z'_i \\ \sum_{i=1}^{I} -w_i X'_i Y'_i & \sum_{i=1}^{I} w_i (X_i'^2 + Z_i'^2) & \sum_{i=1}^{I} -w_i Y'_i Z'_i \\ \sum_{i=1}^{I} -w_i X'_i Z'_i & \sum_{i=1}^{I} -w_i Y'_i Z'_i & \sum_{i=1}^{I} w_i (X_i'^2 + Y_i'^2) \end{bmatrix}^{-1} . \end{aligned} \tag{10.206}$$

If the moment matrix $M' = \sum_i w_i \boldsymbol{X}'_i \boldsymbol{X}_i{}'^{\mathsf{T}}$ of the transformed points is diagonal, the off-diagonal elements vanish as well in the right 3×3 matrix in (10.206), and the precision around the three axes increases with the average squared distance of the points from the corresponding axis.

As each direction pair induces two constraints and the number of unknown parameters using (10.199) is three, the estimated variance factor is

$$\widehat{\sigma}_0^2 = \frac{\Omega(\widehat{R})}{2I - 3} \,. \tag{10.207}$$

Proof: From the linearized model with the approximate rotation $\widehat{R}^a$ $\boldsymbol{X}'_i + \boldsymbol{V}'_i = R(\Delta \boldsymbol{r})\widehat{R}^a(\boldsymbol{X}_i + \boldsymbol{V}_i)$, using $R(\Delta \boldsymbol{r}) \approx I_3 + S(\Delta \boldsymbol{r})$ and the approximate rotation $\widehat{R}^a$, we first obtain

$$\boldsymbol{X}'_i + \boldsymbol{V}'_i = (I_3 + S(\Delta \boldsymbol{r}))\widehat{R}^a(\boldsymbol{X}_i + \boldsymbol{V}_i) \,. \tag{10.208}$$

With the rotated directions

$$\overline{\boldsymbol{X}}'_i = \widehat{R}^a \boldsymbol{X}_i \,, \tag{10.209}$$

and neglecting higher-order terms, this can be written as

$$\boldsymbol{X}'_i + \boldsymbol{V}'_i = \widehat{R}^a(\boldsymbol{X}_i + \boldsymbol{V}_i) + S(\Delta\boldsymbol{r})\overline{\boldsymbol{X}}'_i\,. \tag{10.210}$$

This leads to the linearized model

$$\Delta\boldsymbol{V}_i(\Delta\boldsymbol{r}) = \boldsymbol{V}'_i - \widehat{R}^a\boldsymbol{V}_i = \underbrace{-\boldsymbol{X}'_i + \widehat{R}^a\boldsymbol{X}_i}_{-\Delta\boldsymbol{l}_i} + S^{\mathsf{T}}(\overline{\boldsymbol{X}}'_i)\Delta\boldsymbol{r} \quad \text{with} \quad \Sigma_{l_i l_i} = (\sigma_i^2 + \sigma_i'^2)\,I_3 = \frac{1}{w_i}I_3\,. \tag{10.211}$$

Minimizing $\Omega(\Delta\boldsymbol{r}) = \sum_{i=1}^{I}\Delta\boldsymbol{V}_i^{\mathsf{T}}(\Delta\boldsymbol{r})\Sigma^{-1}_{\Delta V_i\Delta V_i}\Delta\boldsymbol{V}_i(\Delta\boldsymbol{r})$ results in

$$\widehat{\Delta\boldsymbol{r}} = \Sigma_{\widehat{\Delta r}\widehat{\Delta r}}\sum_{i=1}^{I} w_i S(\overline{\boldsymbol{X}}'_i)(-\boldsymbol{X}'_i + R^{(0)}\boldsymbol{X}_i) \quad \text{with} \quad \Sigma_{\widehat{\Delta r}\widehat{\Delta r}} = \left(\sum_{i=1}^{I} w_i S(\overline{\boldsymbol{X}}'_i)S^{\mathsf{T}}(\overline{\boldsymbol{X}}'_i)\right)^{-1}\,. \tag{10.212}$$

Taking into account that $\boldsymbol{X}'_i \approx \overline{\boldsymbol{X}}'_i$, we obtain (10.206). ◇

The covariance matrix of the uncertain quaternion $\widehat{\underline{\mathbf{q}}}$ representing the uncertain estimated rotation $\widehat{\underline{R}} = R_Q(\widehat{\underline{\mathbf{q}}})$ is given by

$$\Sigma_{\widehat{\mathrm{q}}\widehat{\mathrm{q}}} = \frac{1}{4}\overline{\mathsf{M}}_{\widehat{\mathrm{q}}}\begin{bmatrix} 0 & \mathbf{0}^{\mathsf{T}} \\ \mathbf{0} & \Sigma_{\widehat{\Delta r}\widehat{\Delta r}} \end{bmatrix}\overline{\mathsf{M}}_{\widehat{\mathrm{q}}}^{\mathsf{T}}\,, \tag{10.213}$$

since $\widehat{\underline{\mathbf{q}}} = \underline{\Delta\mathbf{q}}\,\mathbb{E}(\underline{\mathbf{q}}) = \overline{\mathsf{M}}_{\mathbb{E}(\underline{\mathbf{q}})}\underline{\Delta\mathbf{q}}$ and $\underline{\Delta\mathbf{q}} = 1/2[1, \underline{\Delta\boldsymbol{r}}^{\mathsf{T}}]^{\mathsf{T}}$ with the matrix $\overline{\mathsf{M}}$ from (8.44), p. 334.

Algorithm 12 summarizes the procedure for the least squares estimation of the rotation from direction pairs.

Algorithm 12: Direct least squares estimation of a rotation in 3D from independent direction pairs with isotropic uncertainty.
$[\widehat{R}, \Sigma_{\widehat{\Delta r}\widehat{\Delta r}}, \widehat{\sigma}_0^2, R]$=`direct_LS_rotation_from_direction_pairs` $(\{\boldsymbol{X}, \sigma, \boldsymbol{X}', \sigma'\}_i)$
Input: list of $I \geq 2$ direction pairs $\in \mathbb{R}^n$: $\{\boldsymbol{X}_i, \sigma_i[\mathrm{rad}], \boldsymbol{X}'_i, \sigma'_i[\mathrm{rad}]\}$, $|\boldsymbol{X}_i| = |\boldsymbol{X}'_i| = 1$.
Output: mean rotation $\{\widehat{R}, \Sigma_{\widehat{\Delta r}\widehat{\Delta r}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = 2I - 3$;
2 **if** $R < 0$ **then stop**, not enough direction pairs;
3 Weights $w_i = 1/(\sigma_i^2 + \sigma_i'^2), i = 1, ..., I$;
4 Cross moment matrix $H = \sum_i w_i \boldsymbol{X}_i \boldsymbol{X}'^{\mathsf{T}}_i$;
5 SVD: $[U, D, V] = \mathtt{svd}(H)$;
6 Estimated rotation $\widehat{R} = VU^{\mathsf{T}}$;
7 Covariance matrix of differential rotation $\Sigma_{\widehat{\Delta r}\widehat{\Delta r}} = (\sum_i w_i(I_3 - \boldsymbol{X}'_i\boldsymbol{X}_i{}'^{\mathsf{T}}))^{-1}$;
8 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \sum_i w_i|\boldsymbol{X}'_i - \widehat{R}\boldsymbol{X}_i|^2/R$ **else** $\widehat{\sigma}_0^2 = 1$.

10.5.4.3 Direct Least Squares Estimation of a 3D Similarity Transformation

Estimating a motion also has a direct solution if both point sets are uncertain, as we have seen for estimating a rotation from pairs of directions. However, estimating a similarity from point pairs only has a direct solution if one point set is fixed and the other set has an isotropic uncertainty.

We discuss this direct least squares solution and analyse its theoretical precision and its ability to detect outliers.

The Direct Least Squares Solution. Let I point pairs $(\mathcal{X}_i, \mathcal{X}'_i), i = 1, ..., I \geq 3$, be given. Their relation is modelled as a similarity,

$$\mathbb{E}(\underline{\boldsymbol{X}}'_i) = \lambda R \boldsymbol{X}_i + \boldsymbol{T}\,. \tag{10.214}$$

The points $\mathcal{X}_i$ are assumed to be fixed whereas the points $\mathcal{X}'_i$ are observed and uncertain, namely mutually independent with covariance matrices $\sigma_i^2 I_3$.

It is easiest to work with the following model:

$$\mathbb{E}(\underline{\boldsymbol{X}}'_i) - \boldsymbol{X}'_0 = \lambda R(\boldsymbol{X}_i - \boldsymbol{Z})\,, \tag{10.215}$$

with the weighted centroid of the observed coordinates $\boldsymbol{X}'_i$,

$$\boldsymbol{X}'_0 = \frac{\sum_{i=1}^I w'_i \boldsymbol{X}'_i}{\sum_{i=1}^I w'_i}\,. \tag{10.216}$$

The translation $\boldsymbol{T}$ here is replaced by the *unknown shift* $\boldsymbol{Z}$ related by

$$\boldsymbol{T} = \boldsymbol{X}'_0 - \lambda R \boldsymbol{Z}\,, \tag{10.217}$$

which will turn out to be the weighted centroid of the fixed points $\mathcal{X}_i$.

Therefore, we determine the best similarity by minimizing

$$\Omega(\boldsymbol{Z}, R, \lambda) = \sum_{i=1}^I w_i |\,(\boldsymbol{X}'_i - \boldsymbol{X}'_0) - \lambda R(\boldsymbol{X}_i - \boldsymbol{Z})\,|^2 \tag{10.218}$$

with the weights $w_i = 1/\sigma_i^2$. This can be achieved in three steps:

1. The translation results from

$$\widehat{\boldsymbol{Z}} = \frac{\sum_{i=1}^I w_i \boldsymbol{X}_i}{\sum_{i=1}^I w_i} =: \boldsymbol{X}_0\,. \tag{10.219}$$

   If we choose as given point $\boldsymbol{X}_i = \boldsymbol{X}_0 = \widehat{\boldsymbol{Z}}$ in the basic model $\widehat{\boldsymbol{X}}'_i - \boldsymbol{X}'_0 = \widehat{\lambda}\widehat{R}(\boldsymbol{X}_i - \widehat{\boldsymbol{Z}})$ (cf. (10.215)), we obtain for the transformed point $\widehat{\boldsymbol{X}}'_i - \boldsymbol{X}'_0 = \boldsymbol{0}$: hence, the weighted centroid $\boldsymbol{X}_0$ of the fixed coordinates is transformed into the centroid $\boldsymbol{X}'_0$ of the observed coordinates.

2. The optimal rotation can be determined using the scheme from the section before (see the text after (10.201), p. 407):

$$\widehat{R} = \operatorname{argmax}_R \sum_{i=1}^I w_i \overline{\boldsymbol{X}}_i{}'^{\mathsf{T}} R \overline{\boldsymbol{X}}_i \tag{10.220}$$

   using the centred coordinates

$$\overline{\boldsymbol{X}}_i = \boldsymbol{X}_i - \boldsymbol{X}_0 \qquad \overline{\boldsymbol{X}}'_i = \boldsymbol{X}'_i - \boldsymbol{X}'_0\,. \tag{10.221}$$

3. The optimal scale is obtained from

$$\widehat{\lambda} = \frac{\sum_{i=1}^I w_i \overline{\boldsymbol{X}}_i{}'^{\mathsf{T}} \widehat{R} \overline{\boldsymbol{X}}_i}{\sum_{i=1}^I w_i \overline{\boldsymbol{X}}_i^{\mathsf{T}} \overline{\boldsymbol{X}}_i}\,. \tag{10.222}$$

Proof: We set $\partial\Omega/\partial\lambda = 0$ using centred coordinates:

$$\frac{\partial\Omega}{\partial\lambda} = \frac{\partial}{\partial\lambda} \sum_{i=1}^I w_i (\overline{\boldsymbol{X}}_i{}'^{\mathsf{T}} \overline{\boldsymbol{X}}'_i - 2\lambda \overline{\boldsymbol{X}}_i{}'^{\mathsf{T}} R \overline{\boldsymbol{X}}_i + \lambda^2 \overline{\boldsymbol{X}}_i^{\mathsf{T}} \overline{\boldsymbol{X}}_i) = -\sum_{i=1}^I w_i \overline{\boldsymbol{X}}'_i R \overline{\boldsymbol{X}}_i + \lambda \sum_{i=1}^I w_i \overline{\boldsymbol{X}}_i^{\mathsf{T}} \overline{\boldsymbol{X}}_i = 0.$$

This leads to (10.222). ◇

Using the substitution $R\overline{\boldsymbol{X}}_i \approx \lambda^{-1}\overline{\boldsymbol{X}}'_i$, we obtain an approximate symmetric estimate

$$\widehat{\lambda^2} = \frac{\sum_{i=1}^{I} w_i \left|\overline{\boldsymbol{X}}'_i\right|^2}{\sum_{i=1}^{I} w_i \left|\overline{\boldsymbol{X}}_i\right|^2}, \tag{10.223}$$

independent of the rotation. The difference between both estimates for λ is usually negligible.

The estimated variance factor is

$$\widehat{\sigma}_0^2 = \frac{\Omega(\widehat{\boldsymbol{Z}}, \widehat{R}, \widehat{\lambda})}{3I - 7}. \tag{10.224}$$

If the redundancy $R = 3I - 7$ is large enough, say > 30, and no outliers are present, the estimate is reliable enough; otherwise, we should use the prior $\sigma_0 = 1$.

The three estimated elements, the centroid $\widehat{\boldsymbol{Z}} = \boldsymbol{X}_0$, the estimated rotation $\widehat{R}$, and the estimated scale $\widehat{\lambda}$, are statistically mutually independent, which can be shown by analysing the normal equation matrix of an iterative estimation scheme. The coordinates of the centroid are also mutually independent and have standard deviations

Exercise 10.19

$$\sigma_{\widehat{X}_0} = \sigma_{\widehat{Y}_0} = \sigma_{\widehat{Z}_0} = \frac{\sigma_0^2}{\sum_{i=1}^{I} w_i}, \tag{10.225}$$

which simplifies to $\widehat{\sigma}_0^2/I$ in case all points are weighted with 1. For the uncertainty of the rotation we can use (10.206). As the directions in (10.206) are normalized vectors, we need to use the weights $w_i|\overline{\boldsymbol{X}}'_i|^2$ and obtain the covariance matrix for the rotation correction $\widehat{\Delta\boldsymbol{r}}$,

$$\Sigma_{\widehat{\Delta r}\widehat{\Delta r}} = \sigma_0^2 \left(\sum_{i=1}^{I} w_i \left(|\overline{\boldsymbol{X}}'_i|^2 I_3 - \overline{\boldsymbol{X}}'_i \overline{\boldsymbol{X}}_i{}'^{\mathsf{T}} \right) \right)^{-1}. \tag{10.226}$$

Finally, the standard deviation of the scale factor is

$$\sigma_{\widehat{\lambda}} = \frac{\sigma_0}{\sqrt{\sum_{i=1}^{I} w_i |\overline{\boldsymbol{X}}_i|^2}} = \frac{\sigma_0}{\sqrt{\sum w_i (\overline{X}_i^2 + \overline{Y}_i^2 + \overline{Z}_i^2)}}. \tag{10.227}$$

It is inversely proportional with the average weighted squared distances $|\overline{\boldsymbol{X}}_i|^2$ to the centroid of the points in the given system. This is plausible, since the scale $\lambda_i = |\overline{\boldsymbol{X}}'_i|/|\overline{\boldsymbol{X}}_i|$, determined from one corresponding point pair, would have a standard deviation of $\sigma_{\lambda_i} = \sigma_i/|\overline{\boldsymbol{X}}_i|$.

When the uncertainty of the translation parameter $\widehat{\boldsymbol{T}}$ in (10.214) is needed, (10.217) can be used for variance propagation, due to $\mathrm{d}\boldsymbol{T} = -\mathrm{d}\lambda R\boldsymbol{Z} - \lambda S^{\mathsf{T}}(R\boldsymbol{Z})\mathrm{d}(\Delta\boldsymbol{r}) - \lambda R\mathrm{d}\boldsymbol{Z}$ leading to:

$$\Sigma_{\widehat{T}\widehat{T}} = \widehat{\boldsymbol{Z}}\widehat{\boldsymbol{Z}}^{\mathsf{T}}\sigma_{\widehat{\lambda}}^2 + \widehat{\lambda}^2 S^{\mathsf{T}}(\widehat{R}\widehat{\boldsymbol{Z}})\Sigma_{\widehat{\Delta r}\widehat{\Delta r}} S(\widehat{R}\widehat{\boldsymbol{Z}}) + \widehat{\lambda}^2 I_3 \sigma_0^2 / \sum_i w_i. \tag{10.228}$$

Exercise 10.20

However, this will cause correlations between $\widehat{\boldsymbol{T}}$, $\widehat{R}$, and $\widehat{\lambda}$. The procedure is summed up in Algorithm 13.

If the coordinate system is chosen parallel to the principal axes of the 3×3 moment matrix $\sum_i \overline{\boldsymbol{X}}_i \overline{\boldsymbol{X}}_i^{\mathsf{T}}$, the unknown parameters are statistically independent; thus, all correlations are zero and we can analyse the quality more easily. We obtain the standard deviations of the three rotation angles around the three axes,

Algorithm 13: Direct least squares estimation of a similarity in 3D from independent point pairs with isotropic uncertainty, where one set is non-stochastic.
$[\widehat{R}, \Sigma_{\widehat{\Delta r}\widehat{\Delta r}}, \widehat{\boldsymbol{T}}, \Sigma_{\widehat{T}\widehat{T}}, \widehat{\lambda}, \sigma^2_{\widehat{\lambda}}, \widehat{\sigma}^2_0, R] =$ `direct_LS_3D_similarity`$(\{\boldsymbol{X}, \boldsymbol{X}', \sigma\}_i)$
Input: list of I point pairs in $\mathbb{R}^3$: $\{\boldsymbol{X}_i, \boldsymbol{X}'_i, \sigma_i\}$.
Assumption: point pairs are independent, covariance matrices $\Sigma_{X'_iX'_i} = \sigma_i^2 \boldsymbol{I}_3$ of $\boldsymbol{X}'_i$, coordinates are conditioned.
Output: similarity $(\widehat{R}, \widehat{\boldsymbol{T}}, \widehat{\lambda})$ with variances, estimated variance factor $\widehat{\sigma}^2_0$, redundancy R.

1 Redundancy $R = 3I - 7$;
2 **if** $R < 0$ **then stop**, not enough point pairs;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$;
4 Sum of weights $S_w = \sum_i w_i$;
5 Weighted centroids $\boldsymbol{X}_0 = \sum_i w_i \boldsymbol{X}_i / S_w$, $\boldsymbol{X}'_0 = \sum_i w_i \boldsymbol{X}'_i / S_w$;
6 Centred coordinates $\overline{\boldsymbol{X}}_i = \boldsymbol{X}_i - \boldsymbol{X}_0$, $\overline{\boldsymbol{X}}'_i = \boldsymbol{X}'_i - \boldsymbol{X}'_0$;
7 Cross moment matrix $\boldsymbol{H} = \sum_i w_i \overline{\boldsymbol{X}}_i \overline{\boldsymbol{X}}'^{\mathsf{T}}_i$;
8 Weighted sum of distances squared $S_X = \sum_i w_i |\overline{\boldsymbol{X}}_i|^2$;
9 SVD: $[\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{V}] =$ `svd`$(\boldsymbol{H})$;
10 Estimated rotation $\widehat{\boldsymbol{R}} = \boldsymbol{V}\boldsymbol{U}^{\mathsf{T}}$;
11 Estimated scale $\widehat{\lambda} = \sum_{i=1}^{I} w_i \overline{\boldsymbol{X}}_i'^{\mathsf{T}} \widehat{\boldsymbol{R}} \overline{\boldsymbol{X}}_i / S_X$;
12 Estimated translation $\widehat{\boldsymbol{T}} = \boldsymbol{X}'_0 - \widehat{\lambda}\widehat{\boldsymbol{R}}\boldsymbol{X}_0$;
13 Covariance matrix of centroid $\Sigma_{X_0X_0} = \boldsymbol{I}_3 / S_w$;
14 Covariance matrix of rotation $\Sigma_{\widehat{\Delta r}\widehat{\Delta r}} = (\sum_i w_i (|\overline{\boldsymbol{X}}'_i|^2 \boldsymbol{I}_3 - \overline{\boldsymbol{X}}'_i \overline{\boldsymbol{X}}_i'^{\mathsf{T}}))^{-1}$;
15 Variance of scale $\sigma^2_{\widehat{\lambda}} = 1/S_X$;
16 Covariance matrix of translation $\Sigma_{\widehat{T}\widehat{T}}$ from (10.228);
17 **if** $R > 0$ **then** variance factor $\widehat{\sigma}^2_0 = \sum_i w_i |\boldsymbol{X}'_i - \widehat{\lambda}\widehat{\boldsymbol{R}}\boldsymbol{X}_i - \widehat{\boldsymbol{T}}|^2 / R$ **else** $\widehat{\sigma}^2_0 = 1$.

$$\sigma_\omega = \frac{\sigma_0}{\sqrt{\sum w_i (\overline{Y}_i^2 + \overline{Z}_i^2)}}, \quad \sigma_\phi = \frac{\sigma_0}{\sqrt{\sum w_i (\overline{Z}_i^2 + \overline{X}_i^2)}}, \quad \sigma_\kappa = \frac{\sigma_0}{\sqrt{\sum w_i (\overline{X}_i^2 + \overline{Y}_i^2)}}. \tag{10.229}$$

They mainly depend on the distance of the points from the three axes. Finally, we also want to give the redundancy numbers of the coordinates, again in the coordinate system of the principal axes of the point set. They indicate how the total redundancy is distributed over all observations and are a measure of the detectability of outliers. As an example, the redundancy number for the $\overline{X}_i$ coordinate is

$$r_{\overline{X}_i} = 1 - \frac{w_i}{\sum w_i} - \frac{w_i \overline{X}_i^2}{\sum w_i (\overline{X}_i^2 + \overline{Y}_i^2 + \overline{Z}_i^2)} - \frac{w_i (\overline{Y}_i^2 + \overline{Z}_i^2)}{\sum w_i (\overline{Y}_i^2 + \overline{Z}_i^2)}. \tag{10.230}$$

As the checkability of the coordinates increases with increasing redundancy numbers, points in the centre of the point cloud can be checked more easily than points at the boundary. Actual tests for outliers will not address individual coordinates, but the coordinate vectors $\boldsymbol{X}'_i$, cf. Sect. 4.6.4.2, p. 128.

10.5.5 Best Fitting 3D Line

The best fitting straight 3D line through points or planes is more elaborate, since the Plücker coordinates $\mathbf{L}$ of a 3D line are homogeneous and need to fulfil the quadratic Plücker constraint $\mathbf{L}^{\mathsf{T}}\mathbf{D}\mathbf{L} = 0$. The algebraically optimal solution generally neglects this constraint, which is why a second step is required to enforce it.

Finding the Algebraically Optimal Straight 3D Line. The algebraic closed form solution for the 3D line is useful when in addition to 3D points other constraints are given, for instance, the line has to lie on given planes.

Let I 3D points $\mathcal{X}_i, i = 1, ..., I$, and J planes $\mathcal{A}_j$ be given which are supposed to be incident with an unknown 3D line $\mathcal{L}$. Then we need to fulfil the following constraints:

$$\begin{aligned} \mathcal{X}_i \in \mathcal{L}: \quad & \overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X}_i)\mathbf{L} = \mathbf{0} && (10.231)\\ \mathcal{A}_j \ni \mathcal{L}: \quad & \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A}_j)\mathbf{L} = \mathbf{0} && (10.232)\\ \text{Normalization}: \quad & \mathbf{L}^{\mathsf{T}}\mathbf{L} - 1 = 0 && (10.233)\\ \text{Plücker}: \quad & \mathbf{L}^{\mathsf{T}} D \mathbf{L} = 0 && (10.234) \end{aligned}$$

(cf. Sect. 7.2.2, p. 304). It is not necessary to use selected independent constraints or reduced coordinates, see the remark after (10.172), p. 401. If we neglect the Plücker constraint, we obtain the algebraically best parameters $\widehat{\mathbf{L}}^a$ from the right singular vector of the $4I + 4J \times 6$ matrix corresponding to the smallest singular value of

$$\begin{bmatrix} \{\overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X}_i)\} \\ \{\mathsf{\Pi}^{\mathsf{T}}(\mathbf{A}_j)\} \end{bmatrix} . \tag{10.235}$$

In general, the 6-vector $\widehat{\mathbf{L}}^a$ does not fulfil the Plücker constraint. However, if we only have points, it can be shown that the right singular vectors of $\{\overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X}_i)\}$ all fulfil the Plücker constraint. In order to enforce the Plücker constraint, we can use the procedure in ((10.79), p. 381).

Exercise 10.17

Direct Least Squares Estimation of a 3D Line from Statistically Independent 3D Points. Now let the I 3D points $\mathcal{X}_i$ with coordinates $\boldsymbol{X}_i$ be statistically independent and have isotropic uncertainty $\Sigma_{X_i X_i} = \sigma_i^2 I_3$. Let the 3D line be represented by some point $\boldsymbol{Z}$ on the line and the normalized direction $\boldsymbol{D}$. Then we have the following result:

isotropic uncertainty of 3D points

1. The optimal 3D line passes through the weighted centroid of the 3D points,

$$\widehat{\boldsymbol{Z}} = \boldsymbol{X}_0 = \frac{\sum_{i=1}^{I} w_i \boldsymbol{X}_i}{\sum_{i=1}^{I} w_i} . \tag{10.236}$$

2. The optimal direction is the eigenvector of the (central) moment matrix corresponding to its largest eigenvalue,

$$M = \sum_{i=1}^{I} w_i (\boldsymbol{X}_i - \boldsymbol{X}_0)(\boldsymbol{X}_i - \boldsymbol{X}_0)^{\mathsf{T}} , \tag{10.237}$$

in full analogy to the best fitting 2D line in Sect. 10.5.2.2, p. 397.

Proof: The squared distance of the point $\mathcal{X}_i$ from the line is

$$d_i^2 = |(\boldsymbol{X}_i - \boldsymbol{Z}) \times \boldsymbol{D}|^2 = \boldsymbol{D}^{\mathsf{T}} \mathsf{S}_{X_i - Z}^{\mathsf{T}} \mathsf{S}_{X_i - Z} \boldsymbol{D} = (\boldsymbol{X}_i - \boldsymbol{Z})^{\mathsf{T}} \mathsf{S}^{\mathsf{T}}(\boldsymbol{D}) \mathsf{S}(\boldsymbol{D}) (\boldsymbol{X}_i - \boldsymbol{Z}) . \tag{10.238}$$

Minimizing the weighted sum of the squared distances $\Omega(\boldsymbol{Z}, \boldsymbol{D}) = \sum_{i=1}^{I} w_i d_i^2$ can be achieved by setting the partial derivatives of Ω w.r.t. the parameters to zero, following the same argument as for the 2D line above. We first obtain the condition

$$\frac{\partial \Omega(\boldsymbol{Z}, \boldsymbol{D})}{\partial \boldsymbol{Z}} = -2 \mathsf{S}^{\mathsf{T}}(\boldsymbol{D}) \mathsf{S}(\boldsymbol{D}) \sum_{i=1}^{I} w_i (\boldsymbol{X}_i - \boldsymbol{Z}) = \mathbf{0} . \tag{10.239}$$

It is satisfied by the weighted centroid $\boldsymbol{X}_0$ of all points $\mathcal{X}_i$, i.e., $\boldsymbol{Z} = \boldsymbol{X}_0$. Thus, we can determine the position of the 3D line without knowing its direction. Of course, every other point $\boldsymbol{Z} + t\boldsymbol{D}$ with some arbitrary t would also fulfil the condition. Knowing $\boldsymbol{X}_0$, we now directly determine the best estimate for $\boldsymbol{D}$ by minimizing

$$\Omega(\boldsymbol{D} \mid \boldsymbol{X}_0) = \boldsymbol{D}^\mathsf{T} \overline{M} \boldsymbol{D} \quad \text{with} \quad \overline{M} = \sum_{i=1}^{I} w_i \mathsf{S}^\mathsf{T}_{X_i - X_0} \mathsf{S}_{X_i - X_0}, \tag{10.240}$$

due to (10.238). For an arbitrary vector $\boldsymbol{U}$, the relation $\mathsf{S}^\mathsf{T}(\boldsymbol{U})\mathsf{S}(\boldsymbol{U}) = I_3|\boldsymbol{U}|^2 - \boldsymbol{U}\boldsymbol{U}^\mathsf{T}$ holds, and, therefore, the matrix $\overline{M}$ can be written as

$$\overline{M} = \sum_{i=1}^{I} w_i \left(|\boldsymbol{X}_i - \boldsymbol{X}_0|^2 I_3 - (\boldsymbol{X}_i - \boldsymbol{X}_0)(\boldsymbol{X}_i - \boldsymbol{X}_0)^\mathsf{T} \right) = \sum_{i=1}^{I} w_i |\boldsymbol{X}_i - \boldsymbol{X}_0|^2 I_3 - M, \tag{10.241}$$

with the moment matrix M of the given points. Consequently, the optimal estimate $\widehat{\boldsymbol{D}}$ minimizing $\Omega(\boldsymbol{D}) = \boldsymbol{D}^\mathsf{T} \overline{M} \boldsymbol{D}$ at the same time maximizes $\boldsymbol{D}^\mathsf{T} M \boldsymbol{D}$; thus, it is the eigenvector belonging to the largest eigenvalue of the moment matrix. ◇

The uncertainty of the line can be characterized by the variance of the position of $\boldsymbol{X}_0$ across the line and the covariance matrix of the direction $\boldsymbol{D}$. Both vectors are statistically independent. This is because generally estimated parameters and the estimated residuals are stochastically independent, and here we estimate the mean and derive the moment matrix from the residuals, i.e., the centred coordinates, from which the optimal direction is taken.

The derivation of the uncertainty of the position and of the direction uses the same procedure as for the 2D line; the coordinate axes are chosen to lie close to the principal directions, i.e., the eigenvectors of the point set.

uncertainty of best fitting 3D line from points with isotropic uncertainty

The theoretical covariance matrix of the centroid is

$$\Sigma_{X_0 X_0} = \sigma_q^2 I_3 \quad \text{with} \quad \sigma_q^2 = \frac{1}{\sum_{i=1}^{I} w_i}. \tag{10.242}$$

It is isotropic and identical to the uncertainty of the weighted mean of I points. The theoretical covariance matrix of the direction $\widehat{\boldsymbol{D}}$ also is isotropic, with

$$\Sigma_{\widehat{D}\widehat{D}} = \frac{1}{\lambda_1} (I_3 - \widehat{\boldsymbol{D}} \widehat{\boldsymbol{D}}^\mathsf{T}), \tag{10.243}$$

where λ_1 is the largest eigenvalue of the moment matrix, cf. (10.166), p. 399. The covariance matrix is singular, as the length of $\widehat{\boldsymbol{D}}$ is normalized to 1, which means that the direction $\boldsymbol{D}$ has angular variance

$$\sigma_\phi^2 = \sigma_\psi^2 = \frac{1}{\lambda_1} \tag{10.244}$$

in all directions. The angles ϕ and ψ denote rotations of the 3D line around the two minor axes of the 3D point set.

If we have enough points, we can use the estimated variance factor

$$\widehat{\sigma}_0^2 = \frac{\lambda_2 + \lambda_3}{2I - 4}, \tag{10.245}$$

as the deviations of the points from the line in both directions across the line have to be taken into account, and we have $\lambda_2 + \lambda_3 = \Omega(\widehat{\boldsymbol{Z}}, \widehat{\boldsymbol{D}})$ (cf. (10.167), p. 399). Then the estimated covariance matrix for the direction is

$$\widehat{\Sigma}_{\widehat{D}\widehat{D}} = \frac{1}{2I - 4} \frac{\lambda_2 + \lambda_3}{\lambda_1} (I_3 - \widehat{\boldsymbol{D}} \widehat{\boldsymbol{D}}^\mathsf{T}), \tag{10.246}$$

or, with isotropic directional uncertainty,

$$\widehat{\sigma}_\phi^2 = \widehat{\sigma}_\psi^2 = \frac{1}{2I - 4} \frac{\lambda_2 + \lambda_3}{\lambda_1}. \tag{10.247}$$

As we have full rotational symmetry of the uncertainty, the surface of constant probability density for a point on the 3D line is a rotational hyperboloid of one sheet. This is a consequence of the simple uncertainty model for the given 3D points.

Generally, i.e., for 3D points with non-isotropic uncertainty, the uncertainty of the resulting 3D line will have the more general shape, as shown in Fig. 10.16, p. 379, which is not a hyperboloid.

Algorithm 14: Direct least squares estimation of a 3D line from statistically independent 3D points with isotropic uncertainty.
$[\widehat{\boldsymbol{X}}_0, \Sigma_{\widehat{X}_0\widehat{X}_0}, \widehat{\boldsymbol{D}}, \Sigma_{\widehat{D}\widehat{D}}, \widehat{\sigma}_0^2, R]$= `direct_LS_3D_line_from_points`$(\{\boldsymbol{X}, \sigma\}_i)$
Input: list of $I \geq 2$ 3D points $\{\mathcal{X}_i(\boldsymbol{X}_i, \sigma_i)\}$.
Assumption: coordinates are conditioned.
Output: best fitting 3D line in point direction form $\mathcal{L}(\widehat{\boldsymbol{X}}_0, \widehat{\boldsymbol{D}})$ with uncertainty, estimated variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = 2I - 4$;
2 **if** $R < 0$ **then stop**, not enough points;
3 Weights $w_i = 1/\sigma_i^2, i = 1, ..., I$;
4 Weighted centroid $\boldsymbol{X}_0 = \sum_i w_i \boldsymbol{X}_i / \sum_i w_i$;
5 Centred coordinates $\overline{\boldsymbol{X}}_i = \boldsymbol{X}_i - \boldsymbol{X}_0, i = 1, ..., I$;
6 Weighted moment matrix $M = \sum_i w_i \overline{\boldsymbol{X}}_i \overline{\boldsymbol{X}}_i^\mathsf{T}$;
7 Eigenvalue decomposition $R\boldsymbol{\Lambda}R^\mathsf{T} = M$, $\boldsymbol{\Lambda} = \mathrm{Diag}([\lambda_i])$, $\lambda_1 \geq \lambda_2 \geq \lambda_3$;
8 Estimated direction $\widehat{\boldsymbol{D}}$: column of R belonging to largest eigenvalue;
9 Covariance matrix of centroid $\Sigma_{\widehat{X}_0\widehat{X}_0} = 1/\sum_i w_i I_3$;
10 Covariance matrix of direction $\Sigma_{\widehat{D}\widehat{D}} = 1/\lambda_1 (I_3 - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{D}}^\mathsf{T})$;
11 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = (\lambda_2 + \lambda_3)/R$ **else** $\widehat{\sigma}_0^2 = 1$.

10.6 Iterative Solutions for Maximum Likelihood Estimation

10.6.1 Estimation on Curved Manifolds with Reduced Coordinates 415
10.6.2 Vanishing Point Determination 417
10.6.3 Estimating a Homography from Corresponding Points 424
10.6.4 Estimating a Symmetric Roof from a Point Cloud 429

Iterative techniques for maximum likelihood estimation (cf. Sect. 4, p. 75) start from sufficiently good approximate values and perform a Taylor expansion to obtain a linear substitute problem whose solution yields improved approximations until convergence is reached. Depending on the type of problem, we use the Gauss–Markov model or the Gauss–Helmert model, possibly with constraints among the unknown parameters or among the observations. As we assume that the distribution of the observations is close to a Gaussian distribution, we use the principle of maximum likelihood estimation. It leads to minimizing the Mahalanobis distance

$$\Omega(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}}) = \frac{1}{2} \boldsymbol{v}^\mathsf{T}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}})\, \Sigma_{ll}^{-1}\, \boldsymbol{v}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}})\,, \tag{10.248}$$

where the given constraints may be of the form

$$\widehat{\boldsymbol{l}} = \boldsymbol{f}(\widehat{\boldsymbol{x}})\,, \qquad \boldsymbol{g}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{l}}) = \boldsymbol{0}\,, \qquad \boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}\,, \qquad \boldsymbol{h}_l(\widehat{\boldsymbol{l}}) = \boldsymbol{0}\,, \tag{10.249}$$

where we explicitly mentioned the constraints $\boldsymbol{h}_l(\widehat{\boldsymbol{l}}) = \boldsymbol{0}$ only involving observations $\widehat{\boldsymbol{l}}$.

In our context, both, observations and parameters, may be either Euclidean or homogeneous entities. Using homogeneous entities during parameter estimation causes two problems.

1. Homogeneous entities in the role of observations inherently have a singular covariance matrix Σ_{ll}. This prevents the simple classical formulation of the estimation problem in (10.248).
2. Homogeneous entities in the role of unknown parameters require normalization constraints in the form of $\boldsymbol{h}(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$, which in addition to causing the redundant representation with homogeneous entities increases the number of unknown parameters in the estimation process due to the use of Lagrangian multipliers.

Both problems can be solved by using the minimal representation of homogeneous entities introduced in Sect. 10.2.2.1. First, in general the reduced homogeneous parameters all have a regular covariance matrix. Second, the representation is minimal, not requiring any constraints for enforcing the correct normalization.

We will give three examples and discuss various aspects when using homogeneous entities during estimation.

1. For estimating vanishing points from straight line segments in a real image we address the following: (1) Tracking the uncertainty from the found edge elements to the final estimates; (2) Estimating the line segments from edge elements and changing from the centroid representation to homogeneous coordinates; (3) Estimating the vanishing points, possibly at infinity, when using reduced coordinates for the line segments and the Gauss–Helmert model estimation (model D); and (4) Jointly estimating the three vanishing points in the image of a 3D object with orthogonal structure lines by enforcing the orthogonality constraints between the directions to the vanishing points. The estimation model C of constraints between observations only is applied and the increase in precision is evaluated.
2. Estimating a homography between two simulated point sets allows the following: (1) Exploiting reduced coordinates and the minimal representation using estimation models A and D, the Gauss–Markov and the Gauss–Helmert model, respectively; (2) Disclosing the difference between algebraically and statistically optimal estimation; and (3) Proving the validity of the resulting covariance information based on the covariance matrices of the minimal representation.
3. Estimating a symmetric roof from a point cloud allows the following: (1) Comparison of the stepwise estimation with the joint estimation and (2) Estimation using model E with additional constraints between the parameters.

We will first discuss how to modify the general estimation scheme in the Gauss–Helmert model when using reduced coordinates.

10.6.1 Estimation on Curved Manifolds with Reduced Coordinates

When estimating geometric entities represented by homogeneous vectors or when estimating transformations, we actually perform an estimation on curved manifolds, which are characterized by their specific constraints, e.g., when vectors have length 1, line vectors have to fulfil the Plücker constraint, or matrices are orthogonal or have determinant 1.

These constraints usually are handled with a Gauss–Markov model with constraints. But using this model has the following disadvantage: Introducing constraints increases the number of unknowns by twice the number of constraints, first because of the redundant representation, second because of the introduction of Lagrangian multipliers. While for small estimation problems this is fully acceptable, in problems with thousands of unknown geometric entities, e.g., 3D points, this method increases the normal equation system and the solution time by an unacceptable amount. Therefore, we exploit the concept of reduced coordinates, which has been introduced to arrive at a minimum representation for geometric entities.

The estimation procedures given before slightly change: In each iteration we have up to now improved the current approximate value for the estimated parameters and fitted

values. When estimating on a curved manifold we now need to embed this single step into a previous projection and subsequent reprojection, also called retraction by Absil et al. (2008). Thus, we perform the following three steps, as already mentioned when introducing the reduced coordinates:

1. *Projection:* The observations $\boldsymbol{l}$ are projected onto their tangent space at the current approximate value for their estimates $\widehat{\boldsymbol{l}}^a$ together with their uncertainty, leading to $\{\boldsymbol{l}_r, \Sigma_{l_r l_r}\}$. The same type of projection is performed with the unknown parameters. The approximate values in the tangent space are all zero, both for fitted observations and for the parameters.
2. *Estimation:* The corrections $\Delta\boldsymbol{l}_r$ and $\Delta\boldsymbol{x}_r$ for the observations and the parameters, possibly together with their covariance matrix, are determined within an estimation procedure.
3. *Reprojection:* These corrections are reprojected from the tangent space $\mathcal{T}(\mathcal{M})$ to the manifold $\mathcal{M}$, leading to the improved estimates $\widehat{\boldsymbol{l}}$ and $\widehat{\boldsymbol{x}}$.

Algorithm 15 gives the essential steps in a Gauss–Markov model when using reduced coordinates. Additional constraints between the parameters and a reweighting scheme are omitted for clarity. Most of the steps are similar to those of Algorithm 4, except for a few lines, which need some explanation.

Algorithm 15: Estimation in the Gauss–Helmert model with reduced coordinates.
$[\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}, \widehat{\sigma}_0^2, R] =$ `GaussHelmertModell_reduced`$(\boldsymbol{l}, \Sigma_{ll}, c_g, \boldsymbol{x}^a, \boldsymbol{\sigma}^a_{\widehat{x}}, T_x,$ `maxiter`$, u_x, u_l)$
Input: observed values $\{\boldsymbol{l}, \Sigma_{ll}\}$, number N,
constraint functions $[\boldsymbol{c}_g, \mathsf{A}, \mathsf{B}] = c_g(\boldsymbol{l}, \widehat{\boldsymbol{l}}, \widehat{\boldsymbol{x}})$, number G,
approximate values $\widehat{\boldsymbol{x}}^a_u$, possibly $\sigma^a_{\widehat{x}_u}$,
parameters T_x, `maxiter` for controlling convergence,
update function u_x for parameters, update function u_l for fitted observations.
Output: estimated parameters $\{\widehat{\boldsymbol{x}}, \Sigma_{\widehat{x}\widehat{x}}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

1 Redundancy $R = G - U$;
2 **if** $R < 0$ **then stop**, not enough constraints;
3 Initiate: iteration $\nu = 0$, $\widehat{\boldsymbol{l}}^{(\nu)} = \widehat{\boldsymbol{l}}^a = \boldsymbol{l}$, $\widehat{\boldsymbol{x}}^{(\nu)} = \boldsymbol{x}^a$, stopping variable: $s = 0$;
4 **repeat**
5 Residuals and Jacobians for constraints $\boldsymbol{g}$: $[\boldsymbol{c}_{g_r}, \mathsf{A}_r, \mathsf{B}_r, \Sigma_{l_r l_r}] = c_{g_r}(\boldsymbol{l}, \widehat{\boldsymbol{l}}^{(\nu)}, \widehat{\boldsymbol{x}}^{(\nu)}, \Sigma_{ll})$;
6 Weight of constraints: $\mathsf{W}_{g_r g_r} = (\mathsf{B}_r^\mathsf{T} \Sigma_{l_r l_r} \mathsf{B}_r)^{-1}$;
7 Updates of parameter vector: $\widehat{\Delta\boldsymbol{x}}_r = (\mathsf{A}_r \mathsf{W}_{g_r g_r} \mathsf{A}_r)^{-1} \mathsf{A}_r \mathsf{W}_{g_r g_r} \boldsymbol{c}_{g_r}$;
8 Set iteration: $\nu := \nu + 1$;
9 **if** $\max |\widehat{\Delta x}_{ru}| / \sigma^a_{\widehat{x}_{ru}} < T_x$ or $\nu =$ `maxiter` **then** $s = 2$;
10 Corrections for fitted observations: $\widehat{\Delta\boldsymbol{l}}_r$, see (4.448);
11 Update parameters: $\widehat{\boldsymbol{x}}^{(\nu)} = u_x(\widehat{\boldsymbol{x}}^{(\nu-1)}, \widehat{\Delta\boldsymbol{x}}_r)$;
12 Update fitted observations: $\widehat{\boldsymbol{l}}^{(\nu)} = u_l(\widehat{\boldsymbol{l}}^{(\nu-1)}, \widehat{\Delta\boldsymbol{l}}_r)$;
13 **until** $s \equiv 2$;
14 Covariance matrix of estimated parameters: $\Sigma_{\widehat{x}\widehat{x}}$ (4.455);
15 **if** $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \boldsymbol{c}_{g_r}^\mathsf{T} (\mathsf{B}_r^\mathsf{T} \Sigma_{l_r l_r} \mathsf{B}_r)^{-1} \boldsymbol{c}_{g_r} / R$ **else** $\widehat{\sigma}_0^2 = 1$.

Comments:

1 The number U is the total degrees of freedom, not the number of parameters in some redundant representation. For example, when estimating the homogeneous 4-vector of a 3D point, we have $U = 3$.

5,6 The function c_g provides vectors and matrices for the estimation in the tangent space. For each observational group we have

$$\underset{G_i \times U}{A_{ri}} = A_i \; J_r(\widehat{\boldsymbol{x}}^a) \tag{10.250}$$

$$\underset{G_i \times N}{B_{ri}^{\mathsf{T}}} = B_i^{\mathsf{T}} \; \mathrm{Diag}(\{J_r(\widehat{\boldsymbol{l}}_j^a)\}) \tag{10.251}$$

$$\boldsymbol{c}_{g_{ri}} = -\boldsymbol{g}_i(\widehat{\boldsymbol{l}}^a, \widehat{\boldsymbol{x}}^a) - B_{ri}^{\mathsf{T}} \; [J_r^{\mathsf{T}}(\widehat{\boldsymbol{l}}_j^a) \; \boldsymbol{l}_j] \tag{10.252}$$

$$\Sigma_{l_{ri}l_{ri}} = J_r^{\mathsf{T}}(\widehat{\boldsymbol{l}}_i^a) \; R_{ab}(\boldsymbol{l}_i, \widehat{\boldsymbol{l}}_i^a) \; \Sigma_{l_i l_i} \; R_{ab}^{\mathsf{T}}(\boldsymbol{l}_i, \widehat{\boldsymbol{l}}_i^a) \; J_r(\widehat{\boldsymbol{l}}_i^a) \tag{10.253}$$

$$W_{g_{ri}g_{ri}} = (B_{ri}^{\mathsf{T}} \Sigma_{l_r l_r} B_{ri})^{-1} \, . \tag{10.254}$$

We assume that each constraint may refer to all unknown parameters $\boldsymbol{x}$ which may be partitioned in groups of parameters of different type, e.g., homogeneous coordinates of 3D points or the rotation matrices. Therefore the Jacobian $J_r(\widehat{\boldsymbol{x}}^a)$ in (10.250) consists of a block diagonal matrix referring to the individual groups of parameters.

We assume that each constraint refers to non-overlapping vectors of observations. Each of these vectors refers to the original observational groups, which may be of different type, e.g., homogeneous vectors of 2D and 3D points. Therefore the Jacobian $J_r(\widehat{\boldsymbol{l}}^a)$ in (10.251) is a block diagonal matrix of Jacobians J_r referring to the individual observational groups. The observational groups referring to the same constraint may be correlated. For transformations, the Jacobians A_r and B_r for the reduced parameters need to be derived individually. The individual constraints may be vector-valued, e.g., when 3D points are constrained to lie on a 3D line.

7 The normal equation should be regular.

11,12 The update functions u_l and u_x perform the reprojection from tangent space to curved manifold. For each group of observations or parameters, e.g., for observed 2D points or unknown rotation parameters,

$$\widehat{\boldsymbol{x}}_u = u_l(\widehat{\boldsymbol{x}}_u^a, \widehat{\Delta \boldsymbol{x}}_{ru}) = \mathsf{N}(\widehat{\boldsymbol{x}}_u^a + J(\widehat{\boldsymbol{x}}_u^a)\widehat{\Delta \boldsymbol{x}}_{ru}) \tag{10.255}$$

$$\widehat{R} = \; u_x(\widehat{R}^a, \widehat{\Delta \boldsymbol{r}}) \; = R_R(\widehat{\Delta \boldsymbol{r}}) \; \widehat{R}^a \, , \tag{10.256}$$

with R_R from (8.59), p. 336.

For an example of a Gauss–Markov model with reduced parameters, see the estimation of projection matrix, Alg. 17, p. 499. For an example of observational groups involved in one constraint, cf. Alg. 21, p. 600.

10.6.2 Vanishing Point Determination

Let an image be given showing a man-made scene with three mutually perpendicular lines, as in Fig. 10.27, p. 423, upper left; let the line segments be extracted and grouped into sets assumed to belong to one of the three vanishing points. The task is to accurately estimate these vanishing points. If the calibration of the camera is known, additional constraints on the vanishing points can be enforced, as we will discuss in a second step.

Estimating vanishing points is a problem if they are close to infinity and Euclidean representation is used. The estimated coordinates become extremely uncertain, especially in the direction of the vanishing point, and lines may even intersect on the opposite side, indicating the description of the uncertainty with a Gaussian distribution is not adequate. Performing the estimation on the unit sphere eliminates these problems and leads to a stable and unbiased solution. Furthermore, we will show how to enforce the orthogonality constraints on the result if the camera calibration is known.

10.6.2.1 Estimating a Single Vanishing Point

We start with the estimation of a single vanishing point from observed line segments. This demonstrates how the uncertainty given by some line finding algorithm can be exploited within an estimation procedure that allows us to handle vanishing points close to infinity.

Let I lines $l_i, i = 1, ..., I$, be given. They represent finite line segments derived from an image using some image processing module. Let us assume it provides us with the centroid form of the lines (10.48), p. 375, namely centre points $\mathbf{x}_{0i}$ and the directions α_i of the line segments, together with the standard deviations σ_{q_i} of the positions across the lines and the standard deviations σ_{α_i} of the directions. This information may be achieved from algorithm 8, p. 401. We first need to condition the observations. Here, we scale the coordinates of the centroids $\mathbf{x}_{0i}$ such that the absolute value of the maximum coordinate of all points is 1. Then, using (10.47), we first get the distance d and the position m_0 of $\mathbf{x}_0$ on the line; from this, with (10.45), we calculate the covariance $\sigma_{\phi d}$; then we obtain σ_d from (10.46) and the covariance matrix $\Sigma_{\mathbf{l}^e\mathbf{l}^e}$ from (10.49). As we want to perform the estimation with spherically normalized coordinates, we determine $\mathbf{l}_i^s$ and their covariance matrix $\Sigma_{\mathbf{l}^s\mathbf{l}^s}$ from (10.52), p. 376. From now on, we assume that all homogeneous vectors are spherically normalized and skip the superscript ^{s} to simplify notation.

We assume the lines intersect in a common point $\mathbf{x}$. Determining approximate values in this case is simple. We use a pair $(\mathbf{l}_{i'}, \mathbf{l}_{i''})$ of lines, for which the inner product $\mathbf{l}_{i'} \cdot \mathbf{l}_{i''}$ of the spherically normalized coordinates is small enough, and determine $\widehat{\mathbf{x}}^a = \mathsf{N}(\mathbf{l}_{i'} \times \mathbf{l}_{i''})$.

The non-linear Gauss–Helmert model for estimating $\widehat{\mathbf{x}}$ reads

$$\widehat{\mathbf{l}}_i^{\mathsf{T}}\, \widehat{\mathbf{x}} = 0\,, \qquad i = 1, ..., I\,. \tag{10.257}$$

We now derive the linearized model for the reduced coordinates $\boldsymbol{l}_{ri}$ and $\boldsymbol{x}_r$ of the given lines and the unknown point. Starting from approximate values $\widehat{\mathbf{l}}_i^a$ and $\widehat{\mathbf{x}}^a$ with (4.427), p. 163, we obtain

$$\widehat{\mathbf{l}}_i = \mathbf{l}_i + \widehat{\mathbf{v}}_i = \widehat{\mathbf{l}}_i^a + \widehat{\Delta\mathbf{l}}_i\,, \tag{10.258}$$

and with $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}^a + \widehat{\Delta\mathbf{x}}$, the usual linearized model reads as follows:

$$\widehat{\mathbf{l}}_i^{a\mathsf{T}}\widehat{\mathbf{x}}^a + \widehat{\mathbf{l}}_i^{a\mathsf{T}}\Delta\widehat{\mathbf{x}} + \widehat{\mathbf{x}}^{a\mathsf{T}}\Delta\widehat{\mathbf{l}}_i = 0\,. \tag{10.259}$$

With the Jacobian $J_r(\mathbf{a}) = \mathsf{null}(\mathbf{a}^{\mathsf{T}})$ from (10.25), p. 370 applied to both 3-vectors, using the corrections $\widehat{\Delta\boldsymbol{l}}_{ri}$ and $\widehat{\Delta\boldsymbol{x}}_r$ of the reduced coordinates

$$\boldsymbol{l}_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a)\,\mathbf{l}_i \qquad \boldsymbol{x}_r = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}^a)\,\mathbf{x}\,, \tag{10.260}$$

of the lines and the unknown point, we can express the original corrections as

$$\Delta\widehat{\mathbf{x}} = J_r(\widehat{\mathbf{x}}^a)\,\widehat{\Delta\boldsymbol{x}}_r\,, \qquad \Delta\widehat{\mathbf{l}}_i = J_r(\widehat{\mathbf{l}}_i^a)\,\widehat{\Delta\boldsymbol{l}}_{ri}\,, \tag{10.261}$$

and, therefore, obtain the linearized model for the minimal parametrization of both, the unknown point coordinates $\mathbf{x}$ and the fitted values of $\mathbf{l}_i$,

$$\widehat{\mathbf{l}}_i^{a\mathsf{T}}\,\widehat{\mathbf{x}}^a + \boldsymbol{a}_{ri}^{\mathsf{T}}\,\widehat{\Delta\boldsymbol{x}}_r + \boldsymbol{b}_{ri}^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}}_{ri} = 0\,, \qquad i = 1, ..., I \tag{10.262}$$

(cf. (4.429), p. 164), with the Jacobians, which are the 2-vectors

$$\boldsymbol{a}_{ri}^{\mathsf{T}} = \widehat{\mathbf{l}}_i^{a\mathsf{T}}\, J_r(\widehat{\mathbf{x}}^a) \qquad \boldsymbol{b}_{ri}^{\mathsf{T}} = \widehat{\mathbf{x}}^{a\mathsf{T}}\, J_r(\widehat{\mathbf{l}}_i^a)\,. \tag{10.263}$$

If we compare the Jacobians in (10.263) to the Jacobians $\boldsymbol{a}_i^{\mathsf{T}} = \widehat{\mathbf{l}}_i^{a\mathsf{T}}$ in (10.259) and $\boldsymbol{b}_i^{\mathsf{T}} = \widehat{\mathbf{x}}^{a\mathsf{T}}$ of the original model (10.257), we see that the 3×2 Jacobians J_r developed in the previous section reduce the number of parameters in the linearized model to the minimum number in a straightforward manner.

We now minimize

$$\Omega(\widehat{\mathbf{x}}) = \sum_{i=1}^{I} \widehat{\boldsymbol{v}}_{ri}^{\mathsf{T}} \, (\Sigma_{l_{ri}l_{ri}}^{a})^{-1} \, \widehat{\boldsymbol{v}}_{ri} \tag{10.264}$$

under the constraints (10.262).

Due to $J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a)\widehat{\mathbf{l}}_i^a = \mathbf{0}$, the reduced residuals are

$$\widehat{\boldsymbol{v}}_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a)\widehat{\boldsymbol{v}}_i = J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a)(\widehat{\Delta \boldsymbol{l}}_i - \Delta \boldsymbol{l}_i) = J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a)(\widehat{\Delta \boldsymbol{l}}_i - (\mathbf{l}_i - \widehat{\mathbf{l}}_i^a)) = \widehat{\Delta \boldsymbol{l}}_{ri} - \Delta \boldsymbol{l}_{ri} \,. \tag{10.265}$$

Equation (10.264) exploits the regularity of the covariance matrix $\Sigma_{l_{ri}l_{ri}}^{a}$ of the reduced coordinates $\boldsymbol{l}_{ri}$. It is given by

$$\Sigma_{l_{ri}l_{ri}}^{a} = J_i^{\mathsf{T}} \Sigma_{\mathbf{l}_i\mathbf{l}_i}^{a} J_i \,, \qquad J_i = J_r(\widehat{\mathbf{l}}_i^{a\mathsf{T}}) \, R_{ab}(\mathbf{l}_i, \widehat{\mathbf{l}}_i^a) \,, \tag{10.266}$$

with the covariance matrix $\Sigma_{\mathbf{l}_i\mathbf{l}_i}$ referring to the original observations $\mathbf{l}_i$ transferred into the tangent space at $\widehat{\mathbf{l}}_i^a$ and the minimal rotation $R_{ab}(\bullet, \bullet)$ between two vectors as in (8.76), p. 340.

Due to the use of a minimal parametrization, we now do not have constraints between the unknown parameters anymore. Therefore the 2×2 normal equation system reads as $N\widehat{\Delta \boldsymbol{x}}_r = \boldsymbol{n}$, with

$$N = \left(\sum_{i=1}^{I} \boldsymbol{a}_{ri} (\boldsymbol{b}_{ri}^{\mathsf{T}} \Sigma_{l_{ri}l_{ri}}^{a} \boldsymbol{b}_{ri})^{-1} \boldsymbol{a}_{ri}^{\mathsf{T}} \right) \,, \qquad \boldsymbol{n} = \sum_{i=1}^{I} \boldsymbol{a}_{ri} (\boldsymbol{b}_{ri}^{\mathsf{T}} \Sigma_{l_{ri}l_{ri}}^{a} \boldsymbol{b}_{ri})^{-1} \, c_{g_i} \,, \tag{10.267}$$

and the residual of the constraint, cf. (4.444), p. 165,

$$c_{g_i} = -\widehat{\mathbf{l}}_i^{a\mathsf{T}} \, \widehat{\mathbf{x}}^a - \boldsymbol{b}_{ri}^{\mathsf{T}} \, J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^{a\mathsf{T}}) \, \mathbf{l}_i \tag{10.268}$$

regarding (10.258) and the fact $\widehat{\boldsymbol{l}}_{ri}^{a} = \mathbf{0}$. The corrections to the observations are (cf. (4.448), p. 165)

$$\widehat{\Delta \boldsymbol{l}}_{ri} = \Sigma_{l_{ri}l_{ri}} \, \boldsymbol{b}_{ri} \, (\boldsymbol{b}_{ri}^{\mathsf{T}} \Sigma_{l_{ri}l_{ri}}^{a} \boldsymbol{b}_{ri})^{-1} (c_{g_i} - \boldsymbol{a}_{ri}^{\mathsf{T}} \, \widehat{\Delta \boldsymbol{x}}_r) + J_r^{\mathsf{T}}(\widehat{\mathbf{l}}_i^a) \, \mathbf{l}_i \,. \tag{10.269}$$

They will be used to update the estimated observations. For this, the update equations (cf. (10.24), p. 370 and (10.55), p. 376) in iteration $(\nu + 1)$, where $\widehat{\mathbf{l}}_i^{\nu} = \widehat{\mathbf{l}}_i^a$ and $\widehat{\mathbf{x}}^{\nu} = \widehat{\mathbf{x}}^a$, yield

$$\widehat{\mathbf{x}}^{(\nu+1)} = \mathsf{N}(\mathbf{x}^{\nu} + J_r(\mathbf{x}^{\nu})\widehat{\Delta \boldsymbol{x}}_r) \,, \qquad \widehat{\mathbf{l}}_i^{(\nu+1)} = \mathsf{N}(\mathbf{l}_i^{\nu} + J_r(\mathbf{l}_i^{\nu})\widehat{\Delta \boldsymbol{l}}_{ri}) \,, \tag{10.270}$$

with the minimal representation for both observations and unknown parameters.

After convergence, we determine the estimated variance factor from

$$\widehat{\sigma}_0^2 = \frac{\Omega(\widehat{\mathbf{x}})}{I - 2} \tag{10.271}$$

and the estimated covariance matrix of the coordinates of the vanishing point from

$$\widehat{\Sigma}_{\widehat{x}_r\widehat{x}_r} = \widehat{\sigma}_0^2 \, N^{-1} \,, \qquad \widehat{\Sigma}_{\widehat{x}\widehat{x}} = J_r(\widehat{\mathbf{x}}) \, \widehat{\Sigma}_{\widehat{x}_r\widehat{x}_r} \, J_r^{\mathsf{T}}(\widehat{\mathbf{x}}) \,, \tag{10.272}$$

using N from (10.267).

The following example compares the results of estimating a vanishing point using different optimization functions w.r.t. the achievable accuracy, i.e., the precision and the bias. Moreover, it demonstrates how to check the correctness of the different models following Sect. 4.6.8, p. 139.

Example 10.6.39: ***Comparing different optimization functions.***

The intersection point $\mathbf{x}$ of I 2D lines $\mathbf{l}_i$ can be determined using different optimization functions. We compare the following alternatives:

1. Minimizing the algebraically motivated error with spherically normalized homogeneous line coordinates (denoted by ALGs in short). It uses the direct solution from Sect. 10.5.3.1, p. 401,

$$\widehat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x},|\mathbf{x}|=1} \sum_{i=1}^{I} \mathbf{x}^{\mathsf{T}} \mathbf{l}_i^s \,. \tag{10.273}$$

   In the given example, the result for the Euclideanly normalized line coordinates does not differ significantly.
2. Minimizing the sum of the squared Euclidean distances of the point from the lines (denoted by SSDe in short), using the direct solution of Sect. 10.5.3.2, p. 401:

$$\widehat{\boldsymbol{x}} = \operatorname{argmin}_x \sum_{i=1}^{I} d^2(l_i, \mathcal{X}) \,. \tag{10.274}$$

3. Maximizing the likelihood of the Euclidean point coordinates $\boldsymbol{x}$ (denoted by MLEe in short) , or minimizing the weighted sum of the squares of the Euclidean point from line distances,

$$\widehat{\boldsymbol{x}} = \operatorname{argmin}_x \sum_{i=1}^{I} \left(\frac{d(l_i, \mathcal{X})}{\sigma_{d_i}} \right)^2 , \tag{10.275}$$

   taking the standard deviations σ_{d_i} of the distances into account.
4. Maximizing the likelihood of the reduced point coordinates $\boldsymbol{x}_r$ in the tangent space of the spherically normalized homogeneous vectors (denoted by MLEs in short); this is equivalent to minimizing the weighted sum of the squared distances of the lines from the unknown point in the tangent space, cf. (10.264),

$$\widehat{\boldsymbol{x}}_r = \operatorname{argmin}_{x_r} \sum_{i=1}^{I} \widehat{\boldsymbol{v}}_{ri}^{\mathsf{T}} \, (\boldsymbol{\Sigma}^a_{l_{ri} l_{ri}})^{-1} \, \widehat{\boldsymbol{v}}_{ri} \,. \tag{10.276}$$

We compare the four estimation results for the vanishing point in two cases, namely for a point close to the lines (Fig. 10.24) and for one far from the lines (Fig. 10.26).

The first figure, 10.24, shows the results of the four estimators for a vanishing point close to 50 generated lines. We assume a square image with 2000×2000 pixels. The line centres were randomly distributed but oriented towards the point $[1300, 1700]$ pixel referring to the unconditioned image, yielding lengths s_i in the range of $[100, 800]$ pixel. The lines were randomly perturbed in direction and position with Gaussian noise with individual standard deviations $\sigma_{\alpha_i} = 1/\sqrt{s_i^3 [\text{pixel}]/12}$ pixel and $\sigma_{q_i} = 1/\sqrt{s_i \, [\text{pixel}]}$ pixel, following (10.170), p. 400. The lines were conditioned such that the maximal coordinates are 1, which is also the scale in the figure. The figure shows the generated lines and the resultant coordinates of the intersection point of 100 samples, magnified around the centre by a factor of 1 000. For all estimators except the geometric one, we also give the threefold standard ellipse, which in the case of a Gaussian distribution contains 98.9% of the points. The lower right figure for the MLEs estimate on the sphere also shows the threefold standard ellipse of the algebraic solution (dashed lines).
We observe the following:

- The algebraically and the geometrically motivated solutions ALGs and SSDe show comparable precision. The same holds for the ML estimates MLEe and MLEs.
- The two estimates MLEe and MLEs are significantly better than the other two estimates, approximately by a factor of 3, see also the close-up Fig. 10.25.
- The empirical covariance matrices do not significantly differ from the theoretical prediction. Using the test (4.358), p. 140 in Sect. 4.6.8.2, we obtain the test statistic for the ALGs, for the MLEe and for the MLEs estimates

$$X^2_{\mathsf{ALGs}} \approx 5.8 \,, \qquad X^2_{\mathsf{MLEe}} \approx X_{\mathsf{MLEs}} \approx 3.7. \tag{10.277}$$

  All test statistics are well below the 99% critical value $\chi^2_{3,0.99} \approx 11.3$. This indicates that *the theoretically derived covariance matrices are reliable for the algebraic and the ML estimation.*
- The covariance matrices for the two MLEe estimates differ by less than 1 permille. This is to be expected, as the estimated point does not really lie far from the observed lines.

We now compare the results for a point far from the given lines, namely with the coordinates $[0, 200]$ in the conditioned coordinate system far outside the image frame. This corresponds to a maximal parallactic angle between two image lines of 0.5°. Figure 10.26 shows the scatter plots of 200 samples. The scales for the different estimators are different due to the different accuracies obtained. The true point is indicated by a crosshair, the empirical mean with a circle. Bias, precision, and accuracy of the four solutions are collected in the following Table 10.5.
We observe the following:

1. The precision of the algebraically and geometrically optimal estimates ALGs and SSDe are again significantly lower than for the ML estimates MLEe and MLEs by a factor of approximately 2.5.

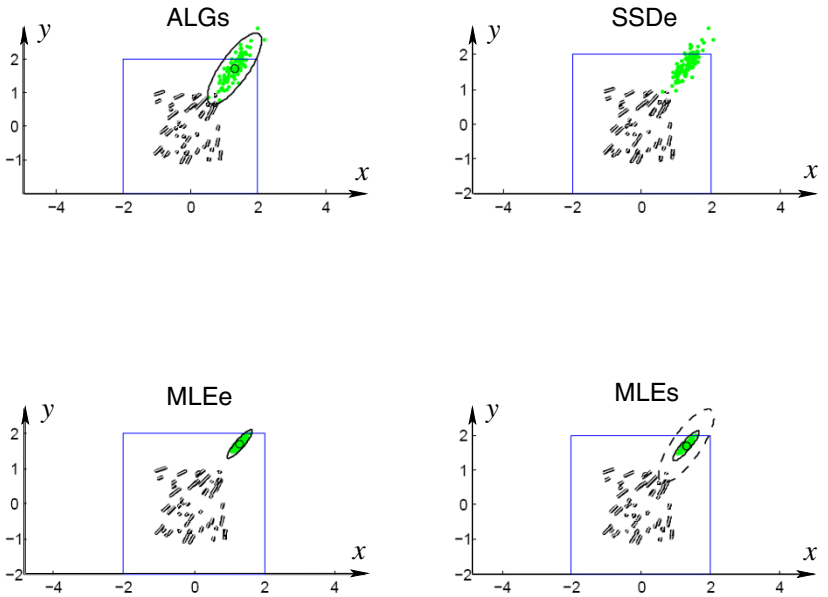


Fig. 10.24 Comparison of the four optimization functions ALGs, SSDe, MLEe, and MLEs in determining a vanishing point close to the generating lines. The point is estimated from 50 line segments lying in the square $[-1, +1]^2$. The scatter plot shows the result of 200 estimates. The ellipses shown are three times the standard ellipse of the theoretical covariance matrix

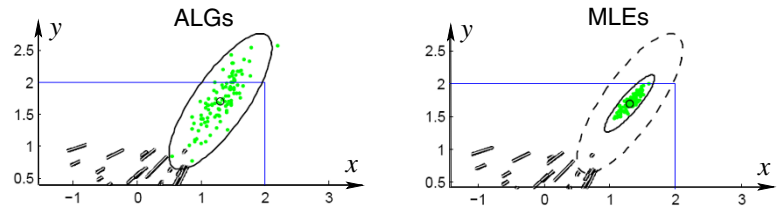


Fig. 10.25 Close up: left ALGs, right MLEs, see Fig. 10.24

Table 10.5 Estimated bias $\widehat{b}_y$, standard deviation $\widehat{\sigma}_{\widehat{y}}$, and accuracy $\widehat{s}_y$ of four estimates of the coordinates of an intersection point far from the line segments, derived from 1000 samples

	ALGs	SSDe	MLEe	MLEs
bias $\widehat{b}_y = \widehat{y} - \tilde{y}$	-0.9	28.7	-7.4	-0.2
precision $\widehat{\sigma}_{\widehat{y}}$	8.4	7.3	3.4	3.0
accuracy $\widehat{s}_y = \sqrt{\widehat{b}_y^2 + \widehat{\sigma}_{\widehat{y}}^2}$	8.5	29.7	8.2	3.1

2. The SSDe and MLEe estimates show a large bias of approximately 4σ and 2σ, respectively. The biases are caused by omitting higher-order terms, which, when working with the Euclidean coordinates, increase with decreasing parallactic angle.
3. The algebraic optimum of ALGs and the statistical optimum of MLEs using spherically normalized coordinates for the 2D lines practically show no bias.

The results confirm the superiority of the statistical estimation scheme MLEs based on spherically normalized homogeneous coordinates. ◇

Example 10.6.40: ***Estimating three vanishing points in an image.*** Fig. 10.27, top left, shows an image with all extracted straight line segments. The coordinate system of the image is in the upper left corner with the x-axis pointing downwards and the y-axis pointing to the right. The image size is 2304 × 3072 pixels, the focal length is approximately 3100 pixels. We have three vanishing points, the vanishing point v_1 points to the right, the vanishing point v_2 points to the left, and the vanishing point v_3 points downwards to the nadir point. The three groups of line segments pointing to the three vanishing points are shown in Fig. 10.27, top right. Applying the maximum likelihood estimation with reduced coordinates, as explained at the beginning of this section, we obtain the spherically normalized coordinates of the independently determined vanishing points as columns of the 3×3 matrix

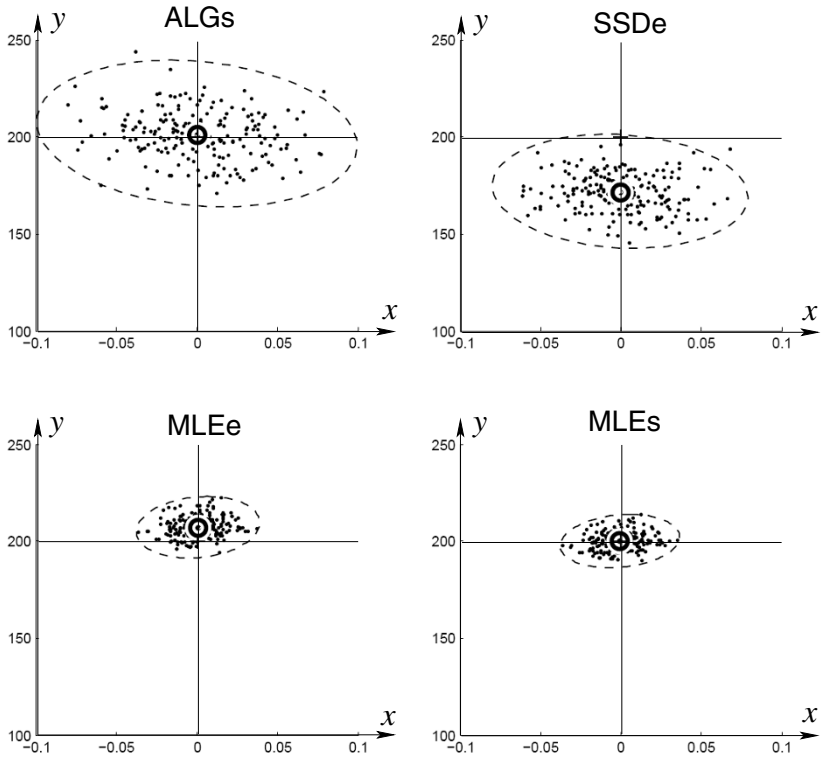


Fig. 10.26 Comparison of the accuracy for a far vanishing point given in the image coordinate system. Observe the different scales for x- and y-coordinates. Shown are the true value (crosshair), the scatter plots of 200 samples, and the derived estimated mean value (circle)

$$V = [\widehat{\mathbf{x}}_1^s, \widehat{\mathbf{x}}_2^s, \widehat{\mathbf{x}}_3^s] = \begin{bmatrix} -0.0810 & -0.0907 & +0.9957 \\ +0.8553 & -0.5182 & +0.0341 \\ +0.5117 & +0.8504 & +0.0865 \end{bmatrix} . \tag{10.278}$$

This corresponds to image coordinates

$$[\widehat{\boldsymbol{x}}_1, \widehat{\boldsymbol{x}}_2, \widehat{\boldsymbol{x}}_3] = \begin{bmatrix} -491 & -331 & 35684 \\ 5182 & -1889 & 1222 \end{bmatrix} [\text{pixel}] . \tag{10.279}$$

The x'-coordinates of the vanishing point v_3, pointing towards the nadir point, are close to infinity compared to the image size of 2000 pixels.

The semi-axes of the standard ellipses of the three directions are

$$1 : (0.1652°, 0.0401°) \qquad 2 : (0.0856°, 0.0373°) \qquad 3 : (0.1029°, 0.0226°) . \tag{10.280}$$

Clearly, the directions are quite precise. The uncertainty is visualized in Fig. 10.27, lower left. The vanishing point v_1 pointing to the left is more uncertain than the vertical vanishing point. The vertical vanishing point v_3 is very close to the nadir or the zenith; the confidence ellipse passes the line at infinity of the image plane.

The three vectors $\widehat{\mathbf{v}}_i$ should be mutually orthogonal, cf. Sect. 12.3.4.2, p. 531. However, they do not form a rotation matrix, since

$$V^\mathsf{T} V = \begin{bmatrix} 1.000000 & -0.000752 & -0.007163 \\ -0.000752 & 1.000000 & -0.034452 \\ -0.007163 & -0.034452 & 1.000000 \end{bmatrix} . \tag{10.281}$$

The three angles between the three directions deviate by $\alpha_{12} = 0.0431° = \arccos(0.000752)$, $\alpha_{23} = 0.410°$ and $\alpha_{31} = 1.97°$ from the nominal 90°. ◇

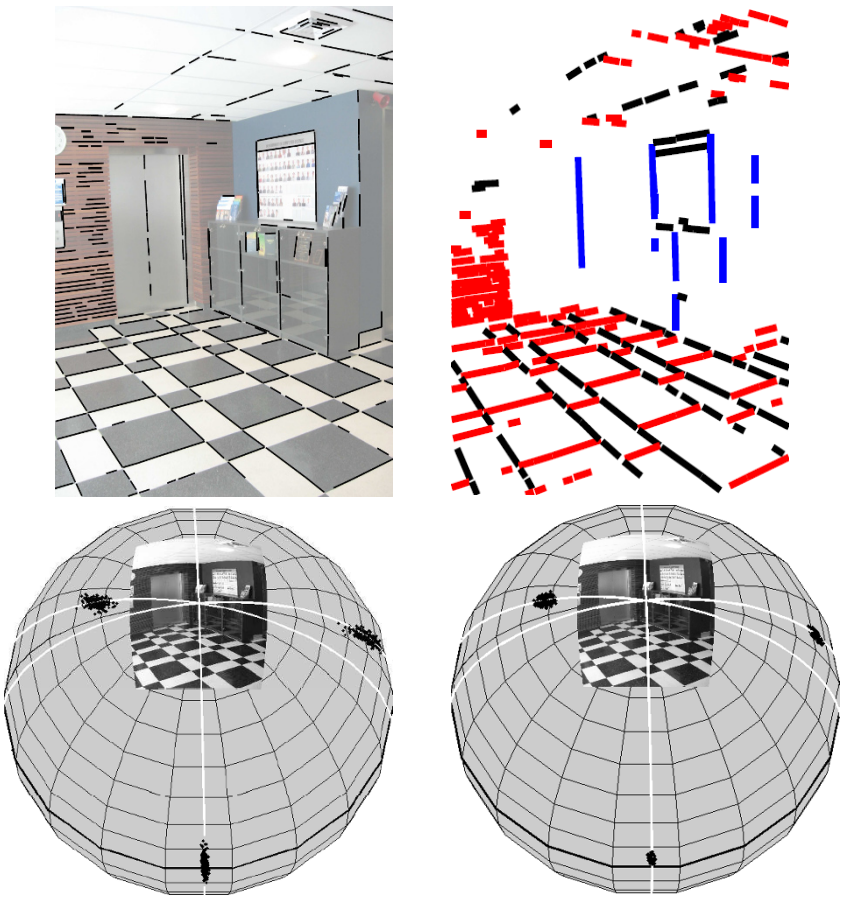


Fig. 10.27 Estimation of three vanishing points. **Upper left**: Extracted edges. **Upper right**: Edges classified as belonging to one of the three vanishing points. **Bottom left**: Image on viewing sphere together with uncertain vanishing points in three clusters estimated individually. The line at infinity of the image plane is located at the equator of the sphere visualized as thick black line. The uncertainty is shown by a sample of 250 points (see black dots), using a blow-up factor of 800. **Bottom right**: Uncertainty of vanishing points satisfying orthogonality constraints. Obviously, applying the orthogonality constraints to the approximate vanishing directions increases the precision

10.6.2.2 Enforcing Orthogonality Constraints

We therefore want to enforce the three orthogonality constraints which interrelate the three vanishing points provided the images are calibrated, cf. (12.260), p. 531. We now treat the estimated coordinates $\widehat{\mathbf{x}}_j^s$, $j = 1, 2, 3$, of the vanishing points (achieved in the previous step) together with their covariance matrices as observations $\mathbf{x}_j^s$, $j = 1, 2, 3$, in a second estimation process which aims at finding fitted observations which fulfil the orthogonality constraints. Observe, we (1) omit the hat, in order to characterize the coordinates $\mathbf{x}_j^s$ as given entities, and (2) leave out the naming of the variables, though they now play the role of observations. This leads to an estimation model of type C with constraints between the observations only, cf. Sect. 4.8.1, p. 162.

Based on approximate values $\widehat{\mathbf{x}}_j^a$ we again want to find $\widehat{\mathbf{x}}_j = \mathbf{x}_j + \widehat{\mathbf{v}}_j = \widehat{\mathbf{x}}_j^a + \widehat{\Delta \mathbf{x}}_j$, $j = 1, 2, 3$, which now fulfil the orthogonality constraints. The model for enforcing the three orthogonality constraints is $\boldsymbol{g}([\widehat{\mathbf{x}}_j]) = \mathbf{0}$, with

$$g_1 = \widehat{\mathbf{x}}_2^\mathsf{T} \widehat{\mathbf{x}}_3 \,, \quad g_2 = \widehat{\mathbf{x}}_3^\mathsf{T} \widehat{\mathbf{x}}_1 \,, \quad g_3 = \widehat{\mathbf{x}}_1^\mathsf{T} \widehat{\mathbf{x}}_2 \,. \tag{10.282}$$

After reducing the observations $\widehat{\boldsymbol{x}}_{rj} = J_x^\mathsf{T}(\widehat{\mathbf{x}}_j)\, \mathbf{x}_j, j = 1, 2, 3$, in order to be able to handle the singularity of the covariance matrices $\boldsymbol{\Sigma}_{\mathbf{x}_j \mathbf{x}_j}$, we obtain the reduced model $\boldsymbol{g}(\widehat{\boldsymbol{x}}_{rj}) = \mathbf{0}$,

$$g_1 = \widehat{\boldsymbol{x}}_{r2}^{\mathsf{T}} \widehat{\boldsymbol{x}}_{r3} = 0\,, \quad g_2 = \widehat{\boldsymbol{x}}_{r3}^{\mathsf{T}} \widehat{\boldsymbol{x}}_{r1} = 0\,, \quad g_3 = \widehat{\boldsymbol{x}}_{r1}^{\mathsf{T}} \widehat{\boldsymbol{x}}_{r2} = 0\,. \tag{10.283}$$

The linearized model, therefore, is $\boldsymbol{c}_g(\widehat{\boldsymbol{x}}^a) + \boldsymbol{B}_r^{\mathsf{T}} \widehat{\Delta \boldsymbol{x}}_r = \boldsymbol{0}$, or, explicitly,

$$\begin{bmatrix} \widehat{\boldsymbol{x}}_{r2}^{a\mathsf{T}} \widehat{\boldsymbol{x}}_{r3}^{a} \\ \widehat{\boldsymbol{x}}_{r3}^{a\mathsf{T}} \widehat{\boldsymbol{x}}_{r1}^{a} \\ \widehat{\boldsymbol{x}}_{r1}^{a\mathsf{T}} \widehat{\boldsymbol{x}}_{r2}^{a} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}^{\mathsf{T}} & \widehat{\boldsymbol{x}}_{r3}^{a\mathsf{T}} & \widehat{\boldsymbol{x}}_{r2}^{a\mathsf{T}} \\ \widehat{\boldsymbol{x}}_{r3}^{a\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} & \widehat{\boldsymbol{x}}_{r1}^{a\mathsf{T}} \\ \widehat{\boldsymbol{x}}_{r2}^{a\mathsf{T}} & \widehat{\boldsymbol{x}}_{r1}^{a\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \widehat{\Delta \boldsymbol{x}}_{r1} \\ \widehat{\Delta \boldsymbol{x}}_{r2} \\ \widehat{\Delta \boldsymbol{x}}_{r3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} .$$

The reduced covariance matrices $\Sigma^a_{x_{rj}x_{rj}}$ of the observations are taken from the previous estimation step.

Minimizing

$$\Omega = \sum_{j=1}^{3} (\widehat{\boldsymbol{x}}_{rj} - \boldsymbol{x}_{rj})^{\mathsf{T}} (\Sigma^a_{x_{rj}x_{rj}})^{-1} (\widehat{\boldsymbol{x}}_{rj} - \boldsymbol{x}_{rj}) \tag{10.284}$$

with the three constraints

$$\widehat{\Delta \boldsymbol{x}}_{rj} = \Sigma^a_{x_{rj}x_{rj}} \boldsymbol{B}_{rj} (\boldsymbol{B}_{rj}^{\mathsf{T}} \Sigma_{x_{rj}x_{rj}} \boldsymbol{B}_{rj})^{-1} (\boldsymbol{c}_{g_j} + \boldsymbol{B}_{rj}^{\mathsf{T}} \boldsymbol{x}_{rj}) + \boldsymbol{x}_{rj}\,, \quad j = 1, 2, 3 \tag{10.285}$$

yields the classical solution for the improved fitted observations, cf. Table 4.10, p. 172. They are used to obtain improved approximate values for the fitted values of the vanishing point coordinates. In spite of low redundancy of $R = 3$, it is useful to determine and report the estimated variance factor $\widehat{\sigma}_0^2 = \Omega/3$.

Example 10.6.41: ***Estimating the rotation matrix.*** Applying this procedure to the result of the previous example leads to a set of mutually orthogonal vanishing point directions, the columns of an exact rotation matrix. After applying the constraints, the precision of the directions is higher, indicated by the lower values for the semi-axes of their standard ellipses,

$$1: (0.0584°, 0.0238°)\,, \quad 2: (0.0617°, 0.0344°)\,, \quad 3: (0.0416°, 0.0205°). \tag{10.286}$$

Compared to (10.280), the maximum length of the semi-axes drops from 0.165° to 0.058°, which is an improvement by nearly a factor of 3. ◇

10.6.3 Estimating a Homography from Corresponding Points

The basic model for the homography between two sets of I corresponding points $(\mathcal{X}_i, \mathcal{X}_i')$, $i = 1, ..., I$, reads $\mathbf{x}_i' = \mathsf{H}\mathbf{x}_i$. In the following we assume all coordinate vectors to be spherically normalized without indicating this by a superscript ^{s}. Further we imply that the point pairs are stochastically independent. However, depending on the situation, we need to distinguish between two cases concerning the stochastical model of the coordinates of one pair:

1. One point set, namely $\{\mathcal{X}_i\}$, is fixed, the other is observed, i.e., is uncertain. Then only the coordinates of the $\{\mathcal{X}_i'\}$ need to be taken as samples of random variables. The mapping already is in the form $\mathbb{E}(\underline{\boldsymbol{l}}) = \boldsymbol{f}(\boldsymbol{x})$ of the Gauss–Markov model. Similarly to the previous example on vanishing point estimation, reduced coordinates need to be used for the transformed points.
2. Both point sets, $\{\mathcal{X}_i\}$ and $\{\mathcal{X}_i'\}$, are observed. Then all coordinates need to be treated as samples of random variables. They may be correlated due to the nature of the observation process. Here, we need the Gauss–Helmert model for estimating the optimal homography, as there is no simple way to express all observed coordinates as a function of the unknown parameters of the transformation, cf. the discussion in Sect. 4.8.3.2, p. 172.

The first model can obviously be treated as a special case of the second one. However, arriving at two constraints per point pair can be achieved either by selecting independent constraints, or by using reduced homogeneous coordinates only. We demonstrate the

selection of constraints for building a functional model of the correct rank, though the Gauss–Markov model is simpler to handle and transfers to the general setup of a bundle adjustment, cf. Sect. 15.4, p. 674.

The following example demonstrates (1) the use of correlated point pairs when estimating transformations, (2) the practical use of the representation of uncertain transformations, and (3) the validation of the predicted covariance matrix for both algebraically and statistically optimal estimates of a homography. We start by elaborating case 1, as it uses the Gauss–Helmert model, like the previously discussed estimation of the vanishing point.

10.6.3.1 Homography from Uncertain Point Pairs

When estimating a homography between two point sets, we generally assume the points have different covariance matrices (e.g., due to the point detector working on different resolutions of the image). Concerning the correlation between the coordinates of one point pair, however, we need to distinguish between two cases: (1) the two points of a point pair are detected independently in two images, and (2) the point $\boldsymbol{x}_i$ is detected in one image and the coordinates in the second image are determined by finding the best parallax $\boldsymbol{x}'_i - \boldsymbol{x}_i$, as then the two coordinate pairs are not independent anymore. To allow for both cases, we will treat the coordinates of a point pair of two corresponding points as a statistical unit. For the statistically optimal estimation, we also assume an approximation $\widehat{\mathsf{H}}^a$ for the unknown homography $\mathcal{H}(\mathsf{H})$. Exercise 10.21

The non-linear model then reads

$$\mathsf{S}^{(s)}(\widehat{\mathbf{x}}'_i)\,\widehat{\mathsf{H}}\,\widehat{\mathbf{x}}_i = \mathbf{0}\,, \qquad \mathbb{D}\left(\begin{bmatrix}\mathbf{x}_i\\ \mathbf{x}'_i\end{bmatrix}\right) = \underset{4\times 4}{\Sigma_{ii}}\,, \qquad i = 1, ..., I\,. \tag{10.287}$$

Here, we assumed the reduction of the constraints to two linearly independent constraints has been adopted, indicated by the superscript (s), cf. Sect. 7.4.1, p. 317. In the following, we assume that the selection is realized by pre-multiplication of the original constraint equation, $\mathsf{S}(\widehat{\mathbf{x}}'_i)\,\widehat{\mathsf{H}}\,\widehat{\mathbf{x}}_i = \mathbf{0}$, cf. (7.113), p. 315, with a suitable 2×3 matrix $[\boldsymbol{e}^{[3]}_{i'}, \boldsymbol{e}^{[3]}_{i''}]^\mathsf{T}$.

For linearization, we rewrite it in three forms, moving the fitted observations and individual unknown parameters to the right, cf. (7.140), p. 321,

$$\boldsymbol{g}_i(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}'_i, \widehat{\mathsf{H}}) = \mathsf{S}^{(s)}(\widehat{\mathbf{x}}'_i)\,\widehat{\mathsf{H}}\,\widehat{\mathbf{x}}_i = -\mathsf{S}^{(s)}(\widehat{\mathsf{H}}\,\widehat{\mathbf{x}}_i)\,\widehat{\mathbf{x}}'_i = (\widehat{\mathbf{x}}_i^\mathsf{T} \otimes \mathsf{S}^{(s)}(\widehat{\mathbf{x}}'_i))\,\mathrm{vec}(\widehat{\mathsf{H}}) = \mathbf{0}\,. \tag{10.288}$$

Therefore the linearized model is

$$\begin{aligned}\mathbf{0} = \mathsf{S}^{(s)}(\widehat{\mathbf{x}}'^a_i)\,\widehat{\mathsf{H}}^a\,\widehat{\mathbf{x}}^a_i + \mathsf{S}^{(s)}(\widehat{\mathbf{x}}'^a_i)\,\widehat{\mathsf{H}}^a\,\widehat{\Delta\mathbf{x}_i} &\qquad (10.289)\\ - \mathsf{S}^{(s)}(\widehat{\mathsf{H}}^a\,\widehat{\mathbf{x}}^a_i)\,\widehat{\Delta\mathbf{x}'_i} &\qquad (10.290)\\ + (\widehat{\mathbf{x}}_i^{a\mathsf{T}} \otimes \mathsf{S}^{(s)}(\widehat{\mathbf{x}}'^a_i))\,\widehat{\Delta\mathbf{h}}\,, &\qquad (10.291)\end{aligned}$$

with the corrections $\widehat{\Delta\mathbf{x}_i}$ and $\widehat{\Delta\mathbf{x}'_i}$ to the observed homogeneous coordinates and the correction $\widehat{\Delta\mathbf{h}} = \mathrm{vec}(\widehat{\Delta\mathsf{H}})$ to the homography. This form of the linearized model refers to the redundant representation of the homogeneous entities, and does not show the envisaged minimal parametrization.

We now combine the two vectors $\widehat{\mathbf{x}}_i$ and $\widehat{\mathbf{x}}'_i$ of fitted observations and use the updates in the notation of the standard estimation procedure with the Gauss–Helmert model,

$$\widehat{\boldsymbol{l}}_i \doteq \begin{bmatrix}\widehat{\mathbf{x}}_i\\ \widehat{\mathbf{x}}'_i\end{bmatrix} = \begin{bmatrix}\mathsf{N}(\widehat{\mathbf{x}}^a_i + J_r(\widehat{\mathbf{x}}^a_i)\,\widehat{\Delta\boldsymbol{x}_{ri}})\\ \mathsf{N}(\widehat{\mathbf{x}}'^a_i + J_r(\widehat{\mathbf{x}}'^a_i)\,\widehat{\Delta\boldsymbol{x}'_{ri}})\end{bmatrix}\,, \tag{10.292}$$

with the minimal four corrections

$$\widehat{\Delta \boldsymbol{l}}_{ri} \doteq \begin{bmatrix} \widehat{\Delta \boldsymbol{x}}_{ri} \\ \widehat{\Delta \boldsymbol{x}}'_{ri} \end{bmatrix} \tag{10.293}$$

for each point pair. The update for the homography is

$$\widehat{\mathsf{H}} = \exp\left\{\mathsf{K}(\widehat{\Delta \boldsymbol{p}})\right\} \, \widehat{\widehat{\mathsf{H}}}^a \,, \tag{10.294}$$

with the 8-vector $\Delta \boldsymbol{p}$ of parameters realizing a minimal representation and guaranteeing a traceless matrix K. We are not using the classical notation $\boldsymbol{x}$ for the unknown parameters in an estimation problem here, in order not to get confused with the symbol for the coordinates. With the linearized updates (cf. (10.25) and (10.289))

$$\Delta \mathbf{x}_i(\boldsymbol{x}_{ri}) = J_r(\widehat{\mathbf{x}}_i^a) \Delta \boldsymbol{x}_{ri} \,, \qquad \Delta \mathbf{x}'_i(\boldsymbol{x}'_{ri}) = J_r(\widehat{\mathbf{x}}_i'^a) \Delta \boldsymbol{x}'_{ri} \,, \tag{10.295}$$

$$\Delta \mathbf{h}(\Delta \boldsymbol{p}) = (\widehat{\mathbf{x}}_i^{a\mathsf{T}} \otimes \mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i'^a)) \, J_{h,\Delta p} \, \Delta \boldsymbol{p} \tag{10.296}$$

and using the Jacobian

$$J_{h,\Delta p} = (\widehat{\mathsf{H}}^a \otimes I_3) \begin{bmatrix} I_8 \\ -1 \mid 0 \mid 0 \mid 0 \mid -1 \mid 0 \mid 0 \mid 0 \end{bmatrix} \tag{10.297}$$

from (10.107), p. 385, we arrive at the final form of the linearized model,

$$\mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i'^a) \, \widehat{\mathsf{H}}^a \, \widehat{\mathbf{x}}_i^a + A_{ri}^{\mathsf{T}} \widehat{\Delta \boldsymbol{p}} + B_{ri}^{\mathsf{T}} \widehat{\Delta \boldsymbol{l}}_{ri} = \mathbf{0} \qquad i = 1, ..., I \,, \tag{10.298}$$

with

$$A_{ri}^{\mathsf{T}} = (\widehat{\mathbf{x}}_i^{a\mathsf{T}} \otimes \mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i'^a)) \, J_{h,\Delta p} \tag{10.299}$$

$$B_{ri}^{\mathsf{T}} = \left[\mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i'^a) \, \widehat{\mathsf{H}}^a \, J_r(\widehat{\mathbf{x}}_i^a), \; -\mathsf{S}(\widehat{\mathsf{H}}^a \widehat{\mathbf{x}}_i^a) \, J_r(\widehat{\mathbf{x}}_i'^a)\right] . \tag{10.300}$$

Observe that the parametrization of the update of the homography is minimal, so no additional constraint is required. With these derivations, we are now prepared to apply algorithm 15, p. 416.

Remark: The Gauss–Helmert model with reduced homogeneous coordinates starts from the relation

$$(\widehat{\mathbf{x}}'_i)^s - \mathsf{N}(\widehat{\mathsf{H}} \, \widehat{\mathbf{x}}_i) = \underset{3\times 1}{\mathbf{0}} \,, \quad i = 1, ..., I \tag{10.301}$$

using spherically normalized coordinates in both coordinate systems. Its linearization assumes approximate values, especially $\widehat{\mathbf{x}}_i'^a$ for $\widehat{\mathbf{x}}'_i$; thus, it can start from

$$J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i^a) \left(\widehat{\mathbf{x}}'_i - \mathsf{N}(\widehat{\mathsf{H}} \, \widehat{\mathbf{x}}_i)\right) = \underset{2\times 1}{\mathbf{0}} \,, \quad i = 1, ..., I \,, \tag{10.302}$$

since $J_r(\mathbf{x}'_i) = [\mathbf{x}'_i]^\perp$ (cf. Hartley and Zisserman, 2000, Sect. 4.9.2(ii)). Obviously, both models realize the selection by a projection on the tangent space of $\widehat{\mathbf{x}}_i^s$: in the previous model (10.287), p. 425, using $\mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i)$, in the current model with $J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i)$, where both matrices are the transpose of the null space of $\widehat{\mathbf{x}}_i^{\mathsf{T}}$, as $\widehat{\mathbf{x}}_i^{\mathsf{T}} \mathsf{S}^{(s)}(\widehat{\mathbf{x}}_i) = \widehat{\mathbf{x}}_i^{\mathsf{T}} J_r(\widehat{\mathbf{x}}_i) = \mathbf{0}$, cf. the remark when introducing the reduced coordinates after (10.26), p. 370.

◇

10.6.3.2 Homography from Pairs of Uncertain and Fixed Points

We now assume the point set $\{\mathbf{x}'_i\}$ is uncertain, whereas the coordinates of the points $\{\mathbf{x}_i\}$ are fixed, i.e., they are non-stochastic. Then we can write the model for the homography as

$$\widehat{\mathbf{x}}'_i = \mathsf{N}(\widehat{\mathsf{H}} \, \mathbf{x}_i) \,, \quad \mathbb{D}(\underline{\mathbf{x}}'_i) = \Sigma_{\mathbf{x}'_i \mathbf{x}'_i} \,, \quad i = 1, ..., I \,. \tag{10.303}$$

This is the form of the non-linear Gauss–Markov model. Again we assume that all coordinate vectors are spherically normalized without indicating this by a superscript s.

As the covariance matrix of $\underline{\mathbf{x}}'_i$ is singular, we first reduce these equations to the individual tangent spaces,

$$\widehat{\boldsymbol{x}}'_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i'^a)\,\mathsf{N}(\widehat{\mathsf{H}}\,\mathbf{x}_i)\,, \quad \mathbb{D}(\underline{\boldsymbol{x}}'_{ri}) = \boldsymbol{\Sigma}_{x'_{ri}x'_{ri}}\,, \quad i = 1, ..., I\,. \tag{10.304}$$

The predicted coordinates are

$$\widehat{\mathbf{x}}_i'^a = \widehat{\mathsf{H}}^a\,\mathbf{x}_i = (\mathbf{x}_i^{\mathsf{T}} \otimes I_3)\,\widehat{\mathbf{h}}^a\,, \tag{10.305}$$

with $\mathbf{h} = \mathrm{vec}(\mathsf{H})$, based on the approximate estimated homography $\widehat{\mathsf{H}}^a$ and the regular covariance matrix

$$\boldsymbol{\Sigma}_{x'_{ri}x'_{ri}} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_{ri}'^a)\,\boldsymbol{\Sigma}_{\mathrm{x}'_i\mathrm{x}'_i}\,J_r(\widehat{\mathbf{x}}_{ri}'^a)\,. \tag{10.306}$$

Again using $\widehat{\mathsf{H}} = \exp(\mathsf{K}(\Delta\boldsymbol{p})\widehat{\mathsf{H}}^a$, cf. (10.98), the linearization of (10.303) yields

$$\widehat{\boldsymbol{x}}'_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i'^a)\,\mathsf{N}((\mathbf{x}_i^{\mathsf{T}} \otimes I_3)\,\widehat{\mathbf{h}}^a) + \underbrace{J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i'^a)J_{\mathrm{x}^s\mathrm{x}}(\widehat{\mathbf{x}}_i'^a)\,J_{\mathrm{xh}}(\widehat{\mathbf{x}}_i)\,J_{\mathrm{h},\Delta p}(\widehat{\mathbf{h}}^a)}_{A_i^{\mathsf{T}}\ 2\times 8}\,\Delta\widehat{\boldsymbol{p}}\,, \tag{10.307}$$

with the Jacobians

$$J_{\mathrm{x}^s\mathrm{x}}(\mathbf{x}) = \frac{1}{|\mathbf{x}|}\left(I_3 - \frac{\mathbf{x}\mathbf{x}^{\mathsf{T}}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}}\right) \quad \text{and} \quad J_{\mathrm{xh}}(\mathbf{x}) = \mathbf{x}^{\mathsf{T}} \otimes I_3 \tag{10.308}$$

(cf. (10.18), p. 368 and (10.305)) and $J_{\mathrm{h},\Delta p}$ from (10.107), p. 385 or (10.297).

This yields the normal equation system $N\Delta\widehat{\boldsymbol{p}} = \boldsymbol{n}$ for the corrections $\Delta\widehat{\boldsymbol{p}}$ of the unknown parameters of the homography, with

$$N = \sum_i A_i \Sigma_{x_{ri}x_{ri}}^{-1} A_i^{\mathsf{T}}\,, \qquad \boldsymbol{n} = \sum_i A_i \Sigma_{x_{ri}x_{ri}}^{-1} \boldsymbol{x}'_{ri}\,. \tag{10.309}$$

It can be shown that this normal equation system is identical to the one of the previous model, Sect. 10.6.3.1, if the stochastical model is adapted, i.e., $\boldsymbol{\Sigma}_{x_ix_i} = \boldsymbol{0}$ is chosen.

Example 10.6.42: ***Algebraically and statistically optimal estimates.*** The example is meant to (1) demonstrate the check of the implementation following Sect. 4.6.8, p. 139, (2) investigate the effect of correlations between the two points of a correspondence pair, and (3) evaluate the quality of the theoretical covariance matrices of the estimated homography when performing an algebraically and a statistically optimal estimation.

We assume that the coordinates are perturbed by random noise, and that the coordinates might be highly correlated. For a simulation, we use a 3×3 grid $[-1, 0, +1]^2$ and transform it with the true homography,

$$\tilde{\mathsf{H}} = \begin{bmatrix} 1 & -0.2 & 2.6 \\ 0.1 & 1.2 & -0.3 \\ 0.25 & 0.2 & 1 \end{bmatrix}. \tag{10.310}$$

We add Gaussian noise to the coordinates with the following covariance matrix,

$$\mathbb{D}\left(\begin{bmatrix} \underline{x}_i \\ \underline{y}_i \\ \underline{x}'_i \\ \underline{y}'_i \end{bmatrix}\right) = \begin{bmatrix} \sigma^2 & 0 & \rho\sigma^2 & 0 \\ 0 & \sigma^2 & 0 & \rho\sigma^2 \\ \rho\sigma^2 & 0 & \sigma^2 & 0 \\ 0 & \rho\sigma^2 & 0 & \sigma^2 \end{bmatrix} = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \otimes I_2\,, \tag{10.311}$$

with $\sigma = 0.001$. We have two correlations, $\rho = 0$ and $\rho = +0.98$ for the two cases, respectively. A positive correlation appears if one of the two points is identified and the coordinate differences with the other point are derived, e.g., by some correlation technique, cf. Exerc. 21, p. 435. Here we can assume that the precision of the estimated homography is superior to the case where the correlation is zero for the following reason: Assume the homography is close to a unit matrix. Then the Jacobian $B_{ri}^{\mathsf{T}} \approx [1, -1] \otimes S^{(s)}(\widehat{\mathbf{x}}_i^a)J_r(\widehat{\mathbf{x}}_i^a))$, cf. (10.299). Therefore the covariance $\boldsymbol{\Sigma}_{gg} = B^{\mathsf{T}}\boldsymbol{\Sigma}_{ll}B$ for the residuals of the constraints is smaller for positive correlations than for negative correlations. For a correlation of $\rho = 0.98$ we can expect the resultant

standard deviations of the homography parameters to be smaller by a factor of up to $1/\sqrt{1-\rho} \approx 7$, cf. Exerc. 15, p. 58.

We first check the implementation based on a simulation study. We generate $K = 1,000$ sample values and derive the algebraically and statistically optimal estimates, leading to two sets of estimates, $\widehat{\mathsf{H}}_{a,k}$ and $\widehat{\mathsf{H}}_{s,k}$, $k = 1, 2, ..., K$.

We start with the test on the validity of the *variance factor* of the statistically optimal estimation from K sample estimations. It leads to a test statistic of $F = \sum_k \sigma_{0k}^2/K = 0.9464$, which is well in the 99% nonrejection region $[F_{0.005,1000,\infty}, F_{0.995,1000,\infty}] = [0.8886, 1.1190]$. This is a first indication that we do not have to doubt the implementation of the homography estimation, cf. Sect. 4.6.8.1, p. 140.

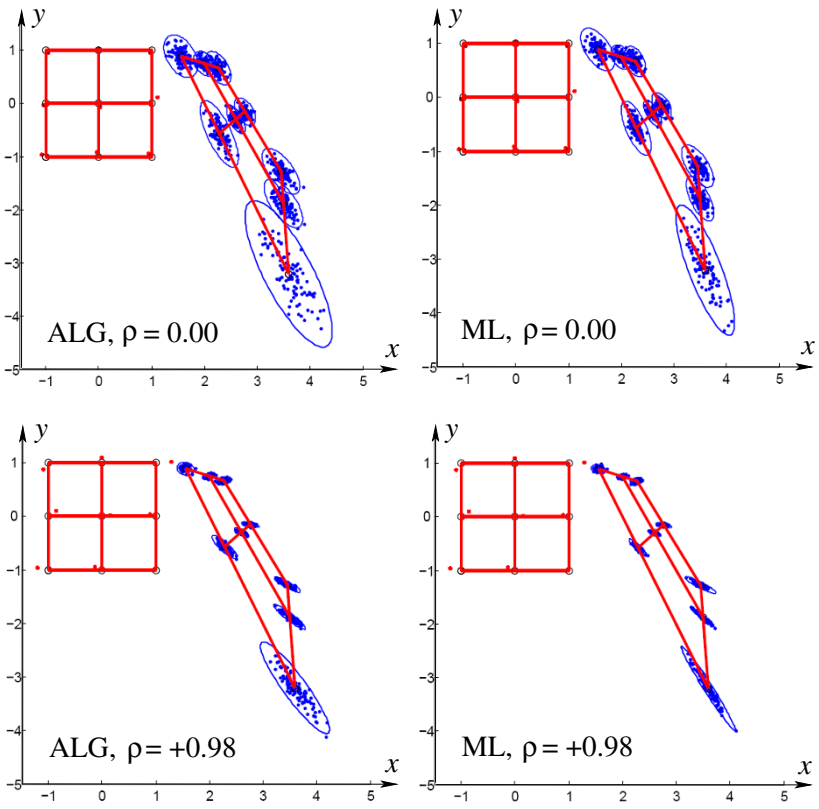


Fig. 10.28 Correctness of the theoretical covariance matrix for optimal homography estimation from nine correlated point pairs. **Left:** the algebraic optimization. **Right:** statistical optimization. **Top:** Correlation $\rho = 0.00$. **Bottom:** $\rho = +0.98$. The square is transformed into the quadrangle. The result of 100 samples for the transformed points and the threefold standard ellipses indicate the consistency of the theoretical and the empirical precision. Scatter plots and threefold standard ellipses are magnified by a factor of 100

The check on the correctness of the implementation continues with the evaluation of the theoretical covariance matrix, cf. Sect. 4.6.8.2, p. 140. For this we compare the empirical covariance matrix, derived from the sample, and the theoretical covariance matrix. Following the evaluation scheme in Sect. 4.6.8.2, p. 140, we determine the empirical means $\widehat{\boldsymbol{\mu}}_{\mathrm{h}_a}$ and $\widehat{\boldsymbol{\mu}}_{\mathrm{h}_s}$ and the empirical covariance matrices $\widehat{\boldsymbol{\Sigma}}_{\widehat{\mathrm{h}}_a\widehat{\mathrm{h}}_a}$ and $\widehat{\boldsymbol{\Sigma}}_{\widehat{\mathrm{h}}_s\widehat{\mathrm{h}}_s}$ in order to compare them with the theoretical values. For the mean, this is $\tilde{\mathbf{h}} = \mathrm{vec}(\tilde{\mathsf{H}})$. The comparison of the covariance matrices requires care as they are singular since the degree of freedom is 8, whereas the size of the matrices is 3×3. Therefore, we compare the covariance matrices of the minimal parameters $\widehat{\Delta \boldsymbol{p}}$. The covariance matrix of the minimal parameters,

$$\widehat{\Delta \boldsymbol{p}} = J_{\Delta p,\mathrm{h}}\, \widehat{\mathbf{h}}, \tag{10.312}$$

can be derived from

$$\boldsymbol{\Sigma}_{\widehat{\Delta p}\widehat{\Delta p}} = J_{\Delta p,\mathrm{h}}\, \boldsymbol{\Sigma}_{\widehat{\mathrm{h}}\widehat{\mathrm{h}}}\, J_{\Delta p,\mathrm{h}}^{\mathsf{T}} \tag{10.313}$$

using the Jacobian $J_{\Delta p,\mathrm{h}}$ given in (10.113), p. 386. The covariance matrix of the minimal parameters has full rank 8 in general. For the algebraically optimal estimate, we use (4.521), p. 181 as the theoretical covariance matrix, cf. also (10.126), p. 388. The test statistic $X_{U,K-1} = 41.1$ from (3.57), p. 72 does not differ from 0 significantly, since the critical value is $\chi^2_{0.99,36} = 58.6$, as the degrees of freedom of the

chi-square distribution is $U(U+1)/2 = 36$, with $U = 8$ unknown parameters. This indicates the usefulness of (4.521), p. 181 for evaluating the result of an algebraically optimal solution. The hypothesis that the empirical covariance of the statistically optimal estimates is identical to the theoretical one cannot be rejected either, as the test statistic with 23.5 does not significantly differ from 0. Thus, there is no reason to doubt the usefulness of the theoretical covariance matrix resulting from the variance propagation within the estimation procedure.

Next, we check the correctness of the implementation by testing for bias; cf. Sect. 4.6.8.3, p. 141. We compare the empirical mean with the true value, which in the case of a simulation can be taken as error-free "ground truth". For calculating the Mahalanobis distance between the mean estimate $\boldsymbol{\mu}_{\underline{\widehat{\mathrm{h}}}}$ and the true value $\tilde{\mathbf{h}}$, we need to reduce the mean to the minimal parametrization (cf. (10.312)) to be able to invert the covariance matrix of the mean estimate. The squared Mahalanobis distance in both cases does not differ significantly from 0, as the two test statistics with $F_a = 13.0$ and $F_s = 10.9$ are below the critical value $F_{0.99,8,\infty} = 20.1$.

Finally we visualize the samples. Because a visualization of the covariance matrix of the homography is difficult, we visualize the threefold standard ellipses for the transformed points $\widehat{\underline{\mathrm{H}}}\,\underline{\mathbf{x}}_i$, taking both uncertainties into account, i.e., the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}_i\mathbf{x}_i}$ of the points $\underline{\mathbf{x}}_i$ and the covariance matrix $\boldsymbol{\Sigma}_{\widehat{\mathrm{h}}\widehat{\mathrm{h}}}$ of the estimated homography $\widehat{\underline{\mathrm{H}}}$. Figure 10.28 shows the result for the algebraically and the statistically optimal estimation for a zero and a positive correlation coefficient $\rho = +0.98$ between the coordinates.

Observe, the resultant homography is more precise if the two points of a point pair are positively correlated, as expected. The maximum gain in standard deviation is given by the square root of largest eigenvalue of $\boldsymbol{\Sigma}^{(\rho=0.00)}_{\widehat{h}\widehat{h}} \left(\boldsymbol{\Sigma}^{(\rho=0.98)}_{\widehat{h}\widehat{h}}\right)^{-1}$, which is $\sqrt{\lambda_{\max}} \approx 6.4$, and close to the prediction for the factor 7 made above. The gain in standard deviation for the individual coordinates is lower, but still more than by a factor of 3, as can be seen in the figure. ◇

10.6.4 Estimating a Symmetric Roof from a Point Cloud

With the following example we want to illustrate:

I The stepwise estimation, similar to the joint estimation of vanishing points and their orthogonality constraints (Sect. 10.6.2.2).
II The use of the Gauss–Helmert model with constraints between the parameters (estimation model E).

Assume a symmetric gable roof with left and right roof planes $\mathcal{A}$ and $\mathcal{B}$ is observed with an airborne laser range finder. Let the N and M points $\mathcal{X}_n$ and $\mathcal{Y}_m$ be observed with a common standard deviation of σ in all coordinates, an assumption which can easily be generalized. The task is to find optimal estimates for the plane parameters while taking the constraints for the symmetric gable roof into account.

A symmetric gable roof can be represented by two equally sloped planes passing through a horizontal 3D line. Then we need four parameters: Three for the gable line, a horizontal straight line (two parameters) at a certain height (one parameter), and one for the slope of the symmetric roof planes. Such an explicit modelling of roofs is cumbersome when addressing the many types of buildings appearing as polyhedra in reality.

Therefore, it is easier to represent roofs as parts of buildings, i.e., by the parameters for the planes, three independent parameters for each plane and additional constraints between the roof planes. As the incidence relations between the observed points and the unknown planes establish constraints between observations and unknown parameters, and the additional constraints refer to the unknown parameters, this leads in a natural way to a Gauss–Helmert model with constraints.

In our case, we need two constraints on the plane parameters, namely on the homogeneous 4-vectors $\mathbf{A}$ and $\mathbf{B}$:

1. The first constraint guarantees that the slopes s_i of the two planes are the same. Using the normal vectors $\boldsymbol{A}_h = [A_1, A_2, A_3]^\mathsf{T}$ and $\boldsymbol{B}_h = [B_1, B_2, B_3]^\mathsf{T}$ of the two planes for determining their slopes (see Fig. 5.11, p. 212) $s_A = \tan\alpha_{Z_A} = A_3/\sqrt{A_1^2 + A_2^2}$ and $s_B = \tan\alpha_{Z_B} = B_3/\sqrt{B_1^2 + B_2^2}$, we obtain the constraint $s_A = s_B$, or, without fractions,

$$A_3^2(B_1^2 + B_2^2) - (A_1^2 + A_2^2)B_3^2 = 0\,. \tag{10.314}$$

2. The second constraint guarantees that the gable line is horizontal. The direction of the gable line $\mathcal{L}$ is $\boldsymbol{L}_h = \boldsymbol{A}_h \times \boldsymbol{B}_h$. The slope of the 3D line $\mathcal{L}$ is zero if $L_3 = L_{h3} = 0$, or, explicitly if the third element of the cross product of the normals vanishes,

$$A_1 B_2 - A_2 B_1 = 0\,. \tag{10.315}$$

We do not need to include any constraints on the length of the homogeneous vectors, as the estimation is performed using reduced homogeneous coordinates. With the observations and the unknown parameters

$$\boldsymbol{l} = \begin{bmatrix} \{\mathbf{X}_n\} \\ \{\mathbf{Y}_n\} \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \tag{10.316}$$

the nonlinear Gauss–Helmert model reads as

$$\boldsymbol{g}(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{x}}) = \mathbf{0}: \qquad \begin{bmatrix} \{\tilde{\mathbf{X}}_n^\mathsf{T}\tilde{\mathbf{A}}\} \\ \{\tilde{\mathbf{Y}}_n^\mathsf{T}\tilde{\mathbf{B}}\} \end{bmatrix} = \mathbf{0} \tag{10.317}$$

$$\boldsymbol{h}(\tilde{\boldsymbol{x}}) = \mathbf{0}: \quad \begin{bmatrix} \tilde{A}_3^2(\tilde{B}_1^2 + \tilde{B}_2^2) - (\tilde{A}_1^2 + \tilde{A}_2^2)\tilde{B}_3^2 \\ \tilde{A}_1\tilde{B}_2 - \tilde{A}_2\tilde{B}_1 \end{bmatrix} = \mathbf{0}\,. \tag{10.318}$$

The numerical solution can be achieved in a two-step procedure (in the following example denoted by *procedure I*): first estimate the plane parameters independently by a Gauss–Helmert estimation (model type D), and in a second step treat the estimated plane parameters as observations and apply the constraints, using an estimation model with constraints between the observations only (model type C). Generally, it will be much more efficient than the one-step approach (in the example denoted by *procedure II*) using the Gauss–Helmert model with constraints (model type E). This is of advantage if only a small data set has to be processed.

Example 10.6.43: ***Numerical example.*** We simulate a specific situation (see Fig. 10.29) and discuss intermediate results, which also can be used for checking an implementation. Let the two roof

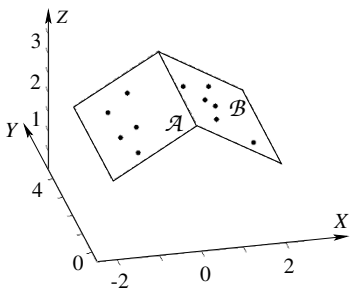


Fig. 10.29 Symmetric roof determined from 5 and 6 points

planes $\mathcal{A}$ and $\mathcal{B}$ cover the region $[-2 \cdots + 2] \times [0 \ldots 4]$; the gable has a height of 3, assuming all units are in meters. The slope of the roof is assumed to be $s = 0.6$. Then the true values for the two roof planes are

$$\tilde{\mathbf{A}} = \begin{bmatrix} 2.4 \\ 0 \\ -4 \\ 12 \end{bmatrix}, \qquad \tilde{\mathbf{B}} = \begin{bmatrix} -2.4 \\ 0 \\ -4 \\ 12 \end{bmatrix}. \tag{10.319}$$

The gable line has the Plücker coordinates

$$\tilde{\mathbf{L}} = \begin{bmatrix} \tilde{\boldsymbol{L}}_h \\ \tilde{\boldsymbol{L}}_0 \end{bmatrix} = [0, 1, 0 \mid -3, 0, 0]^\mathsf{T}, \tag{10.320}$$

which is horizontal (as $L_3 = 0$) and has the distance 3 to the origin (as $|\boldsymbol{L}_0|/|\boldsymbol{L}_h| = 3$). Assume plane $\mathcal{A}$ has been observed by $I_A = 5$ and plane $\mathcal{B}$ by $I_B = 6$ 3D points which have normally distributed noise in all three coordinates with $\sigma_X = 0.1$, and no correlations, cf. Table 10.6.

Table 10.6 Observed coordinates of the 3D points on the two planes

	$\mathcal{A}$			$\mathcal{B}$		
i	X	Y	Z	X	Y	Z
1	-0.8749	3.8996	2.3502	0.5370	2.1825	2.6343
2	-1.6218	3.2287	2.0395	0.8666	1.3679	2.2913
3	-1.6059	1.6108	1.8873	0.9658	3.6694	2.4097
4	-1.4275	0.0609	2.0249	1.0134	2.3504	2.2827
5	-1.2100	0.8542	2.4765	-0.0517	2.4755	2.9952
6				1.6559	-0.0646	2.0355

We now compare the two results, the two-step estimation with *procedure I* for the two planes separately and the subsequent application of the symmetry constraints, and the one-step estimation in *procedure II* with the Gauss–Helmert model with constraints.

Before beginning, we generate approximate values, best by using the algebraical optimum from the five and six point-plane incidences (cf. Sect. 10.5.2.1, p. 396), which yields

$$\widehat{\mathbf{A}}_{\text{alg}} = \begin{bmatrix} -0.9818 \\ 0.0247 \\ 1.0000 \\ -3.5338 \end{bmatrix}, \quad \widehat{\mathbf{B}}_{\text{alg}} = \begin{bmatrix} 0.5738 \\ 0.0106 \\ 1.0000 \\ -2.9306 \end{bmatrix}. \tag{10.321}$$

The two-step *procedure I* starts with separately determining the statistically optimal planes using the Gauss–Helmert model, after five and three iterations resulting in

$$\widehat{\mathbf{A}}^{(1)} = \begin{bmatrix} -0.8055 \\ 0.0336 \\ 1.0000 \\ -3.3162 \end{bmatrix}, \quad \widehat{\mathbf{B}}^{(1)} = \begin{bmatrix} 0.5799 \\ -0.0001 \\ 1.0000 \\ -2.9265 \end{bmatrix}, \tag{10.322}$$

normalized (for an easy comparison) such that $A_3 = B_3 = 1$. They are obviously not symmetric. Morever, due to the low redundancies of $R_A = I_A - 3 = 2$, $R_B = I_B - 3 = 3$, and due to the relatively poor distributions of the 3D points on the two roofs, their normals differ substantially from the given true values, namely by 8.0° and 0.8°. This is confirmed by the major axes of the standard ellipses of the normals,

$$\sigma_{A1} = 7.73^\circ, \quad \sigma_{A2} = 1.81^\circ \tag{10.323}$$

$$\sigma_{B1} = 4.85^\circ, \quad \sigma_{B2} = 1.90^\circ. \tag{10.324}$$

These standard deviations are derived by interpreting the first three elements of the plane coordinates $\mathbf{x} = \widehat{\boldsymbol{A}}_h$ and $\mathbf{y} = \widehat{\boldsymbol{B}}_h$ together with their 3×3 covariance matrix as uncertain 2D points on the unit sphere, thus as elements of $\mathbb{P}^2$, and determining the square roots of the eigenvalues of the reduced 2×2-covariance matrices $\boldsymbol{\Sigma}_{x_r x_r}$ and $\boldsymbol{\Sigma}_{y_r y_r}$, which indicate the extreme uncertainties of the normal directions.

We obtain the reduced covariance matrices of the plane estimates

$$\boldsymbol{\Sigma}^{(1)}_{\widehat{A}_r \widehat{A}_r} = 10^{-4} \begin{bmatrix} 1.3195 & -1.3290 & -0.0351 \\ -1.3290 & 14.0146 & -11.3422 \\ -0.0351 & -11.3422 & 10.8475 \end{bmatrix}, \tag{10.325}$$

$$\boldsymbol{\Sigma}^{(1)}_{\widehat{B}_r \widehat{B}_r} = 10^{-4} \begin{bmatrix} 2.3882 & -2.7349 & 0.7961 \\ -2.7349 & 5.5200 & -3.7663 \\ 0.7961 & -3.7663 & 4.4511 \end{bmatrix}; \tag{10.326}$$

the estimated variance factors for the two planes in this step 1 are $(\widehat{\sigma}^{(1)}_{0,A})^2 = 2.71$, $(\widehat{\sigma}^{(1)}_{0,B})^2 = 0.71$.

In step 2 of *procedure I* we will use the plane estimates $\{\widehat{\mathbf{A}}^{(1)}, \boldsymbol{\Sigma}^{(1)}_{\widehat{A}_r \widehat{A}_r}\}$ and $\{\widehat{\mathbf{B}}^{(1)}, \boldsymbol{\Sigma}^{(1)}_{\widehat{B}_r \widehat{B}_r}\}$ together with their covariances as observations. Step 2 after five iterations leads to the estimates

$$\widehat{\mathbf{A}} = \begin{bmatrix} -0.6075 \\ 0.0101 \\ 1.0000 \\ -3.0130 \end{bmatrix}, \quad \widehat{\mathbf{B}} = \begin{bmatrix} 0.6075 \\ -0.0101 \\ 1.0000 \\ -2.9300 \end{bmatrix}. \tag{10.327}$$

As expected, the symmetry constraints are fulfilled. The major axes of the standard ellipses of the two normals

$$\sigma_{A1} = \sigma_{B1} = 3.54^\circ\,, \qquad \sigma_{A2} = \sigma_{B2} = 1.31^\circ \tag{10.328}$$

are necessarily identical for both, since we enforced two symmetry constraints onto their relation.

The Plücker coordinates of the estimated gable line are

$$\widehat{\mathbf{L}} = [0.0161, 0.9999, -0.0000, -2.9673, 0.0477, -0.0610]^\mathsf{T}\,, \tag{10.329}$$

which is horizontal, as required. The standard deviation of the azimuth α of the gable is

$$\sigma_\alpha = 0.13^\circ\,. \tag{10.330}$$

The one-step *procedure II* with the Gauss–Helmert model with constraints after five iterations yields the estimates for the plane vectors normalized such that $A_3 = B_3 = 1$,

$$\widehat{\mathbf{A}} = \begin{bmatrix} -0.6039 \\ 0.0100 \\ 1.0000 \\ -3.0016 \end{bmatrix}\,, \quad \widehat{\mathbf{B}} = \begin{bmatrix} 0.6039 \\ -0.0100 \\ 1.0000 \\ -2.9272 \end{bmatrix}\,. \tag{10.331}$$

Obviously, the two planes are symmetric, as $A_1 = -B_1$ and $A_2 = -B_2$. The major axes of the standard ellipses of the two normals are identical to the result of the two-step estimation procedure.

The gable is represented in 3D by the Plücker coordinates,

$$\widehat{\mathbf{L}} = [0.0166, 0.9999, 0.0000, -2.9640, 0.0493, -0.0616]^\mathsf{T}. \tag{10.332}$$

The direction vector $[0.0166, 0.9999 \mid 0]^\mathsf{T}$ is horizontal, however, not exactly parallel to the Y-axis and in height 2.93 instead of 3. The directional error of the gable line is

$$\sigma_\alpha = 0.13^\circ\,. \tag{10.333}$$

It can be derived by interpreting the 3-vector $\widehat{\boldsymbol{L}}_h$ as homogeneous coordinates of a 2D point z, which due to $z_3 = 0$ is at infinity, and thus represents a direction, and deriving the covariance $\boldsymbol{\Sigma}_{z_r z_r}$. Due to the horizontality constraint it is singular; its maximum eigenvalue yields the variance σ_α^2 of the azimuth angle α.

The estimated variance factor $\widehat{\sigma}_0^2 = 1.1219$, as expected, does not indicate any discrepancies between the model and the data.

The results of the two estimation procedures obviously differ slightly. The reasons are linearization effects in the second estimation step, since measuring the Mahalanobis distance between the given points and the unknown planes under the constraints and measuring the Mahalanobis distance between the derived planes and the fitted planes refer to two different points of linearization. If the same experiment is repeated with a tenth of the noise standard deviation of the given points, thus $\sigma_X = 0.01$, which represents a relative accuracy of 1 permille, the differences practically vanish when restricting the comparison to four valid digits. ◇

10.7 Exercises

Basics

1. (1) Given are two uncertain rotations $\{\mathbb{E}(\underline{R}_i), \boldsymbol{\Sigma}_{\Delta r_i \Delta r_i}\}$. Show that the concatenation $R = R_2 R_1$ leads to $\mathbb{E}(\underline{R}_2\underline{R}_1, R_2\boldsymbol{\Sigma}_{\Delta r_1 \Delta r_1} R_2^\mathsf{T} + \boldsymbol{\Sigma}_{\Delta r_2 \Delta r_2})$.
2. (1) Given are two uncertain motions $\{\mathbb{E}(\underline{\mathsf{M}}_i), \boldsymbol{\Sigma}_{\Delta\xi_i \Delta\xi_i}\}$ where $\xi_i = [\Delta\boldsymbol{r}^\mathsf{T}, \Delta\boldsymbol{T}^\mathsf{T}]^\mathsf{T}$. Show that the concatenation $\mathsf{M} = \mathsf{M}_2\mathsf{M}_1$ leads to $\mathbb{E}(\underline{\mathsf{M}}_2\underline{\mathsf{M}}_1, \mathsf{M}_{\Delta\xi_2}\boldsymbol{\Sigma}_{\Delta\xi_1\Delta\xi_1}\mathsf{M}_{\Delta\xi_2}^\mathsf{T} + \boldsymbol{\Sigma}_{\Delta\xi_2\Delta\xi_2})$, where $\mathsf{M}_{\Delta\xi_2} = \mathsf{M}_L$ from (6.54), p. 259.
3. (3) An algorithm provides you with three rotation angles, α, β and γ, together with their standard deviations, σ_α, σ_β, and σ_γ, which are used to determine the complete rotation matrix $R = R_3(\gamma)R_2(\beta)R_1(\alpha)$. Your own software only can handle uncertain quaternions.

   You want to derive the quaternion representation $\mathbf{q}$ for the complete rotation $R_q(\mathbf{q}) := R$ together with its covariance matrix.

a. Is the information given by the algorithm sufficient to derive the covariance matrix of the quaternion?
b. Derive an expression for the quaternion $\mathbf{q}$.
c. Derive an expression for the covariance matrix of the quaternion $\mathbf{q}$.

Hint: start with the case that β and γ are 0 with variance 0.

4. (3) Give an explicit expression for the uncertainty $\Sigma_{x'x'}$ of a rotated 2D point $\boldsymbol{x}' = R\boldsymbol{x}$ if both the point $\boldsymbol{x}$ and the rotation matrix R are uncertain. Assume the uncertain 2D point is given by $\{\boldsymbol{x}, \Sigma_{xx}\}$.

   a. Let the uncertain rotation $R(\alpha)$ be given with $\{\alpha, \sigma_\alpha^2\}$.
   b. Let the rotation matrix be given by the homogeneous 2-vector $\mathbf{a} = [a, b]^\mathsf{T}$:

$$R(\mathbf{a}) = \frac{1}{\sqrt{a^2 + b^2}} \begin{bmatrix} a & -b \\ b & a \end{bmatrix}; \tag{10.334}$$

   thus assume $\{\mathbf{a}, \Sigma_{\mathrm{aa}}\}$ is given.

   Compare the expressions $R(\alpha)$ and $R(\mathbf{a})$ concerning simplicity .

5. (2) Derive the covariance matrix $\Sigma_{\mathrm{L}^e\mathrm{L}^e}$ (see (10.74), p. 380) for the 3D line passing through the point $\mathcal{X}$ in the XY-plane and having the direction $\boldsymbol{L}_h$ with latitude ϕ and longitude λ, measured by a clinometer:

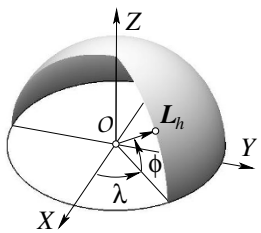


Fig. 10.30 Geographic coordinates for specifying the direction $\boldsymbol{L}_h$ of a 3D line

$$D\left(\begin{bmatrix} X \\ Y \end{bmatrix}\right) = \begin{bmatrix} 0.25 & 0.1 \\ 0.1 & 0.25 \end{bmatrix} [\mathrm{m}^2] \qquad D\left(\begin{bmatrix} \phi \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [(^\circ)^2]; \tag{10.335}$$

see Fig. 10.30.

Show that the null space of the covariance matrix is given by (10.70), p. 380.

6. (3) Show that the envelope of the lines, which are represented by the points of the standard ellipse of the line parameters ϕ and d, is a hyperbola with centre $\boldsymbol{x}_0$, opening angle $2\sigma_\phi$ and width of the waist $2\sigma_q$, cf. 10.2.2.3, p. 373.
7. (1) Refer to Sect. 10.3.1.1, p. 386 and derive explicit algebraic expressions for the Jacobian of the parameters $\mathbf{A}$ of a plane $\mathcal{A}$ passing through three points $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, i.e., $\partial\mathbf{A}/\partial\boldsymbol{t}$ with $\boldsymbol{t}^\mathsf{T} = [\mathbf{X}^\mathsf{T}, \mathbf{Y}^\mathsf{T}, \mathbf{Z}^\mathsf{T}]$.
8. (2) Derive tests for the constraints discussed in Sects. 7.1.2, p. 295, ff and 7.2.2, p. 304, ff which are not contained in the two Tables 10.3 and 10.4, p. 395, ff, especially:

   a. (1) oriented parallelity of two planes.
   b. (1) antiparallelity of two planes.
   c. (1) perpendicularity of two lines $\mathcal{l}$ and $\mathcal{m}$ with a left turn from $\mathcal{l}$ to $\mathcal{m}$.
   d. oriented orthogonality of a 3D line and a plane, indicating the direction of the line points in the direction of the normal of the plane.
   e. (2) collinearity of three 2D points.
   f. (2) concurrence of three 2D lines.
   g. (2) coplanarity of four 3D points.

h. (2) concurrence of four planes.

Assume the given entities are stochastically independent.

9. (1) Derive a statistical test on the identity of two homographies with given covariance matrices for their parameters.
10. (1) Derive a statistical test on the identity of two uncertain rotations $\mathcal{R}(\mathbf{p})$ and $\mathcal{R}(\mathbf{q})$ with given covariance matrices.
11. (2) Derive a statistical test on the identity of two uncertain 2D motions $\mathcal{M}(\alpha, \boldsymbol{s})$ and $\mathcal{M}(\beta, \boldsymbol{t})$, assuming the 2D rotation to be represented with angles.
12. (2) Derive a statistical test on the identity of two uncertain 3D motions $\mathcal{M}(\mathbf{p}, \boldsymbol{S})$ and $\mathcal{M}(\mathbf{q}, \boldsymbol{T})$.
13. (2) In a digital image you can assume that in a first approximation the pixels on an edge segment of length L are equally spaced with spacing Δs and the positions of the individual edge pixels have the same precision. Then the line passing through these edge pixels can be derived using the following approximate model: Given a set of I equally spaced points $\mathbf{x}_i$ on a straight line with covariance matrix $\sigma^2 \, I_2$.

    a. Show that the theoretical variance σ_q of the position of the line and the variance σ_ϕ^2 of the direction are given by (10.170), p. 400.
    b. Show that the two coordinates q_1 and q_2 of the end points of the line segment at distance $\pm I \, \Delta s/2$ across the line segment for large I have standard deviation and correlation (10.171), p. 400.
    c. What approximations are contained in the model?

14. (2) Derive the mutual transformations between the Hessian normal form of an uncertain plane and the centroid representation, see Sect. 10.2.2.4, p. 377. Specifically, show that given a plane in centroid representation the covariance matrix of the Euclideanly normalized plane vector $\mathbf{A}^e$ is

$$\Sigma_{\mathrm{A}^e\mathrm{A}^e} = \begin{bmatrix} \sigma_\alpha^2 \boldsymbol{r}_1 \boldsymbol{r}_1^\mathsf{T} + \sigma_\beta^2 \boldsymbol{r}_2 \boldsymbol{r}_2^\mathsf{T} & -\sigma_\alpha^2 \boldsymbol{r}_1 \boldsymbol{r}_1^\mathsf{T} \boldsymbol{X}_0 - \sigma_\beta^2 \boldsymbol{r}_2 \boldsymbol{r}_2^\mathsf{T} \boldsymbol{X}_0 \\ -\sigma_\alpha^2 \boldsymbol{X}_0^\mathsf{T} \boldsymbol{r}_1 \boldsymbol{r}_1^\mathsf{T} - \sigma_\beta^2 \boldsymbol{X}_0^\mathsf{T} \boldsymbol{r}_2 \boldsymbol{r}_2^\mathsf{T} & \sigma_\alpha^2 \boldsymbol{X}_0^\mathsf{T} \boldsymbol{r}_1 \boldsymbol{r}_1^\mathsf{T} \boldsymbol{X}_0 + \sigma_\beta^2 \boldsymbol{X}_0^\mathsf{T} \boldsymbol{r}_2 \boldsymbol{r}_2^\mathsf{T} \boldsymbol{X}_0 + \sigma_q^2 \end{bmatrix} . \tag{10.336}$$

    Show that if $\sigma_\alpha = \sigma_\beta$, this simplifies to

$$\Sigma_{\mathrm{A}^e\mathrm{A}^e} = \begin{bmatrix} \sigma_\alpha^2 P & -\sigma_\alpha^2 P \boldsymbol{X}_0 \\ -\sigma_\alpha^2 \boldsymbol{X}_0^\mathsf{T} P & \sigma_\alpha^2 \boldsymbol{X}_0^\mathsf{T} P \boldsymbol{X}_0 + \sigma_q^2 \end{bmatrix} , \tag{10.337}$$

    with the projection matrix $P = \boldsymbol{r}_1 \boldsymbol{r}_1^\mathsf{T} + \boldsymbol{r}_2 \boldsymbol{r}_2^\mathsf{T} = I_3 - \boldsymbol{N} \boldsymbol{N}^\mathsf{T}$.
15. (2) Given a spatial rectangular region with sides a and b, the task is to recover a plane from three uncertain points within the rectangle such that it has best accuracy, namely that the variance of the normal direction is smallest. Where should these three points be placed?
16. (2) Assume I points with common weight $w = 1$ are distributed uniformly in a spatial planar disk of radius R. Derive an algebraic expression for the expected variances σ_q^2, σ_ϕ^2 and σ_ψ^2 of the centralized plane parameters. Hint: Show the expected moments are $\lambda_1 = \lambda_2 = \frac{\pi}{4} \, I \, R^4$.
17. (2) When estimating a 3D line from points or planes using an algebraic minimization, generally the Plücker constraint is not enforced, see (10.235), p. 412.
    Show that the matrix $B = [\{\overline{\mathsf{TT}}(\mathbf{X}_i)\}]$, when specializing (10.235), p. 412, to 3D points, only has right singular vectors which fulfil the Plücker constraint.
    Hint: Show that $B^\mathsf{T} B$ has the structure

$$B^\mathsf{T} B = \begin{bmatrix} a I_3 - G & S(\boldsymbol{b}) \\ -S(\boldsymbol{b}) & c I_3 + G \end{bmatrix} \tag{10.338}$$

with arbitrary scalars $a > 0$ and $c > 0$, 3-vector $\boldsymbol{b}$ and symmetric positive semi-definite matrix G. Using this result show that if $\mathbf{L}$ is an eigenvector belonging to λ_1 then also $D_6\mathbf{L}$ is an eigenvector to some other eigenvalue λ_2; thus, the vector $\mathbf{L}$ fulfills the Plücker constraint. What relation do λ_1 and λ_2 have?

18. (1) The covariance of $\underline{R} = [\underline{\boldsymbol{c}}_1, \underline{\boldsymbol{c}}_2, \underline{\boldsymbol{c}}_3]$ can be specified by $\mathrm{vec}\underline{R} = [\underline{\boldsymbol{c}}_1^\mathsf{T}, \underline{\boldsymbol{c}}_2^\mathsf{T}, \underline{\boldsymbol{c}}_3^\mathsf{T}]^\mathsf{T}$ using the columns of R. Derive Σ_{rr} from $\Sigma_{\Delta r \Delta r}$ in (10.85) and show

$$\mathrm{vec}\underline{R} \approx \mathrm{vec}(\mathbb{E}(R)) + \begin{bmatrix} S^\mathsf{T}(\boldsymbol{\mu}_{c_1}) \\ S^\mathsf{T}(\boldsymbol{\mu}_{c_2}) \\ S^\mathsf{T}(\boldsymbol{\mu}_{c_3}) \end{bmatrix} \underline{\Delta \boldsymbol{r}}\,, \tag{10.339}$$

which allows us to derive the rank 3 covariance matrix of $\mathrm{vec}\underline{R}$ if a good estimate of the mean rotation is available.

19. (2) Set up the design matrix and the covariance matrix for the Gauss-Markov model of the similarity transformation (10.215), p. 409 and show that the normal equation matrix is sparse and the three parameters $\widehat{\lambda}$, $\widehat{R}$ and $\widehat{\boldsymbol{Z}}$ are uncorrelated.
20. (2) Use (10.217), p. 409 and derive the covariance matrix of all seven parameters $\widehat{\lambda}$, $\widehat{R}$ and $\widehat{\boldsymbol{T}}$.
21. (1) Using key point pairs $(\boldsymbol{x}, \boldsymbol{x}')$ for homography determination may also exploit methods for measuring the parallaxes $\boldsymbol{p} = \boldsymbol{x}' - \boldsymbol{x}$. Assume a key point detector yields points $\boldsymbol{x}$ in one image with a covariance matrix Σ_{xx}. After the corresponding points in the other image are found the parallaxes $\boldsymbol{p} = \boldsymbol{x}' - \boldsymbol{x}$ are determined and used to yield the coordinates $\boldsymbol{x}'$. Let the parallax be determined with an accuracy represented by the covariance matrix Σ_{dd}.

    a. Determine the covariance matrix of the combined vector $\boldsymbol{y}^\mathsf{T} = [\boldsymbol{x}^\mathsf{T}, \boldsymbol{x}'^\mathsf{T}]$.
    b. Assume the key point detector yields points with a standard deviation of $\sigma_x = 0.3$ pixel in both coordinates, and the parallax determination is achieved with a standard deviation of $\sigma_d = 0.15$ pixel. What standard deviation do the coordinates of $\boldsymbol{x}'$ have? What is the correlation between the coordinates $\boldsymbol{x}$ and $\boldsymbol{x}'$?
    c. Which of the standard deviations will mainly influence the determination of a homography between the two images? How large is the expected gain, measured in ratio of standard deviations, if you use the measured parallaxes instead of just using the independently detected points in both images.

22. (2) The setup of the estimation of 3D similarities for point pairs assumes that the points are not at infinity.

    a. If you only want to estimate motions, can you also handle points at infinity? What do you need to change?
    b. Does your modification transfer to the estimation of similarities?

23. (2) Show that the Jacobians of the angle α between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are

$$\frac{\delta\alpha}{\delta \boldsymbol{a}} = -\frac{\boldsymbol{a}^\mathsf{T}(\boldsymbol{a}\boldsymbol{b}^\mathsf{T} - \boldsymbol{b}\boldsymbol{a}^\mathsf{T})}{|\boldsymbol{a}|^2\,|\boldsymbol{a} \wedge \boldsymbol{b}|}\,, \qquad \frac{\delta\alpha}{\delta \boldsymbol{b}} = \frac{\boldsymbol{b}^\mathsf{T}(\boldsymbol{a}\boldsymbol{b}^\mathsf{T} - \boldsymbol{b}\boldsymbol{a}^\mathsf{T})}{|\boldsymbol{b}|^2\,|\boldsymbol{a} \wedge \boldsymbol{b}|}\,. \tag{10.340}$$

Proofs and Problems

24. (1) Prove (10.18), p. 368.
25. (1) Let two uncertain rotations $\mathcal{R}_p$ and $\mathcal{R}_q$ be given by their uncertain quaternions, $\{\mathbf{p}, \Sigma_{\mathrm{pp}}\}$ and $\{\mathbf{q}, \Sigma_{\mathrm{qq}}\}$. Give an explicit expression for the uncertainty of the concatenations (see Table 6.4, p. 262): $R_A = R_q R_p$, $R_B = R_p R_q$, $R_C = R_p^\mathsf{T} R_q^\mathsf{T}$, and $R_D = R_q^\mathsf{T} R_p^\mathsf{T}$.

Computer Experiments

26. (2) Generate a set of N random 3D points $\boldsymbol{x}_n, n = 1, ..., N$, with $x_n \in [-1, 1]$ and a random rotation matrix R. Determine the rotated points $\boldsymbol{x}'_n = \mathsf{R}\boldsymbol{x}_n$. Add random noise to $\boldsymbol{x}'_n, n = 1, ..., N$, with a standard deviation $\sigma = 0.01$.
    a. Find the best rotation based on the model $\mathbb{E}(\boldsymbol{x}'_n) = \mathsf{R}\boldsymbol{x}_n$ using (10.202), assuming that $\boldsymbol{x}_n$ are fixed values.
    b. Take random triplets of points and determine the rotation matrix, using (8.74) and (8.77), and compare it with the results from (a).
27. (2) Let an uncertain 3D point $\{\boldsymbol{X}, \Sigma_{XX}\}$ and an uncertain rotation $\{\mathbf{q}, \Sigma_{\mathrm{qq}}\}$ be given. Give an algorithm for the uncertainty $\Sigma_{X'X'}$ of the rotated point $\boldsymbol{X}' = \mathsf{R}\boldsymbol{X}$.
28. (3) Given are I uncertain 3D points $\mathcal{X}_i, i = 1, ..., I$, with $\{\boldsymbol{X}_i, \sigma_i^2 \mathsf{I}_3\}$.
    a. Show that the best fitting plane $\mathcal{A}(\mathbf{A})$ passes through the weighted centroid $\boldsymbol{X}_0$, that its normal $\boldsymbol{A}_h$ is the eigenvector of the moment matrix belonging to the smallest eigenvalue, and that it is given by $\boldsymbol{A}_h^{\mathsf{T}}(\boldsymbol{X} - \boldsymbol{X}_0) = 0$.
    b. Show the *theoretical variances* of the parameters of a plane through I equally weighted ($w_i = 1$) 3D points $\mathcal{X}_i$ with standard deviation σ for all coordinates can be determined from
    $$\sigma_q^2 = \frac{\sigma^2}{I} \qquad \sigma_\phi^2 = \frac{\sigma^2}{\lambda_1} \qquad \sigma_\psi^2 = \frac{\sigma^2}{\lambda_2}, \tag{10.341}$$
    where σ_q^2 is the variance of the position of the plane in the direction of the normal and σ_ϕ^2 and σ_ψ^2 are the variances of rotations around the two principle axes of the point set.
    Hint: Translate the point cloud into the origin and rotate it such that the two major axes of the moment matrix fall into the X- and the Y-coordinate axes. Then apply the reasoning from the chapter on the best fitting 2D line.
    c. Show that the *estimated variance* of the plane's position q perpendicular to the plane and the two principal normal directions are given by
    $$\sigma_q^2 = \frac{1}{I-3}\frac{\lambda_3}{I} \qquad \sigma_\phi^2 = \frac{1}{I-3}\frac{\lambda_3}{\lambda_1} \qquad \sigma_\psi^2 = \frac{1}{I-3}\frac{\lambda_3}{\lambda_2}. \tag{10.342}$$
    d. Derive the covariance matrix of the homogeneous vector $\mathbf{A}$ of the plane. *Hint:* use the rotation matrix R spanned by the three eigenvectors of the moment matrix.
29. (2) Assume you have two point clouds in 3D whose relative motion is unknown. Using some segmentation and matching procedure you are able to derive two sets of corresponding planes $(\mathcal{A}, \mathcal{A}')_i$ with their parameters and covariance matrix.
    a. Derive the Jacobian J_M for the linearized model of the mutual motion, $\mathsf{M}^{\mathsf{T}}\mathbf{A}'_i = \mathbf{A}$.
    b. Write a computer program to optimally estimate the motion parameters.
    c. Assume you have a program for estimating the motion for a given set of corresponding 3D points with arbitrary covariance matrix. Could you use it for determining the motion for two sets of corresponding planes? Why? Under what conditions would you be able to use it without changing the program?

Part III
Orientation and Reconstruction

This part provides the tools for camera orientation and geometric scene reconstruction. We focus on Euclidean scene reconstruction and on statistically rigorous methods for estimation and evaluation using the basic tools provided in Parts I and II.

The scene model consists of a set of geometric features, possibly surfaces. Cameras are assumed to follow a central projection, leaving cameras with rolling shutter aside. Images therefore also are assumed to be a set of geometric features. The analysis pipeline is well-structured: Starting from camera calibration, we first determine the camera's orientation, in the most general case employing what is called bundle adjustment providing a statistically optimal estimate of all camera poses and scene features, which then may be densified with some technique for surface reconstruction. In all cases the representation of the resulting geometric image interpretation is a set of parameters, mostly camera poses and scene features together with covariance matrices and other measures for evaluating the quality of the result.

The separation of the pipeline into a sequence of smaller components has positive side effects: The necessary separation of the entire orientation task into the orientation of several smaller sets of images and the joint estimation of all views using bundle adjustment (1) allows more efficient outlier detection, (2) allows for using closed form solutions for determining approximate values, (3) provides simplified procedures and techniques for 3D reconstruction (inverse perspective, binocular/trinocular stereo), and (4) efficiently provides a sparse surface representation and camera self-calibration. Dense surface reconstruction here is focused on determining $2^1/_2$D surfaces from a point cloud.

For didactical reasons we start with the geometry of the single image, the image pair and the image triple, assuming the cameras to be at least partially calibrated. The camera calibration is treated in the context of self-calibrating bundle adjustment, as this is the technique which is pretty standard and most efficient.

In all steps we perform uncertainty reasoning: We therefore can track the uncertainty from the original image features to the orientation and the reconstructed scene. Due to low signal to noise ratios and well-understood models, we obtain close to optimal solutions in practice.

Chapter 11
Overview

11.1 Scene, Camera, and Image Models 441
11.2 The Setup of Orientation, Calibration, and Reconstruction 449
11.3 Exercises 453

This chapter gives an overview of the specific models required for orientation and reconstruction based on images of a scene. It first addresses geometric models for the scene, the cameras, and the images as the result of the projection process. Geometric image analysis tasks such as camera calibration, camera pose estimation and scene reconstruction can exploit a joint model. It is the key to what is known as *self-calibrating bundle adjustment*, which yields statistically optimal parameters for calibration, orientation, and scene features. Small sets of images allow us to solve special tasks, such as the prediction of geometric loci from given image features, the determination of approximate values of parameters using direct solutions, and outlier detection. Depending on the specific context, such as whether cameras are calibrated or not, or whether we have only points or also line segments, we arrive at specific solutions. In all cases we provide the means to evaluate the quality of the resultant parameters in a statistically rigorous fashion. The integration of image analysis procedures into orientation and reconstruction procedures is the topic of the second volume of the book.

11.1 Scene, Camera, and Image Models

11.1.1 Modelling the Image Acquisition and Analysis Process 441
11.1.2 Geometric Scene Models 442
11.1.3 Geometric Camera Models 443
11.1.4 Geometric Image Models 447
11.1.5 Models for Geometric Image Analysis and Interpretation 448

11.1.1 Modelling the Image Acquisition and Analysis Process

A meta model for image analysis has been discussed in the introduction. Here we specify its components for geometric image analysis, assuming scene and image entities can be sufficiently well-described using their geometric properties.

The scene model, which is assumed to be composed of geometric objects, and the geometric model for the sensor leads to the geometric image model. The analysis model essentially consists of statistical parameter estimation. The interpretation therefore is a set of parameters describing the envisaged geometric aspects of the scene.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_11

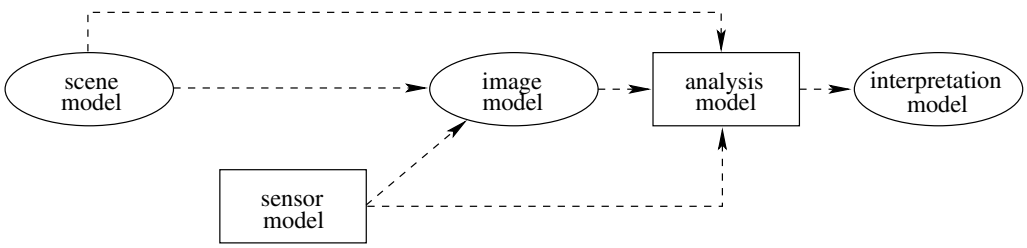


Fig. 11.1 Meta model showing the dependencies between models within image interpretation, see Fig. 1.6. Adapted from Förstner (1993)

11.1.2 Geometric Scene Models

Depending on the viewer's goal, a scene may be described in many different ways.

Take, for example, a physicist: He or she may describe the scene as a function in space by specifying the material and its properties. The surface of an object, say a building, is assumed to be the transition between solid material and air. The boundary between two surface parts, say two parts of the roof, is regarded as the transition of some possibly purely geometric surface property, e.g., the normal. This view can be seen as a *field-based description* of the scene, since the space and possibly also time coordinates are key to addressing the properties of the scene. The notion transfers to semantic raster maps, e.g., when providing the land cover derived from a remote sensing image.

field-based scene description

Aggregating material with the same properties, e.g., material density or velocity, leads to a complete partition of the scene into a set of regions whose spatial extent may be described by the form of their boundary and possibly their common velocity and whose spatial relations may be used to describe the interaction with other regions. Reasoning about these regions and their relations is simplified by categorizing them and giving them names. This can be seen as an *object-based description*, as the objects are keys to addressing the scene's content. This view at the same time may purposely neglect the physical properties inside the regions or objects.

object-based scene description

In the following, we restrict the discussion to scenes which can be represented as

- a set of 3D points and a set of straight 3D line segments (an object-based description) when performing orientation tasks and
- a set of smooth regions or piecewise smooth surfaces (a field-based description) when performing surface reconstruction.

Thus the scene description may be very sparse compared to human perception of it.

A set of *3D points* is the basic scene model used in bundle adjustment. It aims at simultaneously recovering the scene and the cameras, in the computer vision community called structure and motion. If the 3D points are representative of a smooth surface, a set of surface elements, called *surfels*, consisting of a 3D point and the corresponding surface normal, may also be useful.

Man-made objects often are represented as a *polyhedron*, consisting of a set of 3D points, 3D line segments, and planar regions, see Fig. 11.2, left. Due to occlusions, the polyhedron may not be closed or connected. In that case the surface is incompletely described by a set of polyhedral patches. Often only a set of one type, say planar regions, is used to describe the scene.

Generally the boundaries of real objects may be represented as piecewise smooth surfaces with smooth regions having piecewise smooth boundaries. As real surfaces are rough, possibly including fractional parts, we assume the representation of the surface relates to a certain scale, e.g., to the resolution of a mesh, where surface details smaller than the mesh are not relevant and therefore smoothed away by the observation and reconstruction processes (cf. the discussion by Koenderink, 1990). This includes polyhedra as special cases. However, it allows the boundary between two smooth regions to vary between sharp and smooth, see Fig. 11.2, right.

boundary model

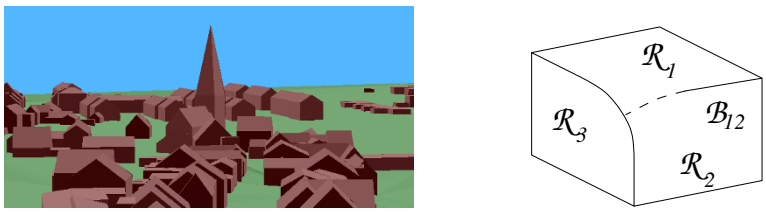


Fig. 11.2 **Left:** City model represented as polyhedron. **Right:** Surface with three regions, region $\mathcal{R}_1$ and region $\mathcal{R}_2$ are separated by boundary $\mathcal{B}_{12}$, which is only partly sharp

11.1.3 Geometric Camera Models

Cameras map the 3D space to a set of light-sensitive sensor elements via some optical system from which a 2D image is derived. We discuss the various forms of cameras to motivate our focus.

11.1.3.1 Cameras

Figure 11.3 shows cameras with quite different optical systems. They range from consumer cameras with a wide angle lens, via stereo cameras, multi-camera systems, and cameras with mirrors, to cameras which contain an array or a line of light-sensitive sensor elements.

Fig. 11.3 Cameras with different viewing geometry, approximate diameter and weight. **First row, 1:** consumer camera with central shutter (approximately 12 cm, 25 g); **2:** stereo camera with two optical systems (14 cm, 300 g); **3:** high-resolution multi-spectral camera with eight optical systems (80 cm, 65 kg); **4:** omnidirectional camera with six optical systems (20 cm, 2.5 kg). **Second row, 5:** catadioptric panorama camera with a single lens system and a parabolic mirror (25 cm, 1 kg); **6:** panorama camera with rotating line sensor the rotation axis passing through the centre of the lens (40 cm, 5 kg); **7:** smart phone camera with rolling shutter (1 cm, 10 g), **8:** viewing planes (forward, down, backward) of high-resolution multi-spectral three line sensor camera (80 cm, 150 kg)

Classical cameras with photographic film map the 3D scene to an image plane via a perspective mapping in a first approximation, since they are refined pinhole cameras. The light-sensitive film physically carries the image presented to the human user. The possibility of exploiting the digital image information and the flexibility of modern optics leads

computational camera

to the general concept of a *computational camera*, see Fig. 11.4, p. 444. A computational

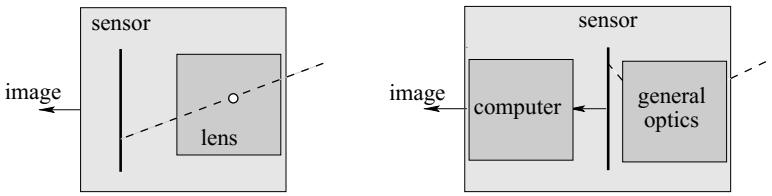


Fig. 11.4 Principle of a traditional and a computational camera following Nayar (2006). **Left**: The sensor of the traditional camera immediately gives the perceivable image. **Right**: The image of a computational camera is derived from the raw sensor data by geometric and radiometric transformations performed by the camera's internal computer. Examples are given in Fig. 11.5

camera, in addition to the optics and a digital sensor, contains a computer to transform the captured raw data into an image which can be perceived by a human or further processed. Such a transformation can refer to radiometry or geometry when processing the raw sensor data arranged in the classical *Bayer pattern*, where each pixel has only one of three colours, to achieve a colour image, where each pixel is represented by three colours, or when processing a *fish-eye* image to achieve an undistorted perspective image, see Fig. 11.5, or a panorama.

Bayer pattern

Fig. 11.5 Radiometric and geometric operations in a computational camera. **Left:** The light-sensitive sensor elements in a normal digital camera are sensitive to different colours according to the Bayer pattern (first row: green/blue, second row: red/green); the three colour values of each pixel are determined by interpolation, best viewed in colour. **Middle and right:** Rectification of a fish-eye image, from Abraham and Förstner (2005)

The cameras can be distinguished by the arrangement of their pixels (see Fig. 11.6).

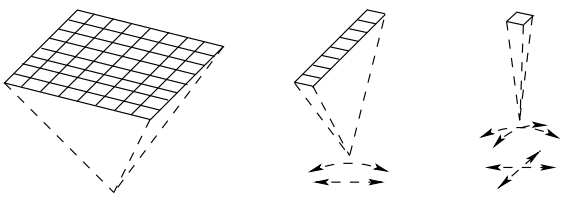


Fig. 11.6 Cameras with frame, line and point sensor. In order to achieve a 2D image, line and point cameras need to be rotated or moved

frame cameras

- In *frame cameras* the pixels are arranged in one or several rectangular grids. Examples of cameras with two or more such area sensors are stereo cameras or omnidirectional camera systems, such as the Ladybug 3 with six single view video cameras, or the Ultracam System of Vexcel, integrating four panchromatic cameras for achieving higher resolution and four monochrome colour cameras, namely blue, green, red, and infra-red (see Fig. 11.3, camera 3).

line cameras

- A *line camera* can be interpreted as an area camera where the sensor array is de-

generated to a line of sensor elements. Such a line camera produces a 2D image by moving it across the line sensor. One coordinate in that 2D image refers to the position on the line sensor, the other to the time the line has been illuminated. A similar situation occurs in cameras with rolling shutters. Though they have a 2D array of light-sensitive pixels, the image is generated line by line over the complete exposure time, not simultaneously.

Examples are the panorama camera of Rollei producing one 2D image, smart phone cameras and the three line camera system ADS of Leica, which consists of three individual line cameras with individual optics producing three 2D images.

- In *point cameras* we only have a single or several isolated sensor elements. Images of this type are common for satellite scanners, but also are generated by airborne laser scanners if the intensity data of the reflecting laser pulse are collected into a 2D image. *point cameras*

A camera system with multiple cameras arranged in order to capture a large field of view is called a *polycamera* (cf. Swaminathan and Nayar, 2000).

11.1.3.2 Camera Models

A camera model is an abstraction of the real camera sufficiently simplified for solving a task.[1] In our context we aim at modelling the geometry of the relation between positions of a set of image points in the sensor area and the corresponding bundle of viewing rays.

Specifically, we assume a unique relation between a picture element at position $\mathbf{x}'$ and all points $\mathbf{X}$ in 3D mapped to this pixel which lie on a straight line, the projection ray $\mathcal{L}_{x'}$ if we neglect atmospheric effects. The set $\{\mathcal{L}_{x'}\}$ of all projection rays in most cameras is structured differently for each optical system used, see Fig. 11.7. The set of projection

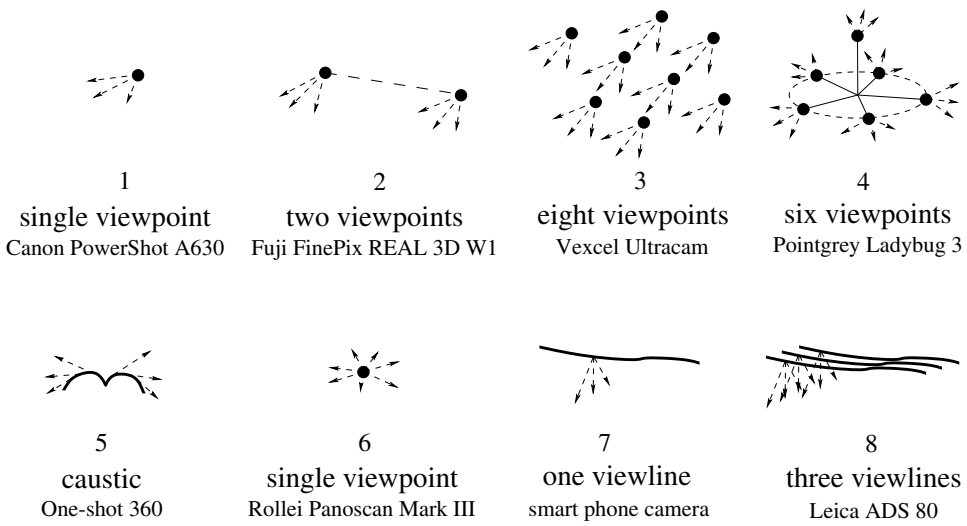


Fig. 11.7 Viewpoints for the cameras in Fig. 11.3. The smart phone camera is assumed to move during exposure

rays of a pinhole camera has a common point. We call this the effective viewpoint or just the *viewpoint* $\mathbf{O}$ of the camera, the letter $\mathbf{O}$ standing for the Latin word '*oculus*', the eye. Classical lens cameras without too much distortion approximately have a single viewpoint. The viewpoint in a first approximation coincides with the centre of the lens. *viewpoint*

[1] As long as this does not cause confusion, we call camera models just cameras in a mathematical sense. They may be used to generate artificial images with the corresponding properties in order to be taken as approximations of real images. Some of the camera models are used only on the computer.

A stereo camera system consisting of two classical cameras obviously has two viewpoints. The omnidirectional camera system Ladybug 3 consists of six standard cameras, and thus has six viewpoints.

catadioptric optics

If the optical system also contains a mirror it is called a *catadioptric optics*. For example, let the camera have a planar mirror in front of the lens; this mirror is treated as part of the optical system. Then all projection rays still may pass through one viewpoint, which in a first approximation is the mirror point of the centre of the lens. Thus in this case the viewpoint and the centre of the lens are distinct points.

caustic

The general case is where most projection rays do not meet but touch a common surface, the *caustic*, such as for the camera 'One-shot 360' or extreme wide angle lens systems. Cameras where the projection rays do not meet in a single viewpoint are called *generic cameras* (cf. Grossberg and Nayar, 2001; Pless, 2003).

generic camera

For line cameras the structure of the set of projection rays depends on the type of motion. We have different characteristics of the set of projection rays:

- If the line sensor is rotated around the spatially fixed centre of the lens, we obtain a 2D image with a single viewpoint, as for the Rollei Panoscan Mark III (cf. Fig. 11.3).
- If the centre of the lens moves along a line, all projection rays pass through this line. One could say we have a viewing line, instead of one or several viewing points.
- In the case of a camera system consisting of several line sensors we obtain one viewing line for each line sensor, as in the three line scanner system ADS 80 of Leica in Fig. 11.3.

In the following we focus on central area cameras or, in short, *central cameras*, i.e., cameras with a single viewpoint, see the classification of camera models in Fig. 11.8. Due

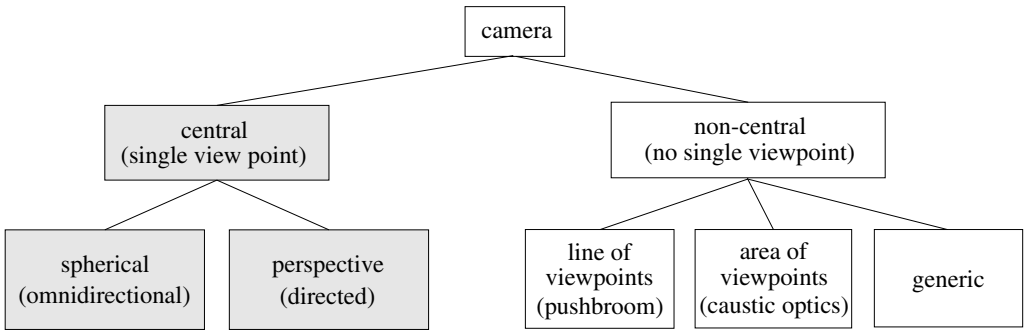


Fig. 11.8 Camera models. We distinguish between central cameras with a single viewpoint and noncentral cameras without a single viewpoint. Central cameras are perspective cameras, thus straight line-preserving with a viewing field less than a hemisphere, or spherical cameras, such as central omnidirectional cameras. We distinguish between three types of noncentral cameras: (1) with a line of viewpoints, such as push-broom cameras; (2) where the envelope of the incoming rays are tangents at a surface, called caustic; and (3) generic cameras otherwise

to their dominant role in applications, the main emphasis is on cameras with a *perspective mapping* onto a sensor plane which is characterized by being straight line-preserving. They

perspective camera

are modelled as *perspective cameras* (cf. Fig. 11.9, left). Their field of view is limited to a proper subset of a hemisphere, similarly to classical cameras, which are not able to observe points behind the camera or points in a direction perpendicular to the viewing direction. Therefore the classical camera model can take the viewing ray as a full line, including points behind the camera, as they do not appear in the image.

Omnidirectional cameras may also have a unique projection centre. In this case the viewing rays need to be treated as *half lines* from the projection centre to the scene point, establishing a *bundle of oriented rays*, see Fig. 11.9, right. Such cameras are modelled as

spherical camera

spherical cameras. In contrast to perspective cameras, their field of view may be significantly larger than a hemisphere (Fig. 11.10). They allow us to exploit the full potential of oriented projective geometry.

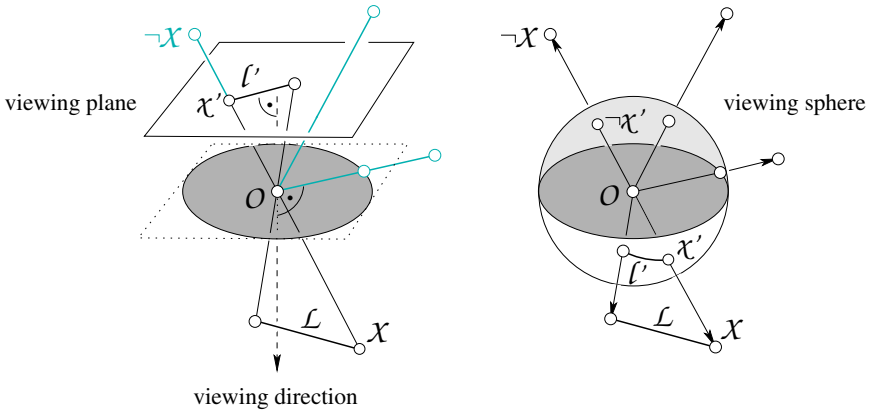


Fig. 11.9 Central camera models. **Left:** *perspective camera model.* Scene points $\mathcal{X}$ are mapped to points $\mathcal{x}'$ in a sensor plane. 3D lines $\mathcal{L}$ are mapped into 2D lines $\mathcal{l}'$. The viewing ray is a 3D line through the projection centre and the scene point. Its intersection with the sensor plane yields the image point $\mathcal{x}'$. Points on or behind the shaded plane, which is parallel to the sensor plane through the projection centre $\mathcal{O}$, are not mapped to the sensor, shown as a rectangle. The model cannot distinguish between $\mathcal{X}$ and its antipodal point $\neg\mathcal{X}$, both sitting on the projection ray and mapping to the same image point $\mathcal{x}'$. **Right:** *spherical camera model.* Scene points are mapped to the unit sphere. 3D lines $\mathcal{L}$ are mapped into great circles $\mathcal{l}'$. The viewing rays are half lines from the projection centre to the scene point. Its intersection with the viewing sphere yields the image point $\mathcal{x}'$. Any scene point, except the projection centre, has an image point. Especially, the antipodal point $\neg\mathcal{X}$ to the point $\mathcal{X}$ has image point $\neg\mathcal{x}'$, distinct from $\mathcal{x}'$. Adapted from (Mičušík, 2004)

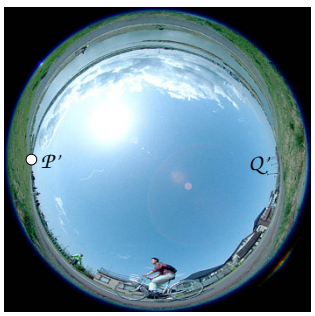


Fig. 11.10 Image taken with a Nikkor Fish-eye lens having a viewing angle of 200°. The image points $\mathcal{P}'$ and $\mathcal{Q}'$ refer to two points on the horizon in opposite directions. Taken from https://hadidankertas101.blogspot.de/2016/02/normal-0-false-false-false-en-us-x-none_16.html, last visited August 28, 2016

In both cases the mappings with real cameras will deviate more or less from the ideal projection, i.e., they may not be straight line-preserving when using a perspective camera model. In all cases an ideal mapping can be achieved by a proper rectification of the observed image.

As we also handle poly-cameras, we easily can generalize to camera systems with multiple central cameras, such as stereo video cameras with two or more viewpoints (see Fig. 11.8). Although we don't present models for catadioptric cameras, the presented methods can be applied if they have a single viewpoint.

11.1.4 Geometric Image Models

Geometric image analysis requires a geometric image model. It specifies the structure of the geometric description of a single image or of multiple images used for solving the task at hand. We give some representative examples:

Fig. 11.11 Image with automatically extracted image features. **Left:** Points. The radius of the circle indicates the image region responsible for the position. **Centre:** Straight line segments. Curved lines are approximated by polygons. **Right:** Regions. The image region is partitioned into homogeneous regions having approximately the same colour

1. Let us assume (1) the scene can be described by a surface consisting of polyhedral, cylindrical or spherical regions, such that the boundaries between regions are either straight lines or circles; (2) the reflectance function is constant within each surface polygon; and (3) the light source is point type. Then the model for an ideal image taken with a straight line-preserving camera is a map consisting of nonoverlapping regions covering the image domain and bounded by straight lines or conics. Whereas edges in 3D must lie on the two neighbouring faces, the boundary between two polygonal regions may belong to one or more of the following classes (cf. Binford, 1981):
   a. it is the image of the boundary of two neighbouring surface regions, or
   b. it is the boundary between a lit and a shadow region on the same 3D surface region, or
   c. it is the image of an occluding 3D edge.
2. Let us assume the scene is describable by sets of mutually parallel 3D lines. Then the model for an ideal image taken with a straight line-preserving camera consists of sets of concurrent 2D lines.
3. Let us assume the scene can be described by a set of 3D points observed by a set of images. The model for a set of images taken with a straight line-preserving camera for each image is a set of 2D points, such that projection rays from the projection centre through the image points intersect in the 3D points. If the camera is not straight line-preserving but has a single viewpoint, the projection rays of corresponding image points also should intersect in a common 3D point, but there is a one-to-one relation between the projection rays and the corresponding (distorted) image points.

symbolic image description

We assume methods are available to convert the image into a *symbolic description* leading to

- a set of 2D points or
- a set of 2D lines or
- a set of possibly open 2D polygons or
- a set of segments of conics.

Examples for such a transition from an iconic to a symbolic image description are given in Fig. 11.11. We assume the feature extraction procedure yields estimates for the parameters of the features and their uncertainty. Examples are given in Sect. 12.2.1, p. 490.

11.1.5 Models for Geometric Image Analysis and Interpretation

Geometric image analysis follows the probabilistic and statistical reasoning discussed in the introductory section on probabilistic and statistical reasoning, p. 9.

We need all techniques for parameter estimation, mainly the Gauss–Markov model but also the Gauss–Helmert model, especially for small image sets. Statistical testing is required for outlier detection, often in its multivariate version. Variance components are useful for determining the noise model of the geometric entities. Direct and indirect variance propagation is used in all chapters. Finally, direct solutions and robust methods are needed to address especially small image sets. In other words, all methods from Parts I and II find their application here, especially Chap. 4, p. 75 on estimation and Chap. 10, p. 359 on uncertain projective geometry.

We address (1) theoretical accuracies of 3D scene points and orientation parameters for one, two, and multiple images useful for view planning, (2) the evaluation of the result's sensitivity w.r.t. parts of parameters, namely calibration parameters and coordinates of scene points, (3) the covariance matrix for minimum solutions necessary to efficiently use RANSAC, and finally (4) the exploitation of the sparsity of matrices within bundle adjustment.

In the following we discuss these general aspects in more detail.

11.2 The Setup of Orientation, Calibration, and Reconstruction

11.2.1 Estimation Tasks 449
11.2.2 Prediction and Testing Tasks 451
11.2.3 Aspects of Orientation and Reconstruction 451

We will now discuss the general aspects of modelling geometric image analysis, especially for camera orientation, camera calibration and scene reconstruction.

general projection model

The basis is a simplified but sufficiently detailed model for the projection of scene features $\mathcal{F}_i$, such as points or lines, into an image leading to image features,

$$f'_{it} = \mathcal{P}_t(\mathcal{F}_i; \mathcal{O}_t, \mathcal{C}_t)\,. \tag{11.1}$$

The projection into the camera at time t is denoted by $\mathcal{P}_t$. It depends on the pose or orientation $\mathcal{O}_t$ of the camera in 3D space, i.e., rotation and translation, and on the internal geometric properties $\mathcal{C}_t$ of the camera at time t as specified in the camera model. The scene is described by a set $\{\mathcal{F}_i\}$ of features, say points or lines, and possibly by a surface. In the computer vision literature the sets $\{\mathcal{F}_i\}$ and $\{\mathcal{O}_t\}$ are called *structure* and *motion*, respectively. If the scene is assumed to be static, as in most situations discussed here, there is conceptually no need to interpret the index t for the camera as a time stamp, but it can be viewed just as a name or number of the image taken with some camera.

We now can easily name the tasks for which we need to develop models.

11.2.1 Estimation Tasks

We have the following parameter estimation tasks:

- *orientation* or *motion from structure*: derive the parameters of the projections $\mathcal{P}_t$, especially $\mathcal{O}_t$, from given correspondences $\{\mathcal{F}_i, f'_{it}\}$. Methods differ for one, two, three, and many images.

  camera orientation, motion from structure
- *calibration*: derive the internal properties $\mathcal{C}_t$ of the camera from given image features f'_{it}, possibly given corresponding scene features $\mathcal{F}_i$.

  camera calibration

  In computer vision, the task of orientation sometimes is called calibration, reflecting the fact that a vision system usually is part of a larger system which needs to be calibrated. We stick to the notions used in photogrammetry, where orientation refers to the pose of the camera w.r.t. the scene coordinate system and calibration refers to the internal structure of the camera or camera system.

scene reconstruction, structure from motion

- *reconstruction* or *structure from motion*: derive the scene's structure represented by the scene features $\mathcal{F}_i$, possibly by a surface, from image features f'_{it} and camera information $\mathcal{P}_t$.

relative orientation of images, photogrammetric model

- *relative orientation*: determine the geometric relation of two images, called their relative orientation, and derive the relative pose of the cameras, say $\mathcal{P}_2\mathcal{P}_1^{-1}$, and a local scene description $\{\mathcal{F}_i\}$ from corresponding image features, say $\{f'_{i1}, f'_{i2}\}$, in two images. The problem can be generalized to the relative pose of many cameras.

  No complete scene reconstruction will be possible as long as no relation to a scene coordinate system is available. However, generally the intrinsic shape of the scene can be recovered up to a global transformation, and serve as a scene description and may be useful in certain applications.

absolute orientation, control information

- *absolute orientation:* derive the transformation between the coordinate system of the local scene description and a reference coordinate system using *control information* (3D features) available in the local and the reference coordinate systems.

bundle adjustment, simultaneous localization and mapping

- *bundle adjustment*: derive both the camera poses, $\mathcal{O}_t$, possibly the internal properties $\mathcal{C}_t$ of the cameras, and the scene structure $\mathcal{F}_i$ from corresponding image features $\{f_{it}\}$ and some control information, i.e., 3D features, cf Fig. 11.12. Some 3D features need to

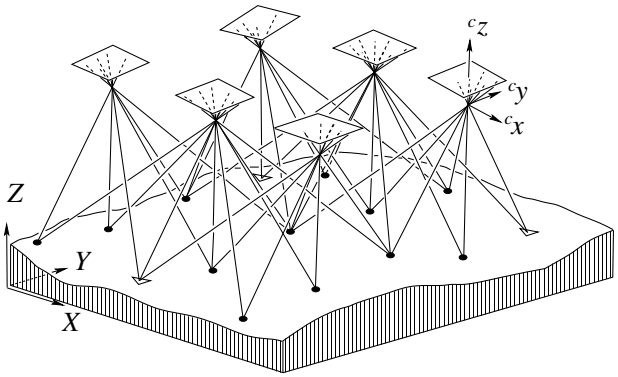


Fig. 11.12 Bundle adjustment: Given the image coordinates of 3D points in many images and the coordinates of some scene points, the task is to recover the poses (rotation, translation) of the cameras during image exposure and the coordinates of the other scene points (after Ackermann et al., 1972, Fig. 3, p. 1637). Control points are shown as triangles

be known to relate the camera poses to the scene coordinate system. In the photogrammetric community they are called *control points* or, more generally, *control features*. They may (in principle) be replaced by some direct measurements of poses, e.g., using GPS if the relation between the GPS coordinate system and the scene coordinate system is known. Bundle adjustment is closely related to *simultaneous localization and mapping* (SLAM) in robotics, where the robot's poses and the spatial structure of the environment are determined from interwoven sensor data, usually in an incremental mode.

control features

Obviously, the task of bundle adjustment generalizes the previous ones. For example, if no 3D features are available, only the relative orientation and a local scene description can be derived. When the internal structure of the cameras is also determined, the task is called self-calibrating bundle adjustment.

self-calibrating bundle adjustment

All tasks have variants where either only partial solutions are required or where additional knowledge is available and allows for simplified methods.

11.2.2 Prediction and Testing Tasks

Prior to or during estimation, methods for outlier detection are required. They rely heavily on geometric constraints and on prediction of geometric positions useful for testing.

We have the following prediction and testing tasks. In our context they refer to one, two or more images and the used image features, namely image points and straight image line segments (see Fig. 11.13), and possibly curved line segments such as conics.

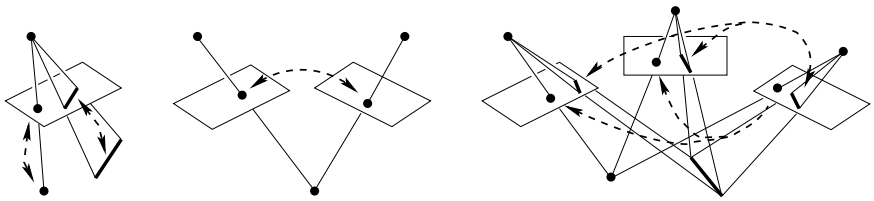


Fig. 11.13 Prediction of geometric entities and constraints between corresponding geometric entities in single images, image pairs, and image triplets. **Left**: Prediction of image points and lines in a single image and reconstruction of projection rays and planes from observed image points and lines. **Middle**: Transfer of points from one image into another and checking of points in two images for correspondence. **Right**: Prediction of points and lines in a third image and checking of points and lines in three images for correspondence

1. We will establish explicit expressions for the *prediction* of points, lines and planes in the images and in the scene.
   These predictions will support measurements and automatic matching processes. In the case of one image, we discuss the prediction, i.e., the projection, of given points and lines in object space into the image, leading to image points and image lines. This will also be the basis for the orientation of single and multiple images.
   We also discuss the prediction, i.e., the back projection, of 3D lines and planes from image points and lines into object space, leading to projection rays and projection planes. This inverse task to projection will later enable us to reconstruct 3D entities from observed image entities.
   In the case of image pairs and image triplets we discuss the predictive determination of image points and possibly lines if they are given in one or two other images.
   Including straight lines as well as points as observed entities is mandatory in today's methods of digital photogrammetry, as lines can be automatically extracted very reliably from digital images.
2. We will establish explicit expressions for the *constraints* between *corresponding* geometric features, such as points or lines in single images, image pairs and image triplets. This will enable us to determine the mutual or *relative orientation* of images without any information about the observed scene. In the case of known correspondences between the images and the scene, it is then possible to determine the *absolute orientation* of the cameras at the time of image capture.

11.2.3 Aspects of Orientation and Reconstruction

The procedures for orienting cameras and reconstructing the geometry of an object depend on the available knowledge about the camera and the object. We always aim at obtaining optimal results in a statistical sense, and at automatic procedures handling random and gross errors. No unified solution for these problems exists. Therefore, various aspects need to be addressed:

1. We want to distinguish between *calibrated* and *uncalibrated* cameras. In all cases we assume the cameras to have a single viewpoint. For modelling and orientation of rolling shutter cameras, cf., e.g., Hedborg et al. (2012), for generic cameras, cf. Chen and Chang (2004); Lee et al. (2013); Ventura et al. (2014); Sweeney et al. (2014).
2. We want to distinguish between solutions which require approximate values and usually are *iterative* and those which are *direct*.
3. Direct solutions with the *minimum number of observations* are needed for gross error detection.
4. Critical configurations need to be known during planning and require means for an a posteriori self-diagnosis.
5. For important cases, we give the quality of parameters for planning purposes.
6. We give solutions not only for points, but also for lines if possible.

In detail we cover the following aspects:

1. We always distinguish between *calibrated* cameras and *uncalibrated* cameras.

The orientation of calibrated cameras generally is more stable, as only six orientation parameters must be determined. The orientation of straight line-preserving uncalibrated cameras often occurs in close range applications, especially when exploiting the zoom capabilities of consumer cameras.

In Sects. 12, p. 455, to 14, p. 621, on one, two or three cameras, we assume that nonlinear distortions are eliminated in a preprocessing step. The distortions are often small and may be neglected in a first step of orientation. If they are known, they can be used to correct the coordinates of the image features at least to such an extent that the image features can be treated as if coming from a perspective or ideal spherical camera.

2. We always distinguish whether *approximate values* for the orientation are available or not.

If we have approximate values, we may directly apply a statistically optimal estimation procedure which exploits the nonlinear relations. Nearly all these estimation procedures are iterative, improving the approximate values in each iteration step. The result in the last step may be evaluated statistically based on the various techniques discussed in Sect. 4.6, p. 115. The determination of approximate values often turns out to be much more difficult than the optimal estimation of the final parameters. In contrast to the optimal estimation, there is no general technique for determining approximate values. Therefore we have to discuss this topic for each situation separately.

In practice, approximate values are often available either from the design of the observation process (from the flight plan for aerial imaging) or from direct measurements (GPS, INS). The accuracy of this information is usually sufficient to initiate an iterative solution. As this information also contains uncertainty, it may be integrated into the orientation process within an optimal estimation procedure in a Bayesian manner.

3. We also give *direct solutions with minimum number of observations*.

They are useful in case no approximate values are available *and* gross errors in the observations are to be expected. A large number of direct solutions have been developed in recent decades, cf. the collection on the page `http://cmp.felk.cvut.cz/minimal`. Random or systematic sampling of the observations may be used to search for a smallest set of observations free from gross errors together with good approximate values for the parameters (cf. Sect. 4.7.7, p. 153). As the algorithmic complexity of the search for good observations increases exponentially with the size of the sample, procedures with a minimum number of observations are useful. We provide information on the precision of these direct solutions in the form of a covariance matrix, useful when performing RANSAC.

4. The solutions, both optimal and suboptimal direct ones, generally fail for certain *critical configurations*.

Close to these configurations the solutions will be unstable, i.e., heavily affected by random perturbations in the given observations. Even in a stable configuration, there may be multiple solutions.

Often these configurations are simple to describe, as when all 3D points are collinear; however, some of them are algebraically complicated. We will mention these configurations; so they can be avoided when designing the configuration; however, we give no proofs. If no a priori information is available about the configuration, we may check the configuration a posteriori. This can be done most easily by investigating the statistical uncertainty of the resultant parameters with respect to an acceptable reference configuration 4.6.2.3, p. 120. An example is given in Sect. 12.2.4.1, p. 516 for the orientation of a single image.

5. For planning purposes we discuss the *quality* of the main orientation and reconstruction procedures: the *precision* indicating the effect of random errors and the *checkability* indicating the ability to identify gross errors.

This type of analysis requires the specification of certain configurations. We refer to the *normal cases* of orientation, namely the vertical view of one and two images and the resulting reconstruction of 3D points from image measurements.

For this purpose we also give *algebraic expressions* for the Jacobians of the nonlinear relations. This gives insight into the geometric structures of the orientation procedures. For checking the algebraic expressions it is recommended to determine the Jacobians by numerical differentiation, i.e., by replacing the differentials by finite differences.

6. *Self-calibrating bundle adjustment* may be used for ego-motion determination, for camera calibration or for (sparse) scene reconstruction. In all cases the other parameter sets can be interpreted either as nuisance parameters, or as parameters for adapting the model to the data, e.g., when performing self-calibration during ego-motion determination. The geometric structure of large sets of images may vary from very regular, especially when view planning can be performed ahead of taking the images, to very irregular, if pre-planning is not possible or the images are collected from the internet. We will discuss both aspects: (1) view planning for capturing certain representative objects completely and (2) evaluating the degree of geometric stability of irregularly positioned image sets.

7. *Surface reconstruction* can be based on a large set of scene features. Mostly points are used, which are determined via intersection from two or more images, assuming image points corresponding to the same scene point are identified. Starting from 3D scene points, the task is to find an optimally interpolating surface. This requires some pre-knowledge about the surface, e.g., about its smoothness. We will discuss surface reconstruction for graph surfaces $z = f(x, y)$ and their accuracy, which is depending on the point distribution.

11.3 Exercises

1. (2) Assume the following scenario: You use the camera of a mobile phone to derive a 3D model of your home consisting of a dense point cloud, each point coloured with the colour of the corresponding image points in the mobile phone. Specify the five models in Fig. (11.1), p. 442 with not more than three sentences or 30 words each. Identify those parts of the models of which you are certain, and those where you need more information. *Hint:* Have a look at the home page of Google's Tango project.
2. (1) What camera class does the camera of your mobile phone belong to? Refer to Fig. (11.8), p. 446.
3. (2) Assume you build the following camera system: It consists of a consumer camera and a mirroring sphere, like a Christmas bauble. The spherical mirror is mounted in front of the camera at a large enough distance. You can expect to obtain images similar to Escher's *Self-Portrait in Spherical Mirror*. What type of camera is this, see Fig. (11.8), p. 446? *Hint:* Follow the classification tree of that figure.
4. (1) Characterize the camera of your mobile phone w.r.t. its being a computational camera, see Fig. (11.4), p. 444. What tasks does the computer in your phone perform between taking and displaying the image? Name at least three of them.

5. (2) Give a practical example for each of the estimation tasks mentioned in Sect. (11.2.1), p. 449. For each task, name the relevant elements in (11.1), p. 449.
6. (2) Under what conditions can you call a consumer camera, such as the Canon PowerShot (see Fig. (11.3), p. 443), a calibrated camera? Name three relevant situations where it definitely cannot be treated as a calibrated camera.
7. (1) Name three distinct geometric tasks for which you know a direct solution. For which of these tasks does the number of observations need to be minimal?
8. (1) Give a critical configuration for determining the similarity between two 3D point sets with more than two points.
9. (1) How many points in general position do you need in order to estimate a conic and to identify one outlier?

Chapter 12
Geometry and Orientation of the Single Image

12.1 Geometry of the Single Image 456
12.2 Orientation of the Single Image 489
12.3 Inverse Perspective and 3D Information from a Single Image 523
12.4 Exercises 537

A single image of a scene is useful in various applications, such as ego-motion determination or partial scene reconstruction. This is due to the strong geometric rules of the imaging process, which, in the simplest case, is a perspective projection in spite of losing one dimension when mapping the 3D scene to a 2D image.

The geometry of the imaging process can be expressed algebraically in compact form using the tools from algebraic projective geometry. The bundle of light rays from the scene through the lens is fixed using the imaging sensor and, depending on the type of optics, may be a bundle of rays in (nearly) all directions, which is the basis for the model of a spherical camera.

As real cameras generally only approximate the perspective or the spherical camera model, we discuss the relation of the sensor w.r.t. the lens, and also models for lens distortion, which make it possible to correct the position of image features such that they obey the perspective or spherical model with sufficient accuracy. This allows us to reconstruct the bundle of viewing rays from the corresponding set of image points.

The bundle of rays can be used to infer the *relative pose* of the camera and the scene at the time of exposure. This relative pose may be used for determining the *pose of the camera* w.r.t. the scene coordinate system, taking control points known in the scene and observed in the image. Depending on the structure of the scene, parameters describing the interior geometry of the camera may also be recovered from a single image. Alternatively, the relative pose may be used to determine the *pose of a scene object* of known form with respect to the camera, e.g., when tracking a 3D object in a video sequence.

When the camera pose is known, the image may be back projected to the scene. In case the scene is planar, it is a straight line-preserving mapping. If the camera's pose is not known, at least partial information about the scene can be recovered if it is well-structured, e.g., as in cities, where the scene can be modelled using planes with three mutually perpendicular directions, which often are called Manhattan or Legoland scenes. Here methods of what is called *inverse perspective* are of great value, which, among other things, exploit the existence of vanishing points and the cross ratio as an invariant of projective mappings.

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_12

12.1 Geometry of the Single Image

12.1.1 Basic Terms About Perspective Images 456
12.1.2 General Aspects on Modelling the Projection 459
12.1.3 Modelling Central Cameras 465
12.1.4 Extending the Perspective Projection Model 476
12.1.5 Overview on the Different Camera Models 479
12.1.6 Mapping of Straight 3D Lines 480
12.1.7 Inverse Relations 482
12.1.8 Mapping of Curved 3D Lines 484
12.1.9 Nonstraight Line-Preserving Mappings 484

To exploit the image content for reconstructing 3D objects using a computer, we need a mathematical model which approximates the physical reality of the projection to a sufficient extent. In the following we reduce it to physical models of geometric optics.

This section describes the geometric model of the projection of points and lines in 3D into the image generated by a real camera. This has to take into account (1) the pose of the camera, i.e., the spatial position and *attitude of the camera* during exposure, (2) the projection through the optics, (3) the effects of lens distortion, and possibly (4) the effects of refraction, especially for large distances between camera and scene.

This geometric model allows us to invert the projection process by inferring the spatial direction to 3D points and lines from their observed images. We use such directions to determine the spatial position of the camera and the 3D position of the observed points and lines. We thus will be able to reconstruct all visible details of the scene (cf. Sect. 12.2, p. 489), especially on planes or other surfaces, as long as they are identifiable in images.

We focus on *central cameras*, i.e., cameras having a single viewpoint, thus following a central projection model. Central cameras are either *spherical cameras* or *perspective cameras* (cf. Sect. 11.1.3.2). Perspective cameras have a planar sensor and are straight line-preserving. Spherical cameras use specific optics and are able to look in (almost) all directions. Real cameras will deviate from these models which can be taken into account during modelling. Though the light-sensitive area generally is planar, the camera can be imagined as having a spherical sensor, so that straight lines in 3D are mapped to great circles.

central, spherical and perspective cameras

After collecting basic terms about perspective, i.e., straight line-preserving images, we describe cameras following the central projection model and make the projection of points, straight lines, and quadrics explicit. We then model deviations from this model, addressing real cameras with imperfect lenses. Generalization to spherical cameras, i.e., cameras with a single viewpoint and a large field of view, is straightforward. We close with a discussion on cameras having more than a single viewpoint, such as polycameras or line cameras.

12.1.1 Basic Terms About Perspective Images

We usually distinguish between different types of images, depending on the orientation or pose of the camera in space:

- *nadir view:* The image plane is strictly horizontal and the viewing direction is in the nadir direction. Taking an image in the opposite direction yields a *zenith view*.
- *vertical view:* The image plane is approximately horizontal, typical for aerial images.
- *oblique view:* The image plane is tilted.
- *horizontal view:* The image plane is approximately vertical, typical for terrestrial images.

We use the following notation, much of which is related to aerial images (Figs. 12.1, 12.2):

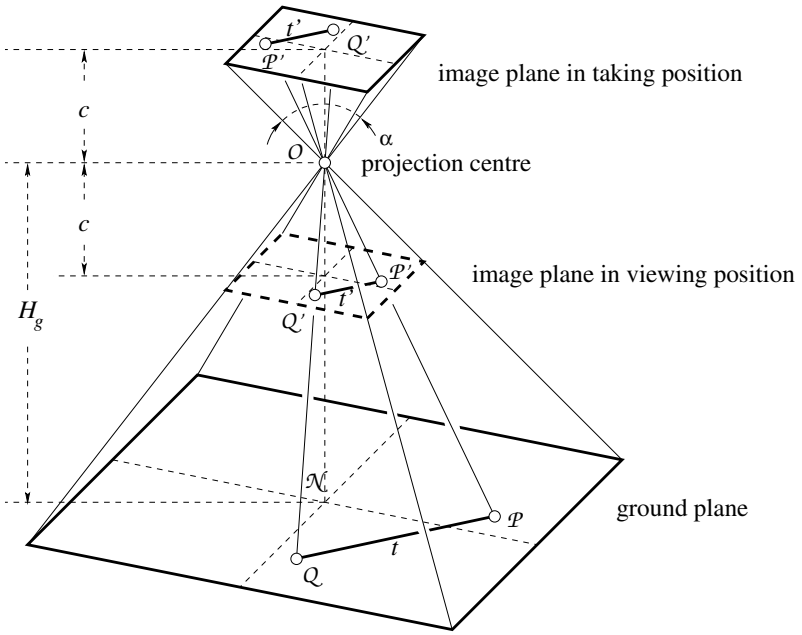


Fig. 12.1 Basic points and lines in a nadir image: projection centre O, image plane in taking position, image plane in viewing position, ground plane, nadir point N, principal distance c, flight height over ground H_g, image scale $t'/t = c/H_g$. If the ground surface is not flat, the image scale varies from image point to image point. Viewing angle α referring to the diagonal

- The image rays all intersect in one point, the *projection centre* O. It lies between the object space and the image plane during exposure of the image (*taking position*). The scene is shown upside down, with the image reflected as a mirror image. Therefore the image usually is drawn in *viewing position*, where the scene is shown in its correct orientation.[1] *projection centre*
- In the case of a nadir image of a horizontal plane, the image is similar to the object. Distances $t = \overline{\mathcal{PQ}}$ in object space are mapped to distances $t' = \overline{\mathcal{P}'\mathcal{Q}'}$, reduced by the *scale number* $S = t/t' = H_g/c$, where H_g is the flight height above ground and c is the *principal distance*, close to the focal length. The *image scale*, $s = 1/S$, is the inverse of the scale number. For digital cameras the principal distance usually is measured in pixels, then the scale number $S = H_g/c$ has unit [m/pixel]. Its value is identical to the *ground sampling distance* GSD given in meters:

$$S\ [\text{m/pixel}] = \frac{H_g\ [\text{m}]}{c\ [\text{pixel}]}\,, \qquad \text{GSD}\ [\text{meter}] = S \times 1\ [\text{pixel}]\,. \tag{12.1}$$

  For nadir images and flat terrain S and GSD are constant over the whole image.

[1] In classical literature on perspective mapping the projection centre is denoted by O. The notation stands for the Latin word *oculus* (eye). Its use can be traced back to perspective drawing devices used by painters during the Renaissance, such as Leonardo da Vinci (1452-1519) and Albrecht Dürer (1471-1528), who are regarded as pioneers of projective geometry (Stocker and Schmid, 1966; Slama, 1980; Faugeras and Luong, 2001). We follow this notation and specify the projection centre by $\mathcal{O}([X_O, Y_O, Z_O])$. According to our convention, its *coordinate vector* should read $\boldsymbol{O}$ or the homogeneous $\mathbf{O}$. This leads to confusion with the zero scalar 0, the zero vector $\mathbf{0}$ and the zero matrix $\boldsymbol{0}$. Therefore we choose $\boldsymbol{Z}$ as the name for the coordinate vector of $\mathcal{O}$, taken from the German word "Projektions**z**entrum", analogously to the $\boldsymbol{C}$ used by Hartley and Zisserman (2000) for "projection **c**entre". Thus the projection centre is given by $\mathcal{O}(\boldsymbol{Z})$, and its coordinate vector is $\boldsymbol{Z} = [X_O, Y_O, Z_O]^\mathsf{T}$.

For vertical images of flat, smooth terrain, the image scale varies slightly from point to point. Here the image scale s is the average value of the local image scales.
In rough terrain or oblique views the image scale and thus the ground sampling distance vary greatly from point to point, due to the relief or the tilt of the image.

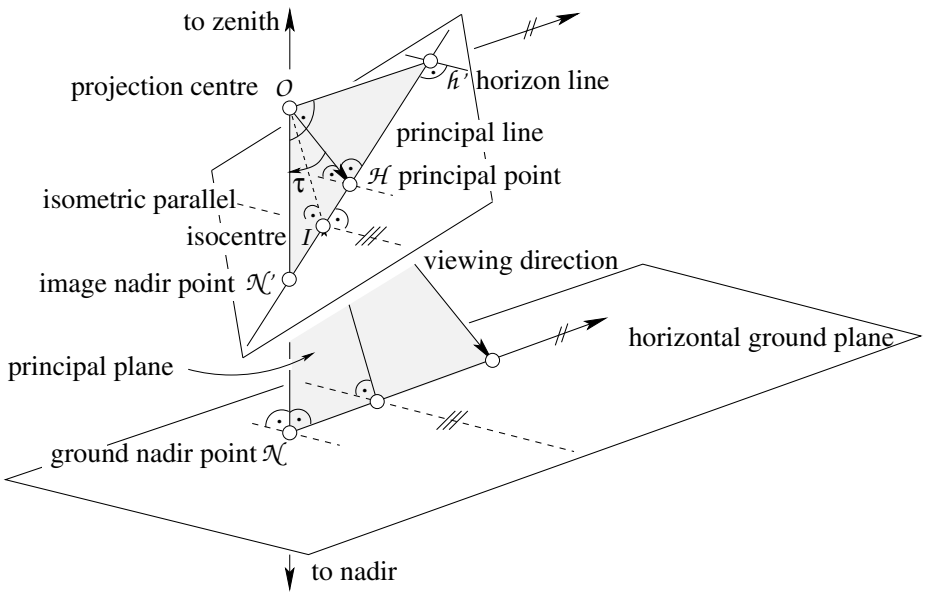


Fig. 12.2 Basic points and lines in a tilted image: projection centre O, principal point $\mathcal{H}$, viewing direction $\overrightarrow{O\mathcal{H}}$, nadir points $\mathcal{N}$ and $\mathcal{N}'$, tilt angle τ, isocentre I = bisector of nadir and viewing direction intersected with image plane, isometric parallel = horizontal line in the image through the isocentre I, horizon line h', principal line $\mathcal{N}'\mathcal{H}$

- The *viewing angle* or *field of view*is the angular extent of the scene seen by a camera. It may be related to the horizontal, vertical, or diagonal diameter of the image. The angle α can be derived from the chosen diameter d and the principal distance c by

$$\alpha = 2 \arctan \frac{d}{2c} . \tag{12.2}$$

- The *principal point* is named $\mathcal{H}$,[2] and is the point in the image plane closest to the projection centre (Fig. 12.2). The *viewing direction* is the direction of the line $O\mathcal{H}$ towards the object.
- The *nadir points* $\mathcal{N}$ and $\mathcal{N}'$ are the intersections of the plumb line through the projection centre O in the ground plane and the image plane, respectively.
- The *principal plane* is defined by the nadir line $\mathcal{N}\mathcal{N}'$ and the viewing direction $O\mathcal{H}$. This plane stands perpendicular on the horizontal ground plane and intersects the image plane in the principal line.
- The *principal line* is the line of maximum slope in the image plane and passes through the principal point and the nadir point.
- The *horizon line* is the intersection of the horizontal plane through the projection centre O and the image plane and is perpendicular to the principal line. It is the image of the horizon.

[2] Since the letters p and $\mathcal{P}$ are used in various contexts, we adopt the letter $\mathcal{H}$ for the principal point, following the German name Hauptpunkt.

- The *tilt angle* τ is the angle between the viewing direction $\mathcal{OH}$ and the plumb line. The *swing angle* (not shown in Fig. 12.2) is the angle between the principal line and the y' coordinate axis of the image coordinate system.
- The *isocentre* is the intersection of the bisector of $\mathcal{ON}$ and $\mathcal{OH}$ with the image plane. At this point the local image scale for infinitesimal distances is independent of their directions and of the tilt angle τ. The image scale for horizontal terrain at this point is identical to c/H_g. It is the only conformal point of the plane perspective projection. Exercise 12.3
- The *isometric parallel* is a horizontal line through the isocentre. For horizontal terrain the image scale along this line is equal to c/H_g. The local image scale is larger than c/H_g below the isometric parallel and smaller above it.

Depending on the field of view it is common to characterize the lens system:

- *Normal* or *standard lenses* cover angles between 40°and 60°.
- *Wide angle lenses* cover angles between 60°and 85°.
- *Ultra wide angle lenses* cover angles up to approximately 120°.
- *Fish-eye lenses* cover angles up to more than 180°.
- *Narrow angle lenses* cover angles between 30°and 40°.
- *Telelenses* cover angles below 30°.

Zoom lenses are able to change the focal length. They may cover very large ranges of viewing angles.

Sketches of a horizontal and a vertical view are shown in Fig. 12.3. The horizontal view (left) shows the horizon line h' with two vanishing points v'_1 and v'_2 that form the image

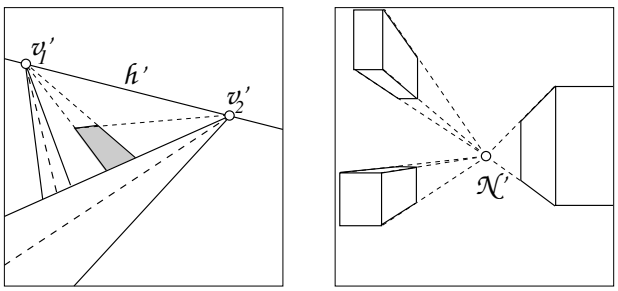


Fig. 12.3 Horizontal (left) and vertical (right) views with horizon line h', vanishing points v'_1 and v'_2 and nadir point $\mathcal{N}'$

of the intersection of groups of parallel 3D lines in object space. Obviously the position of the horizon line allows inference of the tilt and swing angles. The vertical view (right) shows the image nadir point $\mathcal{N}'$, which is the vanishing point of plumb lines in object space. In spite of having a vertical view, we can see the vertical walls of the buildings and realize that points having the same planimetric coordinates but different heights have significantly different positions in the image. This *relief displacement*, also present but not visible in images of smooth terrain, can be used to infer the heights of the buildings. The local image scale obviously differs for points at ground level and at the tops of the buildings.

The image geometry thus needs to be modelled in detail in order to cope with all these effects and to fully exploit the geometric information of the perspective image.

12.1.2 General Aspects on Modelling the Projection

The basic modelling of a projection refers to points. The projection of straight lines or quadrics can be derived from the basic projection equations for points.

12.1.2.1 Interior and Exterior Orientation

As the position of the camera in space usually varies much more quickly than the geometry and the physics of the camera itself, we usually distinguish between two sets of parameters in modelling:

exterior orientation, pose of camera

1. *Extrinsic Parameters*, sometimes called extrinsics, describe the *pose* of the camera in space. They always contain the six parameters of the *exterior orientation* (EO), namely the three coordinates of the projection centre, or the translation of the camera from the origin to its position during exposure, and the three parameters describing the rotation, e.g., as rotation angles around the three camera axes.
   The exterior orientation parameters vary with the motion of the camera in space or may be constant over time if the camera is fixed, as when using a tripod.
   The parameters of the exterior orientation may be directly measured; however, they are usually determined by orientation procedures.
2. *Intrinsic Parameters*, sometimes referred to as intrinsics, are all parameters necessary to model the geometry and the physics of the camera in order to be able to infer the direction of the projection ray towards an object point given an image point and the exterior orientation.

interior orientation of camera

   The intrinsic parameters describe the *interior orientation* (IO) of the camera. In the most simple case of an ideal camera this may be only the distance of the pinhole from the image plane; in the most refined model of a generic camera this may be several dozens of parameters. The interior orientation is determined by *calibration*.
   The interior orientation in photogrammetric applications is usually held fixed. A camera with fixed interior orientation is called a *metric camera*, and we can assume that the calibration leads to intrinsic parameters, which are valid for a certain time. On the other hand, images taken with a camcorder cannot be assumed to have a stable interior orientation due to zooming and the fact that its CCD chip may be not in a fixed relation with the camera lens.

12.1.2.2 Calibrated and Uncalibrated Cameras

It is useful to look at the interior orientation of a camera from its state of calibration, which depends on the envisaged task. We may distinguish between three different states, leading to different camera models:

calibrated camera

1. The intrinsic parameters of the camera are completely known up to an accuracy which is sufficient for the envisaged task. We call this a *calibrated camera*.

metric camera

   This implies that the calibration has been performed and the camera is stable over the time of its usage. It is then called a *metric camera*. Most photogrammetric cameras are metric cameras. For metric cameras the relation between observable image points and projection rays is available in the camera frame.
   If the camera has a viewing angle significantly below 180° and is calibrated, we can employ the model of a calibrated perspective camera, especially if we want to keep the relation to the sensor coordinates. For calibrated cameras we can always use the model of a *spherical camera* if we transfer the sensor coordinates together with their uncertainty into ray directions using the information of the camera calibration.

partially calibrated camera

2. Some of the intrinsic parameters of the camera have been determined by calibration with an accuracy sufficient for the envisaged task. We call this a *partially calibrated camera*.
   The parameters which disturb the straight line property of the camera also may be assumed to be negligible, so that the mapping from object space to image space sufficiently preserves straight lines. Then we arrive at a straight line-preserving *perspective camera*. This particular state of calibration is conceptually simple. In the area of *com-*

puter vision a camera is often called *uncalibrated* if nothing is assumed to be known about the camera except that it is free from nonlinear distortions.

3. The intrinsic parameters of the camera are completely unknown. We call this an *uncalibrated camera.* *uncalibrated camera*

Observe, any mathematical model can be used for the calibration if it is sufficient for the application.

12.1.2.3 The Geometry of the Thick Lens Representing the Projection of Optical Systems

We now describe the mapping from object into image space to a degree which is sufficient for most applications. It is based on the model of a thick lens from geometric optics as shown in Fig. 12.4. Although simple, this model effectively reflects the geometric projection of sophisticated optical systems in photogrammetric cameras (cf. McGlone, 2013, Sect. 4).

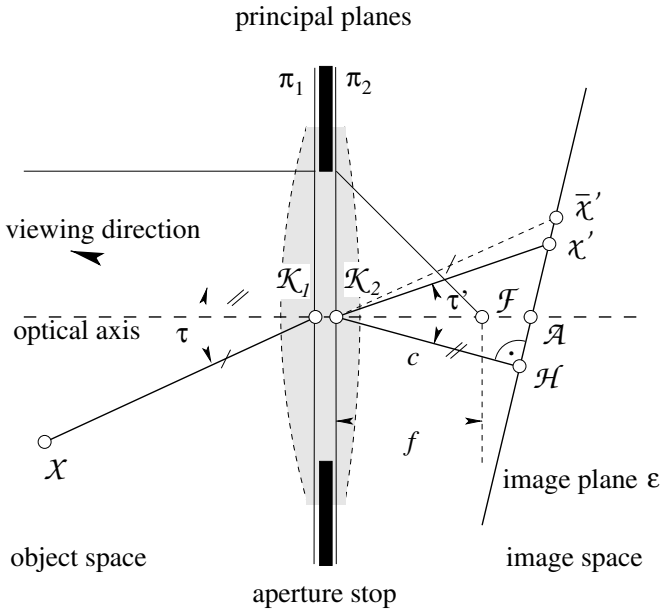


Fig. 12.4 Geometry of optical mapping

The model assumes the optics to be rotationally symmetric around the *optical axis.* The *focal point* $\mathcal{F}$ lies on the optical axis at the distance of the *focal length* f from the principal plane π_2. The image plane generally is not perpendicular to the optical axis, nor does it pass through the focal point $\mathcal{F}$. Due to lens distortion, the projection ray does not have the same direction in object space (left) and in image space (right). The essential parts of the ray from X to $\mathcal{K}_1$ and from $\mathcal{K}_2$ to the observable image point x' are displaced by a certain amount, as the two principal planes of the lens are not identical and may be separated by up to several centimetres.

bundle of rays for orientation and scene reconstruction

For camera orientation and scene reconstruction we use only the *bundle of rays* in object space passing through $\mathcal{K}_1$. As the geometric relation between image plane and object space is of no concern, we may mentally shift the principal plane π_2 to the left together with the part of the figure to the right of it until both principal planes π_1 and π_2 coincide and $\mathcal{K}_1 = \mathcal{K}_2$. This point is then used as the *centre of the camera coordinate system* S_c. It will be called *projection centre* and denoted by O.

To simplify the mapping relations we choose the ${}^cX$ and the ${}^cY$ axes parallel to the image plane such that they form a right-handed Cartesian coordinate system with the ${}^cZ$ axis then perpendicular to the image plane. The direction of the optical axis therefore has no direct effect on modelling. However, the intersection point $\mathcal{A}$ of the optical axis with the image can be taken as the *point of symmetry* when modelling lens distortion, as this distortion in a first approximation can be assumed to be rotationally invariant.

Since the image plane generally will not pass through the focal point $\mathcal{F}$, e.g., when focusing on a point at finite distance, the *principal distance* c is essential, since it is generally different from the focal length. The principal distance is determined computationally and approximates the distance of the projection centre $\mathcal{O} = \mathcal{K}_2$ from the image plane. The point on the image plane closest to the projection centre is the *principal point* $\mathcal{H}$. The direction $\mathcal{HO}$ is the *viewing direction* and coincides with the optical axis only if the image plane is perpendicular to it.

Observe, that zooming not only changes the principal distance of a camera but generally also the other parameters of the interior orientation, e.g., the principal point. Moreover, when analysing colour images, care has to be taken, since the geometry of the optical mapping varies with the colour (cf. Willson and Shafer, 1994).

We now address the different steps of the projection.

12.1.2.4 Modelling Perspective Cameras with Distortion

We now develop the general model for cameras having a unique viewing point. This will include as intermediate steps the model of a *spherical camera* and the model of a *perspective camera*. Both will allow us to exploit the tools from projective geometry.

For modelling the projection, we will need to represent points in different coordinate systems. We refer to Fig. 12.5, where the image is shown in viewing position. We assume all the coordinate systems to be Cartesian and right-handed.

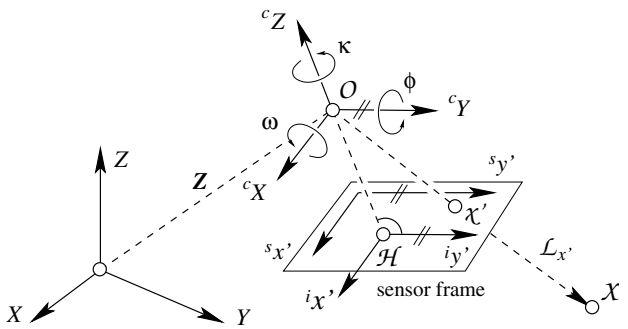


Fig. 12.5 Perspective projection of an object point $\mathcal{X}$ with a camera into the image point $\mathcal{X}'$. Coordinate systems: object coordinate system $[X, Y, Z]$, projection centre O, camera coordinate system $[{}^cX, {}^cY, {}^cZ]$, image coordinate system $S_i, [{}^ix', {}^iy']$, with origin in the principal point $\mathcal{H}$, sensor coordinate system $[{}^sx', {}^sy']$. We assume the y-axis of the sensor and the y-axes of the camera to be parallel to the rows of the sensor. When taking the normalized directions $O\mathcal{X}'$ as observable entities in the camera system we arrive at the spherical projection. The right-hand rotations around the three coordinate axes of the camera system are denoted by ω, ϕ, and κ

We refer to the following coordinate systems:

1. The scene or object coordinate system S_o. As we only deal with static objects in a fixed object coordinate system, we will omit the superscript, thus $\mathbf{X} \doteq {}^o\mathbf{X}$.
   The choice of the object coordinate system is up to the user. For numerical reasons it is often useful to choose an object coordinate system with the origin at the centre

of the area of interest and with a unit larger than half the diameter of the area of interest. We will not always make such conditioning explicit.

2. The camera coordinate system S_c. Its origin is at the *projection centre* O, close to the centre of the lens, cf. the discussion of Fig. 12.4, p. 461.
3. The sensor coordinate system S_s of the camera. For digital cameras its origin is at the position of the centre of pixel $(0, 0)$ or pixel $(1, 1)$, depending on the convention.
4. The centred or image coordinate system S_i. It is parallel to the sensor coordinate system. Its origin is in the image plane in the prolongation of the ${}^cZ$-axis.

The direction of the y-axis of the camera system, of the image coordinate system and of the sensor system are parallel, and are defined by the direction of the rows of the sensor. This is motivated by the fact that the x'-coordinates of the pixels are electronically defined, whereas the y'-coordinates are defined by the hardware, which can be assumed to be more stable. During the derivation we will use other coordinate systems, such as the one centred at the principal point, which will be explained later.

The mapping can be split into four steps (see Fig. 12.6):

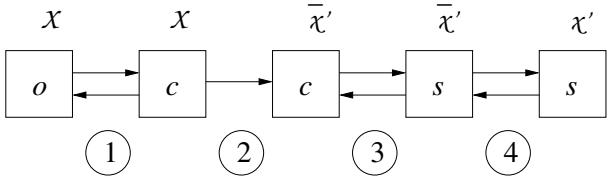


Fig. 12.6 Steps of mapping with a camera. The scene point $\mathcal{X}$ is transformed from the scene system S_o to the camera system S_c, mapped to the ideal image point $\bar{\mathcal{X}}'$, transformed into the sensor system S_s, and distorted to achieve the observable image point $\mathcal{X}'$

1. Transformation of the space point $\mathcal{X}(\mathbf{X})$ from S_o to S_c leads to $\mathcal{X}({}^c\mathbf{X})$ specifying the exterior orientation of the camera. It consists of the three coordinates (X_O, Y_O, Z_O) of the camera centre O in the reference coordinate system S_o and a rotation from the reference system S_o to the camera system S_c specified by three Euler angles ω, ϕ, and κ. The sequence (ω, ϕ, κ) is similar to the sequence (o, p, q) in the Latin alphabet.
2. Generate a half ray $\mathcal{L}_{x'}$ from the projection centre O to the scene point $\mathcal{X}$. Its direction vector, ${}^c\bar{\mathbf{x}}'$, is expressed in the camera coordinate system. The *bundle of rays* $\{{}^c\bar{\mathbf{x}}'^s_i\}, i = 1, ..., I$, can be treated as the *ideal* representation of the information, which we intend to obtain from the points $\mathcal{X}'_i$ measured in the sensor. It is ideal in the sense that it is free of any random or systematic perturbation.
   In this step we lose the depth information, which needs to be recovered using scene information or other images.

   bundle of rays
3. Transformation of the direction ${}^c\bar{\mathbf{x}}'$ of the camera ray into the coordinate system of the digital camera, i.e., the sensor coordinate system S_s, leads to the undisturbed *ideal image point* $\bar{\mathcal{X}}'({}^s\bar{\mathbf{x}}')$.[3] We will do this in several steps which give rise to specific camera models.

   ideal image point
4. Shift the ideal image point $\bar{\mathcal{X}}'({}^s\bar{\mathbf{x}}')$ to the *observable image point* $\mathcal{X}'({}^s\mathbf{x}')$ represented in the sensor system, in this way realizing the modelled imaging errors.

   observable image point

The mapping is *not invertible*, since one dimension, the depth information, is lost completely.

The mapping from space to image can be modelled in different ways. The camera models differ in the number and type of intrinsic parameters. We distinguish between the following

[3] The notion "ideal image point" is not to be confused with the notion of *ideal point* in projective geometry denoting a point at infinity.

camera models,[4] which are more or less good approximations of a real camera (see Fig. 12.7).

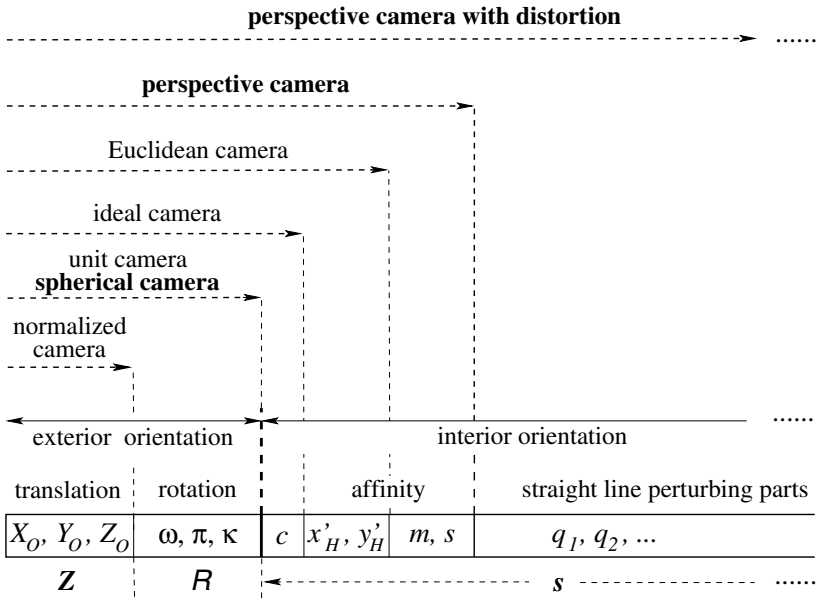


Fig. 12.7 Overview of the mapping steps for perspective and spherical cameras, the involved parameters and the naming of the camera models: rotation matrix $\boldsymbol{R}$ with rotation angles, coordinates $\boldsymbol{Z}$ of the projection centre, parameters $\boldsymbol{s}$ for interior orientation, partitioned into five parameters for line-preserving mapping and further parameters $\boldsymbol{q}$ for the straight line perturbing parts. The unit camera and the ideal spherical camera without distortions are both characterized by the six parameters of the exterior orientation only

- The *perspective camera with nonlinear distortions* has a planar sensor. It does not preserve straight lines. For its specification we need the six parameters of the exterior orientation and a number, N_s, of parameters $\{s_i\}$ for specifying the interior orientation.

  additional parameters

  These *additional parameters* can be subdivided into two sets: five parameters specifying an affine transformation between the camera and the sensor system and further parameters q_i required to describe the image errors which perturb straight lines, especially due to lens distortion.[5]

- The *perspective camera* or *camera with affine sensor coordinate system* also has a planar sensor, and is characterized solely by the invariance of straight lines. Here the image coordinate system S_s – besides having an offset from the principal point – may have a shear s or a scale difference m in x' and y' with respect to the physical situation.[6]

  perspective camera is straight line-preserving

- The *Euclidean camera*, where the geometric elements in the image plane follow Euclidean geometry, with coordinate system S_e, differs from a perspective camera only by the lack of shear and scale difference between the axes of the camera and the sensor coordinate system.

  The only parameters of the interior orientation are the principal distance c and the coordinates $[x'_H, y'_H]$ of the principal point $\mathcal{H}$.

  Again, this is a good model for a pinhole camera with planar sensor if the coordinates of the image points are measured in an arbitrary Cartesian coordinate system. It

[4] The notions used in this introductory section are later explained in detail.

[5] cf. the concatenation $\mathsf{K} = \mathsf{K}_i {}^i\mathsf{K}$ in (12.31), p. 471 of ${}^i\mathsf{K}$ in (12.21), p. 469 and of K_i in (12.29), p. 471.

[6] This notion of an affine camera model is not to be confused with the one introduced by Hartley and Zisserman (2000) and Faugeras and Luong (2001), which is a *parallel projection* onto an image plane with an *affine* coordinate system.

is also a good model for a digital camera if the pixel distances are the same in both coordinate directions, there is no skew, and the lens does not produce any unacceptable distortion.[7]

The following camera models are special cases of the perspective camera model; thus, all are straight line-preserving.

- The *ideal camera* (Fig. 12.8, p. 469) with coordinate system S_i is a Euclidean camera with *principal point*, which is the origin of a Cartesian coordinate system. The camera is characterized by the *principal distance* c as the only parameter of the interior orientation.

  The ideal camera is a good model for a pinhole camera with a planar sensor if the principal point is the origin of the coordinate system
- The (ideal) *unit camera* with coordinate system S_c is a camera with principal distance $c = 1$. There are no additional parameters describing the interior orientation as long as the viewing field is not too large. The image coordinates do not require conditioning for further processing.

  When normalizing the homogeneous image vectors ${}^c\mathbf{x}'$ spherically, we obtain the above-mentioned bundle of direction vectors, expressed in the camera coordinate system, pointing towards the scene points. When treating the normalized homogeneous image vectors ${}^c\mathbf{x}'^s$ as oriented quantities, we arrive at the model for the ideal *spherical camera*, ideal as we assume no distortions are present. Here the scene points are projected to the viewing sphere instead of to an image plane (see Fig. 11.9, p. 447). This camera is the basis for modelling omnidirectional cameras with a unique projection centre.

  Remark: Although the image of a spherical camera is the unit sphere, the ideal spherical mapping can be treated as straight line-preserving, since 3D lines map to great circles and straight lines on the sphere are great circles, or, equivalently, since three collinear 3D points $\mathcal{X}_i$ map to three collinear 2D points as their direction vectors are coplanar. We therefore could treat both mappings, the perspective and the ideal spherical mappings, as straight line-preserving ones. ◇

  In the following, however, when referring to a perspective camera model we mean a straight line-preserving mapping with a planar sensor.
- The *normalized camera* with coordinate system S_n has principal distance $c = 1$ and rotation matrix $R = I_3$. Its coordinate system is centred at the principal point and is parallel to the scene coordinate system.

  This camera can be used for computational purposes if the rotation parameters of the camera are known. The *normal case* for the image pair as it is defined in photogrammetry motivates this definition (Sect. 13.2.4, p. 561).[8]

All models for single cameras represent a mapping from object space to image space with a unique projection centre, as the projection rays all pass through a single point. This is valid for both perspective as well as spherical cameras.

In the following we present all mappings in homogeneous coordinates and, for comparison with classical textbooks, in inhomogeneous coordinates.

12.1.3 Modelling Central Cameras

12.1.3.1 Exterior Orientation

The exterior orientation transforms the coordinates $\boldsymbol{X} := {}^o\boldsymbol{X}$ from the scene or object coordinate system S_o into the camera system S_c. This can be achieved in two steps:

[7] This notion of a Euclidean camera model is not to be confused with the one introduced by Hartley and Zisserman (2000) and Faugeras and Luong (2001), which is a *parallel projection* onto an image plane with a Cartesian coordinate system.

[8] The notion differs from that used in Hartley and Zisserman (2000), where an ideal camera with $c = 1$ and arbitrary rotation is called a normal camera.

1. Translation of the object coordinate system S_o into the projection centre O with three coordinates $\boldsymbol{Z} = [X_O, Y_O, Z_O]^\mathsf{T}$ as parameters. This yields the normalized camera coordinate system S_n; cf. the definition of the normalized camera in the last section.
2. Rotation of the normalized coordinate system S_n into the system S_c. The rotation matrix $R \doteq {}^cR_o$ can be represented by three independent parameters.

In inhomogeneous coordinates, the coordinate transformation from the object coordinate system to the camera coordinate system is

$$ {}^c\boldsymbol{X} = R(\boldsymbol{X} - \boldsymbol{Z}) \,. \tag{12.3}$$

This representation does not allow for scene points at infinity. In homogeneous coordinates this reads as

$$ \begin{bmatrix} {}^c\boldsymbol{X} \\ 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} I_3 & -\boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix} ; \tag{12.4}$$

or, with the 4×4 matrix M which performs the motion of the object system into the camera system,

$$ \mathsf{M}(R, \boldsymbol{Z}) = \begin{bmatrix} R^\mathsf{T} & \boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} R & -R\boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}^{-1} , \tag{12.5}$$

we have the compact representation

$$ {}^c\mathbf{X} = \mathsf{M}^{-1}\, \mathbf{X} \,. \tag{12.6}$$

For points at infinity, $\mathcal{X}_\infty([\boldsymbol{X}_{\infty 0}^\mathsf{T}, 0])$, we have

$$ {}^c\boldsymbol{X}_{\infty 0} = R\boldsymbol{X}_{\infty 0} \,, \tag{12.7}$$

which is a pure rotation, as expected.

The definition of the coordinate system in object space or with respect to the physical body of the camera is left open at this point. Only if distances or angles are measured at the camera body do we need to fix the camera coordinate system with respect to the camera body.

on the meaning of R in the basic equation (12.3)

Remark: The definition of the motion matrix and thus the basic equation (12.3) requires an explanation. If we follow our convention from Sect. 6.3.2.2, p. 262 on transformations that any basic transformation is a displacement of an object or a coordinate system, the matrix M in (12.5) describes a displacement of the camera coordinate system w.r.t. the scene or object coordinate system. Observe, the rotation of this displacement is described by R^T, not by R. The reason is that we started with the coordinate transformation of a point $\mathcal{X}$ from the object coordinate system into the camera system in (12.3) using the rotation matrix R. We choose this convention for compatibility with the literature on Computer Vision (cf. Hartley and Zisserman, 2000).[9] Generally, this convention for defining the rotation does not cause trouble.

The chosen convention for the rotation matrix has an *important consequence w.r.t. the signs when estimating the rotation angles* ω, ϕ, and κ, as we will discuss in Sect. 12.2.2.3, p. 501. The rotational displacement of the object to the camera system can be written as a function of the rotation angles ω, ϕ, and κ around the three coordinate axes X, Y, and Z as

$$ R^\mathsf{T} = R_3(\kappa)R_2(\phi)R_1(\omega) \quad \text{or} \quad R = R_1(-\omega)R_2(-\phi)R_3(-\kappa) \,. \tag{12.8}$$

The rotation matrix R, which we usually refer to, thus is a function of the negative rotation angles ω, ϕ and κ.[10]

[9] Also observe, for simplicity we omitted the indices characterizing the motion matrix and their rotation and translation parameters. Taking all indices into account we would have ${}^c\mathsf{M}_o({}^cR_o^\mathsf{T}, {}^c\boldsymbol{T}_o)$, with ${}^c\boldsymbol{T}_o = \boldsymbol{Z}$, indicating that the motion of the coordinate systems is from the scene coordinate system S_o to the camera coordinate system S_c.

[10] We arbitrarily chose this specific sequence of rotation axes. For differential rotation angles the sequence of the three rotation axes has no effect. For larger angles the sequence should be chosen by the user, e.g., following the specifications of an inertial system.

When referring to the vector $[\omega, \phi, \kappa]^\mathsf{T}$, which we denote by $\boldsymbol{r}$ in the context of camera orientation,[11] we will therefore estimate the rotation matrix $\boldsymbol{R}$ in (12.8) using the multiplicative relation for improving an approximate rotation $\widehat{\boldsymbol{R}}^a$,

multiplicative update and linearization of the rotation matrix of a camera

$$\widehat{\boldsymbol{R}} = \boldsymbol{R}(-\widehat{\Delta\boldsymbol{r}})\widehat{\boldsymbol{R}}^a \approx (\boldsymbol{I} - \boldsymbol{S}(\widehat{\Delta\boldsymbol{r}}))\widehat{\boldsymbol{R}}^a \quad \text{with} \quad \widehat{\Delta\boldsymbol{r}} = \begin{bmatrix} \widehat{\Delta\omega} \\ \widehat{\Delta\phi} \\ \widehat{\Delta\kappa} \end{bmatrix}, \tag{12.9}$$

with the negative sign in the correcting rotation matrix $\boldsymbol{R}(-\widehat{\Delta\boldsymbol{r}})$, cf. (10.84), p. 383. We will refer to this convention for the rotation matrix when explicitly addressing the rotation angles in Sects. 12.2.2.3, p. 501, on estimating pose parameters of a single image, and 13.3.6, p. 588, on the estimation of the relative orientation of the image pair.

When modelling camera systems or explicitly addressing the motion, we recommend sticking to the convention introduced in Sect. 6, p. 247 on transformations and using ${}^c\mathbf{X} = \boldsymbol{R}^\mathsf{T}(\mathbf{X} - \boldsymbol{Z})$ instead of (12.3) (cf. Schneider and Förstner, 2013, Eq. (4), and Kraus, 1993). ◇

12.1.3.2 Perspective and Spherical Projection

The first and most crucial step in the projection from the scene to the image is the generation of projection rays from the camera to the scene.

Note, to keep the notation simple in the following sections, we do not distinguish between the observable image points $\mathcal{X}'$ and the ideal image point $\bar{\mathcal{X}}'$ (see Fig. 12.6, p. 463). Thus points $\mathcal{X}'$ can be interpreted as being observed in a camera without distortion. We will make the distinction between the observed point $\mathcal{X}'$ and the ideal $\bar{\mathcal{X}}'$ explicit when discussing image distortions.

Given the homogeneous camera coordinates with Euclidean and homogeneous parts,

$${}^c\mathbf{X} = \begin{bmatrix} {}^c\boldsymbol{X}_0 \\ {}^cX_h \end{bmatrix}, \tag{12.10}$$

of the scene point $\mathcal{X}$, we can derive the direction from the projection centre, i.e., the origin of the camera coordinate system, to $\mathcal{X}$, by

$${}^c\mathbf{x}' = {}^c\boldsymbol{X}_0 = [\boldsymbol{I}_3|\mathbf{0}]\,{}^c\mathbf{X}\,. \tag{12.11}$$

The 3×4 matrix $\mathsf{P}_4 = [\boldsymbol{I}_3|\mathbf{0}]$ has been used in (7.81), p. 307. Here it can be interpreted as the *projection matrix*

$${}^c\mathsf{P}_c = \mathsf{P}_4 = [\boldsymbol{I}_3|\mathbf{0}]\,, \tag{12.12}$$

which enables us to derive the direction of the viewing ray from the scene coordinates, both expressed in the camera coordinate system:

$${}^c\mathbf{x}' := {}^c\boldsymbol{X}_0 = {}^c\mathsf{P}_c\,{}^c\mathbf{X}\,. \tag{12.13}$$

This yields the *central projection* from the 3D point expressed in scene coordinates to the camera ray or (nonnormalized) *ray direction,*

direction of viewing ray in camera system, ray direction

$${}^c\mathbf{x}' = [\boldsymbol{I}_3|\mathbf{0}]\mathsf{M}^{-1}\mathbf{X} = \boldsymbol{R}(\boldsymbol{X}_0 - X_h\boldsymbol{Z})\,, \tag{12.14}$$

with $\mathbf{X} = [\boldsymbol{X}_0^\mathsf{T}, X_h]^\mathsf{T}$, which in both forms allows for points at infinity with $X_h = 0$. If we want to work with oriented entities, we need to assume $X_h \geq 0$. For points not at infinity (12.14) reduces to

$${}^c\mathbf{x}' = \boldsymbol{R}(\boldsymbol{X} - \boldsymbol{Z})\,. \tag{12.15}$$

Observe, the left-hand side is a homogeneous vector; thus, the equality holds up to scale.

[11] This vector is not to be confused with a rotation vector, except for small rotations; see the discussion on differential rotations in Sect. 8.1.6, p. 336.

We now clarify the properties of the two homogeneous vectors in (12.14). We want to allow for scene points at infinity, e.g., stars or points at the horizon; hence, we need to distinguish between the point at infinity, $\mathbf{X}_{\infty 1} = [\boldsymbol{X}_0^\mathsf{T}, 0]^\mathsf{T}$, with direction $\boldsymbol{X}_0$, and the point at infinity, $\mathbf{X}_{\infty 2} = [-\boldsymbol{X}_0^\mathsf{T}, 0]$, in the opposite direction, $-\boldsymbol{X}_0$. Therefore we treat scene points as elements of an oriented projective space $\mathbb{T}^3$.

Similarly, in order to guarantee that the vector ${}^c\mathbf{x}'$ points from the camera towards the scene point, and not from the camera away from the scene point, we also need to treat ${}^c\mathbf{x}'$ as an element of an oriented projective space, here of $\mathbb{T}^2$. This way we obtain the model of a *central camera*,

model of a central camera

$$ {}^c\mathcal{P} : \mathbb{T}^3 \mapsto \mathbb{T}^2 \qquad {}^c\mathbf{x}' = \lambda\, {}^c\mathsf{P}\mathbf{X}\,, \quad \lambda > 0\,, \tag{12.16}$$

with the projection matrix

$$ {}^c\mathsf{P} = {}^c\mathsf{P}_c\, \mathsf{M}^{-1} = [\boldsymbol{I}_3|\mathbf{0}]\, \mathsf{M}^{-1} = \boldsymbol{R}[\boldsymbol{I}_3| - \boldsymbol{Z}] = [\boldsymbol{R} \mid -\boldsymbol{R}\boldsymbol{Z}]\,. \tag{12.17}$$

Taking into account that all points in $\mathbb{T}^2$ can be represented by normalized vectors, we may spherically normalize the directions ${}^c\mathbf{x}'$. This leads to the model of a *spherical camera*,

model of a spherical camera

$$ {}^c\mathcal{P} : \mathbb{T}^3 \mapsto \mathbb{T}^2 \qquad {}^c\mathbf{x}'^s = \mathsf{N}({}^c\mathsf{P}\mathbf{X})\,. \tag{12.18}$$

Observe, (12.18) eliminates the scale between the predicted and the observed ray directions. Due to the normalization operator on the right-hand side, we do not need to use spherically normalized homogeneous coordinates for the scene points. Equation (12.18) represents the model of an *ideal spherical projection*: *ideal*, as the direction ${}^c\mathbf{x}'^s$ is identical to the one from the projection centre to the scene point – thus the three vectors ${}^c\mathbf{x}'$, ${}^c\mathbf{0}$ and ${}^c\mathbf{X}$ are collinear; *spherical*, as all image points ${}^c\mathbf{x}'^s$ are elements of a unit sphere around the projection centre.

A *spherical camera* therefore is a camera model which maps a set of points to a bundle of normalized directions and does *not* refer to a specific physical setup of the optics or the sensor.

When modelling a spherical camera, the projection matrix represents an oriented transformation, as otherwise the sign of the projection rays would possibly change. But even without normalizing the ray directions we have an oriented projection, as long as the sign of the scale λ in (12.16) is positive. We therefore introduce the following definition.

proper projection matrix

Definition 12.1.27: *Proper projection matrix.* A 3×4 projection matrix $\mathsf{P} = [\mathsf{A}|\mathbf{a}]$ is called *proper* if the left 3×3 submatrix A of P has a positive determinant. ◇

Since we assumed both the scene and the camera coordinate systems to be right-handed, the projection matrix ${}^c\mathsf{P}$ in (12.17) is proper by construction.

12.1.3.3 The Ideal Perspective Camera

We now assume the bundle of rays is narrow enough to be completely captured by a planar sensor. Then the field of view is a true subset of a hemisphere. A mapping is called an *ideal perspective* (see Fig. 12.8) if the lens is free from distortion, the sensor area is planar, the inhomogeneous image coordinate system is centred at the principal point, and the axes of this system, S_i, are parallel to the axes of the camera coordinate system, S_c. In this case the coordinates ${}^ix'$ and ${}^iy'$ of the ideal image point are the same as its ${}^cX$ and ${}^cY$ coordinates in the camera system, since the two-dimensional $({}^cX\,{}^cY)$-system and the S_i-system only differ by a shift along the ${}^cZ$-axis. In the photogrammetric community these coordinates are often called *reduced image coordinates*,[12] since they are are achieved from the observed image coordinates by a reduction step.

[12] This notion of reduction, i.e., centring and correcting for distortion, is not to be confused with the notion of reduced homogeneous coordinates i.e., achieving a minimal representation.

In an ideal mapping, the projection rays are straight and pass through the two points $\mathcal{K}_1$ and $\mathcal{K}_2$, with $\mathcal{K}_1 = \mathcal{K}_2$, in the optics (see Fig. 12.4, p. 461), and project the object points $\mathcal{X}$ into the ideal image points $\overline{\mathcal{X}'}$.

We model this ideal mapping with the coordinate systems in a specifically chosen relation: The image plane is assumed to be perpendicular to the ${}^cZ$-axis of the camera coordinate system. Its signed distance is the principal distance c. The convention for the sign is such that the plane ${}^cZ = c$ is the image plane. For $c < 0$ we have the image plane in viewing position, for $c > 0$ in taking position (see Fig. 12.8, p. 469). The origin of the image coordinate system S_i is in the principal point $\mathcal{H}$. The xy-axes of the image coordinate system are parallel to those of the camera system.

taking position: $c > 0$
viewing position: $c < 0$

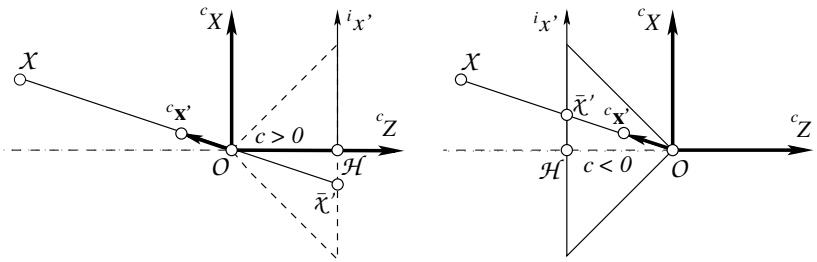


Fig. 12.8 Geometry of ideal perspective mapping. The image plane is given by ${}^cZ = c$. Camera coordinate system S_c and image coordinate system S_i are parallel and mutually shifted by c. **Left**: taking position, situation with $c > 0$. **Right:** viewing position, situation with $c < 0$

For the ideal image point $\overline{\mathcal{X}'}$ we therefore have the homogeneous camera coordinates ${}^c\mathbf{x}' = [{}^cu', {}^cv', {}^cw']^\mathsf{T} = [{}^cX, {}^cY, {}^cZ]^\mathsf{T}$ (cf. (12.13), p. 467). Taking the sign of the principal distance c into account, the (reduced) *image coordinates* ${}^i\boldsymbol{x}'$ of the ideal image point are

image coordinates ${}^i\boldsymbol{x}'$

$$ {}^ix' = c\,\frac{{}^cu'}{{}^cw'}\,, \quad {}^iy' = c\,\frac{{}^cv'}{{}^cw'} \tag{12.19} $$

or

$$ {}^i\mathbf{x}' = {}^i\mathsf{K}_c\,{}^c\mathbf{x}'\,, \tag{12.20} $$

with the 3×3 matrix

$$ {}^i\mathsf{K} := {}^i\mathsf{K}_c = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}\,. \tag{12.21} $$

This matrix contains the first parameter of the interior orientation of the camera. It is called a *calibration matrix*. It transforms the camera rays into homogeneous image coordinates of perspective cameras. For convenience we write the calibration matrix omitting the lower right index $_c$. The calibration matrix is a homogeneous matrix, since a multiplication of K with an arbitrary factor does not change the projection.

We have to take care of the sign of the homogeneous vectors. The relation between the camera rays and the image coordinates ${}^i\boldsymbol{x}' = [{}^ix', {}^iy']^\mathsf{T}$ is ${}^c\mathbf{x}' \stackrel{+}{=} -[{}^ix'/c, {}^iy'/c, 1]^\mathsf{T}$ (see Fig. 12.8). The minus sign guarantees that the *camera ray* ${}^c\mathbf{x}'$ points in the correct direction. In order to be independent of the sign of the principal distance, we have

camera ray ${}^c\mathbf{x}'$

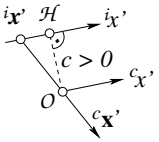
$$ {}^c\mathbf{x}' \stackrel{+}{=} -\text{sign}(c) \begin{bmatrix} {}^ix' \\ {}^iy' \\ c \end{bmatrix}\,. \tag{12.22} $$

If ${}^i\mathbf{x}'$and ${}^i\mathsf{K}$ may have arbitrary signs, this leads to

$$ {}^c\mathbf{x}' \stackrel{+}{=} -\text{sign}(c\,K_{33}\,{}^ix'_3)\;{}^i\mathsf{K}^{-1}\,{}^i\mathbf{x}'\,. \tag{12.23} $$

Thus, deriving correctly signed ray directions from image coordinates needs to take the sign of the principal distance into account, as this sign indicates whether we have an image in taking or viewing position. Only if we do not need the correct signs we may use the direct inversion of (12.20), p. 469,

$^i\boldsymbol{x}' \to {}^c\mathbf{x}'$:



$$ {}^c\mathbf{x}' = \begin{bmatrix} {}^ix' \\ {}^iy' \\ c \end{bmatrix} = {}^i\mathsf{K}^{-1}\, {}^i\mathbf{x}' \,. \tag{12.24} $$

With the exterior orientation (12.6) and the central projection (12.13) in the camera system, we use the projection matrix

$$ {}^i\mathsf{P} = {}^i\mathsf{K}\boldsymbol{R}[\boldsymbol{I}_3| - \boldsymbol{Z}] = {}^i\mathsf{K}\,[\boldsymbol{I}_3|\mathbf{0}]\,\mathsf{M}^{-1} \,, \tag{12.25} $$

to arrive at the perspective projection with an ideal camera,

$$ {}^i\mathbf{x}' = {}^i\mathsf{P}\,\mathbf{X} \,. \tag{12.26} $$

The projection matrix ${}^i\mathsf{P}$ is proper if $|\mathsf{K}| > 0$ or, equivalently, if $K_{33} > 0$.

collinearity equations

The inhomogeneous image coordinates ${}^i\boldsymbol{x}'$ are given by the *collinearity equations*,

$$ {}^ix' = c\,\frac{r_{11}(X - X_O) + r_{12}(Y - Y_O) + r_{13}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)} \tag{12.27} $$

$$ {}^iy' = c\,\frac{r_{21}(X - X_O) + r_{22}(Y - Y_O) + r_{23}(Z - Z_O)}{r_{31}(X - X_O) + r_{32}(Y - Y_O) + r_{33}(Z - Z_O)} \,, \tag{12.28} $$

depending on the principal distance c, the elements r_{ij} of the rotation matrix $\boldsymbol{R}$, the inhomogeneous coordinates $\boldsymbol{Z} = [X_O, Y_O, Z_O]^\mathsf{T}$ of the projection centre, and the scene point $\boldsymbol{X} = [X, Y, Z]^\mathsf{T}$.

Obviously, the unit and the spherical camera models can be thought of having calibration matrix $\mathsf{K} = \boldsymbol{I}_3$, or the principal distance can be thought of being $c = 1$. The following camera models only differ by the form of the calibration matrix, cf. the synopsis in Table 12.1, p. 479.

12.1.3.4 The Perspective Camera

We now extend the projection model to obtain the most general straight line-preserving case.

We use the convention in the field of computer vision and denote the image coordinates by $[x', y']$ without indicating the coordinate system by a left superscript ^{s} of the sensor coordinate system S_s.

The relation between the reduced image coordinates $[{}^ix', {}^iy']$ and the image coordinates $[x', y']$ is affine (see Fig. 12.9). We therefore also call this a *camera with affine sensor*. The affinity is realized by the sequence of the following affine transformations:

1. Translation of the coordinate system into the principal point (x'_H, y'_H) of the image coordinate system S_c. The scale is taken from the ${}^cx'$ axis; thus, the distance Δi between the rows is the length unit, since in digital cameras it is most stable by construction.
2. Correction of the scale of the y' coordinates by the factor $1 + m = \Delta j/\Delta i$. This takes into account the possibly electronically realized spacing of the columns.
3. Shear of the ${}^cy'$ axis. Although this part is usually negligible we include it, as it is straight line-preserving. We use the shear parameter $s = \tan\alpha$ related to the shear angle α.

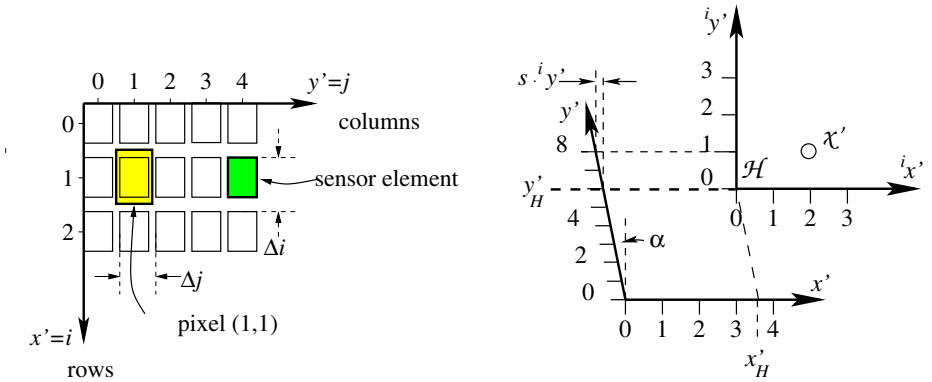


Fig. 12.9 Sensor coordinate system and image coordinate system. **Left:** Sensor coordinate system S_s of the affine sensor. The integer-valued image coordinates of the pixels are denoted by i and j in a right-handed system. Their scales may be different as the distance Δi between the rows and the distance Δj between the columns may differ. The coordinates are assumed to refer to the centre of the sensor element as the definition of the picture element may not be symmetric w.r.t. the sensor element; here the picture element is shown symmetric to the sensor element. The numbering of the pixels usually starts with $(0, 0)$. Image coordinates generally may be real-valued; then they are denoted by x' and y'. **Right:** Mutual position of the Cartesian image coordinate system S_i for ideal camera and affine coordinate system S_s: principal point $\mathcal{H}$, scale difference $s = \tan(\alpha)$ and shear depending on the ${}^i y'$-coordinate. The relation between sensor and picture elements may be chosen differently without changing the relations discussed below

In inhomogeneous coordinates we therefore obtain

$$\begin{aligned} x' &= {}^i x' + s\,{}^i y' + x'_H \\ y' &= {}^i y' + m\,{}^i y' + y'_H \,, \end{aligned}$$

in homogeneous coordinates,

$$\mathbf{x}' = \begin{bmatrix} 1 & s & x'_H \\ 0 & 1+m & y'_H \\ 0 & 0 & 1 \end{bmatrix} {}^i\mathbf{x}' =: \mathsf{K}_i\, {}^i\mathbf{x}' \,. \tag{12.29}$$

This defines the homography K_i of the ideal image point from the coordinate system centred in the principal point to the affine coordinate system of the sensor. The matrix K_i is homogeneous. If the scale correction $m > -1$ then the affine transformation is orientation preserving. Most cameras show only a small scale difference.

The projection into the sensor system therefore reads as

$$\mathbf{x}' = \mathsf{K}_i\, {}^i\mathsf{K} \boldsymbol{R}[\boldsymbol{I}_3 | -\boldsymbol{Z}]\mathbf{X} \,. \tag{12.30}$$

We now concatenate the transformations $\mathcal{K}_i$ and ${}^i\mathcal{K}$ to obtain the combined *calibration matrix* from the camera rays into the sensor system, *calibration matrix*

$$\mathsf{K} \doteq \mathsf{K}_i\, {}^i\mathsf{K} = \begin{bmatrix} c & cs & x'_H \\ 0 & c(1+m) & y'_H \\ 0 & 0 & 1 \end{bmatrix} . \tag{12.31}$$

It is an upper triangular matrix and contains five parameters of the interior orientation:

- the principal distance c,
- the coordinates $[x'_H, y'_H]$ of principal point measured in the sensor system,
- the scale difference m, and
- the shear s.

Sometimes the two first main diagonal terms are named c_1 and c_2, implying two different principal distances in x- and y-direction. The calibration matrix relates the sensor coordinates and the camera coordinates by

$$\mathbf{x}' = \mathsf{K}\,{}^c\mathbf{x}'\,, \tag{12.32}$$

a relation which we will regularly use. The final *projection* therefore reads (cf. Das, 1949)

$$\mathbf{x}' = \mathsf{P}\mathbf{X}\,, \tag{12.33}$$

projection matrix

with the homogeneous *projection matrix*

$$\mathsf{P} = \mathsf{K}\boldsymbol{R}[\boldsymbol{I}_3|-\boldsymbol{Z}] = \mathsf{K}[\boldsymbol{I}_3 \mid \mathbf{0}]\mathsf{M}^{-1}\,. \tag{12.34}$$

It has 11 degrees of freedom, thus depends on 11 parameters, namely five parameters of the interior orientation in K and the six parameters of the exterior orientation in $\mathsf{M}(\boldsymbol{R}, \boldsymbol{Z})$ from (12.5), p. 466. The projection matrix is homogeneous as its scale can be arbitrarily chosen.

The mapping (12.33) with the elements p_{ij} of P is explicitly given as

$$x' = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \tag{12.35}$$

$$y' = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}\,. \tag{12.36}$$

direct linear transformation (DLT) of perspective projection

It is called the *direct linear transformation* (DLT) of the perspective projection (Abdel-Aziz and Karara, 1971), since it directly relates the inhomogeneous coordinates of the object points with the measurable sensor coordinates of the image points of a straight line preserving or perspective camera.

The projection matrix can uniquely be derived (up to scale) if the exterior orientation and the five parameters of the interior orientation are given. The inverse task, deriving the parameters of the exterior and the interior orientation from a given projection matrix, will be discussed in Sect. 21, p. 498.

The projection matrix is proper if $|\mathsf{K}\boldsymbol{R}| > 0$ or, equivalently, if $K_{33} > 0$ and $m > -1$.

12.1.3.5 Mapping of Points at Infinity

Points at infinity $\mathcal{X}_\infty$ are homogeneously represented as $\mathbf{X}_\infty = [\boldsymbol{X}_\infty^\mathsf{T}, 0]^\mathsf{T}$, where $\boldsymbol{X}_\infty$ is the direction to the point $\mathcal{X}_\infty$. Their perspective projection into an image is given by

$$\mathcal{H}_\infty: \quad \mathbf{x}'_\infty = \mathsf{H}_\infty \boldsymbol{X}_\infty\,, \tag{12.37}$$

where the homography matrix is given by

$$\mathsf{H}_\infty = \mathsf{K}\boldsymbol{R}\,. \tag{12.38}$$

infinite homography

Obviously, the position $\boldsymbol{Z}$ of the camera has no influence on the image coordinates $\mathbf{x}'_\infty$. The matrix $\mathsf{H}_\infty = \mathsf{K}\boldsymbol{R}$ represents the *infinite homography* $\mathcal{H}_\infty$ of the projection $\mathcal{P}$. This relation can be used to determine the calibration matrix from observed stars, called star calibration.

12.1.3.6 The Normalized Camera

If the camera is calibrated and its rotation parameters are known, e.g., using an inertial measurement unit and a gyroscope, then we can derive the directions of the camera ray

in object space via

$$^{n}\mathbf{x}' = {}^{n}\mathsf{P}\mathbf{X} = \boldsymbol{X} - \boldsymbol{Z} \tag{12.39}$$

with the projection matrix

$${}^{n}\mathsf{P} = [I_3 \mid -\boldsymbol{Z}]\,. \tag{12.40}$$

A projection matrix having this form represents a *normalized camera*: Its rotation matrix and its calibration matrices are the unit matrix. It is an ideal camera with projection centre at $\boldsymbol{Z}$, with principal distance $c = 1$ pointing in the negative Z-direction, and with the ${}^{c}X{}^{c}Y$ axes of the camera parallel to the scene coordinate. A normalized camera is direction preserving as the vector ${}^{n}\mathbf{x}'$ points in the direction from $\mathcal{O}$ to $\mathcal{X}$. *normalized camera*

Obviously, the normalized camera rays ${}^{n}\mathbf{x}'$ can be derived from the observed sensor coordinates $\mathbf{x}'$ using (12.34), p. 472

$$^{n}\mathbf{x}' = -\text{sign}(c)\,(\mathsf{K}R)^{-1}\mathbf{x}' = -\text{sign}(c)\,R^{\mathsf{T}}\mathsf{K}^{-1}\mathbf{x}'\,. \tag{12.41}$$

This yields the correct sign if $|\mathsf{K}R| > 0$ and the point $\mathbf{x}'$ is positive.

12.1.3.7 Perspective Projection as Singular Projectivity

We can derive the projection equation $\mathbf{x}' = \mathsf{P}\mathbf{X}$ for the perspective projection of the 3D scene into a 2D image, a singular projectivity, solely using projective mappings, which we discussed in Sect. 6, p. 247.

We first represent the image point in a 3D coordinate system, where the not necessarily perpendicular $X'Y'$-coordinate axes span the image plane and the Z'-coordinate points away from the image plane. As the mapping from the scene to the image is assumed to be straight line-preserving, it has to be a 3D projectivity, $\mathbf{X}' = \mathsf{H}\mathbf{X}$, with some adequate 4×4 matrix H. This can be written explicitly as

$$\mathbf{X}' = \begin{bmatrix} U' \\ V' \\ W' \\ T' \end{bmatrix} = \mathsf{H}\mathbf{X} = \begin{bmatrix} \mathbf{A}_1^{\mathsf{T}} \\ \mathbf{A}_2^{\mathsf{T}} \\ \mathbf{A}_3^{\mathsf{T}} \\ \mathbf{A}_4^{\mathsf{T}} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{A}_1^{\mathsf{T}}\mathbf{X} \\ \mathbf{A}_2^{\mathsf{T}}\mathbf{X} \\ \mathbf{A}_3^{\mathsf{T}}\mathbf{X} \\ \mathbf{A}_4^{\mathsf{T}}\mathbf{X} \end{bmatrix}\,. \tag{12.42}$$

Now we force the projected points to lie on the plane $Z' = W'/T' = 0$. Therefore the third row $\mathbf{A}_3^{\mathsf{T}}$ of the homography needs to be zero, guaranteeing $W' = 0$. The resulting homography $\mathsf{H} = [\mathbf{A}_1, \mathbf{A}_2, \mathbf{0}, \mathbf{A}_4]^{\mathsf{T}}$ then is singular, indicating the mapping is a *singular projectivity*, thus not invertible anymore. However, as the image space should be two-dimensional the rank of this matrix needs to be 3.

We finally interpret the coordinates $X' = U'/T'$ and $Y' = V'/T'$ as coordinates in the image and set $x' = X'$ and $y' = Y'$. This is equivalent to omitting the third row in $\mathbf{X}'$ and H and leads to the projection

$$\mathbf{x}' = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} := \begin{bmatrix} U' \\ V' \\ T' \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^{\mathsf{T}} \\ \mathbf{A}_2^{\mathsf{T}} \\ \mathbf{A}_4^{\mathsf{T}} \end{bmatrix} \mathbf{X} =: \mathsf{P}\mathbf{X}\,, \tag{12.43}$$

with a homogeneous 3×4 projection matrix P with rank three. In the next section we show that if the left 3×3 submatrix of P is regular, the mapping is a perspective one, cf. (12.45), p. 475.

12.1.3.8 Properties of the Projection Matrix

The projection matrix P has a number of interesting properties (Das, 1949; Hartley and Zisserman, 2000) which can be derived from different representations of the matrix

$$\mathsf{P} = [p_{ij}] = \begin{bmatrix} \mathbf{A}^\mathsf{T} \\ \mathbf{B}^\mathsf{T} \\ \mathbf{C}^\mathsf{T} \end{bmatrix} = [\mathbf{x}'_{01}, \mathbf{x}'_{02}, \mathbf{x}'_{03} | \mathbf{x}'_{04}] = [\mathsf{A}|\mathbf{a}] = \mathsf{A}[I_3| - \boldsymbol{Z}]\,. \tag{12.44}$$

proper projection matrix

- The projection matrix is proper if $|\mathsf{A}| > 0$, since the camera rays ${}^c\mathbf{x}' = -\text{sign}(c)\,\mathsf{K}^{-1}\mathbf{x}'$ point from the projection centre to the scene points. Therefore we treat two projection matrices P_1 and P_2 as equivalent if they differ by a positive factor, cf. (9.34), p. 355. Both matrices K and R by construction have positive determinants, independent of the sign of the principal distance; $\mathsf{A} = \mathsf{K}R = \mathsf{H}_\infty$ also has a positive determinant by construction. Therefore it is recommended to always normalize the projection matrix such that $|\mathsf{A}| > 0$. In the following we assume the projection matrix is proper if not stated otherwise.

  This reasoning does *not* hold in case the directions of the xy-image coordinate axes are not the same as the directions of the XY-camera axes, as then the determinant of K may be negative.
- As $\mathsf{P}\mathbf{Z} = \mathbf{0}$ the projection centre is the null space of P.

principal camera planes

- The rows $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ represent three *camera planes* $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. They pass through the projection centre O and lead to image points with $u' = 0$, $v' = 0$, and $w' = 0$, respectively. They thus represent planes $\mathcal{A}$ and $\mathcal{B}$ going through the y'- and the x'-axis, respectively. Observe, the principal camera plane $\mathcal{C}$ is parallel to the image plane – see Fig. 12.10 – and the shaded plane in Fig. 11.9, p. 447.

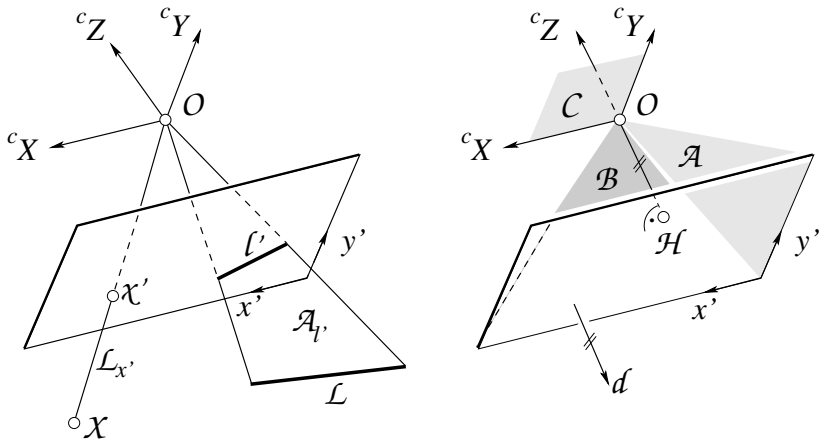


Fig. 12.10 Geometry of the single image. **Left**: 3D point $\mathcal{X}$ and 3D line $\mathcal{L}$, their images $\mathcal{X}'$ and $\mathcal{l}'$, and the corresponding viewing ray $\mathcal{L}_{x'}$ and viewing plane $\mathcal{A}_{l'}$. **Right**: Elements of the projection matrices P. The principal point $\mathcal{H}$ and the viewing direction is the normal of the plane $\mathcal{C}$. Both can be directly derived from P

For example, any object point $\mathcal{X}$ is an element of $\mathcal{A}$ if its u' coordinate is 0: $u' = \mathbf{A}^\mathsf{T}\mathbf{X} = 0$, which shows the plane $\mathcal{A}$ passes through the image line $u' = 0$, i.e., the y'-axis. The projection centre $O(\boldsymbol{Z})$ also yields $u' = \mathbf{A}^\mathsf{T}\mathbf{Z} = 0$; this is why $\mathcal{A}$ passes through O.

For a camera with a diagonal calibration matrix $\mathsf{K} = \text{Diag}([c, c(1+m), 1])$, the three planes are identical to the planes of the camera coordinate system.

- The columns $\mathbf{x}'_{0i}$ are the images of the four points $\mathbf{X}_{0i} = \boldsymbol{e}_i^{[4]}$ defining the object coordinate system, since $\mathbf{x}'_{0i} = \mathsf{P}\boldsymbol{e}_i^{[4]}$. In particular, $\mathcal{x}'_{04}$ is the image of the origin of the scene coordinate system. The two points $\mathcal{x}'_{01}$ and $\mathcal{x}'_{02}$ lie on the image of the horizon of the scene coordinate system.

- The projection centre can be determined from (12.44):

$$\boldsymbol{Z} = -\mathsf{A}^{-1}\mathbf{a} = \mathsf{H}_\infty^{-1}\mathbf{x}'_{04} \qquad \text{or} \qquad \mathbf{Z} = -\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} = \begin{bmatrix} -\mathsf{A}^*\mathbf{a} \\ |\mathsf{A}| \end{bmatrix} \tag{12.45}$$

  since $\mathbf{a} = -\mathsf{A}\boldsymbol{Z}$; the three principal planes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ of the camera pass through the projection centre. We chose the sign such that the homogeneous part Z_h of $\mathbf{Z}$ is positive for a proper camera matrix P. It is $Z_h = |\boldsymbol{A}_h, \boldsymbol{B}_h, \boldsymbol{C}_h| = |\mathsf{A}|$, cf. (5.103), p. 225. How to obtain the rotation and the calibration matrix for a given projection matrix is discussed in Sect. 21, p. 498.

  As the projection centre can be determined for all projection matrices with $\mathsf{A} = \mathsf{H}_\infty$ having full rank, such projectivities from $\mathbb{P}^3 \to \mathbb{P}^2$ are perspectivities (cf. Sect. 6.7, p. 284).
- The image point

$$\mathbf{x}' = U\mathbf{x}'_{01} + V\mathbf{x}'_{02} + W\mathbf{x}'_{03} + T\mathbf{x}'_{04} \tag{12.46}$$

  is the weighted centre of the columns $\mathbf{x}'_{0i}$, where the weights are the homogeneous coordinates $[U, V, W, T]^\mathsf{T}$ of the object point $\mathcal{X}$.
- The viewing direction $\mathbf{d}$ of the camera is orthogonal to the plane $\mathcal{C}(\mathbf{C})$, thus identical to its normal $\boldsymbol{C}_h$, and therefore the third row of A.

$$\mathbf{d}^\mathsf{T} = -[p_{31}, p_{32}, p_{33}]|\mathsf{A}|\,. \tag{12.47}$$

  The minus sign results from the fact that the viewing direction is in the $-{}^cZ$-direction. The factor $|\mathsf{A}|$ makes the viewing direction independent of a scaling of P with a negative factor.
- The principal point $\mathbf{x}'_H$ is the image of the point at infinity in the viewing direction, thus $\mathbf{x}'_H = \mathsf{H}_\infty\mathbf{d}$.

12.1.3.9 Uncertainty of the Projection Matrix

The uncertainty of the projection matrix can be expressed by the covariance matrix Σ_{pp} of the vector $\mathbf{p} = \mathrm{vec}\mathsf{P}$, where $\mathbf{p}$ is an element of the projective space $\mathbb{P}^{11}$. As the matrix is homogeneous and only depends on 11 parameters, the covariance matrix Σ_{pp} will have rank 11. This is the same situation as when representing the uncertainty, say, of an uncertain 2D point using homogeneous coordinates $\underline{\mathbf{x}} \in \mathbb{P}^2$ derived from its uncertain inhomogeneous coordinates $\underline{\boldsymbol{x}}$ (cf. Sect. 10.2.2.1, p. 366). Therefore we may represent an uncertain projection matrix using the reduced covariance matrix

$$\underline{\mathcal{P}}: \ \{\mathbb{E}(\underline{\mathsf{P}}), \Sigma_{p_r p_r}\}\,, \tag{12.48}$$

where the reduced vector of the elements of the projection matrix is

$$\boldsymbol{p}_r = J_r^\mathsf{T}(\boldsymbol{\mu}_p)\,\mathbf{p}\,, \qquad J_r(\boldsymbol{\mu}_p) = \mathsf{null}(\boldsymbol{\mu}_p^\mathsf{T}) \tag{12.49}$$

and therefore

$$\Sigma_{p_r p_r} = J_r^\mathsf{T}(\boldsymbol{\mu}_p)\,\Sigma_{\mathrm{pp}}\,J_r(\boldsymbol{\mu}_p)\,, \qquad \Sigma_{\mathrm{pp}} = J_r(\boldsymbol{\mu}_p)\,\Sigma_{p_r p_r}\,J_r^\mathsf{T}(\boldsymbol{\mu}_p)\,; \tag{12.50}$$

cf. (10.30), p. 371.

Let us now assume the pose and the calibration parameters are given together with their uncertainty. Then we can derive the uncertainty of the projection matrix by variance propagation.

For simplicity, let us assume that the uncertain calibration and rotation matrix are given by

$$\underline{\mathsf{K}}(\underline{\mathbf{k}}) = \begin{bmatrix} \underline{k}_1 & \underline{k}_2 & \underline{k}_4 \\ 0 & \underline{k}_3 & \underline{k}_5 \\ 0 & 0 & \underline{k}_6 \end{bmatrix}, \qquad \underline{R} = R(\underline{\Delta \boldsymbol{r}})\mathbb{E}(\underline{R}), \tag{12.51}$$

where the uncertain upper triangular matrix $\underline{\mathsf{K}}$ depends on the 6-vector $\mathbf{k}$. For generality we also include the element $\underline{k}_6 = \underline{K}_{33}$ in the vector, allowing it to have zero variance if the calibration matrix is Euclideanly normalized. As we can assume that the projection centre is not at infinity, we represent its uncertainty by the covariance matrix of the additive correction in $\underline{\boldsymbol{Z}} = \boldsymbol{\mu}_Z + \underline{\Delta \boldsymbol{Z}}$.

We now collect the stochastic elements in an uncertain 12-vector,

$$\underline{\boldsymbol{h}} = \begin{bmatrix} \underline{\boldsymbol{Z}} \\ \underline{\Delta \boldsymbol{r}} \\ \underline{\mathbf{k}} \end{bmatrix}, \quad \mathbb{D}(\underline{\Delta \boldsymbol{h}}) = \Sigma_{hh} = \begin{bmatrix} \Sigma_{ZZ} & \Sigma_{Z\Delta r} & \Sigma_{Z\mathrm{k}} \\ \Sigma_{\Delta rZ} & \Sigma_{\Delta r\Delta r} & \Sigma_{\Delta r\mathrm{k}} \\ \Sigma_{\mathrm{k}Z} & \Sigma_{\mathrm{k}\Delta r} & \Sigma_{\mathrm{kk}} \end{bmatrix}, \tag{12.52}$$

whose covariance matrix is assumed to be given. Correlations between all elements can then be encoded. With the Jacobian $J_{\mathrm{p}h}$ we derive the covariance matrix Σ_{pp} of $\mathbf{p}$.

For the Jacobian $J_{\mathrm{p}h}$ we use ${}^c\mathsf{P} = R[I_3| - \boldsymbol{Z}]$ and obtain the total differential

$$\mathrm{d}\mathsf{P} = \mathrm{d}\mathsf{K}\,{}^c\mathsf{P} + \mathsf{K}\mathrm{d}S(\boldsymbol{r})\,{}^c\mathsf{P} + \mathsf{A}[\mathsf{0}_{3\times 3}| - \mathrm{d}\boldsymbol{Z}],$$

where all matrices are evaluated at their mean. Therefore

$$\mathrm{d}\mathbf{p} = ({}^c\mathsf{P}^\mathsf{T} \otimes \mathsf{I}_3)\mathrm{d}\mathbf{k} + ({}^c\mathsf{P}^\mathsf{T} \otimes \mathsf{K})\mathrm{d}(\mathrm{vec}S(\boldsymbol{r})) + \mathrm{vec}[\mathsf{0}_{3\times 3}| - \mathsf{A}\mathrm{d}\boldsymbol{Z}]. \tag{12.53}$$

Exercise 12.25 The Jacobian then can be shown to be

$$J_{\mathrm{p}h} = \frac{\partial \mathbf{p}}{\partial \boldsymbol{h}} = \left[J_{\mathrm{p}Z} \,\middle|\, ({}^c\mathsf{P}^\mathsf{T} \otimes \mathsf{K})J_{Sr} \,\middle|\, ({}^c\mathsf{P}^\mathsf{T} \otimes \mathsf{I}_3)J_{\mathrm{Kk}} \right] \tag{12.54}$$

with the Jacobians

$$J_{\mathrm{p}Z} = \frac{\partial \mathbf{p}}{\partial \boldsymbol{Z}} = \begin{bmatrix} \mathsf{0}_{9\times 3} \\ -\mathsf{A} \end{bmatrix}, \quad J_{Sr} = \frac{\partial \mathrm{vec}S(\boldsymbol{r})}{\partial \boldsymbol{r}} = -\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}, \tag{12.55}$$

with $S_i = S\left(\boldsymbol{e}_i^{[3]}\right)$ and

$$J_{\mathrm{Kk}} = \frac{\partial \mathrm{vec}(\mathsf{K})}{\partial \mathbf{k}} = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \tag{12.56}$$

to be evaluated at the mean values.

12.1.4 Extending the Perspective Projection Model

The camera models discussed so far are straight line-preserving. Due to the central theorem of projective geometry we could write the mapping as a linear transformation in homogeneous coordinates, allowing us to fully exploit the power of projective geometry.

As this linear transformation is the most general projective mapping from object to image space, we discussed this model in detail in the previous section.

However, real cameras generally have some degree of lens distortion. Therefore we might need a more extended camera model, including parameters which describe these perturbing effects. Sometimes we call such errors *nonlinear*, referring to their property of not preserving straight lines. An example is given in Fig. 12.11, where the original image shows strong deviations from a perspective mapping.

The causes of nonlinear errors are manifold and include:

- lens distortion,
- nonplanarity of the sensor surface,
- refraction.

All these effects may lead to errors if only a perspective projection or an ideal spherical projection model is assumed. Fortunately, they can be determined without too much effort and can be used to undistort the images, resulting in rectified images which now are straight line-preserving. This is possible because the spatial rays reconstructed from images taken with a real camera still pass through a unique viewpoint. The straight line perturbing effects can therefore be interpreted as image deformations or image errors which, if known, can be eliminated. This is in contrast to images of line cameras generated by a linear motion perpendicular to the lines of the images where no such rectification is possible.

Due to refraction the projection ray is curved. The effect is small for short distances but may reach a few meters at the ground for large flying heights around 10 km. In contrast to most other systematic effects, which are related to the camera, refractive effects depend on the relative position of the camera and the scene points, thus require the pose of the camera to be known approximately. They vary with the weather conditions; however, assuming a standard atmosphere and correcting the image coordinates for this deterministic effect usually is sufficiently accurate. We refer the reader to Kraus (1993, Vol. 1, p. 188ff.).

Fig. 12.11 Original image with strong distortion and rectified image after removing distortion, showing straight lines in 3D, which are mapped to straight lines in the image (courtesy of S. Abraham)

In the following, we discuss how to extend the projection model to cope with moderate deviations from a straight line-preserving mapping using parametrized corrections to the image coordinates. The parameters are part of the interior orientation. How to model these deviations needs to be discussed in the context of determining all parameters of the projection, including parameters of the exterior and the interior orientation. We therefore discuss the modelling in the following Sect. 12.2, p. 489 together with the orientation of the single image.

Cameras with large nonlinear distortions are treated in Sect. 12.1.9, p. 484.

For cameras whose projection is close to that of a perspective model, it is useful to model the distortions as corrections of the image coordinates of a perspective camera, where the corrections depend on some parameters.

To make these corrections explicit, from now on we distinguish between the perspective, undistorted points $\bar{\mathcal{X}}'(\bar{\boldsymbol{x}}')$ and the observable, distorted points $\mathcal{X}'(\boldsymbol{x}')$. Additionally, the camera rays used up to now are undistorted and therefore are denoted by ${}^c\bar{\boldsymbol{x}}'$.

Thus we may determine the distorted coordinates from

$$\boldsymbol{x}' = \boldsymbol{f}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\,. \tag{12.57}$$

The observable point $\boldsymbol{x}'$ depends on the image coordinates ${}^i\bar{\boldsymbol{x}}'$ of the ideal image point $\bar{\mathcal{X}}'$ in a manner specified by the function $\boldsymbol{f}$ parametrized with $\boldsymbol{s}$ (see Fig. 12.7, p. 464). This modelling performs a correction of the coordinates resulting from an ideal camera. We will also have models which correct the observed and therefore distorted image coordinates $\boldsymbol{x}'$ to obtain the undistorted ones, see below.

The modelling of the distortions depends on whether they are small, e.g., when using normal or wide-angle lenses, or whether they are large, e.g., when using fish-eye optics. In all cases, it is convenient to apply these corrections to the reduced image coordinates ${}^i\boldsymbol{x}'$.

When the corrections are small, they are modelled in an additive manner, and otherwise in a multiplicative manner (cf. Sect. 12.1.9, p. 484). For the inhomogeneous sensor coordinates in a perspective camera we then have

$${}^i\boldsymbol{x}' = {}^i\bar{\boldsymbol{x}}' + \Delta\boldsymbol{x}'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q})\,. \tag{12.58}$$

The additive model (12.58) for sensor coordinates can be integrated easily into the projection relations. Starting from the camera rays ${}^c\mathbf{x}'$, this is done in three steps:

1. determine the ideal image coordinates ${}^i\bar{\mathbf{x}}' = {}^i\mathsf{K}_c\,{}^c\bar{\mathbf{x}}'$ using the calibration matrix ${}^i\mathsf{K}(c)$ (12.21), p. 469;
2. correct the ideal image coordinates for lens distortion (12.58) using the matrix

$$\mathsf{K}_{\Delta x}({}^i\bar{\boldsymbol{x}}', \boldsymbol{q}) = \begin{bmatrix} 1 & 0 & \Delta x'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q}) \\ 0 & 1 & \Delta y'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q}) \\ 0 & 0 & 1 \end{bmatrix}, \tag{12.59}$$

   yielding the image coordinates of the observable image point;
3. apply the affine transformation $\mathbf{x}' = \mathsf{K}_i\,{}^i\mathbf{x}'$ between the camera and the sensor system using the calibration matrix K_i from (12.29), p. 471.

This leads to a spatially varying calibration matrix,

$$\mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s}) = \mathsf{K}({}^i\bar{\boldsymbol{x}}', c, x'_H, y'_H, m, s, \boldsymbol{q}) = \mathsf{K}_i(x'_H, y'_H, m, s)\;\mathsf{K}_{\Delta x}({}^i\bar{\boldsymbol{x}}', \boldsymbol{q})\;{}^i\mathsf{K}(c)\,. \tag{12.60}$$

The parameter vector $\boldsymbol{s}$ in the complete calibration matrix $\mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})$ contains the five parameters $[c, x'_H, y'_H, m, s]$ of the perspective calibration matrix K and the further parameters $\boldsymbol{q}$ (see Fig. 12.7, p. 464).

An explicit expression for $\mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})$ contains terms with the corrections $\Delta{}^i x'$ and $\Delta{}^i y'$ multiplied with the shear s and the scale difference m. In many cases s and m are small. Omitting second-order terms, we hence obtain the approximation

Exercise 12.18

$$\mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s}) \approx \begin{bmatrix} c & cs & x'_H + \Delta x'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q}) \\ 0 & c(1+m) & y'_H + \Delta y'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q}) \\ 0 & 0 & 1 \end{bmatrix} = \mathsf{K}_{\Delta x}({}^i\bar{\boldsymbol{x}}', \boldsymbol{q})\,\mathsf{K}\,. \tag{12.61}$$

Thus the addition of the corrections $\Delta\boldsymbol{x}'({}^i\bar{\boldsymbol{x}}', \boldsymbol{q})$ can be interpreted as letting the principal point vary with the position of the image point.

The general mapping from object into image space is now

$$\mathbf{x}' = \mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\;\boldsymbol{R}[\boldsymbol{I}_3|-\boldsymbol{Z}]\mathbf{X} = \mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\;[\boldsymbol{I}_3|\mathbf{0}]\;\mathsf{M}^{-1}\mathbf{X}\,. \tag{12.62}$$

With the general projection matrix

$$\mathsf{P}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s}) = \mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\, R[I_3| - \boldsymbol{Z}] = \mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\, [I_3|\mathbf{0}]\, \mathsf{M}^{-1}\,, \tag{12.63}$$

we have the compact form of the perspective mapping with distortions,

$$\mathbf{x}' = \mathsf{P}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\, \mathbf{X}\,. \tag{12.64}$$

Due to (12.62) the prediction requires three steps. We first determine the ray direction ${}^c\bar{\mathbf{x}}'$, then the reduced image coordinates using (12.19), p. 469, and finally apply the distortions to the ray direction:

$$1)\quad {}^c\bar{\mathbf{x}}' = {}^c\mathsf{P}\mathbf{X} \qquad 2)\quad {}^i\bar{\mathbf{x}}' = {}^i\mathsf{K}^{-1}\, {}^c\bar{\mathbf{x}}' \qquad 3)\quad \mathbf{x}' = \mathsf{K}({}^i\bar{\boldsymbol{x}}', \boldsymbol{s})\, {}^c\bar{\mathbf{x}}'\,. \tag{12.65}$$

If the points are far apart in the image, these three steps need to be calculated for every point.

The inverse relation of (12.64), namely determining the projection ray or projection line $\mathbf{L}_{x'}$, where $\mathbf{X}$ is located for given image coordinates $\mathbf{x}'$, is discussed in Sect. 12.1.7.

12.1.5 Overview on the Different Camera Models

Figure 12.7, p. 464 and the following Table 12.1 present the different camera models together with their main characteristics.

Table 12.1 Camera models: perspective projection with distortion. The coordinates $\mathbf{x}'$ of an observed point generally depend on coordinates $\bar{\mathbf{x}}'$ of the ideal point and additional parameters $\boldsymbol{s}$. Name of camera model and type of projection, calibration matrix with parameters of interior orientation, number N_s of additional parameters. The unit camera, when used to derive spherically normalized camera rays, models a spherical projection

camera model	interior orientation (calibration matrix K)	N_s
unit camera perspective proj. central proj. normalized camera: $R = I_3$	${}^c\mathsf{K} = I_3$ (12.17), p. 468	0
ideal camera perspective proj.	${}^i\mathsf{K} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (12.21), p. 469	1
camera with Euclidean sensor perspective proj.	${}^e\mathsf{K} = \begin{bmatrix} c & 0 & x'_H \\ 0 & c & y'_H \\ 0 & 0 & 1 \end{bmatrix}$ 12.1.2.4, p. 464	3
camera with affine sensor, perspective proj.	$\mathsf{K} = \begin{bmatrix} c & cs & x'_H \\ 0 & c(1+m) & y'_H \\ 0 & 0 & 1 \end{bmatrix}$ (12.31), p. 471	5
perspective proj. with distortion	$\mathsf{K}({}^i\bar{\mathbf{x}}', \boldsymbol{s}) \approx \begin{bmatrix} c & cs & x'_H + \Delta x'({}^i\bar{\boldsymbol{x}}', \boldsymbol{p}) \\ 0 & c(1+m) & y'_H + \Delta y'({}^i\bar{\boldsymbol{x}}', \boldsymbol{s}) \\ 0 & 0 & 1 \end{bmatrix}$ (12.61), p. 478	> 5

12.1.6 Mapping of Straight 3D Lines

All camera models consist of expressions for the projection of scene points into the sensor. For many applications, we also need explicit expressions for the projection of 3D lines. These may be straight 3D lines, general curved lines, contour lines, or conics. They may be derived for the projection relations for 3D points. We start with the mapping of straight lines, which are useful especially when analysing images of urban areas, since straight line segments can be extracted automatically.

We assume the projective model (12.33) to hold and thus assume the image coordinates to be corrected at least for nonlinear distortions.

We first derive an explicit expression for the projection of a 3D line into the image, which gives insight into the geometry of the projection. The expression is quadratic in the elements of the projection matrix P. Therefore, we also derive the constraint for an observed image line segment to lie on the projection of a 3D line which is linear in the elements of P.

12.1.6.1 Perspective Projection of a 3D Line

For the derivation, we assume the 3D line $\mathcal{L}$ to be represented by arbitrary, distinct points $\mathcal{X}$ and $\mathcal{Y}$,

$$\mathbf{L} = \mathbf{X} \wedge \mathbf{Y}\,, \tag{12.66}$$

where we treat the line as an infinite entity. The projected image points are assumed to be

$$\mathbf{x}' = \mathsf{P}\mathbf{X} = \begin{bmatrix} \mathbf{A}^\mathsf{T} \\ \mathbf{B}^\mathsf{T} \\ \mathbf{C}^\mathsf{T} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{A}^\mathsf{T}\mathbf{X} \\ \mathbf{B}^\mathsf{T}\mathbf{X} \\ \mathbf{C}^\mathsf{T}\mathbf{X} \end{bmatrix}, \qquad \mathbf{y}' = \mathsf{P}\mathbf{Y} = \begin{bmatrix} \mathbf{A}^\mathsf{T} \\ \mathbf{B}^\mathsf{T} \\ \mathbf{C}^\mathsf{T} \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{A}^\mathsf{T}\mathbf{Y} \\ \mathbf{B}^\mathsf{T}\mathbf{Y} \\ \mathbf{C}^\mathsf{T}\mathbf{Y} \end{bmatrix}, \tag{12.67}$$

where $\mathbf{A}_i$, etc., are 4-vectors. Thus the image line is

$$\mathbf{l}' = \mathbf{x}' \times \mathbf{y}' = \mathsf{P}\mathbf{X} \times \mathsf{P}\mathbf{Y} = \begin{bmatrix} \mathbf{B}^\mathsf{T}\mathbf{X}\,\mathbf{C}^\mathsf{T}\mathbf{Y} - \mathbf{B}^\mathsf{T}\mathbf{Y}\,\mathbf{C}^\mathsf{T}\mathbf{X} \\ \mathbf{C}^\mathsf{T}\mathbf{X}\,\mathbf{A}^\mathsf{T}\mathbf{Y} - \mathbf{C}^\mathsf{T}\mathbf{Y}\,\mathbf{A}^\mathsf{T}\mathbf{X} \\ \mathbf{A}^\mathsf{T}\mathbf{X}\,\mathbf{B}^\mathsf{T}\mathbf{Y} - \mathbf{A}^\mathsf{T}\mathbf{Y}\,\mathbf{B}^\mathsf{T}\mathbf{X} \end{bmatrix}. \tag{12.68}$$

With the relation (7.61), p. 304

$$\mathbf{X}^\mathsf{T}(\mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T})\mathbf{Y} = \overline{(\mathbf{A} \cap \mathbf{B})}^\mathsf{T}(\mathbf{X} \wedge \mathbf{Y}) = (\mathbf{A} \wedge \mathbf{B})^\mathsf{T}(\mathbf{X} \wedge \mathbf{Y})\,, \tag{12.69}$$

this can be transformed to

$$\mathbf{l}' = \begin{bmatrix} (\mathbf{B} \wedge \mathbf{C})^\mathsf{T} \\ (\mathbf{C} \wedge \mathbf{A})^\mathsf{T} \\ (\mathbf{A} \wedge \mathbf{B})^\mathsf{T} \end{bmatrix} \mathbf{L}\,, \tag{12.70}$$

or, using (7.38), p. 301 with the 3×6 *projection matrix for lines*,

$$\mathsf{Q} = \begin{bmatrix} (\mathbf{B} \wedge \mathbf{C})^\mathsf{T} \\ (\mathbf{C} \wedge \mathbf{A})^\mathsf{T} \\ (\mathbf{A} \wedge \mathbf{B})^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^\mathsf{T}\,\mathsf{\Pi}^\mathsf{T}(\mathbf{B}) \\ \mathbf{A}^\mathsf{T}\,\mathsf{\Pi}^\mathsf{T}(\mathbf{C}) \\ \mathbf{B}^\mathsf{T}\,\mathsf{\Pi}^\mathsf{T}(\mathbf{A}) \end{bmatrix}, \tag{12.71}$$

we have

$$\mathbf{l}' = \mathsf{Q}\mathbf{L}\,. \tag{12.72}$$

Thus also the projection of a 3D line can be realized by a matrix vector product.

12.1.6.2 Spherical Projection of a 3D Line

The spherical projection of a 3D line $\mathcal{L}({}^c\mathbf{L})$ with ${}^c\mathbf{L}^\mathsf{T} = [{}^c\boldsymbol{L}_h^\mathsf{T}{}^c\boldsymbol{L}_0^\mathsf{T}]$ can easily be derived. First we observe: the projecting plane has normal ${}^c\boldsymbol{L}_0$ expressed in the camera system, cf. Sect. 5.4.2, p. 217, and Fig. 12.12. This normal is at the same time the representation of the line l' in the camera system, see Fig. 5.2, p. 200. Therefore we just need to select the sub-vector ${}^c\boldsymbol{L}_0$ from the Plücker coordinates $\mathbf{L}$ of the 3D line and obtain

$$ {}^c\mathbf{l}' = {}^c\boldsymbol{L}_0 = [\mathsf{0}_{3\times3} \mid \mathsf{I}_3] \begin{bmatrix} {}^c\boldsymbol{L}_h \\ {}^c\boldsymbol{L}_0 \end{bmatrix} = [\mathsf{0}_{3\times3} \mid \mathsf{I}_3]\, {}^c\mathbf{L}\,. \tag{12.73} $$

We therefore have the projection matrix for 3D lines,

$$ {}^c\mathsf{Q}_c = [\mathsf{0}_{3\times3} \mid \mathsf{I}_3]\,, \tag{12.74} $$

equivalent to ${}^c\mathsf{P}_c = [\mathsf{I}_3|\mathbf{0}]$ in (12.12), p. 467.

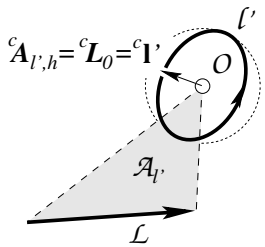


Fig. 12.12 Spherical projection of a 3D line $\mathcal{L}$ into a camera with centre O represented by a viewing sphere (dashed circle). The oriented 3D line $\mathcal{L}$ is projected into the oriented image line l', represented by the oriented circle. The normal $\boldsymbol{A}_{l',h}$ of the oriented projection plane $\mathcal{A}_{l'}$ is identical to the normal of the oriented circle with its homogeneous coordinates $\boldsymbol{A}_{l',h} = {}^c\mathbf{l}'$ observable in the camera coordinate system (not shown) as ${}^c\boldsymbol{A}_{l'h} = {}^c\boldsymbol{L}_0 = {}^c\mathbf{l}'$

We now use the motion matrix M_L for moving 3D lines from the object system into the camera system, cf. (6.54), p. 259, and its inverse for transforming the line coordinates $\mathbf{L}$ from object space into the line coordinates ${}^c\mathbf{L}$ in the camera system. Taking into account that the rotation in M is assumed to be R^T, cf. (12.5), p. 466, we have

$$ \mathsf{M}_L = \begin{bmatrix} R^\mathsf{T} & \mathsf{0}_{3\times3} \\ S(\boldsymbol{Z})R^\mathsf{T} & R^\mathsf{T} \end{bmatrix}, \qquad \mathsf{M}_L^{-1} = \begin{bmatrix} R & \mathsf{0}_{3\times3} \\ RS^\mathsf{T}(\boldsymbol{Z}) & R \end{bmatrix}. \tag{12.75} $$

Using M_L^{-1} for the coordinate transformation, we obtain from (12.73) the spherical projection for 3D lines given in the scene coordinate system,

$$ {}^c\mathbf{l}' = [\mathsf{0}_{3\times3} \mid \mathsf{I}_3]\mathsf{M}_L^{-1}\,\mathbf{L} = [RS^\mathsf{T}(\boldsymbol{Z}) \mid R]\,\mathbf{L} = R[-S(\boldsymbol{Z}) \mid \mathsf{I}_3]\mathbf{L} = {}^c\mathsf{Q}\mathbf{L}\,. \tag{12.76} $$

The same result can be derived using the spherical projection of two line points. Exercise 12.17

12.1.6.3 Properties of the Projection Matrix for Straight 3D Lines

The projection matrix for 3D lines (12.71), (12.76) can be written in several ways,

$$ \mathsf{Q} = [q_{ij}] = \begin{bmatrix} \overline{\mathbf{L}}_1^\mathsf{T} \\ \overline{\mathbf{L}}_2^\mathsf{T} \\ \overline{\mathbf{L}}_3^\mathsf{T} \end{bmatrix} = [\mathsf{l}'_{01}, \mathsf{l}'_{02}, \mathsf{l}'_{03} | \mathsf{l}'_{04}, \mathsf{l}'_{05}, \mathsf{l}'_{06}] = [\mathsf{Y}|\mathsf{N}] = \mathsf{A}^\circlearrowright\,[-S(\boldsymbol{Z}) \mid \mathsf{I}_3]\,, \tag{12.77} $$

with its elements, rows, columns, partitioning, and its relation to the projection matrix for points using $\mathsf{A}^\circ = (\mathsf{K}\boldsymbol{R})^\circ = \mathsf{K}^\circ\boldsymbol{R}$. It has the following properties:

1. We assume the 3D lines to be elements of an oriented projective space $\mathbb{T}^5$, in order to exploit the orientation of the 3D lines. Similarly, we want to treat image lines as oriented, e.g., assuming the area right of the line to be brighter than the area left of the line. Thus we have the mapping now referring to lines,
$$\mathcal{P} : \mathbb{T}^5 \mapsto \mathbb{T}^2 \qquad \mathbf{l}' = \mathsf{Q}\mathbf{L}\,. \tag{12.78}$$
2. For a given projection matrix $\mathsf{P} = \mathsf{K}\boldsymbol{R}[\boldsymbol{I}_3|-\boldsymbol{Z}]$ for points, the corresponding projection matrix for lines is
$$\mathsf{Q} = (\mathsf{K}\boldsymbol{R})^\circ\,[-\boldsymbol{S}(\boldsymbol{Z}) \mid \boldsymbol{I}_3]\,. \tag{12.79}$$
This follows from (12.76) and $\mathbf{x}' = \mathsf{K}\,{}^c\bar{\mathbf{x}}'$; thus, with (6.46), p. 258, we get $\mathbf{l}' = \mathsf{K}^\circ\,{}^c\overline{\mathbf{l}}'$.
3. When partitioning $\mathsf{Q} = [\mathsf{Y}|\mathsf{N}]$ into two 3×3 matrices, the skew matrix of the nonhomogeneous coordinates of the projection centre $\boldsymbol{Z}$ results from
$$\boldsymbol{S}(\boldsymbol{Z}) = -\mathsf{N}^{-1}\mathsf{Y}\,. \tag{12.80}$$

proper projection matrix for lines

4. The projection matrix Q for lines is called proper if the normal of the projection plane $\mathcal{OL}$ is identical to the normal of the plane $\mathcal{O}\ell'$. This is guaranteed if the determinant of the right 3×3 matrix $\mathsf{A}^{*\mathsf{T}}$ is positive or if
$$\text{sign}(|\mathsf{N}|) = \text{sign}(|\mathsf{A}|) > 0\,. \tag{12.81}$$
5. The three rows $\overline{\mathbf{L}}_k, k = 1, 2, 3$, of the projection matrix Q in (12.71) are the dual coordinates of the intersection lines of the camera planes, e.g., we have $\mathcal{L}_1 = \mathcal{B} \cap \mathcal{C}$.
6. The image line can also be expressed as the weighted sum of the columns of Q, where the weights are the homogeneous coordinates L_i of the line $\mathcal{L}(\mathbf{L})$:
$$\mathbf{l}' = \sum_{i=1}^{6} L_i \mathbf{l}'_{0i}\,. \tag{12.82}$$
7. With the columns $\mathbf{x}'_{0i}$ of P we can show
$$\mathsf{Q} = [\mathbf{x}'_{04} \times \mathbf{x}'_{01}, \mathbf{x}'_{04} \times \mathbf{x}'_{02}, \mathbf{x}'_{04} \times \mathbf{x}'_{03}, \mathbf{x}'_{02} \times \mathbf{x}'_{03}, \mathbf{x}'_{03} \times \mathbf{x}'_{01}, \mathbf{x}'_{01} \times \mathbf{x}'_{02}]\,,$$
i.e., the index pairs of the cross products are the same as the ones that generate the 3D line coordinates from two points, cf. (7.36), p. 300. Observe, the last column, $\mathbf{q}'_{06} = \mathbf{x}'_{01} \times \mathbf{x}'_{02}$, is the image of the horizon $\mathbf{L} = [0, 0, 0, 0, 0, 1]^\mathsf{T}$.

These properties of the projection matrix Q for 3D lines are in full analogy to the interpretation of the projection matrix P for 3D points, cf. Sect. 12.1.3.8.

12.1.7 Inverse Relations

The inverse relations serve to infer object information from image information i.e., for back projection of image features. Though we will discuss methods for inverting the perspective in Sect. 12.3, p. 523, we need the projection lines as back projected points, and projection planes as back projected lines, for deriving the image of quadrics in Sect. 12.1.8, p. 484.

Projection Plane. Given a possibly oriented image line $\mathbf{l}' = [a', b', c']^\mathsf{T}$ and a projection matrix P, the *projection plane* $\mathcal{A}_{l'}$ passing through the line ℓ' and the projection centre O

is given by

$$\mathbf{A}_{l'} = \mathsf{P}^\mathsf{T}\mathbf{l}' = a'\mathbf{A} + b'\mathbf{B} + c'\mathbf{C}\,. \tag{12.83}$$

This is because for all points $\mathcal{X}$ on $\mathcal{A}_{l'}$, thus satisfying $\mathbf{A}_{l'}^\mathsf{T}\mathbf{X} = 0$, we also have $\mathbf{l}'^\mathsf{T}\mathbf{x}' = 0$ as $\mathbf{A}_{l'}^\mathsf{T}\mathbf{X} = (\mathbf{l}'^\mathsf{T}\mathsf{P})\mathbf{X} = \mathbf{l}'^\mathsf{T}(\mathsf{P}\mathbf{X}) = \mathbf{l}'^\mathsf{T}\mathbf{x}'$. The projecting plane $\mathcal{A}_{l'}$ passes through O since its coordinate vector is a linear combination of the coordinate planes $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, which pass through O.

If the projection matrix $\mathsf{P} = \mathsf{K}R[I_3| - \boldsymbol{Z}]$ is proper, i.e., $|\mathsf{K}R| > 0$, the normal of the plane $\mathcal{A}_{l_x}$ is identical to the normal of the plane $O\mathcal{l}'$ for $\boldsymbol{A}_h = R^\mathsf{T}\mathsf{K}^\mathsf{T}\mathbf{l}'$ or $\mathbf{l}' = \mathsf{K}^{-\mathsf{T}}R\boldsymbol{A}_h$, thus $\mathbf{l}' = \mathsf{K}^{-\mathsf{T}}\,{}^c\boldsymbol{A}_h$.

Using (12.83) allows us to express the correspondence of the image of a 3D line $\mathcal{L}$ with the observed image line $\mathcal{l}'$ as a constraint which is linear in the elements of the projection matrix P: the 3D line $\mathcal{L}$ needs to lie in the projection plane $\mathcal{A}_{l'}$. With (7.64), p. 305 we have

$$\mathsf{\Gamma}(\mathbf{L})\mathsf{P}^\mathsf{T}\mathbf{l}' = \mathbf{0} \tag{12.84}$$

which are two independent constraints.

Projection Line. Analogously, given an image point $\mathcal{x}'$ and a projection matrix Q for lines, the *projection line* $\mathcal{L}_{x'}$ passing through the point $\mathcal{x}'$ and the projection centre O is given by

$$\mathbf{L}_{x'} = \overline{\mathsf{Q}}^\mathsf{T}\mathbf{x}' = (u'(\mathbf{B}\cap\mathbf{C}) + v'(\mathbf{C}\cap\mathbf{A}) + w'(\mathbf{A}\cap\mathbf{B}))\,, \tag{12.85}$$

where

$$\overline{\mathsf{Q}} = \mathsf{Q}D \tag{12.86}$$

is the dual of Q, cf. (5.116), p. 227. This is because for all lines $\mathcal{M}$ passing through $\mathcal{L}_{x'}$, thus fulfilling $\mathbf{L}_{x'}^\mathsf{T}\overline{\mathbf{M}} = 0$, we also have $\mathbf{x}'^\mathsf{T}\mathbf{m}' = 0$ as $\mathbf{L}_{x'}^\mathsf{T}\overline{\mathbf{M}} = (\mathbf{x}'^\mathsf{T}\overline{\mathsf{Q}})\overline{\mathbf{M}} = \mathbf{x}'^\mathsf{T}(\overline{\mathsf{Q}\mathbf{M}}) = \mathbf{x}'^\mathsf{T}\mathbf{m}'$. Observe, dualizing of the matrix Q is achieved by dualizing its three rows, which can be interpreted as Plücker vectors of 3D lines; therefore, the multiplication with D from right.

The projection line $\mathbf{L}_{x'}$ is the weighted sum of the coordinate lines $\mathbf{B}\cap\mathbf{C}$, $\mathbf{C}\cap\mathbf{A}$, and $\mathbf{A}\cap\mathbf{B}$ of the camera coordinate system, where the weights are the homogeneous coordinates $[u', v', w']$ of the image point $\mathbf{x}'$.

Observe, the projection line $\mathcal{L}_{x'}$ of an image point $\mathbf{x}' = \mathsf{P}\mathbf{X}$ generated from the object point $\mathcal{X}$ is given by

$$\mathbf{L}_{x'} = \overline{\mathsf{Q}}^\mathsf{T}\mathbf{x}' = \overline{\mathsf{Q}}^\mathsf{T}\mathsf{P}\mathbf{X} = \mathsf{\Pi}(\mathbf{Z})\mathbf{X} = \mathbf{Z}\wedge\mathbf{X}\,, \tag{12.87}$$

which is true since the projection line is the join of the projection centre $O(\boldsymbol{Z})$ with the object point $\mathcal{X}$. Thus we have

$$\overline{\mathsf{Q}}^\mathsf{T}\mathsf{P} = \mathsf{\Pi}(\mathbf{Z})\,, \tag{12.88}$$

which verifies (12.79). The relative scale between the two sides is $\lambda = 1$ if the homogeneous coordinates $\mathbf{Z}$ of the projection centre are taken from (12.45), p. 475, right. Exercise 12.10

As the sign of $\mathbf{x}'$ is arbitrary, we will not necessarily obtain the projection ray $\mathbf{L}_{x'}$ with the correct direction from the projection centre to the scene point. However, if the projection is proper, i.e., $|\mathsf{K}| > 0$ and ${}^c\mathbf{x}'$ represents the ray direction ${}^c\mathbf{x}' = -\text{sign}(c)\mathsf{K}^{-1}\mathbf{x}'$, then the direction of $\mathcal{L}_{x'}$ has the correct sign. Due to (12.88), for $\mathbf{L}_{x'}$ to have the correct direction, only the two projection matrices $\mathsf{P} = [\mathsf{A}|\mathbf{a}]$ and $\mathsf{Q} = [\mathsf{Y}|\mathsf{N}]$ need to be consistent w.r.t. their sign, i.e., $\text{sign}(|\mathsf{A}|) = \text{sign}(|\mathsf{N}|)$, cf. (12.77), p. 481.

The derived relations for the projection of points and lines and the inversions are collected in Table 12.2.

Table 12.2 Projection of points and lines from 3D to 2D, projection ray and plane. In case the projection matrices P and Q are proper, the mappings are orientation preserving

operation/entity	point	Equation	line	Equation
projection	$\mathbf{x}' = \mathsf{P}\mathbf{X}$	(12.33)	$\mathbf{l}' = \mathsf{Q}\mathbf{L}$	(12.72)
projection ray, plane	$\mathbf{L}_{x'} = \overline{\mathsf{Q}}^{\mathsf{T}}\mathbf{x}'$	(12.85)	$\mathbf{A}_{l'} = \mathsf{P}^{\mathsf{T}}\mathbf{l}'$	(12.83)

12.1.8 Mapping of Curved 3D Lines

Mapping of Contours of Quadrics. The image of quadrics generally covers a region. The boundary line of this region is the image of the apparent contour of the surface. We now show that this boundary line is a conic.

Let the quadric $\mathcal{Q}$ be given in its dual form, i.e., by the set $\mathcal{A}$ of its tangent planes:

$$\mathbf{A}^{\mathsf{T}}\mathsf{Q}^{\circ}\mathbf{A} = 0\,. \tag{12.89}$$

The planes intersect the image plane in a set of lines which are tangents at the boundary line, thus are projection planes. With the projection planes $\mathbf{A}_{l'} = \mathsf{P}^{\mathsf{T}}\mathbf{l}'$ we thus obtain the relation

$$\mathbf{l}'^{\mathsf{T}}\mathsf{P}\mathsf{Q}^{\circ}\mathsf{P}^{\mathsf{T}}\mathbf{l}' = 0\,, \tag{12.90}$$

which represents a conic $\mathcal{C}$ in dual form:

$$\mathsf{C}^{\circ} = \mathsf{P}\mathsf{Q}^{\circ}\mathsf{P}^{\mathsf{T}}\,. \tag{12.91}$$

Since a conic in 3D can be represented as a quadric, where one semi-axis is zero – which is equivalent to a singular dual quadric – its image can be determined using (12.91).

Exercise 12.21

Mapping of General 3D Curves Represented with NURBS. The mapping of general 3D curves cannot be expressed easily. If, however, the 3D curve is approximated by a nonuniform rational B-Spline, called NURBS, the mapping becomes easy, a unique property of NURBS (cf. Piegl and Tiller, 1997).

The homogeneous coordinates of a point $\mathbf{X}(u)$ on a 3D curve, parametrized by u, can be represented as a weighted sum,

$$\mathbf{X}(u) = \sum_{i=0}^{I} N_{i,p}(u) w_i \mathbf{X}_i\,. \tag{12.92}$$

Here, $\mathbf{X}_i, i = 0, ..., I$, are the control points which define the shape of the curve; their weights w_i control their influence on the curve, and $N_{i,p}(u)$ are basis functions, B-splines of order p, which perform the interpolation.

Projecting all control points into the image $\mathbf{x}'_i = \mathsf{P}\mathbf{X}_i, i = 0, ..., I$, leads to an explicit representation of the 2D curve,

$$\mathbf{x}'(u) = \sum_{i=0}^{I} N_{i,p}(u) w_i \mathbf{x}'_i\,. \tag{12.93}$$

12.1.9 Nonstraight Line-Preserving Mappings

Up to now we have modelled image distortions as corrections to the image coordinates. This implicitly assumes that the distortions are reasonable small, say below 10% of the image size. In this section we describe models for cameras with a central projection where (1) the lens distortions are large or (2) the field of view is larger than 180°.

12.1.9.1 Cameras with Central Projection

Cameras with a single viewing point realize a central projection, cf. Sect. 12.1.3.2, p. 467. Most of these cameras realize the imaging process with an optical system which conceptually is rotationally symmetric. This rotational symmetry includes the sensor plane, which therefore is perpendicular to the symmetry axis, with the intersection point of the symmetry axis with the sensor plane as its principal point (Fig. 12.13).

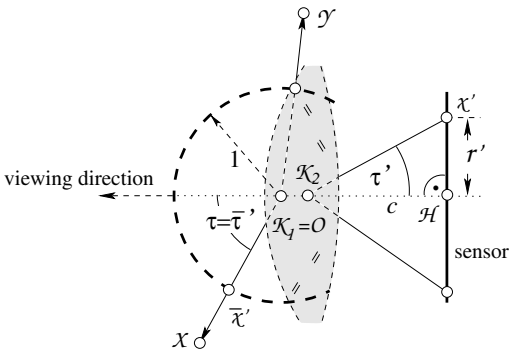


Fig. 12.13 Model of a spherical camera. The viewing rays to the scene points $\mathcal{X}$ intersect in the projection centre $O = \mathcal{K}_1$. The angle $\bar{\tau}'$ between the symmetry axis and the direction $\mathcal{K}_1\bar{\mathcal{X}}'$ to the ideal image point $\bar{\mathcal{X}}'$ on the viewing sphere (dashed) is identical to the angle τ between the symmetry axis and the direction to the scene point $\mathcal{X}$, thus $\tau = \bar{\tau}'$. It may significantly deviate from the angle τ' between the symmetry axis and the direction $\mathcal{K}_2\mathcal{X}'$ to the observable image point $\mathcal{X}'$ lying at a radial distance r' from the principal point $\mathcal{H}$. For a perspective camera, we have the relation $r' = c \tan \tau$. Observe, this optics in principle allows us to observe points with a ray pointing away from the viewing direction, such as to point $\mathcal{Y}$

In a first step we need to model the mapping of the angle $\tau = \bar{\tau}'$ between the optical axis and the camera ray to the radial distance r' of an image point $\mathcal{X}'$ from the principal point $\mathcal{H}$:

$$r' = r'(\tau)\,. \tag{12.94}$$

In addition to this radial distortion model (12.182), we need a calibration matrix $\mathsf{K}({}^c\bar{\mathbf{x}}')$ which allows us to compensate for an affine transformation between the sensor system S_s and the Cartesian system S_i centred at the principal point, and for additional image deformations $\Delta\boldsymbol{x}'({}^c\bar{\mathbf{x}}', \boldsymbol{q})$. Finally, there may be tangential distortions, which can be modelled as in perspective cameras.

Rotating Slit Cameras. Cameras with a rotating line sensor are designed to obtain panorama images. They realize a central projection if the rotation axis passes through the projection centre of the line sensor. If the sensor line intersects the rotation axis, and depending on the angle between the line sensor and the rotation axis, we obtain a projection onto a cone or a cylinder, with the rotation axis as its axis. The cone or the cylinder can be spread in a plane. As the intersection of the projection plane of a 3D line with a cone or a cylinder generally is not a straight line, we achieve a central projection which is not straight line-preserving. The projection can also be realized by an analogue pin-hole camera, in case the film is shaped like a cone or a cylinder, see Fig. 12.14.

Fish-Eye Cameras. Fish-eye lenses have a large viewing angle even beyond 180°. In order to realize such large fields of view while still guaranteeing good resolution, we need optical systems with a large number of lenses. Generally the camera rays do not intersect in a single point but meet in a small area within the lens system. Therefore, strictly speaking, fish-eye lenses do not realize a central projection. However, if the distance to the scene is not too small, the optical system can be modelled by a central projection with sufficient accuracy. This is what we assume in the following.

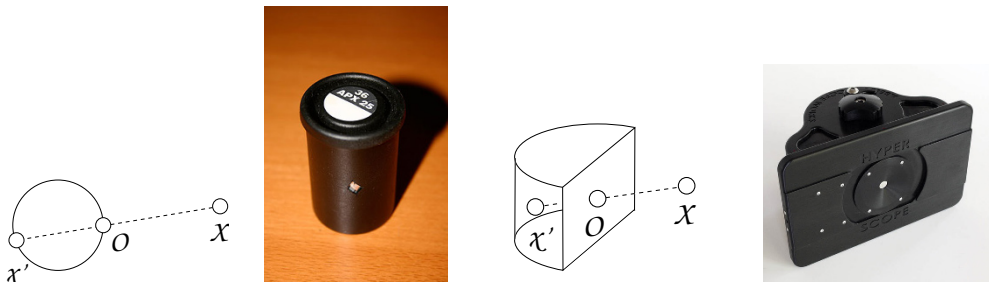


Fig. 12.14 Central nonstraight line-preserving pinhole cameras. (a): The idealized eye or a pinhole camera with a cylindrical sensor area. (b): Pinhole camera with a full cylinder as sensor area (see http://thepinholecamera.com/tipstricks_solargraphy.php, Copyright 2009 Brian J. Krummel) (c): Pin hole camera with a half cylinder as sensor area. (d): Cameras with half cylinder as sensor area (see http://petapixel.com/2012/10/25/hyperscope-a-custom-built-cylindrical-pinhole-camera-for-roll-film/), copyright by Matt Abelson, photographer

In order to map the viewing range onto a sensor, in all cases large distortions i.e., deviations from a straight line-preserving mapping, are to be accepted.

The following models, shown in Fig. 12.15, have been proposed in the literature (cf. Abraham and Förstner, 2005):

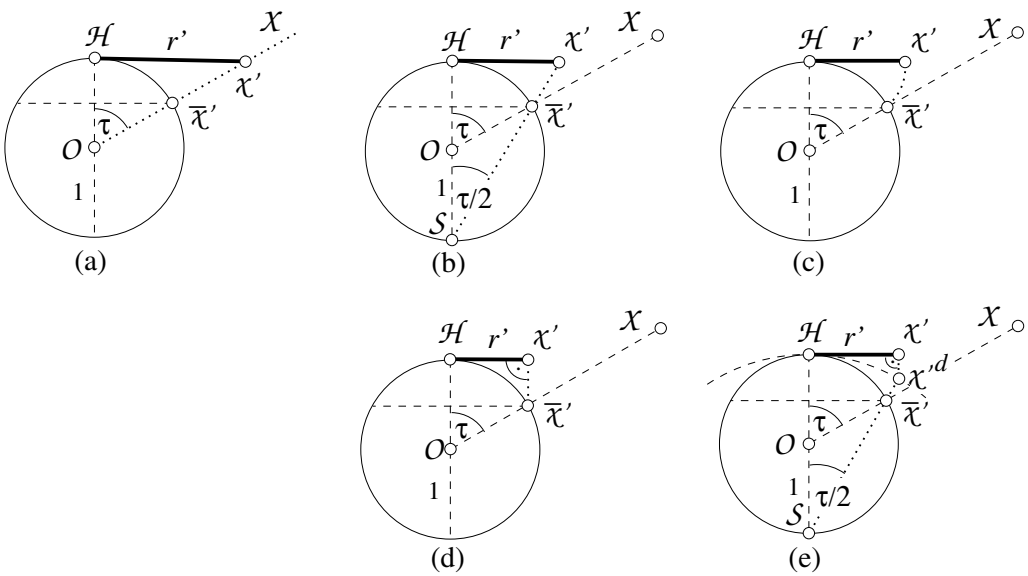


Fig. 12.15 Central models for fish-eye lenses, assuming $c = 1$. Shown are the scene point $\mathcal{X}$ and the radius r' from the symmetry axis of the point $\mathcal{X}'$ in the sensor (thick line). (a) Straight line-preserving perspective model as reference, where the radius r' is the distance in the perspective projection. It can handle viewing fields only close to 180°. (b) Circle preserving stereographic model. It can be viewed as a projection of the spherical point $\bar{\mathcal{X}}'$ onto its tangent plane from the south pole $\mathcal{S}$. (c) Equidistant model. The radius is identical to the arc $\mathcal{H}\bar{\mathcal{X}}'$. (d) Orthographic model. It can only handle viewing fields up to 180°. (e) Equisolid angle model. It first projects the point $\bar{\mathcal{X}}'$ onto the sphere with centre $\mathcal{S}$ and radius $2c$, getting $\mathcal{X}'^d$. The observable point $\mathcal{X}'$ results from an orthographic projection of $\mathcal{X}'^d$ onto the tangent plane

(a) the perspective projection model as reference,

$$\bar{r}' := r'_{\text{persp}} = c \tan \tau \,; \tag{12.95}$$

(b) the stereographic projection model,

$$r'_{\text{stereo}} = 2c \tan \frac{\tau}{2}, \tag{12.96}$$

which is realized in the Samyang 8 mm f/3.5 Aspherical IF MC Fish-eye lens;

(c) the equidistant projection model,

$$r'_{\text{equidi}} = c\,\tau \tag{12.97}$$

which is realized in the OP Fish-eye Nikkor 6 mm f/2.8 lens;

(d) the orthogonal or orthographic projection model,

$$r'_{\text{ortho}} = c \sin \tau\,, \tag{12.98}$$

realized by the OP Fish-eye Nikkor 10 mm f/5.6 lens; and

(e) the equisolid angle projection model,

$$r'_{\text{equiso}} = 2c \sin \frac{\tau}{2}\,, \tag{12.99}$$

which is realized by the Fish-eye Nikkor 10.5 mm DX Lens.

The lens models can be integrated into the projection model similarly to the additive lens distortion model in Sect. 12.1.4, p. 476, but now using the multiplicative model $r' := |{}^i\boldsymbol{x}'|$ and $\bar{r}' := |{}^i\bar{\boldsymbol{x}}'|$; due to the rotational symmetry we obtain

$${}^i\boldsymbol{x}'(\bar{r}') = \frac{r'(\bar{r}')}{\bar{r}'}\, {}^i\bar{\boldsymbol{x}}'\,, \tag{12.100}$$

an approach similar to the one of Scaramuzza (2008).

Catadioptric Cameras. The optical system of catadioptric cameras is composed of lenses and a mirror which map the rays into a perspective camera. The principle of a catadioptric camera with a parabolic mirror is represented in Fig. 12.16, which illustrates the concept of the One Shot 360 optics.

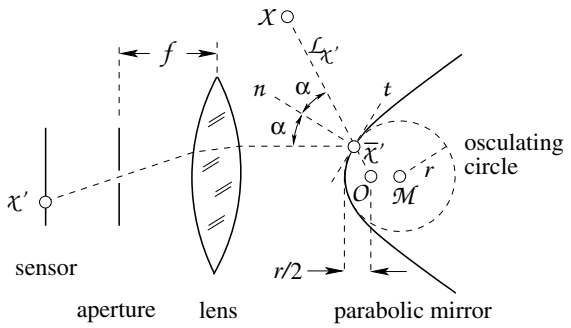


Fig. 12.16 Principle of the central projection with a parabolic catadioptric optical system. The scene point $\mathcal{X}$ is mirrored at point $\bar{\mathcal{X}}'$ of the parabolic mirror. The telecentric lens maps $\bar{\mathcal{X}}'$ to $\mathcal{X}'$. The effective viewpoint, the projection centre O, is the focal point of the parabolic mirror, where all projection rays $\mathcal{L}_{x'}$ intersect

The only catadioptric cameras which realize a central projection are those which have a conic section rotated around their axis and where the projection centre of the camera is at the focal point of the conic. The reason is that rays coming from one focal point of a conic and mirrored at the conic meet at the other focal point; for parabolic mirrors the second

focal point is at infinity. There exist compact camera models with parabolic, elliptic, and hyperbolic mirrors (cf. Nayar, 1997; Baker and Nayar, 1999).

12.1.9.2 Camera Systems

Camera systems consist of several single cameras, which generally have a stable mutual pose. We only refer to systems of perspective or spherical cameras. Such camera systems have a set of viewpoints.

The exterior orientation here refers to the spatial position of the system as a whole, for example, the pose of a stereo head consisting of two cameras mounted on a van or a robot. As before, the parameters of the interior orientation for each camera have to be accounted for, in addition to the *interior system orientation* consisting of the relative poses between the individual cameras.

The mapping of a scene point $\mathcal{X}_i$ into the camera c of the camera system at time t, see Fig. 12.17, is described by

$$\mathcal{X}_{itc} = \mathcal{P}_{tc} \circ \mathcal{M}_c^{-1} \circ \mathcal{M}_t^{-1} \circ \mathcal{X}_i \,. \tag{12.101}$$

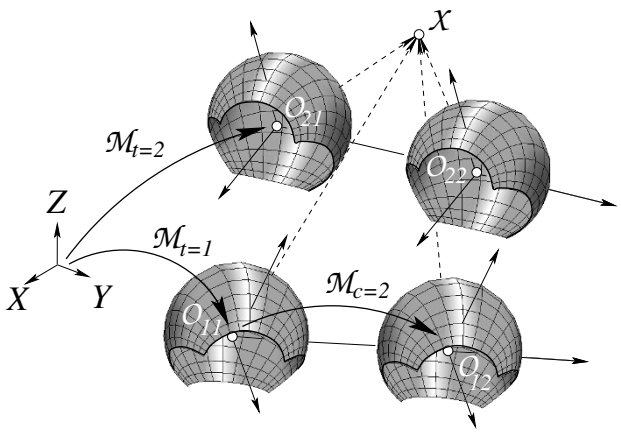


Fig. 12.17 Camera system S_{tc} with two cameras $c = 1, 2$ at two times $t = 1, 2$. The cameras are shown with their viewing spheres and projection centres $\mathcal{O}_{tc}$. The motion of the camera system S_{tc} is composed of the motion $\mathcal{M}_t$ from the origin to the system camera $c = 1$ at time t and the motion $\mathcal{M}_c$ from this camera to the system camera $c = 2$ (from Schneider and Förstner, 2013)

The projection is expressed as

$$\mathbf{x}_{itc} = \mathsf{K}_{tc}\,[I_3|\mathbf{0}]\;\; \mathsf{M}_c^{-1}\;\; \mathsf{M}_t^{-1}\;\; \mathbf{X}_i \,. \tag{12.102}$$

Thus the only difference with the single camera model (12.34), p. 472 is the transformation of the scene point from the reference system of the camera system into the currently relevant camera coordinate system.

In case the cameras of the camera system change their mutual pose over time, e.g., due to vibrations of the sensor platform, the motion matrix M_c would need to be replaced by M_{ct}.

We have touched on a few aspects of modelling cameras, which however are sufficient for a large range of applications. A recent review by Sturm et al. (2011) also addresses the modelling of line and slit cameras, which are common in mobile phones.

12.2 Orientation of the Single Image

12.2.1 Uncertainty of Image and Scene Observations 490
12.2.2 Estimating the Direct Linear Transformation for Orientation 494
12.2.3 Modelling Distortions 505
12.2.4 Spatial Resection 513
12.2.5 Theoretical Precision of Pose Estimation 521

This section collects methods for determining the orientation of a single image from known point or line correspondences. Due to different boundary conditions and scopes we discuss several procedures (cf. Table 12.3):

Table 12.3 Orientation procedures discussed in this section. Procedure names: direct linear transformation (DLT), estimation of parameters for interior and exterior orientation (IO/EO), spatial resection (SRS). Camera models: perspective and spherical, statistical optimality of the solution, direct or iterative solution, minimal solution or solution with redundant observations, relevance of procedure

	name	model	optimality	direct/ iterative	minimal/ redundant	relevance	Sect. /Page
1	DLT	perspective	suboptimal	direct	both	direct, minimal	12.2.2.1, p. 494
2	DLT	perspective	optimal	iterative	both	optimal, possibly IO/EO	12.2.2.2, p. 496
3	IO/EO	perspective	optimal	iterative	both	optimal, self-calibration	p. 501
4	SRS	spherical	–	direct	both	direct	12.2.4.1, p. 513
5	SRS	spherical	optimal	iterative	minimal	optimal	12.2.4.3, p. 520

1. The algebraically optimal solution, Sect. 12.2.2.1, is the classical one, following the procedure for estimating a homography (cf. Sect. 10.5.4.1, p. 406). This solution is statistically suboptimal, but can be used for obtaining approximate values. It can also be used with a minimal set of six points within a RANSAC procedure. We therefore also give the covariance matrix of the estimated parameters. The uncertainty of the derived parameters of the interior and the exterior orientation may be used for checking a solution for plausibility.
2. Estimating the elements of the projection matrix (cf. Sect. 12.2.2.2) in a statistically optimal manner is necessary after having eliminated outliers.
3. Estimating the pose and calibration parameters (cf. Sect. 12.2.2.3) in a statistically optimal manner is necessary for two purposes: (1) for giving insight into the effect of these parameters on the image coordinates and (2) as a basis for the discussion of additional parameters for self-calibration in bundle adjustment in Sect. 15.4, p. 674. We could specialize it for optimally estimating the exterior orientation of calibrated cameras; however, we then would not exploit the potential of using the spherical camera model, cf. below.
4. A minimum solution for spatial resection from points (cf. Sect. 12.2.4.1) is required for outlier detection using RANSAC. It uses the spherical camera model. We discuss this problem to illustrate an algebraic derivation of the covariance matrix which does not depend on the solution method. We also provide a direct solution for the redundant case of the spatial resection.
5. An iterative solution for spatial resection for a spherical camera allows us also to handle perspective images of a calibrated camera (cf. Sect. 12.2.4.3). We analyse the expected accuracy for a schematic setup.

We finally discuss methods of the *inverse perspective* which are useful if only partial information about the orientation or calibration is of interest, and methods which are based on pre-knowledge about the scene and partially infer the scene structure.

In this chapter we always assume the projection is free of distortions whereas the methods of calibration and self-calibration are presented in the context of the bundle-solution in Sect. 15.4, p. 674.

12.2.1 Uncertain Observations for Orientation and Reconstruction

All orientation procedures start from observed, i.e., uncertain, image and scene points and lines. The observation process in most applications will be automatic, using appropriate image analysis techniques discussed in the second volume in detail. Sometimes it may also be manual, e.g., when identifying specific scene points for which 3D coordinates are available and where reliable automatic methods for their detection and location do not (yet) exist. The *stochastical model* for observed image and scene points and lines highly depends on the observation process; it *eventually represents the quality of the assumed camera model.* Small effects not modelled in the functional model of the camera are left for the stochastical model of the estimation process and, e.g., lead to larger standard deviations for the observations, see the discussion in Sect. 4.1, p. 75.

stochastical model for image and scene points and lines

Moreover, the observed entities are not necessarily provided in a representation directly appropriate for statistical estimation. This especially holds for the representation of their uncertainty. Examples are observed image points, which are given by their sensor coordinates but are required as ray directions within a central projection model, or observed image lines, which are given in a centroid representation but are needed in a homogeneous representation. We will therefore exploit the various representations for uncertain points and lines in 2D and 3D given before in Sect. 10.2.2, p. 366 and adapt them to the situation of camera orientation and scene reconstruction.

The transformation of image points and lines depends on whether the cameras are calibrated or not. The orientation of calibrated cameras can use the spherical projection model, thus can be based on camera rays, i.e., normalized directions, which makes it possible to also handle omnidirectional cameras. The orientation of uncalibrated cameras, however, is always based on inhomogeneous sensor coordinates. The transformation of scene points and lines depends on whether we want to include points or lines at infinity, where spherically normalized homogeneous coordinates are the best choice.

In all cases, conditioning as a first step is recommended in order to avoid numerical instabilities.

This section collects the most relevant stochastical models for observed image and scene quantities and the transformations necessary for their orientation and reconstruction. The models should reflect the expected *accuracy* of the observations used within the estimation process. This includes all effects which might cause differences between the observed values and the assumed estimation model. If the assumed accuracy model is adequate, then the estimated variance factor $\widehat{\sigma}_0^2$ will be close to 1 and thus empirically confirm the made assumptions.

We will always model the unavoidable random perturbations during the measurement process. They may result from the used sensor, from the used algorithms or, if so captured, by the skill of human operators to perform the measurement process. But they will also depend on the uncertainty of the definition of the measured point or line: Take, for example, the centre of a tree or the gable of a roof, illuminated from the side. In both cases, the uncertainty of the definition of the image feature is likely to be much higher than the precision of repeated measurements.

uncertainty of feature definition

Specifying the uncertainty conceptually refers to random errors, thus to the *precision* of the measurements. This is why it is represented by a variance or a covariance matrix. It can be empirically derived from the residuals of previous projects, e.g., using the estimated variance factor. However, these estimates include unmodelled systematic effects, thus do not reflect the precision but the *accuracy* of the measurements w.r.t. the assumed mathematical model. Specifying the uncertainty of observations therefore always refers to the assumed mathematical model and the expected random and systematic deviations.

In the following, we give representative examples for such specifications.

12.2.1.1 Uncertainty of Observed Image Points and Lines

Orientation and reconstruction procedures start with image points or lines. This section discusses their stochastic properties, which depend on the details of the observation process.

Uncertainty of Image Points. We assume that we observe image points $\mathcal{X}'_i(\boldsymbol{x}'_i, \Sigma_{x'_i x'_i})$ with their coordinates $\boldsymbol{x}'_i$ and their uncertainty $\Sigma_{x'_i x'_i}$. In all cases we assume that the measured points in one image are mutually independent. The image coordinates of one point in the most simple case are assumed to be uncorrelated and have the same standard deviation, σ'_x, thus assuming $\Sigma_{x'_i x'_i} = \sigma^2_{x'} I_2$. This model is acceptable if we do not have further information about the observation process.

For example, the *key point detector* proposed by Lowe (2004) on average leads to a positional accuracy of approximately $\sigma_{x'} = 0.3$ pixel.

However, points detected at a higher scale of the image pyramid can be seen as having been derived from a blurred image. Then we can expect the uncertainty of the detected key points to increase with the scale s, i.e., the pixel size in a pyramid level expressed in units of the pixel size of the highest resolution. A simple model would be (cf. Zeisl et al., 2009)

effect of scale on uncertainty

$$\sigma_{x'}(s) = s\,\sigma_{x'}(1)\,, \tag{12.103}$$

where the standard deviation in the image of the highest resolution ($s = 1$) is $\sigma_{x'}(1)$ (cf. Läbe et al., 2008). In Sect. 15.4.1.3, p. 679 we will refine this model and experimentally show that the standard deviation of the Lowe key points for small scales is around $\sigma_{x'} = 0.15$ pixel.

When using omnidirectional cameras, instead of the image coordinates $\boldsymbol{x}'$, we use the directions of the image rays, namely normalized direction vectors, say $\mathbf{u}'$. The most simple assumption is that these directions have uniform uncertainty; thus, we can assume their covariance matrix to be $\sigma^2 I_3$ (cf. Lhuillier, 2006). As they only are uncertain across the direction, we can enforce the norm constraint. This yields the rank 2, 3×3 covariance matrix

$$\Sigma_{\mathrm{uu}} = (I_3 - \mathbf{u}\mathbf{u}^\mathsf{T})\sigma^2\,, \tag{12.104}$$

as we already discussed when spherically normalizing uncertain vectors, cf. (10.19), p. 368.

The relation between the uncertainty of image coordinates and directions is shown in Fig. 12.18. The geometric situation is radially symmetric around the viewing direction

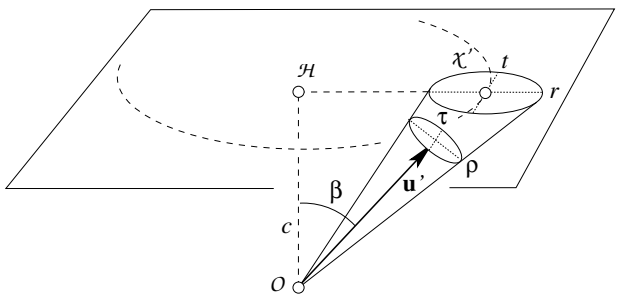


Fig. 12.18 Relation between uncertainty of image coordinates and ray direction. The situation is rotationally symmetric w.r.t. the viewing direction $O\mathcal{H}$. We distinguish between radial (r, ρ) and tangential (t, τ) uncertainties of the point $\mathcal{X}$ in the image plane and the direction $\mathbf{u}'$ on the viewing sphere

$O\mathcal{H}$. The angle β between $\mathbf{u}'$ and the viewing direction is decisive. We need to distinguish between the radial and the tangential uncertainty referring to a circle around the viewing direction.

1. The tangential standard deviation $\sigma_{x'_t}$ of the image point $\mathcal{X}'$ and the standard deviation $\sigma_{u'_\tau}$ of the ray direction $\boldsymbol{u}'$ are related by

$$\sigma_{x'_t} = c\frac{\sigma_{u'_\tau}}{\cos\beta}\,. \tag{12.105}$$

2. The radial standard deviation $\sigma_{x'_r}$ of the image point and of the ray direction $\sigma_{u'_\rho}$ are related by

$$\sigma_{x'_r} = c\frac{\sigma_{u'_\rho}}{\cos^2\beta}\,. \tag{12.106}$$

Even for moderate angles, say $\beta = 45°$, the ratios are 0.7 and 0.5, leading to weight differences between the image coordinates and the components of the ray direction of 1:2 and 1:4 (see Figs. 10.9, p. 369 and 10.11, p. 372). They need to be taken into account and therefore lead to correlations between the image coordinates.

ray direction from sensor coordinates

Sign and Uncertainty of Ray Directions. Ray directions generally can be derived from

$${}^c\mathbf{x}' \stackrel{+}{=} -\text{sign}(c\,K_{33}\,x'_3)\mathsf{K}^{-1}\,\mathbf{x}'\,; \tag{12.107}$$

cf. (12.23), p. 469. If the point is positive, i.e., $x_3 > 0$, and the calibration matrix is proper, which often can be assumed, we obtain

$${}^c\mathbf{x}' \stackrel{+}{=} -\text{sign}(c)\;\mathsf{K}^{-1}\,\mathbf{x}' \quad \text{with} \quad K_{33} > 0, x'_3 > 0\,. \tag{12.108}$$

If in addition the principal distance is negative, this simplifies to

$${}^c\mathbf{x}' = \mathsf{K}^{-1}\mathbf{x}'\,; \tag{12.109}$$

cf. (12.32), p. 472. We will assume the image points to be positive and the calibration proper, and therefore refer to (12.108) if the direction of the ray is of concern.

We now have the projection ray $\mathcal{L}_{x'}$ in the camera system,

$${}^c\mathbf{L}'_x = \begin{bmatrix} {}^c\mathbf{x}' \\ \mathbf{0} \end{bmatrix}\,, \tag{12.110}$$

with ${}^c\mathbf{x}'$ from (12.108) or (12.109).

The stochastic properties of the ray direction depend on both the uncertainty of the calibration matrix and the uncertainty of the observed point coordinates $\mathbf{x}'$. When assuming K to be uncertain, all points in an image, or possibly in all images taken with the same camera, would be correlated, which highly increases the numerical effort for all estimation procedures. We determine the uncertainty of the ray direction assuming the calibration of the camera to be perfect and obtain

$$\Sigma_{{}^c\mathbf{x}'{}^c\mathbf{x}'} = \mathsf{K}^{-1}\,\Sigma_{\mathbf{xx}}\,\mathsf{K}^{-\mathsf{T}}\,. \tag{12.111}$$

This approximation usually is acceptable, as quite some effort is put into the calibration of a camera. If the calibration parameters are taken as approximate values within estimation where also these parameters are determined (a self-calibration), then neglecting the uncertainty of the approximate values has no effect on the final estimate, cf. Rao (1967, Lemma 5a) and the discussion of (4.344), p. 138.

Uncertainty of Line Segments. The situation is a bit more complex when using image line segments, as there is no canonical way to represent uncertain line segments. We assume they are extracted with some automatic detector. Most of these detectors determine the parameters of the straight line by a least squares fit through sequences of edge points. This is equivalent to using the algorithm discussed in Sect. 10.5.2.2, p. 397. It yields the line segment in the centroid representation (10.48), p. 375 with centroid coordinates $\boldsymbol{x}_0$ and direction α:

$$\ell: \quad \{\boldsymbol{x}_0, \alpha; \sigma_q, \sigma_\alpha\}\,, \tag{12.112}$$

where the standard deviations σ_q and σ_α represent the uncertainty of the position across the line and its direction. If the line segment is derived from I edge points with an average spacing of the pixel size Δx and a positional uncertainty of σ, e.g., the rounding error $\sigma = \Delta x/\sqrt{12}$, we can use

Exercise 12.13

$$\sigma_q = \frac{\sigma}{\sqrt{I}}, \qquad \sigma_\alpha = \frac{\sigma}{\Delta x}\sqrt{\frac{12}{I^3 - I}}. \tag{12.113}$$

Deriving homogeneous coordinates $\mathbf{l}$ for the line can be achieved via the representation using the Hessian form of the line. With the relations in Sect. 10.2.2.3, p. 373, we obtain the parameters from

$$\boldsymbol{h} = \begin{bmatrix} \phi \\ d \end{bmatrix} = \begin{bmatrix} \alpha + \pi/2 \\ d \end{bmatrix}, \qquad \begin{bmatrix} m_0 \\ d \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{12.114}$$

with the covariance matrix

$$\Sigma_{hh} = \begin{bmatrix} \sigma_\alpha^2 & -m_0\sigma_\alpha^2 \\ -m_0\sigma_\alpha^2 & m_0^2\sigma_\phi^2 + \sigma_q^2 \end{bmatrix}. \tag{12.115}$$

The transition to homogeneous parameters for the lines then results from the relations in Sect. 10.2.2.3, p. 375.

bias of theoretical variances

These theoretical covariance matrices usually are biased by a factor which is common for all points and all line segments, but different for the set of points and the set of lines. If only one type of observation is used, e.g., only image points, assuming this factor σ_0 to be 1 has no influence on the estimation, as discussed in Sect. 4.2.3, p. 89. Then the factor σ_0 can be estimated and used in subsequent estimations. If both types of observations are used in the estimation, the two factors, say σ_{0p} and σ_{0l} for points and lines, may be estimated either in two separate estimations by only using points or line segments, or simultaneously by variance component estimation (cf. Sect. 4.2.4, p. 91). In order to achieve precise enough estimates for these variance factors the redundancy should be large enough (cf. Sect. 4.83, p. 90).

variance component estimation

A generalization of using straight line segments would use ellipses as images of 3D circles (cf. Sect. (12.3.5), p. 534). Here the image processing algorithm determining the ellipse segments should provide some information about the uncertainty of the segment. If the method is integrated into an orientation procedure, the accuracy of the image of the centre of a circle may be specified.

12.2.1.2 Uncertainty of Observed Scene Points and Lines

We also assume we observe scene points $\mathcal{X}_i(\boldsymbol{X}_i, \Sigma_{X_iX_i})$ with their inhomogeneous coordinates and some covariance matrix. In the photogrammetric community such points are called control points or control lines. Only in special cases can we assume these coordinates to be of superior quality, and assume the coordinates to have zero variance. Generally, we have some knowledge about the accuracy of the measuring process, which we should encode in the covariance matrix. Coordinates derived by GPS usually are only precise in the range of 2-20 m if no differential GPS procedure is used (Hofmann-Wellenhof et al., 2008). Otherwise accuracies below 0.1 m down to 0.01 m can be achieved.

control points and lines

differential GPS for control points

In all cases it is helpful to assume that the scene coordinates are uncertain, since we then obtain residuals for the observed 3D coordinates, which can be tested statistically. If, however, the coordinates are treated as given, fixed values, their measurement deviations show only indirectly in the residuals of the image coordinates.

Points at infinity, e.g., points at the horizon or stars, are spatial directions $\boldsymbol{D}$, with $|\boldsymbol{D}| = 1$, which need to be represented using homogeneous coordinates. In many cases, we want to assume the uncertainty of the spatial directions to be isotropic, i.e., the same

points at infinity as control directions

in all directions across $\boldsymbol{D}$. Then we represent the uncertainty of the normalized homogeneous vector $\mathbf{D} = [\boldsymbol{D}^\mathsf{T}, 0]^\mathsf{T}$ by the covariance matrix $\mathrm{Diag}(\{\sigma_d^2 I_3, 0\})$, where the standard deviation σ_d denotes the directional uncertainty in radians.

Observed 3D lines $\mathcal{L}$ mostly are generated by the join of two observed scene points, say $\mathcal{X}_s$ and $\mathcal{X}_e$, possibly enforcing some constraint, e.g., horizontality. We may directly use the two points in further processing. We may force their coordinates to follow the constraint, e.g., by averaging the vertical coordinates of the starting and end points, which yields smaller standard deviations at the expense of correlations between the vertical coordinates. Alternatively we can use (7.38), p. 301 to derive the Plücker coordinates $\mathbf{L}$ and their covariance matrix Σ_{LL} from given 3D points.

horizon as 3D control line

An exception is the horizon, which may be visible in the image. It can serve as a control line. Its Plücker coordinates in a local map coordinate system are $\mathbf{L} = [0, 0, 0, 0, 0, 1]^\mathsf{T}$, cf. (5.178), p. 244, and can be assumed to be fixed.

Conditioning, as described in Sect. 6.9, p. 286, is absolutely necessary if we work with homogeneous coordinates in the sensor coordinate system, since image point coordinates and the distances of image line segments from the origin usually are given in pixels. The same holds for 3D points and lines.

Since conditioning contains at least a scaling, and often also a translation, the covariance matrices of the conditioned coordinates need to be derived by variance propagation. This conditioning is necessary even if the homogeneous coordinates are spherically normalized afterwards.

12.2.2 Estimating the Direct Linear Transformation for Orientation

12.2.2.1 Direct Estimation of Projection Matrix

direct linear transformation

We first give a direct solution for determining the projection matrix from points using the direct linear transformation (DLT), cf. (12.33), p. 472. Since image points have two degrees of freedom, and we want to determine the 11 parameters specifying an arbitrary projection matrix, we need at least six points. This minimum number of observed entities already yields a redundancy of 1.

The Algebraically Optimal Solution. For given correspondences $(\mathcal{X}_i', \mathcal{X}_i), i = 1, ..., I$, between image and scene points we have the direct linear transformation based on their inhomogeneous coordinates,

$$x_i' = \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = \frac{u_i'}{w_i'}, \qquad y_i' = \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = \frac{v_i'}{w_i'}. \tag{12.116}$$

Multiplying both sides with the denominator and collecting the coefficients for the elements

$$\mathbf{p}^\mathsf{T} = (\mathrm{vec}\mathsf{P})^\mathsf{T} = [P_{11}, P_{21}, P_{31}, P_{12}, P_{22}, P_{32}, P_{13}, P_{23}, P_{33}, P_{14}, P_{24}, P_{34}] \tag{12.117}$$

of the projection matrix from

$$w_i' x_i' - u_i' = 0, \qquad w_i' y_i' - v_i' = 0, \tag{12.118}$$

we arrive at the following two constraints:

$$A_i^\mathsf{T}\mathbf{p} = \begin{bmatrix} \boldsymbol{a}_{yi}^\mathsf{T} \\ \boldsymbol{a}_{xi}^\mathsf{T} \end{bmatrix} \mathbf{p} = \mathbf{0}, \tag{12.119}$$

with the Jacobian $\mathbf{A}_i^\mathsf{T}$ for each point

$$\mathbf{A}_i^\mathsf{T} = \begin{bmatrix} 0 & -X_i & X_i y_i' & 0 & -Y_i & Y_i y_i' & 0 & -Z_i & Z_i y_i' & 0 & -1 & y_i' \\ X_i & 0 & -X_i x_i' & Y_i & 0 & -Y_i x_i' & Z_i & 0 & -Z_i x_i' & 1 & 0 & -x_i' \end{bmatrix} . \tag{12.120}$$

Observe, the first row refers to y_i', the second to x_i'.

These constraints can also be derived from their compact form,

$$\mathsf{S}(\mathbf{x}_i')\widehat{\mathsf{P}}\mathbf{X}_i = \mathsf{S}^\mathsf{T}(\widehat{\mathsf{P}}\mathbf{X}_i)\mathbf{x}_i' \stackrel{!}{=} \mathbf{0} . \tag{12.121}$$

The constraint enforces the identity $\mathbf{x}_i \equiv \mathcal{P}(\mathcal{X}_i)$ of the observed point $\mathbf{x}_i'$ and the projected point $\mathcal{P}(\mathcal{X}_i)$ with coordinates $\mathsf{P}\mathbf{X}_i$. The constraints (12.121) only have two degrees of freedom. For the algebraically optimal solution we may, but do not need to, select independent constraints following Sect. 7.4.1, p. 317. Since we want to provide a different derivation for (12.120), we use the skew symmetric matrix $\mathsf{S}^{(s)}(\mathbf{x}_i')$ with selected rows. Since the observed image points do actually lie within the image area, the third coordinate is guaranteed to be not 0. We therefore obtain the two independent constraints for the observed scene points,

$$\mathbf{A}_i^\mathsf{T}\,\widehat{\mathbf{p}} \stackrel{!}{=} \mathbf{0} , \qquad \mathbf{A}_i^\mathsf{T} = \mathbf{X}_i^\mathsf{T} \otimes \mathsf{S}^{(s)}(\mathbf{x}_i') , \tag{12.122}$$

with

$$\mathbf{p} = \mathrm{vec}(\mathsf{P}) , \quad \mathsf{S}^{(s)}(\mathbf{x}_i') = \begin{bmatrix} \boldsymbol{e}_1^\mathsf{T} \\ \boldsymbol{e}_2^\mathsf{T} \end{bmatrix} \mathsf{S}(\mathbf{x}_i') . \tag{12.123}$$

The first two rows of the matrix $\mathbf{A}_i^\mathsf{T}$ in (12.122) are identical to the ones in (12.120).

If we have six or more points we obtain an algebraic solution $\widehat{\mathbf{p}}$ as the right singular vector of the $2I \times 12$ matrix $\mathbf{A} = [\mathbf{A}_i^\mathsf{T}]$ belonging to the smallest singular value using an SVD (cf. Sect. 4.9.2, p. 177). Observe, the SVD yields a spherically normalized 12-vector $\mathbf{p}$ having only 11 d.o.f.

Critical Configurations for the Estimation of the Projection Matrix with Points. The solution is not possible if the 3D points are coplanar, i.e., when the matrix A has rank 8 (Faugeras, 1993, p. 61). This can be seen in (12.120), where three columns of A become 0, e.g., if all Z_i are zero, which can be assumed without loss of generality.

If the object is planar and the interior orientation is known then the camera orientation can be determined (cf. Sect. 12.2.4, p. 513).

If the points and the projection centre are on a *twisted cubic curve* (which is very unlikely), no unique or stable solution is possible either (cf. Hartley and Zisserman, 2000; Wrobel, 2001).

If the points are far from a critical configuration the solution will be stable, e.g., if the points are evenly distributed on two planes. The acceptability of a configuration can be evaluated by an analysis of the covariance matrix of the parameters of the projection matrix with respect to the criterion covariance matrix of an acceptable reference configuration (cf. Sect. 4.6.2.3, p. 121). An example of the accuracy of the orientation with eight points is given in Sect. 12.2.5, p. 521.

Covariance Matrix of the Estimated Projection Matrix. In all cases we can give the covariance matrix of the algebraically estimated projection matrix using $\boldsymbol{\Sigma}_{\mathrm{pp}} = \mathbf{A}^+ \mathbf{B} \boldsymbol{\Sigma}_{ll} \mathbf{B}^\mathsf{T} \mathbf{A}^{+\mathsf{T}}$, cf. (4.521), p. 181, with the constraint matrix $\mathbf{A}$. Starting from the model $\boldsymbol{g}(\mathbf{p}, \boldsymbol{l}) = \mathbf{0}$, the Jacobian w.r.t. the observations generally is $\mathbf{B}(\mathbf{p}, \boldsymbol{l}) = \partial \boldsymbol{g}(\mathbf{p}, \boldsymbol{l}) / \partial \boldsymbol{l}$ with the observations $\boldsymbol{l} = [[\mathbf{X}_i^\mathsf{T}], [\mathbf{x}_i'^\mathsf{T}]]^\mathsf{T}$. The matrix consists of two block diagonal matrices referring to the observed values $\mathbf{X}_i$ and $\mathbf{x}_i'$. We have

$$\mathbf{B} = \mathrm{Diag}(\{ \underbrace{\mathbf{S}^{(s)}(\mathbf{x}_i')\mathsf{P}}_{\mathbf{B}_{\mathbf{X}_i}^\mathsf{T}\ 2\times 4} , \underbrace{\mathbf{S}^{(s)\mathsf{T}}(\mathsf{P}\mathbf{X}_i)}_{\mathbf{B}_{\mathbf{x}'_i}^\mathsf{T}\ 2\times 3} \}) . \tag{12.124}$$

The covariance matrix Σ_{ll} is the block matrix containing the covariance matrices of the observed entities,

$$\Sigma_{ll} := \text{Diag}(\{\text{Diag}(\{\Sigma_{\mathrm{X}_i\mathrm{X}_i}, \Sigma_{\mathrm{x}'_i\mathrm{x}'_i}\})\}) . \tag{12.125}$$

covariance matrix of projection matrix

Since we assume the observations to be mutually independent, the covariance matrix is given by

$$\Sigma_{\widehat{\mathrm{p}}\widehat{\mathrm{p}}} = \sum_{i=1}^{I} (A^+)_i \left(B_{\mathrm{X}_i}^{\mathsf{T}} \Sigma_{\mathrm{X}_i,\mathrm{X}_i} B_{\mathrm{X}_i} + B_{\mathrm{x}'_i}^{\mathsf{T}} \Sigma_{\mathrm{x}'_i,\mathrm{x}'_i} B_{\mathrm{x}'_i} \right) (A^+)_i^{\mathsf{T}} , \tag{12.126}$$

where the matrix $(A^+)_i$ is the submatrix of the pseudo-inverse A^+ belonging to observation i. Observe, we can work with the homogeneous coordinates themselves, even if their covariance matrices are singular, as only the projection onto the constraint is relevant. Equation (12.126) can also be used if we assume the control points $\mathcal{X}_i$ to be fixed, as then $\Sigma_{\mathrm{X}_i\mathrm{X}_i} = \mathbf{0}$.

Algorithm 16 summarizes the estimation method. It assumes the given image and scene coordinates to be conditioned. The estimated conditioned projection matrix therefore has to be unconditioned.

Algorithm 16: Algebraic estimation of uncertain conditioned projection matrix from $I \geq 6$ conditioned observed scene points
$[\widehat{\mathsf{P}}, \Sigma_{\widehat{\mathrm{p}}\widehat{\mathrm{p}}}] = \texttt{P_algebraically_from_x_X}(\{\mathbf{x}'_i, \Sigma_{\mathrm{x}'_i\mathrm{x}'_i}, \mathbf{X}_i, \Sigma_{\mathrm{X}_i\mathrm{X}_i}\})$.
Input: coordinates of corresponding points with covariance matrix $\{\mathbf{x}'_i, \Sigma_{\mathrm{x}'_i\mathrm{x}'_i}\}$, $\{\mathbf{X}_i, \Sigma_{\mathrm{X}_i\mathrm{X}_i}\}$.
Assumption: the coordinates are conditioned.
Output: conditioned projection matrix with uncertainty $\widehat{\mathsf{P}}$, $\Sigma_{\widehat{\mathrm{p}}\widehat{\mathrm{p}}}$.

1 SVD: $UAV^{\mathsf{T}} = \texttt{svd}([\mathbf{X}_i^{\mathsf{T}} \otimes \mathsf{S}^{(s)}(\mathbf{x}'_i)])$;
2 Parameters $\widehat{\mathbf{p}}$: right singular vector to smallest singular value;
3 Transformation, transfer vector $\mathbf{p}$ into 3×4 matrix: $\widehat{\mathbf{p}} \to \widehat{\mathsf{P}}$;
4 Pseudo inverse: $A^+ = VA^+U^{\mathsf{T}}$, A^+ has only 11 nonzero diagonal elements;
5 Covariance matrix $\Sigma_{\widehat{\mathrm{p}}\widehat{\mathrm{p}}}$ (12.126).

Stellar Calibration. Given an image of stars we can determine the interior orientation of a perspective camera by a direct linear transformation. Equation (12.37) is a mapping of the plane at infinity, namely the directions $\boldsymbol{X}_\infty$, to the sensor. Given four or more points, we may determine the homography at infinity, H_∞. The decomposition according to (12.146), p. 499 then provides the elements of the interior orientation.

12.2.2.2 Statistically Optimal Estimation of the Projection Matrix

The statistically optimal estimation of the projection matrix P may be performed in two ways, both with the same number (11) of unknowns:

1. We may estimate the elements of the projection matrix. This follows the estimation of a planar homography in Sect. 10.6.3.2, p. 426.
2. We may estimate pose and calibration parameters. This allows an analysis of the effect of these parameters on the image and has the advantage of immediately obtaining the covariance matrix of these parameters.

We will treat both cases. For the following derivation we assume the measured image coordinates to be uncertain, and the coordinates of the known 3D points to be fixed.

Estimating the Elements of the Projection Matrix. The estimation task is similar to the one of estimating a 2D homography (cf. Sect. 10.6.3.1, p. 425). There we used the

fact that a homography is an element of a group, which allowed us to represent a small homography using a matrix exponential. Projection matrices do not form a group, so we treat vecP as element of an eleven-dimensional projective space $\mathbb{R}^{11}$, representing P by its spherically normalized vector $\mathbf{p}^s = \mathsf{N}(\mathrm{vec}\mathsf{P})$, which is equivalent to normalizing the matrix to Frobenius norm 1. Then we can use to advantage the concept of reduced coordinates for the estimation of P.

$\mathrm{vec}\mathsf{P} \in \mathbb{R}^{11}$ *used for estimation*

We express the inhomogeneous sensor coordinates, which always can be assumed to be finite, as functions of the unknown projection $\mathcal{P}(\mathsf{P})$ and the given, fixed 3D points $\mathcal{X}_i(\mathbf{X}_i)$. We use the function $\boldsymbol{x} = \boldsymbol{c}(\mathbf{x})$, cf. (5.31), p. 206, which maps homogeneous coordinates $\mathbf{x} = [\boldsymbol{x}_0^\mathsf{T}, x_h]^\mathsf{T}$ to inhomogeneous ones,

$$\boldsymbol{x} = \boldsymbol{c}(\mathbf{x}) := \frac{\boldsymbol{x}_0}{x_h} \,. \tag{12.127}$$

Then we have the model $\boldsymbol{x}'_i + \widehat{\boldsymbol{v}}_i = \boldsymbol{c}(\widehat{\mathsf{P}}\,\mathbf{X}_i)$ or

$$\boldsymbol{x}'_i + \widehat{\boldsymbol{v}}_i = \boldsymbol{c}((\mathbf{X}_i^\mathsf{T} \otimes I_3)\widehat{\mathbf{p}}) \,, \quad \mathbb{D}(\underline{\boldsymbol{x}}'_i) = \Sigma_{x'_i x'_i} \,, \quad i = 1, ..., I \,. \tag{12.128}$$

It is in the form of a nonlinear Gauss–Markov model, where the observations are the sensor coordinates $\boldsymbol{x}'_i$ and the unknowns are the parameters $\mathbf{p}$ of the projection matrix. We assume approximate values $\widehat{\mathsf{P}}^a$ for the estimated projection matrix $\widehat{\mathsf{P}}$ to be available. We obtain the Jacobians for the functions $\boldsymbol{c}(\mathbf{x})$ and for the corrections $\Delta\mathbf{p}(\Delta\boldsymbol{p}_r)$ of the parameters of the projection matrix,

$$\underset{2\times 3}{J_c(\mathbf{x})} := \frac{\partial \boldsymbol{c}(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{x_h^2}[x_h I_2 | - \boldsymbol{x}_0] \,, \qquad \underset{12\times 11}{J_r(\mathbf{p})} := \frac{\partial \Delta\mathbf{p}(\Delta\boldsymbol{p}_r)}{\partial \Delta\boldsymbol{p}_r} = \mathsf{null}(\mathbf{p}^\mathsf{T}) \tag{12.129}$$

(cf. (10.33), p. 371 and (10.25), p. 370), with the predicted homogeneous and inhomogeneous sensor coordinates

$$\widehat{\mathbf{x}}'^a_i = \widehat{\mathsf{P}}^a \mathbf{X}_i \,, \qquad \boldsymbol{x}'^a_i = \boldsymbol{c}(\widehat{\mathbf{x}}'^a_i) \,. \tag{12.130}$$

Linearization of (12.128) then yields

$$\begin{aligned} \Delta\boldsymbol{x}'_i + \widehat{\boldsymbol{v}}_i &= (\boldsymbol{x}'_i - \widehat{\boldsymbol{x}}'^a_i) + \widehat{\boldsymbol{v}}_i && (12.131)\\ &= \frac{\partial \boldsymbol{x}'_i}{\partial \mathbf{x}'_i} \, \frac{\partial \mathbf{x}'_i}{\partial \widehat{\Delta\mathbf{p}}} \, \frac{\partial \widehat{\Delta\mathbf{p}}}{\partial \widehat{\Delta\boldsymbol{p}}_r} \, \widehat{\Delta\boldsymbol{p}}_r && (12.132)\\ &= J_c(\widehat{\mathbf{x}}'^a_i) \, (\mathbf{X}_i^\mathsf{T} \otimes I_3) \, J_r(\widehat{\boldsymbol{p}}^a) \, \widehat{\Delta\boldsymbol{p}}_r \,, && (12.133) \end{aligned}$$

where all Jacobians are to be taken at the approximate values. We finally have the linearized Gauss–Markov model

$$\Delta\boldsymbol{x}'_i + \widehat{\boldsymbol{v}}_i = A_i^\mathsf{T} \, \widehat{\Delta\boldsymbol{p}}_r \,, \quad i = 1, \ldots, I \tag{12.134}$$

for the unknown 11-vector for the corrections to the reduced estimated parameters $\widehat{\boldsymbol{p}}_r$ of the vectorized projection matrix $\mathbf{p} = \mathrm{vec}\mathsf{P}$. The 2×11 matrices A_i^T, which are the ith component of the design matrix

$$A = \begin{bmatrix} A_1^\mathsf{T} \\ \cdots \\ A_i^\mathsf{T} \\ \cdots \\ A_I^\mathsf{T} \end{bmatrix} \,, \tag{12.135}$$

are

$$A_i^\mathsf{T} = J_c(\widehat{\mathbf{x}}'^a_i) \, (\mathbf{X}_i^\mathsf{T} \otimes I_3) \, J_r(\widehat{\boldsymbol{p}}^a) \,. \tag{12.136}$$

The updates for the reduced parameters are determined from the normal equation system

$$N\widehat{\Delta \boldsymbol{p}}_r = \boldsymbol{n} \qquad N = \sum_i A_i \Sigma_{x_i x_i}^{-1} A_i^{\mathsf{T}}\,, \qquad \boldsymbol{n} = \sum_i A_i \Sigma_{x_i x_i}^{-1}\, \Delta \boldsymbol{x}_i'\,. \tag{12.137}$$

The updated parameters in the νth iteration are

$$\widehat{\mathbf{p}}^{(\nu)} = \mathsf{N}\left(\widehat{\mathbf{p}}^{(\nu-1)} + J_r(\widehat{\mathbf{p}}^{(\nu-1)\mathsf{T}})\widehat{\Delta \boldsymbol{p}}_r^{(\nu)}\right) \tag{12.138}$$

(cf. (10.270), p. 419), which can be rearranged to yield the estimated projection matrix $\widehat{\mathsf{P}}^{(\nu)}$. The iterations have to be performed until the corrections of the parameters fall below a certain percentage of their standard deviation (cf. Sect. 4.8.2.5, p. 167).

Besides the estimated projection matrix, we obtain the estimated residuals

$$\widehat{\boldsymbol{v}}_i = \boldsymbol{c}(\widehat{\mathsf{P}}\mathbf{X}_i) - \boldsymbol{x}_i\,, \tag{12.139}$$

which allows us to determine the estimated variance factor. With the redundancy

$$R = 2I - 11 \tag{12.140}$$

we obtain

$$\widehat{\sigma}_0^2 = \frac{\sum_{i=1}^{I} \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \Sigma_{x_i x_i}^{-1} \widehat{\boldsymbol{v}}_i}{R}\,. \tag{12.141}$$

interpretation of $\widehat{\sigma}_0^2$

The estimated variance factor indicates the degree of consistency of the given observations with the perspective camera model. It can be statistically tested. If the test fails, we may detect causes by performing individual tests on outliers or deficiencies in the functional model using the techniques discussed in Sect. 4.6, p. 115. If there are no outliers and the perspective model is valid, the factor $\widehat{\sigma}_0^2$ gives the multiplicative bias of the theoretical precision of the observations, as discussed above (cf. Sect. 12.2.1, p. 490). The estimated covariance matrix of the observations, $\widehat{\sigma}_0^2 \Sigma_{x_i' x_i'}$, gives a more realistic value for the uncertainty of the observations.

The theoretical and empirical covariance matrices of the parameters are

$$\Sigma_{\widehat{p}\widehat{p}} = \sigma_0^2 N^{-1} \quad \text{and} \quad \widehat{\Sigma}_{\widehat{p}\widehat{p}} = \widehat{\sigma}_0^2 N^{-1}\,. \tag{12.142}$$

Algorithm 17, p. 499 summarizes the estimation. The approximate standard deviations for the parameters, $\sigma_{p_u}^a$, refer to the elements of the reduced parameter vector. Since the projection matrix refers to the conditioned coordinates in the scene and the image (cf. Sect. 6.9, p. 286), all standard deviations will be in the range of the relative precision of the observations, which usually is between 10^{-3} and 10^{-5}.

Decomposition of the Projection Matrix. We may now decompose the projection matrix P to achieve the calibration matrix, the rotation matrix, and the projection centre together with their joint covariance matrix. Since P has 12 elements, but only 11 degrees of freedom, we can expect the procedure to contain a normalization step. The decomposition of the projection matrix

$$\mathsf{P} = [\mathsf{A}|\mathbf{a}] = [\mathsf{K}R| - \mathsf{K}R\boldsymbol{Z}] \tag{12.143}$$

can be performed easily (cf. Sect. 12.1.3.8).

1. The projection centre is obtained from

$$\boldsymbol{Z} = -\mathsf{A}^{-1}\mathbf{a}\,; \tag{12.144}$$

   cf. (12.45), p. 475.
2. The rotation matrix can be obtained from a QR-decomposition of A. As both matrices R and K should have a positive determinant, we first ensure that the determinant of

Algorithm 17: Optimal estimation of a projection matrix from observed image points
$[\widehat{\mathsf{P}}, \Sigma_{\widehat{p}_r\widehat{p}_r}, \widehat{\sigma}_0^2, R] = \mathtt{P_from_x}(\{\boldsymbol{x}'_i, \Sigma_{x'_ix'_i}, \boldsymbol{X}_i\}, \mathsf{P}^a, \boldsymbol{\sigma}_p^a, T_x, \mathtt{maxiter})$
Input: I observed image points $\{\boldsymbol{x}'_i, \Sigma_{x'_ix'_i}\}$ and I fixed control points $\{\boldsymbol{X}_i\}$, approximate values $\mathsf{P}^{(0)}$ and $\boldsymbol{\sigma}_{p_u}^{(0)}$, thresholds for convergence T_x, `maxiter`.
Assumption: coordinates are conditioned.
Output: projection matrix $\{\widehat{\mathsf{P}}, \Sigma_{\widehat{p}_r\widehat{p}_r}\}$, variance factor $\widehat{\sigma}_0^2$, redundancy R.

Redundancy: $R := 2I - 11$;
if $R < 0$ **then stop**, not enough observations;
Initiate: iteration $\nu = 0$, $\widehat{\mathbf{p}}^{(0)} = \mathrm{vec}(\mathsf{P}^{(0)})$, stopping condition: $s = 0$;
repeat
  Initiate normal equation system: $\mathsf{N} = \mathsf{0}$, $\boldsymbol{n} = \mathbf{0}$;
  for *all points* **do**
    Approximated fitted values: $\widehat{\boldsymbol{x}}'_i = \boldsymbol{c}(\mathsf{P}^{(\nu)}\mathbf{X}_i)$;
    Linearized observations: $\Delta\boldsymbol{x}'_i = \widehat{\boldsymbol{x}}'_i - \boldsymbol{x}'_i$;
    Coefficient matrices: $\mathsf{A}_i = \mathsf{J}_c(\widehat{\mathsf{P}}^{(\nu)}\mathbf{X}_i)\ (\mathbf{X}_i^\mathsf{T} \otimes \mathsf{I}_3)\ \mathsf{J}_r(\widehat{\mathbf{p}}^{(\nu)})$;
    Intermediate design matrix: $\bar{\mathsf{A}}_i = \mathsf{A}_i\Sigma_{x_ix_i}^{-1}$;
    Update normal equations: $\mathsf{N} := \mathsf{N} + \bar{\mathsf{A}}_i\mathsf{A}_i^\mathsf{T}$, $\boldsymbol{n} := \boldsymbol{n} + \bar{\mathsf{A}}_i\Delta\boldsymbol{x}'_i$;
  end
  Covariance matrix of reduced parameters $\Sigma_{\widehat{p}_r\widehat{p}_r} = \mathsf{N}^{-1}$;
  Estimated reduced parameters $\widehat{\Delta\boldsymbol{p}}_r = \Sigma_{\widehat{p}_r\widehat{p}_r}\,\boldsymbol{n}$;
  if $\max_u |\widehat{\Delta\boldsymbol{p}}_{ru}|/\sigma_{p_u}^a < T_x$ **or** ν=`maxiter` **then** $s = 2$;
  Set iteration: $\nu := \nu + 1$;
  Update parameters: $\widehat{\mathbf{p}}^{(\nu)} = \mathsf{N}(\widehat{\mathbf{p}}^{(\nu-1)} + \mathsf{J}_r(\widehat{\mathbf{p}}^{(\nu-1)})\widehat{\Delta\boldsymbol{p}}_r)$;
  Updated projection matrix: $\widehat{\mathbf{p}}^{(\nu)} \to \widehat{\mathsf{P}}^{(\nu)}$;
until $s \equiv 2$;
Residuals: $\widehat{\boldsymbol{v}}_i = \boldsymbol{c}(\widehat{\mathsf{P}}\mathbf{X}_i) - \boldsymbol{x}_i$;
if $R > 0$ **then** variance factor $\widehat{\sigma}_0^2 = \sum_{i=1}^I \widehat{\boldsymbol{v}}_i^\mathsf{T}\Sigma_{x_ix_i}^{-1}\widehat{\boldsymbol{v}}_i/R$ **else** $\widehat{\sigma}_0^2 = 1$.

A is positive,

$$\bar{\mathsf{A}} = \mathrm{sign}(|\mathsf{A}|)\,\mathsf{A}\,. \tag{12.145}$$

As the canonical QR decomposition yields a multiplication of an orthogonal and an upper right triangular matrix, we first decompose the inverse,

$$\bar{\mathsf{A}}^{-1} = \bar{R}^\mathsf{T}\bar{\mathsf{K}}^{-1}\,, \tag{12.146}$$

and invert the resulting matrices.
The QR-decomposition yields an orthogonal matrix, not necessarily a proper rotation with $|\bar{R}| = 1$. Also, it is not unique, since with some arbitrary diagonal matrix $\mathrm{Diag}([a, b, c])$, with $a, b, c \in [-1, +1]$, $\bar{R}^\mathsf{T}D$, together with $D\bar{\mathsf{K}}^{-1}$, is also a valid QR-decomposition of $\bar{\mathsf{A}}^{-1}$.
Therefore, users need to specify, whether they want to have the camera in taking or in viewing position, i.e., whether in $\mathrm{sign}(\mathrm{diag}(\mathsf{K})) = [s, s, 1]^\mathsf{T}$ we have $s = +1$ or $s = -1$. Given the sign s, we then choose

user needs to specify sign s of principal distance c

$$\mathsf{K} = \bar{\mathsf{K}}D\,, \quad R = D\bar{R}\,, \quad \text{with} \quad D = \mathrm{Diag}(\mathrm{sign}(\mathrm{diag}(\mathsf{K})))\,\mathrm{Diag}([s, s, +1])\,. \tag{12.147}$$

The first factor $\mathrm{Diag}(\mathrm{sign}(\mathrm{diag}(\mathsf{K})))$ of D compensates for negative signs of the diagonal of $\bar{\mathsf{K}}$ and thus guarantees R has a positive determinant; the second factor $\mathrm{Diag}([s, s, +1])$ enforces the correct signs of the diagonals of K^e.

3. The normalized calibration matrix then is given by

$$\mathsf{K}^e = \frac{1}{K_{33}}\mathsf{K}\,. \tag{12.148}$$

Observe, taking the third row of $\mathsf{A} = \mathsf{K}R$ (12.143) leads to $|K_{33}| = |[0, 0, K_{33}]R| = |[p_{31}, p_{32}, p_{33}]|$.

Referring to the construction of the projection matrix P in (12.34), p. 472, we therefore have a one-to-one relationship between the projection matrix and the 11 orientation parameters when taking two conditions into account: (1) The matrix P may be scaled arbitrarily, as it is a homogeneous matrix. (2) The sign of the principal distance needs to be fixed. Only if we do not assume right-handed coordinate systems for both the scene and the sensor system do we need an additional constraint concerning the relative handedness of both coordinate systems.

Uncertainty of Parameters of the Exterior and the Interior Orientation. Given the covariance matrix Σ_{pp} of the elements of the projection matrix, we can derive the uncertainty of the parameters of the interior and the exterior orientation. The covariance matrix of the projection matrix could also have been derived from an algebraic solution (cf. Sect. 12.2.2.1, p. 495). As the algebraic solution can also be used in the minimal case of six points, the derived uncertainties for the orientation parameters can be used in a RANSAC procedure if some pre-knowledge about these parameters is available.

We first collect the orientation parameters in a stochastic 12-vector $\underline{\boldsymbol{h}}$: These are the three parameters $\underline{\boldsymbol{Z}}$ of the projection centre, the three parameters $\underline{\Delta \boldsymbol{r}}$ of the rotation, and the six parameters $\underline{\mathbf{k}}$ of the *nonnormalized* calibration matrix $\underline{\mathsf{K}}$.

From (12.54), p. 476 we obtain the Jacobian,

$$J_{h\mathrm{p}} = \frac{\partial \boldsymbol{h}}{\partial \mathbf{p}} = \left(\frac{\partial \mathbf{p}}{\partial \boldsymbol{h}}\right)^{-1} = J_{\mathrm{p}h}^{-1} \,. \tag{12.149}$$

This allows us to determine the covariance matrix $\mathbb{D}(\boldsymbol{h})$; cf. (12.52), p. 476 for the components $\boldsymbol{Z}$, $\Delta \boldsymbol{r}$, and $\mathbf{k} = [\boldsymbol{k}_0^\mathsf{T}, k_h]^\mathsf{T}$ of the exterior and the interior orientation.

The five uncertain elements $\boldsymbol{k}^e$ of the Euclideanly normalized calibration matrix are obtained using

$$\boldsymbol{k}^e = \frac{\boldsymbol{k}_0}{k_h} \,, \qquad \underbrace{J_{\mathrm{k}\mathrm{k}}}_{5\times 6} = \frac{\partial \boldsymbol{k}}{\partial \mathbf{k}} = \frac{1}{k_h^2}[k_h \mathsf{I}_5 \mid -\boldsymbol{k}_0] \,. \tag{12.150}$$

As the given 12 parameters are generally correlated, we obtain the covariance matrix Σ_{hh} of the minimum set of the 11 parameters from

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{Z} \\ \Delta \boldsymbol{r} \\ \boldsymbol{k}^e \end{bmatrix} \,, \quad \Sigma_{gg} = J_{gh} \Sigma_{hh} J_{gh}^\mathsf{T} \,, \quad \text{with} \quad J_{gh} = \begin{bmatrix} \mathsf{I}_6 & \mathsf{0}_{6\times 5} \\ \mathsf{0}_{5\times 6} & J_{\mathrm{k}\mathrm{k}} \end{bmatrix} \,. \tag{12.151}$$

Algorithm 18 summarizes the decomposition.

Algorithm 18: Decompose P, enforce correct sign of principal distance; $[\boldsymbol{Z}, R, \mathsf{K}^e, \Sigma_{gg}]$=`KRZ_from_P`$(\mathsf{P}, \Sigma_{\mathrm{pp}}, s)$
Input: $\{\mathsf{P}, \Sigma_{\mathrm{pp}}\}$, sign $s \in \{-1, +1\}$ of c.
Output: $\boldsymbol{Z}$, R, K^e, Σ_{gg}, $\boldsymbol{g} = [\boldsymbol{Z}^\mathsf{T}, \Delta \boldsymbol{r}^\mathsf{T}, \boldsymbol{k}^{e\mathsf{T}}]^\mathsf{T}$.

1 Partition: $[\mathsf{A}|\mathbf{a}] = \mathsf{P}$;
2 Projection centre: $\boldsymbol{Z} = -\mathsf{A}^{-1}\mathbf{a}$;
3 Normalize A: $\mathsf{A} := \mathrm{sign}(|\mathsf{A}|)\,\mathsf{A}$;
4 Decompose A: $[B, C] = \text{QR-decomposition}(\mathsf{A}^{-1})$;
5 Rotation matrix: $R = B^{-1}$;
6 Calibration matrix: $\mathsf{K} = \mathsf{C}^{-1}$;
7 Enforce correct signs of diag(K): $D = \mathrm{Diag}([\mathrm{sign}(\mathrm{diag}(\mathsf{K}))])\mathrm{Diag}([s, s, +1])$, $R := DR$, $\mathsf{K} := \mathsf{K}D$;
8 Normalized calibration matrix: $\mathsf{K}^e = \mathsf{K}/\mathsf{K}_{33}$;
9 Build Jacobian: $J_{gp} = J_{gh} J_{hp}$ from (12.149) and (12.151);
10 Covariance matrix of parameters: $\Sigma_{gg} = J_{gp} \Sigma_{\mathrm{pp}} J_{gp}^\mathsf{T}$.

The derivation of the uncertainty of the internal parameters (c, x'_H, y'_H, s, m) of the calibration is left to the reader. Alternatively, the covariance matrix can be determined using the result of Sect. 12.2.2.3, p. 501, namely A_i^T in (12.157), p. 501.

12.2.2.3 Statistically Optimal Estimation of the Pose and Calibration Parameters

Instead of taking the 12 elements of the projection matrix P, cf. Sect. 12.2.2.1, p. 494, we can directly take the 11 pose and calibration parameters as unknown variables by defining the 11-vector $\boldsymbol{p}$ as follows,

$$\boldsymbol{p} = [X_O, Y_O, Z_O,\ \omega, \phi, \kappa,\ c, x'_H, y'_H, m, s]^\mathsf{T} = [\boldsymbol{Z}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}, \boldsymbol{k}^\mathsf{T}]^\mathsf{T}\,, \tag{12.152}$$

including the six parameters of the exterior orientation and the five parameters of the interior orientation, cf. (12.34), p. 472.

Here we assume that none of the given, fixed 3D points is at infinity. For the estimation we need the Jacobian of the observed sensor coordinates $\boldsymbol{x}'_i$ w.r.t. the 11 parameters. We need to be careful when interpreting $\boldsymbol{r}$ in terms of Euler angles (cf. Sect. 12.1.3.1, p. 465) around the three axes of the camera coordinate system following the right-hand rule. We assumed the rotation of the scene coordinate system to the camera system to be $R^\mathsf{T} = R^{-1}$, cf. (12.5), p. 466. Therefore the multiplicative update with the differential rotation angles $\mathrm{d}\boldsymbol{r} = [\mathrm{d}\omega, \mathrm{d}\phi, \mathrm{d}\kappa]^\mathsf{T}$ is given by $\mathrm{d}R = -S(\mathrm{d}\boldsymbol{r})R$, cf. (12.9), p. 467.

With the function $\boldsymbol{c}$ in (5.31), the differential of the inhomogeneous sensor coordinates,

$$\boldsymbol{x}'_i = \boldsymbol{c}(\mathsf{K}R(\boldsymbol{X}_i - \boldsymbol{Z}))\,, \tag{12.153}$$

with respect to the observations and the unknown parameters yields

$$\mathrm{d}\boldsymbol{x}'_i = J_c(\widehat{\mathbf{x}}'^a_i)\,(\mathrm{d}\mathsf{K}\ R(\boldsymbol{X}_i - \boldsymbol{Z}) - \mathsf{K}\ S(\mathrm{d}\boldsymbol{r})R\ (\boldsymbol{X}_i - \boldsymbol{Z}) + \mathsf{K}R\ \mathrm{d}(\boldsymbol{X}_i - \boldsymbol{Z}))\ , \tag{12.154}$$

to be evaluated at the approximate values for K, R, and $\boldsymbol{Z}$, using the 2×3 matrix J_c from (12.129), p. 497. With $\mathbf{X}_i = [\boldsymbol{X}_i^\mathsf{T}, 1]^\mathsf{T}$ and the unnormalized camera rays ${}^c\mathbf{x}'_i = R(\boldsymbol{X}_i - \boldsymbol{Z})$, we have

$$\mathrm{d}\boldsymbol{x}'_i = J_c(\widehat{\mathbf{x}}'^a_i)\,\left(\mathrm{d}\mathsf{K}\ {}^c\mathbf{x}'_i - \mathsf{K}\ S(\mathrm{d}\boldsymbol{r})\,{}^c\mathbf{x}'_i - \mathsf{K}R\ \mathrm{d}\boldsymbol{Z}\right)\ . \tag{12.155}$$

Here we use the additive Taylor expansion $\mathsf{K} = \widehat{\mathsf{K}}^a + \widehat{\mathrm{d}\mathsf{K}}$ with the differential calibration matrix

$$\mathrm{d}\mathsf{K} = \begin{bmatrix} \mathrm{d}c & s\ \mathrm{d}c + c\ \mathrm{d}s & \mathrm{d}x'_H \\ 0 & (1+m)\ \mathrm{d}c + c\ \mathrm{d}m & \mathrm{d}y'_H \\ 0 & 0 & 0 \end{bmatrix} \tag{12.156}$$

depending on the calibration parameters $\boldsymbol{k} = [c, x'_H, y'_H, m, s]^\mathsf{T}$, cf. (12.31), p. 471.

The Jacobian $A_i^\mathsf{T} = \partial \boldsymbol{x}'_i / \partial \boldsymbol{p}$ is therefore given in

$$\mathrm{d}\boldsymbol{x}'_i = \underbrace{J_c(\widehat{\mathbf{x}}'^a_i)\,\left[-\widehat{\mathsf{K}}^a\widehat{R}^a \mid \mathsf{K}^a S({}^c\widehat{\mathbf{x}}'^a) \mid J_{\mathrm{xk}}(\widehat{c}^a, \widehat{m}^a, \widehat{s}^a; {}^c\widehat{\mathbf{x}}'^a)\right]}_{A_i^\mathsf{T}} \begin{bmatrix} \mathrm{d}\boldsymbol{Z} \\ \mathrm{d}\boldsymbol{r} \\ \mathrm{d}\boldsymbol{k} \end{bmatrix}\ , \tag{12.157}$$

with the 3×5 Jacobian of the homogeneous sensor coordinates w.r.t. the calibration parameters,

$$J_{\mathrm{xk}}(c, m, s; {}^c\mathbf{x}') = \frac{\partial \mathbf{x}'_i}{\partial \boldsymbol{k}} = \begin{bmatrix} {}^cu' + s\ {}^cv' & {}^cw' & 0 & 0 & c\ {}^cv' \\ (1+m)\ {}^cv' & 0 & {}^cw' & c\ {}^cv' & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\ . \tag{12.158}$$

The Jacobian A_i^T may be used within an iterative estimation scheme for estimating the 11 parameters of the pose and the calibration, as in the estimation procedure for the elements of the projection matrix discussed before.

We will take this model for comparing the theoretical precision of the pose estimation using an uncalibrated and a calibrated camera in Sect. 12.2.5, p. 521.

visualization of derivatives

Effect of Pose and Calibration Parameters on the Image Coordinates. The visualization of the derivatives from (12.157), p. 501 gives insight into the effect of the 11 parameters of pose and calibration on the image coordinates.

We specialize these relations for a vertical view with the following approximations: rotation matrix $R^a = I$, and an ideal camera model with $\mathsf{K}^a = \mathrm{Diag}([c, c, 1])$. We will use this vertical view for the investigation of the theoretical precision.

We obtain for (12.155)

$$\mathrm{d}\boldsymbol{x}' = J_c(\widehat{\mathbf{x}}'^a)\left(-\widehat{\mathsf{K}}^a\,\mathrm{d}\boldsymbol{Z} + \widehat{\mathsf{K}}^a S({}^c\widehat{\mathbf{x}}'^a)\mathrm{d}\boldsymbol{r} + \mathrm{d}\mathsf{K}\,{}^c\mathbf{x}'\right), \tag{12.159}$$

with

$${}^c\widehat{\mathbf{x}}'^a = \begin{bmatrix} {}^cX \\ {}^cY \\ {}^cZ \end{bmatrix} = \begin{bmatrix} X - X_O \\ Y - Y_O \\ Z - Z_O \end{bmatrix} \quad \text{and} \quad \widehat{\mathbf{x}}'^a = \begin{bmatrix} c\,{}^cX \\ c\,{}^cY \\ {}^cZ \end{bmatrix} = \begin{bmatrix} c(X - X_O) \\ c(Y - Y_O) \\ Z - Z_O \end{bmatrix}. \tag{12.160}$$

From (12.157) we obtain the Jacobian $J_{\mathbf{x}'p}$ of the homogeneous coordinates ${}^c\mathbf{x}'$ w.r.t. the parameters $\boldsymbol{p} = [\boldsymbol{Z}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}, \boldsymbol{k}^\mathsf{T}]^\mathsf{T}$,

$$J_{\mathbf{x}'p} = \frac{\partial \mathbf{x}'}{\partial \boldsymbol{p}} = \left[\begin{array}{ccc|ccc|ccccc} -c & 0 & 0 & 0 & -c\,{}^cZ & c\,{}^cY & {}^cX & {}^cZ & 0 & 0 & c\,{}^cY \\ 0 & -c & 0 & c\,{}^cZ & 0 & -c\,{}^cX & {}^cY & 0 & {}^cZ & c\,{}^cY & 0 \\ 0 & 0 & -1 & -{}^cY & {}^cX & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{12.161}$$

all elements $[{}^cX, {}^cY, {}^cZ]$ to be evaluated at the approximate values. With

$$J_c(\mathbf{x}') = \frac{1}{{}^cZ^2}\begin{bmatrix} {}^cZ & 0 & -c\,{}^cX \\ 0 & {}^cZ & -c\,{}^cY \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} {}^ix' \\ {}^iy' \end{bmatrix} = \frac{c}{{}^cZ}\begin{bmatrix} {}^cX \\ {}^cY \end{bmatrix} \tag{12.162}$$

from (12.129) and (12.19), p. 469, we therefore obtain the Jacobian $J_{x'p}$ of the inhomogeneous coordinates $\boldsymbol{x}'$ w.r.t. the parameters $\boldsymbol{p}$ by multiplication with $J_c(\mathbf{x}')$,

$$J_{x'p} = [J_{x'Z} \mid J_{x'r} \mid J_{x'k}] = J_c(\mathbf{x}')\, J_{\mathbf{x}'p}\,, \tag{12.163}$$

with the Jacobians

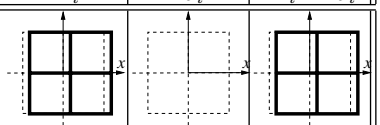
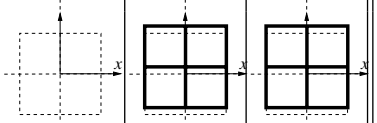
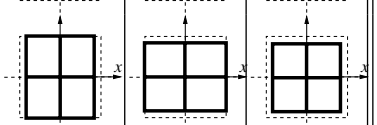
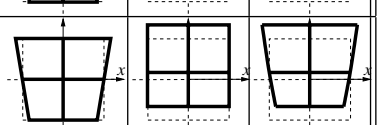
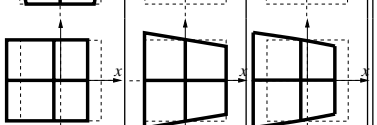
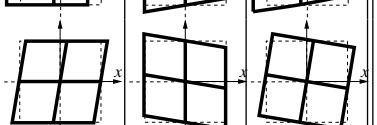
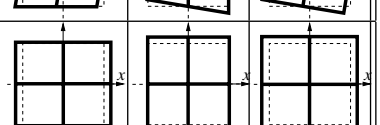
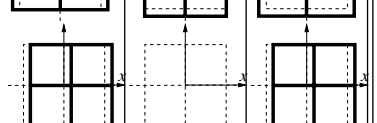
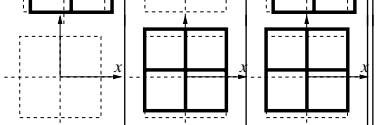
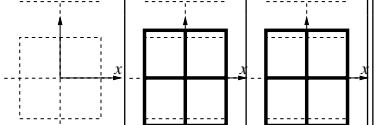
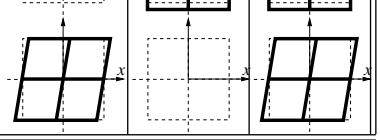
$$J_{x'Z} = \frac{1}{Z - Z_O}\begin{bmatrix} -c & 0 & {}^ix' \\ 0 & -c & {}^iy' \end{bmatrix} \tag{12.164}$$

$$J_{x'r} = \frac{1}{c}\begin{bmatrix} {}^ix'{}^iy' & -(c^2 + {}^ix'^2) & c\,{}^iy' \\ c^2 + {}^iy'^2 & -{}^ix'{}^iy' & -c\,{}^ix' \end{bmatrix} \tag{12.165}$$

$$J_{x'k} = \frac{1}{c}\begin{bmatrix} {}^ix' & c & 0 & 0 & c\,{}^iy' \\ {}^iy' & 0 & c & c\,{}^iy' & 0 \end{bmatrix}. \tag{12.166}$$

Table 12.4 shows these effects for a 3×3 grid of horizontal scene points with $Z_i = 0$, and thus ${}^cZ_i = Z_i - Z_O = -Z_O$, observed in a nadir view of an ideal camera with $Z_O > 0$ above the ground, and the image in taking position with $c > 0$.

Table 12.4 Influence of the 11 parameters of pose and calibration on the image coordinates. Ideal camera in nadir view, image in taking position (principal distance $c > 0$), common Z-coordinate of 3D points $Z_i = 0$

cause	$\mathrm{d}x'_i$	$\mathrm{d}y'_i$	$\mathrm{d}x'_i$	$\mathrm{d}y'_i$	$\mathrm{d}x'_i + \mathrm{d}y'_i$
$\mathrm{d}X_O$	$\frac{c}{Z_O}$	0			
$\mathrm{d}Y_O$	0	$\frac{c}{Z_O}$			
$\mathrm{d}Z_O$	$-\frac{x'_i}{Z_O}$	$-\frac{y'_i}{Z_O}$			
$\mathrm{d}\omega$	$\frac{x'_i y'_i}{c}$	$c\left[1+\left(\frac{y'_i}{c}\right)^2\right]$			
$\mathrm{d}\phi$	$-c\left[1+\left(\frac{x'_i}{c}\right)^2\right]$	$-\frac{x'_i y'_i}{c}$			
$\mathrm{d}\kappa$	y'_i	$-x'_i$			
$\mathrm{d}c$	$\frac{x'_i}{c}$	$\frac{y'_i}{c}$			
$\mathrm{d}x'_H$	1	0			
$\mathrm{d}y'_H$	0	1			
$\mathrm{d}m$	0	y'_i			
$\mathrm{d}s$	y'_i	0			

We observe the following:

interpretation of Jacobian for DLT

- A horizontal shift of the camera leads to a horizontal shift of the image points in the same direction.
- A vertical shift upwards leads to a shrinking of the image.
- The two angles ω and κ lead to the typical perspective distortions. A closer look reveals that the distortions derived from the first-order of the Taylor series are not straight line-preserving due to the quadratic part in $\partial\omega/\partial y'$ and $\partial\phi/\partial x'$. The linearization of the nonlinear but straight line-preserving perspective relations leads to a mapping with nonlinear distortions.

- The effects of the X_O-coordinate and of the rotation angle ϕ around the Y-axis on the x'-coordinate are similar, except for the sign. The same holds for the effect of the Y_O-coordinate and of the rotation angle ω around the X-axis on the y'-coordinate. Therefore these two pairs of parameters are not well-separable during an estimation, which will show in a high correlation, cf. Sect. 12.2.5.1, p. 522 and (12.228), p. 522.
- The rotation of the camera around the Z-axis leads to a rotation of the image in the opposite direction. This is why the affine transformation from the ideal to the sensor coordinates does not contain a rotation; thus, there are only five parameters.
- A change of the principal distance c leads to a change of the image size. Again, due to the special choice of the setup with a flat scene, the effect cannot be distinguished from the effect of changing the height Z_O of the camera.
- A shift of the principal point leads to a shift of all coordinates. In the chosen example the effect cannot be distinguished from a horizontal shift of the camera. The reason is that the scene points all have the same distance from the camera, such that the effects are proportional to the x' and y' coordinates, as $Z_i - Z_O = -Z_O$ is constant in the chosen example. Otherwise the effect, e.g., $\mathrm{d}x'_i = (c/(Z_i - Z_O))\mathrm{d}Z_O$, of a horizontal shift $\mathrm{d}Z_O$ of the camera would depend on the depth $Z_i - Z_O$.
- The scale difference m and the shear s model an affine transformation and do not correlate with one of the other parameters.

12.2.2.4 Direct Linear Transformation Derived from Lines

Lines can easily be integrated into the estimation of the projection matrix.

For observed image and scene lines $(l'_j, \mathcal{L}_j)$, respectively, we have the following constraints, cf. (7.64), p. 305:

$$\mathsf{\Gamma}^\mathsf{T}(\mathbf{L}_j)\mathsf{P}^\mathsf{T}\mathbf{l}'_j = \mathsf{\Pi}^\mathsf{T}(\mathsf{P}^\mathsf{T}\mathbf{l}'_j)\mathbf{L}_j \stackrel{!}{=} \mathbf{0}\,. \tag{12.167}$$

horizon as 3D control line

The constraint enforces the incidence $\mathcal{L}_j \in \mathcal{A}_{l'_j}$ of the scene line $\mathcal{L}_j$ and the projection plane $\mathcal{A}_{l'_j}$ with coordinates $\mathsf{P}^\mathsf{T}\mathbf{l}'_j$. The constraint can be specialized for an observed horizon. Then the projection plane needs to have the normal $[0,0,1]^\mathsf{T}$, leading to

$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \mathsf{P}^\mathsf{T}\mathbf{l}'_j = \mathbf{0}\,, \tag{12.168}$$

here assuming the 3D line to be nonstochastic.

observed scene lines

Alternatively, we obtain two independent constraints for observed scene lines,

$$A_j^\mathsf{T}\,\widehat{\mathbf{p}} \stackrel{!}{=} \mathbf{0}\,, \qquad A_j^\mathsf{T} = \mathsf{\Gamma}^{(s)}(\mathbf{L}_j) \otimes \mathbf{l}_j{}'^\mathsf{T}\,, \tag{12.169}$$

with $\widehat{\mathbf{p}} = \mathrm{vec}\widehat{\mathsf{P}}$ and

$$\mathsf{\Gamma}^{(s)}(\mathbf{L}_j) = \begin{bmatrix} \boldsymbol{e}_{j_1}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{j_2}^{[4]\mathsf{T}} \end{bmatrix} \mathsf{\Gamma}(\mathbf{L}_j)\,, \tag{12.170}$$

where the (j_1, j_2)-element in $\mathsf{\Gamma}(\mathbf{L}_j)$ is the one with the largest absolute value, cf. (7.4.1.3), p. 319.

The Jacobian $\partial \boldsymbol{g}/\partial \boldsymbol{l}$, necessary for determining the covariance matrix of the estimated projection matrix for observed lines, is

$$B = \mathrm{Diag}(\{B_\mathrm{L}^\mathsf{T}, B_{l'_j}^\mathsf{T}\}) = \mathrm{Diag}(\{\underbrace{\mathsf{\Pi}^{r\mathsf{T}}(\mathsf{P}^\mathsf{T}\mathbf{l}'_j)}_{2\times 6}, \underbrace{\mathsf{\Gamma}^{r\mathsf{T}}(\mathbf{L}_j)\mathsf{P}^\mathsf{T}}_{2\times 3}\})\,, \tag{12.171}$$

where

$$\Pi^{r\mathsf{T}}(\mathsf{P}^{\mathsf{T}}\mathbf{l}'_j) = \begin{bmatrix} \boldsymbol{e}_{j_1}^{[4]\mathsf{T}} \\ \boldsymbol{e}_{j_2}^{[4]\mathsf{T}} \end{bmatrix} \Pi^{\mathsf{T}}(\mathsf{P}^{\mathsf{T}}\mathbf{l}'_j)\,, \tag{12.172}$$

with the same indices (j_1, j_2) selecting independent rows. The covariance matrix Σ_{ll} for the observed 3D and 2D lines is a block matrix containing the covariance matrices of the observed entities,

$$\Sigma_{ll} := \mathrm{Diag}(\{\mathrm{Diag}(\{\Sigma_{\mathrm{L}_j\mathrm{L}_j}, \Sigma_{\mathrm{l}'_j\mathrm{l}'_j}\})\})\,. \tag{12.173}$$

If we just have mutually independent observed lines, the covariance matrix is obtained by

$$\Sigma_{\widehat{\mathrm{p}}\widehat{\mathrm{p}}} = \sum_{j=1}^{J} (A^+)_j \left(B_{\mathrm{L}j}^{\mathsf{T}} \Sigma_{\mathrm{L}j\mathrm{L}j} B_{\mathrm{L}j} + B_{\mathrm{l}'j}^{\mathsf{T}} \Sigma_{\mathrm{l}'j\mathrm{l}'j} B_{\mathrm{l}'j} \right)^{-1} (A^+)_j^{\mathsf{T}}\,. \tag{12.174}$$

Again, the matrix $(A^+)_j$ is the submatrix of the pseudo-inverse A^+ belonging to line j. When using both types of observations, points and lines, the Jacobians A_{X_i} and A_{L_j} of points and lines need to be concatenated to A, and the sums taken over both types of observations.

12.2.3 Modelling Distortions

In the last section, 12.2.2.3, p. 502, we analysed the effect of parameters of the exterior and the interior orientation of straight line-preserving cameras on the image coordinates.

We now want to discuss the extension of this perspective model for nonstraight line-preserving components. We refer to Sect. 12.1.4, p. 476, where we showed how to model moderate deviations from the perspective model by adding a parametrized distortion term $\Delta\boldsymbol{x}'$ to the image coordinates. Here we want to discuss the choice of this distortion term in detail. It needs to be seen in the context of estimating all parameters of the projection, both extrinsic and intrinsic. The parameters should be as distinct as possible. Distinctness can be interpreted in two ways, which we will discuss next.

12.2.3.1 Physical and Phenomenological Models

We may follow two views, a physical view or a phenomenological view. Modelling of image distortions may be based on both types of arguments, in both cases by polynomials, possibly augmented by trigonometric terms:

I. *Physical view.* The most intuitive and historically oldest argumentation starts from the physical causes of the image distortions. The effects then should be conceptually distinguishable, i.e., refer to different causes. This has the advantage of being a strong argument for a specific model.
   Its disadvantage is that in practice it is difficult to model all causes. Schilcher (1980) names around 50 of them with respect to cameras in an aeroplane, e.g., the effect of turbulence around the aeroplane or the effect of unflatness of the sensor area. They are not really analysed or are difficult to model. More importantly, the effects are similar and thus often can not be distinguished easily.
II. *Phenomenological view.* This view, proposed in the 1970s, just models the effects and does not model the causes. The corrections are intended only to eliminate or at least reduce image distortions. Thus the *calibration of a camera* should lead to statistically *independent estimates* for the parameters of the image distortions. The advantage is that one needs no physical explanation and has a large number of degrees of freedom for modelling. The choice just needs to be adequate for the task. The disadvantage is that the causes remain unknown.

The choice between models I and II may be based on the goal of the analysis: The goal of model type I is the determination or explanation of causes, as when analysing a lens in a factory. The goal of model type II is the compensation of disturbing effects and the simplest automation of the data evaluation, as when determining 3D objects. Another criterion may be the number of necessary parameters, which affects computational efficiency.

We first give representative examples for each of the models and then discuss the general setup of such models.

12.2.3.2 Physical Distortion Models

The following *physically motivated* model is given by Brown (1971). It aims at modelling unavoidable radial and tangential distortions, mainly caused by nonalignment of the lenses in the lens system.

It refers to the reduced image coordinates ${}^i\boldsymbol{x}'$, thus assumes that shear and scale difference are negligible or corrected for. It is expressed as a correction of the observed, distorted points $\boldsymbol{\mathcal{x}}'$ to obtain the undistorted points $\bar{\boldsymbol{\mathcal{x}}}'$. *Omitting the superscript ^{i} for clarity*, we have

$$\begin{aligned}\bar{x}' &= x' + \Delta x'(\boldsymbol{x}') \\ &= x' + x'(K_1' r'^2 + K_2' r'^4 + K_3' r'^6 + \ldots) \\ &\quad + (P_1'(r'^2 + 2x'^2) + 2P_2' x' y')(1 + P_3' r'^2 + \ldots)\end{aligned} \tag{12.175}$$

$$\begin{aligned}\bar{y}' &= y' + \Delta y'(\boldsymbol{x}') \\ &= y' + y'(K_1' r'^2 + K_2' r'^4 + K_3' r'^6 + \ldots) \\ &\quad + (2P_1' x' y' + P_2'(r'^2 + 2y'^2))(1 + P_3' r'^2 + \ldots)\,,\end{aligned} \tag{12.176}$$

with

$$x' := {}^i x'\,, \qquad y' := {}^i y'\,, \qquad r' = \sqrt{x'^2 + y'^2}\,, \tag{12.177}$$

where r' is the distance of the point $\boldsymbol{\mathcal{x}}'$ from the principal point. The parameters describing this model are the coordinates (x_H', y_H') of the principal point, which are required to derive the reduced image coordinates from the sensor coordinates, and the coefficients K_i' and P_i' of the polynomials. We intentionally added a prime to these parameters, as they refer to the radius r'; later we will refer to a conditioned radius, where we use the corresponding parameters without a prime.

Several remarks are necessary:

1. The polynomial series only contain even terms. This is motivated by optical theory. The first terms with parameters K_i' model *radial distortion*, mainly caused by lens distortion; the second terms with parameters P_i' model *tangential distortions*. They compensate for possible decentring of individual lenses within the lens system.
2. The polynomial for radial distortion starts with the second power of r'. A constant term K_0' would model changes in the principal distance – see the first column of $J_{x'k}$ in (12.166), p. 502 – which is responsible for Δc. This term does not model errors in the lens system and therefore has to be taken into account separately.
3. Conceptually the distortion should refer to the centre of symmetry, i.e., the intersection of the optical axis with the image plane. Brown assumes that the distortion refers to the principal point. This is motivated by the high quality of photogrammetric cameras, where the distance between the principal point and the point of symmetry is very small, say below a few pixels. Otherwise the radius would need to be determined from $r' = \sqrt{(x' - x_A')^2 + (y' - y_A')^2}$, with two additional parameters for the centre $\boldsymbol{\mathcal{x}}_A'$ of symmetry, cf. Fig. 12.4, p. 461.
4. The model has been developed for film cameras, where we can assume the image coordinate system to be Cartesian. Thus the model is only valid for small or negligible

shear and scale differences, s and m, or if these effects are taken into account separately in the calibration matrix.

5. The model starts from the observed distorted image coordinates $\boldsymbol{x}'$ and derives the corrected nondistorted image coordinates $\bar{\boldsymbol{x}}'$ which are functions of the observed values. Taking the coordinates $\boldsymbol{x}'$ as observations leads to the Gauss–Markov model given by Fraser (1997),

$$\tilde{x}' + \Delta x'(\tilde{x}', \tilde{y}') - x'_H = c\,\frac{{}^cX}{{}^cZ} \tag{12.178}$$

$$\tilde{y}' + \Delta y'(\tilde{x}', \tilde{y}') - y'_H = c\,\frac{{}^cY}{{}^cZ}\,. \tag{12.179}$$

where the corrections $\Delta\boldsymbol{x}'(\tilde{\boldsymbol{x}}')$ use the observations $\boldsymbol{x}'$ as fixed values. Since the observed image coordinates appear twice in each equation, the model actually has the form of a Gauss–Helmert model. It is only a Gauss–Markov model if used in the form

$$x' + v'_x = c\frac{{}^cX}{{}^cZ} + x'_H - \Delta x'(\bar{x}', \bar{y}') \tag{12.180}$$

$$y' + v'_y = c\frac{{}^cY}{{}^cZ} + y'_H - \Delta y'(\bar{x}', \bar{y}')\,, \tag{12.181}$$

where the observed values only appear once on the left side and the corrections $\Delta x'$ and $\Delta y'$ are derived using the *corrected* image coordinates, which depend solely on the unknown parameters. The differences in the distortion values $\Delta\boldsymbol{x}'(\bar{\boldsymbol{x}}')$ and $\Delta\boldsymbol{x}'(\boldsymbol{x}')$ generally will be small, which we will take into account when linearizing these equations in Sect. 15.4.1.2, p. 676.

The model (12.62), p. 478 derived in the previous section (12.1.4) has the form of a Gauss–Markov model, as the observable image coordinates are an explicit function of the unknown parameters.

The terms $\Delta\boldsymbol{x}'$ in (12.58) are error terms, not correction terms, so they have the opposite sign from those in (12.180).

We now give common specializations and modifications of this model (12.175). For the following discussion, we assume the perspective projection to be ideal, i.e., the coordinates of the principal point are $\mathbf{0}$. Thus we here model the nonstraight line-preserving central projection as an ideal perspective projection with $\mathsf{K} = \mathrm{Diag}([c, c, 1])$ augmented by additional lens distortions. We have the following models:

- Often only radial distortion is necessary. The model then can be written as

$$ {}^i\bar{\boldsymbol{x}}' = {}^i\boldsymbol{x}'(1 + K'_1 r'^2 + K'_2 r'^4 + K'_3 r'^6 + \ldots) \tag{12.182}$$

only using the parameters K_i. Solving for the observed coordinates,

$$ {}^i\boldsymbol{x}' = \frac{{}^i\bar{\boldsymbol{x}}'}{1 + K'_1 |{}^i\bar{\boldsymbol{x}}'|^2 + K'_2 |{}^i\bar{\boldsymbol{x}}'|^4 + \ldots}\,, \tag{12.183}$$

identifies this model, used by Scaramuzza (2008, Eqs. (2.12), (2.15)), as the general model for omnidirectional catadioptric cameras, cf. Sect. 12.1.9.1, p. 485 and also Bräuer-Burchardt and Voss (2000).
- In the most simple case the first term with K'_1 for radial distortion is sufficient, leading to the model

$$ {}^i\bar{\boldsymbol{x}}' = {}^i\boldsymbol{x}'(1 + K'_1 |{}^i\boldsymbol{x}|^2)\,. \tag{12.184}$$

- The quadratic function $1 + K'_1 r'^2$ of the radius for small enough K'_1 can be replaced by $1/(1 + \lambda r'^2)$ in a first-order approximation, with $\lambda = -K'_1$. Then we obtain the model

$$ {}^i\bar{\boldsymbol{x}}' = \frac{{}^i\boldsymbol{x}'}{1 + \lambda |{}^i\boldsymbol{x}'|^2} , \tag{12.185} $$

which is used frequently due to the ease of algebraic manipulation (cf. Lenz and Fritsch, 1990; Fitzgibbon, 2001; Kúkelová et al., 2010; Steger, 2012). It has the advantage of being algebraically invertible.

12.2.3.3 Phenomenological Distortion Models

As phenomenological distortion models are not intended to describe physical phenomena without referring to their causes, we are free to choose correction terms.

We could start by expressing the distortions as multivariate polynomials, e.g.,

$$ x' = \bar{x}' + \sum_{j=0}^{k} \sum_{l=0}^{k} a_{jl} \bar{x}'^j \bar{y}'^l , \qquad y' = \bar{y}' + \sum_{j=0}^{k} \sum_{l=0}^{k} b_{jl} \bar{x}'^j \bar{y}'^l , \tag{12.186} $$

as used by Brown (1976) for modelling the unflatness of the sensor in conjunction with the above-mentioned physical model for lens distortion.

When using higher-order terms in the mentioned models, camera calibration becomes unreliable, as the determination of the parameters becomes unstable. This indicates that the effects are empirically not really separable. This holds for the polynomials in the lens distortion model but even more for the multivariate polynomials of the phenomenological distortion model.

A classical attempt to eliminate or at least reduce these instabilities when estimating additional parameters is by using orthogonal basis functions. This was already proposed by Ebner (1976), who suggested a set of additional parameters orthogonal to a 3×3 grid of image points. The next section addresses the design of distortion functions which lead to stable estimates of the parameters.

12.2.3.4 Orthogonal Basis Functions for Modelling Systematic Errors

Orthogonal functions for compensating for image distortions are intended to obtain stable estimates, i.e., estimates with low variances and correlations. We need to address three aspects:

- Parameters for modelling image distortions should not model effects of the exterior orientation. This already is taken into account when modelling straight line-preserving cameras: the affine transformation K between the camera coordinates and the sensor coordinates only contains five, not six, parameters, i.e., it does not model a mutual rotation around the viewing direction, as rotations of the sensor w.r.t. the camera body could be compensated for by a rotation of the camera body in space.
- The parameters for modelling image distortions should not model one of the five parameters of the calibration matrix. For example, Brown's model does not contain a parameter for modelling changes of the principle distance, though this would not have required any extra effort, see the discussion above on p. 506.
- Orthogonal functions refer to a certain domain, e.g., the interval $[0, 1]$ or the unit square $[-1, +1]^2$. In our context, orthogonalization aims at a diagonal covariance matrix of the additional parameters, which depends on (1) the spatial distribution of the scene points used, (2) the assumed projection, leading to a certain distribution of the image points, (3) the assumed weights of the observed image coordinates, and (4) the number of images used for determining the parameters. Thus strict orthogonalization would only be possible for prespecified geometric configurations and assumptions about the stochastic properties of the observations.

Therefore it is useful to aim at approximate orthogonalization, which reduces the statistical dependencies of the estimated parameters. This can be achieved by the following means:

1. We assume only one image is used for calibration, as otherwise the geometric configuration of a set of images has too large an impact on the choice of orthogonal functions.
2. We need to distinguish whether the exterior orientation is known or not. The exterior orientation may be determined using a robot or an integrated measuring unit (IMU) using the position from a GPS and the rotations from an inertial sensor.
3. We assume the scene to be flat and use the normal case of the image pose, i.e., the image plane is parallel to the scene plane, as discussed in the previous Sect. 12.2.2.3, p. 502.
   This implies that, if the exterior orientation is not known, the three parameters, principal distance c and principal point $[x'_H, y'_H]$ will not be determinable.
   However, otherwise, the orthogonalization would depend on the specific three-dimensional distribution of the scene points. As camera calibration usually is performed using more that one image, this is no drawback at this point.
4. We assume the image area is homogeneously filled with image points. (This is a realistic assumption. For a detailed discussion, cf. the Sect. 15.5, p. 696 on camera calibration.) Consequently, we can use continuous image domains instead of a set of discrete regularly arranged image points, as done by Ebner (1976).
   We will use the unit square or the unit disk as the domain. This requires that the image coordinates are conditioned, e.g., such that the largest coordinate or the largest radius has value 1.
   Thus we can expect the corresponding parameters to be less correlated if the image points cover the complete image area. Eliminating the remaining correlation can be achieved with an a posteriori orthogonalization, which allows us to evaluate the orthogonalized basis functions individually.

We now address the phenomenological model, using multivariate polynomials.

12.2.3.5 Using Orthogonal Polynomials for a Phenomenological Model

We can express the image deformations with orthogonal multivariate polynomials,

$$\Delta x' = \sum_{j=0}^{k}\sum_{l=0}^{k} a_{jl} f_j(x) f_l(y)\,,\ \Delta y' = \sum_{j=0}^{k}\sum_{l=0}^{k} b_{jl} f_j(x) f_l(y) \tag{12.187}$$

as a phenomenologial distortion model.

We use conditioned image coordinates,

$$x = \frac{{}^{i}x'}{w_{xy}}\,,\qquad y = \frac{{}^{i}y'}{w_{xy}} \quad \text{with} \quad w_{xy} = \max(w_x, w_y)\,, \tag{12.188}$$

with half the maximum of the two side lengths w_x and w_y of the sensor, such that the conditioned coordinates lie in the range $[-1, +1]$.

We take Tschebyscheff polynomials as basis functions,

$$T_0(x) = 1\,,\qquad T_1(x) = x\,,\qquad T_2(x) = 2x^2 - 1 \tag{12.189}$$

$$T_3(x) = x(4x^2 - 3)\,,\quad T_4(x) = 8x^4 - 8x^2 + 1\,,\quad T_5(x) = x(16x^4 - 20x^2 + 5)\,. \tag{12.190}$$

They are orthogonal in the domain $[-1, +1]$ w.r.t. the weighting function $1/\sqrt{1-x^2}$:

$$\int_{x=-1}^{+1} \frac{T_n(x) T_{n'}(x)}{\sqrt{1-x^2}} = \begin{cases} 0 & : n \neq n' \\ \pi & : n = n' = 0 \\ \pi/2 & : n = n' \neq 0 \end{cases}. \tag{12.191}$$

Since their range also is $[-1,+1]$, the coefficients a_{ij} and b_{ij} in (12.187) can immediately be interpreted as maximum effects on the image coordinates.

If necessary, the 2D polynomials of higher-order are easily generated automatically. First we observe that the basis polynomials can be determined recursively. For example, Tschebyscheff polynomials are recursively defined by

$$T_0(x) = 1\,, \quad T_1(x) = x\,, \quad T_n(x) = 2x\,T_{n-1}(x) - T_{n-2}(x)\,, \quad n = 3, 4, \ldots\,. \tag{12.192}$$

This line of thought could also be extended to Legendre polynomials, or to Fourier series, though less easily due to the resulting mixture of polynomials and trigonometric functions.

Modelling Image Distortions for the Case of Given Exterior Orientation If the pose of the camera, i.e., the rotation matrix R, the projection centre $\boldsymbol{Z}$, and an approximate value for the principal distance c, is given, these polynomials can be taken to model the image coordinates ${}^i\boldsymbol{x}'$:

$${}^i\boldsymbol{x}' = {}^i\bar{\boldsymbol{x}}' + \Delta^i\bar{\boldsymbol{x}}'({}^i\bar{\boldsymbol{x}}')\,, \tag{12.193}$$

with ${}^i\bar{\boldsymbol{x}}' = {}^i\mathsf{K}\mathsf{R}(\boldsymbol{X} - \boldsymbol{Z})$. No immediate interpretation of the parameters is provided. However, the six parameters $a_{ij}, b_{ij}, (ij) \in \{(00), (01), (10)\}$ referring to linear distortions in the coordinates can be related to the five parameters of the calibration matrix and an additional rotation between the sensor and the camera body if the rotation matrix R refers to the camera body.

Modelling Image Distortions for the Case of Unknown Exterior Orientation. We now assume the exterior orientation of the camera is modelled together with the interior orientation. When modelling the image distortions, we need to *exclude* those effects already covered by the six parameters of the exterior orientation, or the eleven parameters of the projection matrix, see the proposals by Ebner (1976) and Grün (1978) and the discussion in Blazquez and Colomina (2010).

We start with the effects of the parameters $p_i, i = 1, ..., 6$, of the exterior orientation on the image coordinates, derived in Sect. 12.2.2.3, p. 502. Eliminating common factors, assuming the camera to have principal distance $c = 1$ and – for simplicity – the distance from the camera to the object to be $Z_O = 1$, we have the effects collected in Table 12.5 taken from Table (12.4), p. 503. The effects are modelled as functions of conditioned image

Table 12.5 Effects of exterior orientation on conditioned image coordinates up to a common scale. Principal distance $c = 1$, distance from object $Z_O = 1$

i	p_i	$\partial\Delta x/\partial p$	$\partial\Delta y/\partial p$	name
1	X_0	1	0	X_0-position
2	Y_0	0	1	Y_0-position
3	Z_0	x	y	Z_0-position
4	ω	xy	$1 + y^2$	rotation around X-axis
5	ϕ	$-(1 + x^2)$	$-xy$	rotation around Y-axis
6	κ	y	$-x$	rotation around Z-axis

coordinates, cf. (12.188), p. 509. In their linearized form they contain terms which are up to quadratic in the coordinates.

We now develop a set of correction terms based on orthogonal polynomials. If we preliminarily restrict correction terms to up to bi-quadratic polynomials, we have 18 parameters, $\{a_{ij}, b_{ij}\}, i, j \in \{0, 1, 2\}$, with the basis functions $T_0(x) = 1$, $T_1(x) = x$, and $T_2(x) = 2x^2 - 1$:

$$\Delta x' = \sum_{ij} a_{ij} T_i(x) T_j(y)\,, \quad \Delta y' = \sum_{ij} b_{ij} T_i(x) T_j(y)\,. \tag{12.194}$$

Following Ebner (1976) and starting from these Tschebyscheff polynomials, we now perform a Gram–Schmidt orthogonalization with respect to the six parameters of the exterior orientation and obtain 12 vector-valued polynomials $\boldsymbol{b}_k(x,y) = [b_{xk}(x,y); b_{yk}(x,y)], k = 4, ..., 15$, see the parameters s_4 to s_{15} in Table 12.6. They contain the two terms s_4 and s_5, representing an affine transformation which is straight line-preserving. Parameters s_{10} and s_{13} make it possible to model typical barrel or cushion type image deformations.

Following the derivation, the three effects $s_k \boldsymbol{b}_k(x,y), k = 1, 2, 3$, of a changing principal distance and a principal point are not part of this set. The reason is simply that for the derivation we assumed *flat terrain*, specifically a scene parallel to the image, which did not allow the inclusion of a shift and a scaling of the image coordinates simultaneously with a camera translation.

However, we need these three effects, as the effects of a camera translation are depth-dependent when observing nonflat scenes. Therefore we include the three parameters s_1 to s_3, a shift and a scaling. We finally obtain the following 15 terms modelling the interior orientation:

$$\Delta\boldsymbol{x}' = \sum_{k=1}^{15} s_k \boldsymbol{b}_k(x,y)\,. \tag{12.195}$$

In the case of multiple images or nonflat scenes this model allows a stable determination of the parameters s_k. We have the following orthogonality relations (1) between the parameters p_i of the exterior orientation and the parameters s_4 to s_{15} and (2) among the 15 additional parameters, i.e.,

$$\int_{x=-1}^{x=+1} \int_{y=-1}^{y=+1} \frac{\boldsymbol{b}_k^{\mathsf{T}}(x,y)\boldsymbol{b}_l(x,y)}{\sqrt{(1-x^2)(1-y^2)}} \mathrm{d}x\mathrm{d}y = 0\,, \quad k \neq l, k, l = 1, ..., 15\,. \tag{12.196}$$

The first five parameters, s_1 to s_5, correspond to the five parameters in the calibration matrix, except that the shear and the aspect ratio are modelled symmetrically here. Second-order corrections, $s_6 = q_1$ to $s_{15} = q_{10}$, for the nonstraight line-preserving part are given in Table 12.6.

Observe, the parameters p_6 and p_9 influence both coordinates x' and y', as they need to be orthogonal w.r.t. their quadratic counterparts, namely the two rotations ω and ϕ, which also influence both coordinates.

Remark: The number 15 of additional parameters can also be derived with the following argument: All 18 parameters necessary for modelling up to bi-quadratic terms can be used, except the three covered by the three rotation angles. ◇

Starting with parameter s_{10}, the polynomials $T_i(x)T_j(y)$ of the same degree $i + j$ are sorted with increasing order of $T_i(x)$ and decreasing order of $T_j(x)$, and alternate between affecting the x'- and the y'-coordinates.

Eventually, the model for additional parameters using orthogonal polynomials up to degree 2 is given by

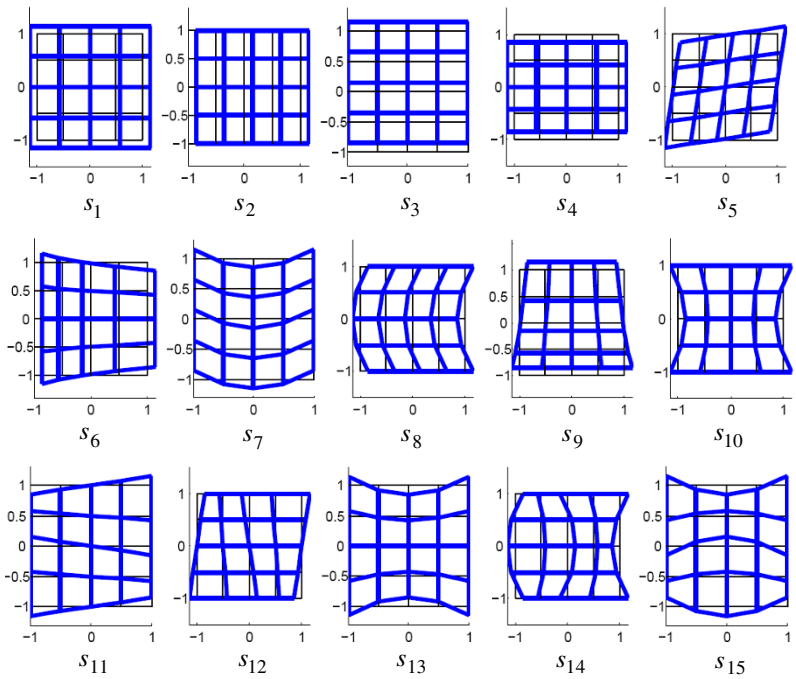
$$\begin{aligned}\Delta_{\text{phen}}x' = {} & q_1\left(2\frac{x'^2}{w^2}-1\right) + q_3\left(2\frac{y'^2}{w^2}-1\right) - q_4\frac{x'y'}{w^2} + q_5\frac{x'}{b}\left(2\frac{y'^2}{w^2}-1\right) \\ & + q_7\left(2\frac{x'^2}{w^2}-1\right)\frac{y'}{b} + q_9\left(2\frac{x'^2}{w^2}-1\right)\left(2\frac{y'^2}{w^2}-1\right)\end{aligned} \tag{12.197}$$

$$\begin{aligned}\Delta_{\text{phen}}y' = {} & q_1\frac{x'y'}{w^2} + q_2\left(2\frac{x'^2}{w^2}-1\right) + q_4\left(2\frac{y'^2}{w^2}-1\right) + q_6\frac{x'}{b}\left(2\frac{y'^2}{w^2}-1\right) + \\ & + q_8\left(2\frac{x'^2}{w^2}-1\right)\frac{y'}{b} + q_{10}\left(2\frac{x'^2}{w^2}-1\right)\left(2\frac{y'^2}{w^2}-1\right)\,.\end{aligned} \tag{12.198}$$

It can be directly compared to the original model of Ebner (1976): The two parameters for shear and scale difference are not included, the arguments in the quadratic terms

Table 12.6 Basis functions for additional parameters with up to bi-quadratic terms in the conditioned image coordinates x and y. Parameters s_1 to s_3 correspond to principal distance $s_1 = c$ and principal point $s_2 = x'_H$ and $s_3 = y'_H$. They only can be determined if the scene is not flat. Parameter s_4 models the aspect ratio, similar to m in the calibration matrix K. Parameter s_5 models the affinity, similar to the shear parameter in the calibration matrix K. The first five parameters thus guarantee an ideal perspective mapping. The next ten parameters $q_k = s_{k-5}$ are responsible for nonlinear distortions. E.g. parameters s_{10} and s_{13} can compensate for barrel or cushion-type distortions. The 12 parameters s_4 to s_{15} correspond to Ebner's set

s_k	q_k	type	$b_{xk}(x,y) = \partial\Delta x_k/\partial s_k$	$b_{yk}(x,y) = \partial\Delta y_k/\partial s_k$
s_1	–	c	x	y
s_2	–	x'_H	1	0
s_3	–	y'_H	0	1
s_4	–	aspect ratio	x	$-y$
s_5	–	shear	y	x
s_6	q_1		$2x^2-1$	xy
s_7	q_2		0	$2x^2-1$
s_8	q_3		$2y^2-1$	0
s_9	q_4		$-xy$	$(2y^2-1)$
s_{10}	q_5	barrel/cushion	$x(2y^2-1)$	0
s_{11}	q_6		0	$x(2y^2-1)$
s_{12}	q_7		$(2x^2-1)y$	
s_{13}	q_8	barrel/cushion	0	$(2x^2-1)y$
s_{14}	q_9		$(2x^2-1)(2y^2-1)$	0
s_{15}	q_{10}		0	$(2x^2-1)(2y^2-1)$



in Ebner's set are $3/2\,x'^2/w^2 - 1$ due to the difference of the domains, and finally, the numbering of the parameters is different.

Orthogonal polynomials, originally proposed for modelling image distortions by Ebner (1976), have been extended by Grün (1978) to up to order 4, leading to a set of 44 parameters. Abraham and Hau (1997) proposed Tschebyscheff polynomials and Tang et al. (2012) proposed Fourier series as basis functions for modelling distortions. Instead of using orthogonal polynomials, splines can be used also. Such a model can take local deviations into account more easily (cf. Rosebrock and Wahl, 2012).

This section completed the modelling of the projection with central cameras. Estimating all parameters of the interior orientation will be discussed in Sects. 15.4 and 15.5, p. 696 on self-calibrating bundle adjustment and calibration, respectively, where we also discuss how to decide on the adequate model to be used.

12.2.4 Spatial Resection

If the interior orientation of the camera is known and thus calibrated, we may determine the six parameters of the exterior orientation of a single image by a *spatial resection* from a set of observed scene points or lines.

In the computer vision community this sometimes is called the *perspective n-point problem* (PnP problem). It assumes n corresponding points are given in 2D and 3D for determining the relative pose of the camera and scene. This suggests that also the pose of the scene w.r.t. the camera could be determined using the spatial resection. An example for both tasks is given in Fig. 12.19.

spatial resection, PnP problem

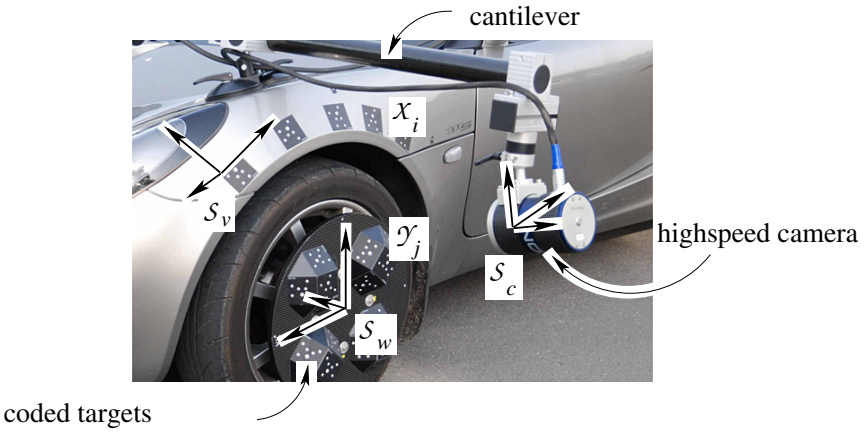


Fig. 12.19 The principle of the system *Wheelwatch* of Aicon for monitoring the rotational motion of a wheel at high speeds. The relative pose of the wheel coordinate system S_w w.r.t. the vehicle system S_v via the camera system S_c with a high speed camera rigidly fixed to the car using a cantilever is based on the image coordinates in each image showing both targeted points on the car body and targeted points on the rotating wheel. The relative pose of the camera system S_c w.r.t. the vehicle system S_v is first determined by a spatial resection with the point correspondences $(\mathcal{X}_i, \mathcal{X}_i')$. The relative pose of the wheel coordinate system S_w w.r.t. the camera coordinate system S_c can be determined by a second spatial resection with the correspondences $(\mathcal{Y}_i, \mathcal{Y}_i')$. The concatenation of the two motions yields the required relative pose of the wheel w.r.t. the car. The establishment of the correspondences is simplified by using coded targets

We discuss a direct solution for observed scene points for the minimal and the redundant case and a statistical solution for observed scene points and lines (cf. Sect. 12.2.4.3). There exists a direct solution of the pose of a calibrated camera from three observed 3D lines (cf. Dhome et al., 1989).

12.2.4.1 Minimum Solution for Spatial Resection from Points

We start with a direct solution with the minimum number of three scene points, which are observed in an image. All solutions start with determining the distances from the projection centre to the three given points. We present the one by Grunert (1841). The second step consists in determining a spatial motion from three points, which is relatively easy.

We assume that three object points $\mathcal{X}_i, i = 1,2,3$, are observed. Either we directly obtain the ray directions ${}^c\mathbf{x}'_i$, or, assuming sensor coordinates $\mathbf{x}'_i$ are given, we determine the normalized ray directions ${}^c\mathbf{x}'^s_i := -\text{sign}(c)\mathsf{N}(\mathsf{K}^{-1}\mathbf{x}'_i)$, cf. (12.108), p. 492. With the coordinates $\boldsymbol{Z}$ of the projection centre O, the rotation matrix R, depending on three parameters, and the three distances $d_i = |\boldsymbol{Z} - \boldsymbol{X}_i|$ to the control points, we obtain the relations

$$d_i {}^c\mathbf{x}'^s_i = R(\boldsymbol{X}_i - \boldsymbol{Z}), \qquad i = 1,2,3. \tag{12.199}$$

These are nine equations for the nine unknown parameters, the three coordinates of $O(\boldsymbol{Z})$, the three distances d_i, and the three parameters for R.

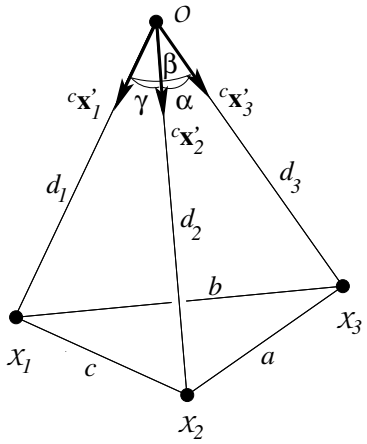


Fig. 12.20 Spatial resection with three given object points, $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$

Step 1: Determination of the Distances. From the 3D points $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ and from the direction vectors ${}^c\mathbf{x}'_i$ we can first determine the three sides a, b, and c and the three angles α, β, and γ, of the tetrahedron. The law of cosines then yields three constraints for the unknown sides d_i (Fig. 12.20),

$$\begin{aligned} a^2 &= d_2^2 + d_3^2 - 2d_2 d_3 \cos\alpha \\ b^2 &= d_3^2 + d_1^2 - 2d_3 d_1 \cos\beta \\ c^2 &= d_1^2 + d_2^2 - 2d_1 d_2 \cos\gamma\,. \end{aligned} \tag{12.200}$$

These equations are invariant to a common sign of the distances. With the substitutions

$$u = \frac{d_2}{d_1} \qquad v = \frac{d_3}{d_1} \tag{12.201}$$

we obtain from (12.200), solving for d_1^2,

$$d_1^2 = \frac{a^2}{u^2 + v^2 - 2uv\cos\alpha} \tag{12.202}$$

$$= \frac{b^2}{1 + v^2 - 2v\cos\beta} \tag{12.203}$$

$$= \frac{c^2}{1 + u^2 - 2u\cos\gamma}\,. \tag{12.204}$$

These are two constraints in u and v, namely the identity of the terms (12.202) and (12.203) and the identity of the terms (12.203) and (12.204). Therefore we may express u in terms of v quadratically; e.g., from the first two equations and after substitution into

the last two equations we obtain a polynomial of degree 4 in v which needs to vanish:

$$A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v^1 + A_0 = 0\,. \tag{12.205}$$

The coefficients (Haralick et al., 1994) depend on the known values a, b, c, α, β, and γ:

$$\begin{aligned}
A_4 &= \left(\frac{a^2-c^2}{b^2} - 1\right)^2 - \frac{4c^2}{b^2}\cos^2\alpha \\
A_3 &= 4\left\{\frac{a^2-c^2}{b^2}\left(1 - \frac{a^2-c^2}{b^2}\right)\cos\beta \right. \\
&\quad \left. + 2\frac{c^2}{b^2}\cos^2\alpha\cos\beta - \left(1 - \frac{a^2+c^2}{b^2}\right)\cos\alpha\cos\gamma\right\} \qquad (12.206)\\
A_2 &= 2\left\{\left(\frac{a^2-c^2}{b^2}\right)^2 + 2\left(\frac{a^2-c^2}{b^2}\right)^2\cos^2\beta + 2\left(\frac{b^2-c^2}{b^2}\right)\cos^2\alpha \right. \\
&\quad \left. + 2\left(\frac{b^2-a^2}{b^2}\right)\cos^2\gamma - 4\left(\frac{a^2+c^2}{b^2}\right)\cos\alpha\cos\beta\cos\gamma - 1\right\} \\
A_1 &= 4\left\{-\left(\frac{a^2-c^2}{b^2}\right)\left(1 + \frac{a^2-c^2}{b^2}\right)\cos\beta \right. \\
&\quad \left. + \frac{2a^2}{b^2}\cos^2\gamma\cos\beta - \left(1 - \left(\frac{a^2+c^2}{b^2}\right)\right)\cos\alpha\cos\gamma\right\} \qquad (12.207)\\
A_0 &= \left(1 + \frac{a^2-c^2}{b^2}\right)^2 - \frac{4a^2}{b^2}\cos^2\gamma\,.
\end{aligned}$$

After determination of up to four solutions for v, we obtain the three distances d_i via u: the distance d_1 from (12.203) and (12.204) and d_2 and d_3 from (12.201). *up to four solutions*

Step 2: Exterior Orientation. After having determined the three distances, we need to determine the six parameters of the exterior orientation of the camera.

We first determine the 3D coordinates of the three points in the camera coordinate system,

$${}^c\boldsymbol{X}_i = d_i\,{}^c\mathbf{x}_i'^s\,. \tag{12.208}$$

Then we have to determine the translation $\boldsymbol{Z}$ and the rotation R of the motion between the object and the camera system from

$${}^c\boldsymbol{X}_i = R(\boldsymbol{X}_i - \boldsymbol{Z}) \qquad i = 1,2,3\,. \tag{12.209}$$

We can use the direct solution for the similarity transformation from Sect. 10.5.4.3, p. 408, omitting the scale estimate.

Figure 12.21 shows an example with four solutions. A fourth given point correspondence can generally be used to choose the correct solution. If the angles $\angle(OX_1X_2) = \angle(OX_1X_3)$ are close to 90^o, two solutions are very close together. More configurations with multiple solutions are discussed by Fischler and Bolles (1981). A characterization of all multiple solutions is given by Gao et al. (2003). Additional direct solutions for spatial resection are collected in Haralick et al. (1994), cf. also Kneip et al. (2011).

Critical Configuration. As in the case of uncalibrated straight line-preserving cameras, here we also have a critical configuration in which ambiguous or unstable solutions are possible. If the projection centre lies on a circular cylinder, the *dangerous cylinder* (Finsterwalder, 1903), through and perpendicular to the triangle $(\mathcal{X}_1\mathcal{X}_2\mathcal{X}_3)$, the solution is unstable, corresponding to a singular normal equation matrix. In this case two solutions of the fourth-order polynomial are identical. If the projection centre is close to the critical cylinder, the solution is not very precise.

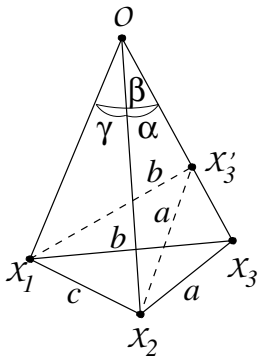


Fig. 12.21 An example of four solutions. For the pyramid shown we have $\alpha = \beta = \gamma$ and $a = b = c$. If we rotate the triangle $\mathcal{X}_1\mathcal{X}_2\mathcal{X}_3$ around $\mathcal{X}_1\mathcal{X}_2$ there exists a position where $\triangle\,\mathcal{X}_1\mathcal{X}_2\mathcal{X}_3 \equiv \triangle\,\mathcal{X}_1\mathcal{X}_2\mathcal{X}_3'$. Thus we have an additional solution. Analogously we find solutions with rotation around $\mathcal{X}_2\mathcal{X}_3$ and $\mathcal{X}_3\mathcal{X}_1$. This is why we have four positions of the triangle with respect to the projection centre and vice versa in this particular configuration

The acceptability of the configuration can again be evaluated by an analysis of the covariance matrix of the unknown parameters with respect to a reference configuration (cf. Sect. 4.6.2.3, p. 120) derived from an iterative solution, as will be shown in the next section.

Direct Determination of the Covariance Matrix of the Projection Centre. After having determined the orientation, we may want to evaluate the solution. This is especially useful if the direct solution is used in a RANSAC procedure and we are interested in evaluating the acceptability of the selected point triple.

The general technique for determining the covariance matrix of a set of unknowns from a set of observations without redundancy has been given in Sect. 2.7.5, p. 43.

Given the nonlinear relation $\boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x}) = \boldsymbol{0}$ between N observations $\boldsymbol{l} = (l_n)$, with covariance matrix Σ_{ll} and N unknown parameters $\boldsymbol{x} = (x_n)$, we obtain the covariance of the parameters using the implicit variance propagation law (cf. Sect. 2.7.5, p. 43),

$$\Sigma_{xx} = A^{-1} B \Sigma_{ll} B^{\mathsf{T}} A^{-\mathsf{T}} \quad \text{with} \quad A = \frac{\partial \boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x})}{\partial \boldsymbol{x}} \quad \text{and} \quad B = \frac{\partial \boldsymbol{g}(\boldsymbol{l}, \boldsymbol{x})}{\partial \boldsymbol{l}} \,. \tag{12.210}$$

In our case we use this relation twice: first for determining the distances $\boldsymbol{d} = [d_1, d_2, d_3]^{\mathsf{T}}$ from the given angles $\boldsymbol{\alpha} = [\alpha; \beta; \gamma]$ and the given, fixed distances $\boldsymbol{a} = [a, b, c]^{\mathsf{T}}$ between the control points using the constraints $\boldsymbol{g}_1$, and second for the determination of the coordinates $\boldsymbol{Z} = [X_O, Y_O, Z_O]^{\mathsf{T}}$ of the projection centre from the distances and the coordinates $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ of the reference points using the constraints $\boldsymbol{g}_2$:

$$\boldsymbol{g}_1(\boldsymbol{\alpha}, \boldsymbol{d}) = \begin{bmatrix} a^2 - d_2^2 - d_3^2 + 2d_2 d_3 \cos\alpha \\ b^2 - d_3^2 - d_1^2 + 2d_3 d_1 \cos\beta \\ c^2 - d_1^2 - d_2^2 + 2d_1 d_2 \cos\gamma \end{bmatrix} \,, \quad \boldsymbol{g}_2(\boldsymbol{d}, \boldsymbol{Z}) = \begin{bmatrix} d_1^2 - |\boldsymbol{Z} - \boldsymbol{X}_1|^2 \\ d_2^2 - |\boldsymbol{Z} - \boldsymbol{X}_2|^2 \\ d_3^2 - |\boldsymbol{Z} - \boldsymbol{X}_3|^2 \end{bmatrix} \,. \tag{12.211}$$

With the Jacobians

$$\begin{aligned} A_1 &= \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{d}} = \begin{bmatrix} 0 & -2d_2 + 2d_3\cos\alpha & -2d_3 + 2d_2\cos\alpha \\ -2d_1 + 2d_3\cos\beta & & -2d_3 + 2d_1\cos\beta \\ -2d_1 + 2d_2\cos\gamma & -2d_2 + 2d_1\cos\gamma & 0 \end{bmatrix} \\ B_1 &= \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\alpha}} = \begin{bmatrix} -2d_2 d_3 \sin\alpha & 0 & 0 \\ 0 & -2d_1 d_3 \sin\beta & 0 \\ 0 & 0 & -2d_1 d_2 \sin\gamma \end{bmatrix} \end{aligned} \tag{12.212}$$

and, similarly,

$$A_2 = \begin{bmatrix} 2(X_1 - X_O) & 2(Y_1 - Y_O) & 2(Z_1 - Z_O) \\ 2(X_2 - X_O) & 2(Y_2 - Y_O) & 2(Z_2 - Z_O) \\ 2(X_3 - X_O) & 2(Y_3 - Y_O) & 2(Z_3 - Z_O) \end{bmatrix} \quad B_2 = \begin{bmatrix} 2d_1 & 0 & 0 \\ 0 & 2d_2 & 0 \\ 0 & 0 & 2d_3 \end{bmatrix}, \tag{12.213}$$

we obtain the covariance matrix for the distances

$$\Sigma_{dd} = A_1^{-1} B_1 \Sigma_{\alpha\alpha} B_1^{\mathsf{T}} A_1^{-\mathsf{T}}, \tag{12.214}$$

and therefore the covariance matrix of the projection centre,

$$\Sigma_{ZZ} = A_2^{-1} B_2 A_1^{-1} B_1 \Sigma_{\alpha\alpha} B_1^{\mathsf{T}} A_1^{-\mathsf{T}} B_2^{\mathsf{T}} A_2^{-\mathsf{T}}. \tag{12.215}$$

Example 12.2.44: ***Uncertainty of projection centre from spatial resection.*** Assume the coordinates of three control points, $\mathcal{X}_1 : [-1000, 0, 0]$ m, $\mathcal{X}_2 : [500, 866, 0]$ m and $\mathcal{X}_3 : [500, -866, 0]$ m, are evenly distributed on a horizontal circle with radius 1000 m and centre, $[80, 0, 0]$ m (Fig. 12.22). The projection centre of an aerial camera with principal distance $c = 1500$ pixel is at a height of 1500 m. The image coordinates are assumed to have a precision of 0.1 pixel, which leads to standard deviations of the angles of approximately $\sigma_\alpha = \sqrt{2} \cdot 0.01/150 \approx 0.0001$ radian.

We want to discuss four positions of the projection centre with respect to the critical cylinder, and give the covariance matrix of the projection centre, the standard deviations of its coordinates, the point error $\sigma_P = \sqrt{\sigma_{X_O}^2 + \sigma_{Y_O}^2 + \sigma_{Z_O}^2}$ and the maximum loss $\sqrt{\lambda_{\max}}$ in precision with respect to an ideal situation, namely the projection centre, which is above the midpoint of the triangle; here $\lambda_{\max}$ is the maximum eigenvalue of the generalized eigenvalue problem, $|\Sigma_{ZZ} - \lambda \Sigma_{ZZ}^{\text{ref}}| = 0$ (cf. (4.263), p. 121).

1. $\mathcal{Z} : [0, 0, 1500]$ m: This is the reference situation.

$$\Sigma_{ZZ}^{(\text{ref})} = \begin{bmatrix} 0.1043 & & \\ & 0.1043 & \\ & & 0.005796 \end{bmatrix} \text{m}^2, \quad \begin{bmatrix} \sigma_{X_O} \\ \sigma_{Y_O} \\ \sigma_{Z_O} \end{bmatrix} = \begin{bmatrix} 0.32 \text{ m} \\ 0.32 \text{ m} \\ 0.076 \text{ m} \end{bmatrix}, \quad \sigma_P = 0.46 \text{ m}.$$

   All coordinates can be determined quite accurately. We choose this covariance matrix to be the reference.

2. $\mathcal{Z} : [-900, 0, 1500]$ m: Here the projection centre is 100 m away from the critical cylinder, i.e.,

$$\Sigma_{ZZ} = \begin{bmatrix} 7.875 & & 6.692 \\ & 0.06064 & \\ 6.692 & & 5.699 \end{bmatrix} \text{m}^2, \quad \begin{bmatrix} \sigma_{X_O} \\ \sigma_{Y_O} \\ \sigma_{Z_O} \end{bmatrix} = \begin{bmatrix} 2.8 \text{ m} \\ 0.25 \text{ m} \\ 2.4 \text{ m} \end{bmatrix}, \quad \sigma_P = 3.7 \text{ m}, \quad \sqrt{\lambda_{\max}} = 32.$$

3. $\mathcal{Z} : [900, 0, 1500]$ m:

$$\Sigma_{ZZ} = \begin{bmatrix} 0.05858 & & -0.03301 \\ & 12.44 & \\ -0.03301 & & 0.02929 \end{bmatrix} \text{m}^2, \quad \begin{bmatrix} \sigma_{X_O} \\ \sigma_{Y_O} \\ \sigma_{Z_O} \end{bmatrix} = \begin{bmatrix} 0.24 \text{ m} \\ 3.5 \text{ m} \\ 0.17 \text{ m} \end{bmatrix}, \quad \sigma_P = 3.5 \text{ m}, \quad \sqrt{\lambda_{\max}} = 11.$$

4. $\mathcal{Z} : [1000, 900, 1500]$ m:

$$\Sigma_{ZZ} = \begin{bmatrix} 6.585 & 4.932 & -4.194 \\ 4.932 & 3.785 & -3.203 \\ -4.194 & -3.203 & 2.725 \end{bmatrix} \text{m}^2, \quad \begin{bmatrix} \sigma_{X_O} \\ \sigma_{Y_O} \\ \sigma_{Z_O} \end{bmatrix} = \begin{bmatrix} 2.6 \text{ m} \\ 1.9 \text{ m} \\ 1.7 \text{ m} \end{bmatrix}, \quad \sigma_P = 3.6 \text{ m}, \quad \sqrt{\lambda_{\max}} = 24.$$

The contour lines of $\sqrt{\lambda_{\max}}$ are given in Fig. 12.22, left. Observe that only within a region of approximately half the diameter of the circle through $(\mathcal{X}_1\mathcal{X}_2\mathcal{X}_3)$ is the loss in precision below a factor of 10. In Fig. 12.22, right, the situation is given for the case where point $\mathcal{X}_1 : [1000, 0, 0]$ m is chosen. The best results are achieved if the projection centre is above the midpoint of $\mathcal{X}_2$ and $\mathcal{X}_3$, although this is still approximately six times worse than the reference configuration.

Both figures confirm the critical cylinder to be a degenerate configuration. Moreover, they show how far away one needs to be from that critical situation in order to achieve a stable result. This just requires knowing the theoretical covariance matrix of the result which can be determined in *all* cases and compared to some prespecified precision. Generally, we might not have such a reference covariance matrix at hand. But it may be derived from an ideal, possibly application-dependent, distribution of the three 3D points using the knowledge of the interior orientation of the camera. This is an a posterior type of quality analysis which follows the general principles of Sect. 4.6, p. 115, and which can be realied for all other geometric estimation problems discussed in this part. We will perform such an analysis for comparing the DLT and

evaluation of final estimates

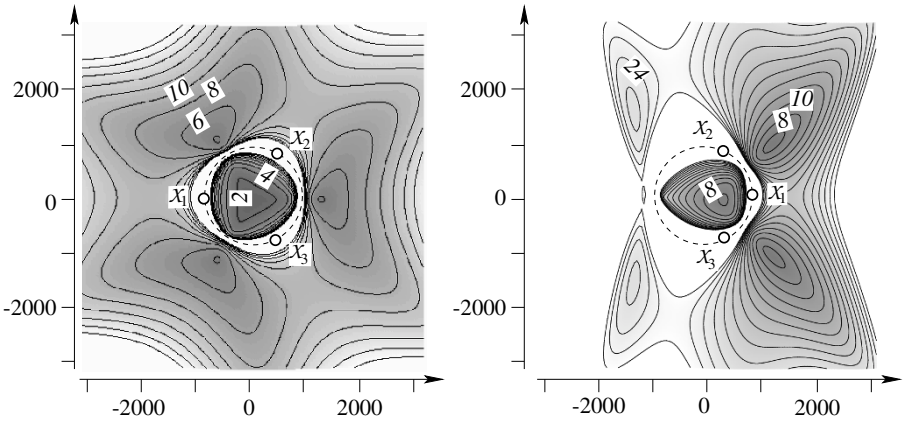


Fig. 12.22 Loss in precision for spatial resection with three given points for varying positions of the projection centre. The symmetric configuration of $X_1X_2X_3$ with the projection centre above their midpoint serves as reference configuration. Contour lines of $\sqrt{\lambda_{\max}}$ are shown. **Left:** symmetric configuration of control points, maximum value 20. **Right:** asymmetric configuration of control points, maximum value 30; distance of isolines is 4. In the white area the values are beyond the maximum. Only at positions far off the critical cylinder can acceptable configurations be achieved: The reference configuration has the projection centre in the middle of an equilateral triangle; therefore, the minimum value of $\sqrt{\lambda_{\max}} = 1$ is at the centre of the circle. Outside the critical cylinder the best values are approximately 4, meaning that the precision is four times worse than in the reference case. In the right configuration the best value for $\sqrt{\lambda_{\max}}$ is approximately 5 and is achieved in the middle between X_2 and X_3

the spatial resection in an overdetermined situation and supplement it with a reliability analysis of the relative orientation of two images. ◇

12.2.4.2 Direct Solution for the Spatial Resection with Four or More Points

The overdetermined spatial resection (PnP), based on more than three observed scene points or lines, though being nonlinear, has direct, i.e., noniterative, solutions. Following the review in Moreno-Noguer et al. (2007) one of the first direct algorithms is given by Quan and Lan (1999). Moreno-Noguer et al. (2007) also provide a fast algorithm, which later was extended to integrate covariance information for the image points (Ferraz et al., 2014) and to handle outliers (Ferraz et al., 2014). A solution based on Gröbner bases is given by Zheng et al. (2013).

The goal of the solution by Moreno-Noguer et al. (2007) is to derive the pose $(R, \boldsymbol{Z})$ of a calibrated camera from I pairs $(\mathcal{X}_i, \mathcal{X}_i)$ related by ${}^c\mathbf{x}'_i = d_i R(\boldsymbol{X}_i - \boldsymbol{Z})$ using a linear relationship between the given observations and some unknown parameters, similar to the DLT, cf. Sect. 12.2.2.1, p. 494. The basic idea is to represent the coordinates, $\boldsymbol{X}_i$, of the 3D points using barycentric coordinates w.r.t. four reference points $\boldsymbol{Y}_j, j = 1, ..., 4$, in the scene coordinate system. In a similar fashion we represent the camera coordinates, ${}^c\boldsymbol{X}_i = R(\boldsymbol{X}_i - \boldsymbol{Z})$, of the scene points with the same barycentric coordinates but with unknown coordinates ${}^c\boldsymbol{Y}_j$ of the reference points since the barycentric coordinates are invariant w.r.t. a spatial motion. The projection relation ${}^c\mathbf{x}'_i = d_i\, {}^c\boldsymbol{X}_i$ then can be used to determine the unknown reference coordinates ${}^c\boldsymbol{Y}_j$ in the camera coordinate system and the parameters R and $\boldsymbol{Z}$ from the now available correspondences $({}^c\boldsymbol{X}_i, \boldsymbol{X}_i)$. We describe the principle, which is the basis for the mentioned algorithm.

With the reference points $Y = [\boldsymbol{Y}_1, ..., \boldsymbol{Y}_4]$, and vectors, $\boldsymbol{\alpha}_i = [\alpha_{i1}, ..., \alpha_{i4}]^\mathsf{T}$, of the barycentric coordinates of each point we have

$$\boldsymbol{X}_i = \sum_{j=1}^{4} \alpha_{ij} \boldsymbol{Y}_j = Y \boldsymbol{\alpha}_i \,. \tag{12.216}$$

The reference coordinates are chosen such that the absolute values of the barycentric coordinates are below or around one, e.g., the centroid $\mathcal{Y}_1$ of the given points $\mathcal{X}_i, i = 1, ..., I$, and additional three points $\mathcal{Y}_j, j = 2, 3, 4$, in the direction of the three coordinate axes (XYZ) . Then the barycentric coordinates can be uniquely derived for each point using Euclideanly normalized homogeneous coordinates via $\boldsymbol{\alpha}_i = \mathsf{Y}^{-1}\mathbf{X}_i$. The same representation is used for the camera coordinates ${}^c\boldsymbol{X}_i$ of the given points, however now with unknown reference coordinates. We need two representations for these: Exercise 12.14

$$ {}^c\mathsf{Y} = [{}^c\boldsymbol{Y}_1, ..., {}^c\boldsymbol{Y}_4] \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \boldsymbol{y}_4 \end{bmatrix} = \operatorname{vec}({}^c\mathsf{Y}) = \begin{bmatrix} {}^c\boldsymbol{Y}_1 \\ {}^c\boldsymbol{Y}_2 \\ {}^c\boldsymbol{Y}_3 \\ {}^c\boldsymbol{Y}_4 \end{bmatrix} . \tag{12.217} $$

Thus we also have

$$ {}^c\boldsymbol{X}_i = \sum_{j=1}^{4} \alpha_{ij} {}^c\boldsymbol{Y}_j = {}^c\mathsf{Y}\boldsymbol{\alpha}_i . \tag{12.218} $$

Using this representation for ${}^c\boldsymbol{X}_i$, the constraint for the projection ${}^c\mathbf{x}'_i = d_i \, {}^c\boldsymbol{X}_i$ can be written as

$$ {}^c\mathbf{x}'_i \times {}^c\boldsymbol{X}_i = \mathsf{S}({}^c\mathbf{x}'_i) \, {}^c\mathsf{Y} \, \boldsymbol{\alpha}_i = \left(\boldsymbol{\alpha}_i^{\mathsf{T}} \otimes \mathsf{S}({}^c\mathbf{x}'_i)\right) \boldsymbol{y} = \underset{3\times12}{M_i^{\mathsf{T}}} \, \underset{12\times1}{\boldsymbol{y}} = \underset{3\times1}{\mathbf{0}} , \tag{12.219} $$

similar to (12.121) and (12.122), p. 495.

Since the barycentric coordinates, $\boldsymbol{\alpha}_i$, and the observed ray directions, ${}^c\mathbf{x}'_i$, are known, the matrices M_i are known and we can derive $\boldsymbol{y}$ by determining the eigenvector of the 12×12 matrix

$$ A = \sum_{i=1}^{I} M_i M_i^{\mathsf{T}} \tag{12.220} $$

that is elonging to the smallest eigenvalue. This solution requires at least six points.

Let this eigenvector be $\boldsymbol{y} = \operatorname{vec}\left([{}^c\boldsymbol{Y}_1, {}^c\boldsymbol{Y}_2, {}^c\boldsymbol{Y}_3, {}^c\boldsymbol{Y}_4]\right)$. It contains the estimated camera coordinates of the four reference points, however, arbitrarily scaled, namely to 1 instead according to the scale of the reference points $\boldsymbol{Y}_j$. With (12.218) this yields arbitrarily scaled camera coordinates for all given points. Thus, from all given correspondences we solve for the sought rotation R and translation $\boldsymbol{Z}$, together with the scale γ, using the similarity transformation

$$ \gamma \, {}^c\boldsymbol{X}_i = R(\boldsymbol{X}_i - \boldsymbol{Z}) , \quad i = 1, ..., I \tag{12.221} $$

This an be found directly using Alg. 13, p. 411.

The three constraints per observed point (12.219) are linearly dependent; thus, two independent constraints per point could be selected, e.g., replacing $\mathsf{S}({}^c\mathbf{x}'_i)$ by $\mathsf{S}^{(s)}({}^c\mathbf{x}'_i)$, cf. Sect. 7.4.1, p. 317. For perspective cameras with ${}^cw'_i \neq 0$ the first two rows can be selected as in Moreno-Noguer et al. (2007). This may slightly decrease the computational effort in (12.220).

As experiments of Moreno-Noguer et al. (2007) show, often more than one eigenvalue of A is small, however, not more than $N = 4$ (cf. Lepetit et al., 2009). Thus $\boldsymbol{y}$ is a weighted sum of the N smallest eigenvectors. This problem can be solved by integrating the estimation of the weights into the similarity transformation. While Moreno-Noguer et al. (2007) give a direct solution for the determination of N and the weighting of the N eigenvectors, Ferraz et al. (2014) propose to fix $N = 4$ and iteratively estimate the weights and thus the unknown coordinates $\boldsymbol{y}$. This way, they can handle four and more points. They also show how to handle outliers. The eigenvectors of A are then determined *robustly* which turns out to be significantly faster than RANSAC procedures. In order to arrive at a (nonrobust) maximum likelihood estimation Ferraz et al. (2014) replace the

determination of the similarity transformation in (12.221) by minimizing the reprojection errors $\boldsymbol{v}^{\mathsf{T}}(\boldsymbol{y})\, W_{ll}\, \boldsymbol{v}(\boldsymbol{y})$ under the constraint $A\boldsymbol{y} = \mathbf{0}$.

12.2.4.3 Iterative Solution for Spatial Resection for a Central Camera

In this section we provide an iterative solution for the overconstrained spatial resection. This solution not only allows us to include very far points or points at infinity. Such scene elements may be very helpful, especially for determining the rotation of the camera. The solution also yields the covariance matrix of the resulting parameters. This can be used for evaluation of the result and for planning purposes. Last not least, the solution is the basis for the orientation of two or more images with the bundle adjustment discussed in Chap. 15, p. 643.

As we know the internal parameters of the camera, we can exploit them to derive viewing rays from observed image points or viewing planes from observed scene lines, as already done for the direct solution. Using viewing rays and planes as basic observations allows us to use the solution for the pose determination of any kind of camera with central projection.

Linearized Observation Equations for Points. Assume that we are given $I \geq 3$ 3D points $\mathcal{X}_i(\mathbf{X}_i)$, which are fixed, i.e., nonstochastic, values. They are given with homogeneous coordinates $\mathbf{X}_i = [\boldsymbol{X}_{0i}^{\mathsf{T}}, X_{hi}]^{\mathsf{T}}$, allowing for points at infinity. They are assumed to be observed, leading to camera rays $\mathcal{K}_i'({}^c\mathbf{x}_i', \Sigma_{{}^c\mathbf{x}'_i{}^c\mathbf{x}'_i})$ represented by homogeneous coordinates and their covariance matrix. We assume all image and scene points to be conditioned and spherically normalized. For simplicity we omit the superscript ^{s} indicating the spherical normalization.

Similarly to (12.128), p. 497, the imaging model then can be written as

$$\mathbb{E}({}^c\underline{\mathbf{x}}_i') = \mathsf{N}(\tilde{R}[I_3 \mid -\tilde{\boldsymbol{Z}}]\,\mathbf{X}_i)\,, \quad \mathbb{D}({}^c\underline{\mathbf{x}}_i') = \Sigma_{{}^c\mathbf{x}_i'{}^c\mathbf{x}_i'}\,, \quad i = 1, ..., I\,, \tag{12.222}$$

where we used the normalization operator $\mathsf{N}(\mathbf{x}) = \mathbf{x}/|\mathbf{x}|$ for fixing the scale.

This is a nonlinear Gauss–Markov model. We have two observations per image point, so that we need to reduce the three constraints to two. If we have approximate values ${}^c\widehat{\mathbf{x}}_i'^a$ for the camera rays we can project the constraints onto the tangent space of the spherically normalized camera rays and obtain the nonlinear constraints (cf. Sect. 10.2.2.1, p. 369)

$$\Delta^c\boldsymbol{x}_{ri}' + \widehat{\boldsymbol{v}}_i = J_r^{\mathsf{T}}({}^c\widehat{\mathbf{x}}_i'^a)\,\mathsf{N}(\widehat{R}[I_3 \mid -\widehat{\boldsymbol{Z}}]\,\mathbf{X}_i)\,, \quad \mathbb{D}({}^c\underline{\mathbf{x}}_i') = \Sigma_{{}^c\mathbf{x}_i'{}^c\mathbf{x}_i'} \quad i = 1, ..., I\,, \tag{12.223}$$

with the reduced observations

$$\Delta^c\boldsymbol{x}_{ri}' = J_r^{\mathsf{T}}({}^c\widehat{\mathbf{x}}_i'^a)\,{}^c\mathbf{x}_i' \quad \text{with} \quad J_r(\mathbf{x}) = \mathsf{null}(\mathbf{x}^{\mathsf{T}})\,. \tag{12.224}$$

Using the multiplicative update of rotations $R = R(\Delta\boldsymbol{r})R^a$ and writing the argument of the normalized vector as $\widehat{R}[I_3 \mid -\widehat{\boldsymbol{Z}}]\,\mathbf{X}_i = \widehat{R}(\boldsymbol{X}_{0i} - X_{hi}\widehat{\boldsymbol{Z}})$ yields the observation equations for the linear substitute Gauss–Markov model, now making the spherical normalized vector ${}^c\mathbf{x}_i'^s$ explicit,

$$\Delta^c\boldsymbol{x}_{ri}' + \widehat{\boldsymbol{v}}_i = \frac{\partial^c\boldsymbol{x}_{ri}'}{\partial{}^c\mathbf{x}_i'^s}\frac{\partial{}^c\mathbf{x}_i'^s}{\partial{}^c\mathbf{x}_i'}\left(\frac{\partial{}^c\mathbf{x}_i'^s}{\partial\boldsymbol{r}}\widehat{\Delta\boldsymbol{r}} + \frac{\partial{}^c\mathbf{x}_i'^s}{\partial\boldsymbol{Z}}\widehat{\Delta\boldsymbol{Z}}\right) \tag{12.225}$$

$$= \underbrace{J_r^{\mathsf{T}}({}^c\widehat{\mathbf{x}}_i'^a)\,J_s({}^c\widehat{\mathbf{x}}_i'^a)\left[-X_{hi}\widehat{R}^a \mid -S\left(\widehat{R}^a(\boldsymbol{X}_{0i} - X_{hi}\widehat{\boldsymbol{Z}}^a)\right)\right]}_{A_i^{\mathsf{T}}}\begin{bmatrix}\widehat{\Delta\boldsymbol{Z}}\\ \widehat{\Delta\boldsymbol{r}}\end{bmatrix}\,, \tag{12.226}$$

and $J_s(\mathbf{x}) = (I_3 - \mathbf{x}\mathbf{x}^\mathsf{T}/|\mathbf{x}|^2))/|\mathbf{x}|$, cf. (10.18), p. 368. This linearized observation equation with the 2×6 Jacobian A_i^T per image point can be used within an iterative estimation scheme.

Exercise 12.13

There exists a critical configuration: If all points, including the projection centre, lie on a horopter curve (Buchanan, 1988; Wrobel, 2001; Hartley and Zisserman, 2000; Faugeras and Luong, 2001), the solution is ambiguous.

critical configuration

For observed scene lines $\mathcal{L}_j(\mathbf{L}_j)$ we can use the following model:

observed scene lines

$$\mathbb{E}({}^c\underline{\mathbf{l}}'^s_j) = \mathsf{N}\left(\widehat{R}\,[S^\mathsf{T}(\widehat{\boldsymbol{Z}}) \mid I_3]\,\mathbf{L}_j\right), \quad \mathbb{D}({}^c\underline{\mathbf{l}}'^s_j) = \Sigma_{\underline{\mathbf{l}}'^s_j \underline{\mathbf{l}}'^s_j}, \quad j = 1, \ldots, J, \tag{12.227}$$

cf. (12.76), p. 481 and Luxen and Förstner (2001); Ansar and Daniilidis (2002); Mirzaei and Roumeliotis (2011).

12.2.5 Theoretical Precision of Pose Estimation

We give the theoretical precision of the orientation parameters for a standard case. We parametrize the configuration, and, as for the direct solution of the spatial resection, analyse the theoretical precision algebraically.

We assume a block with eight corners is observed with a nadir view (cf. Fig. 12.23). We parametrize the block in relation to the height Z_O of the projection centre above the

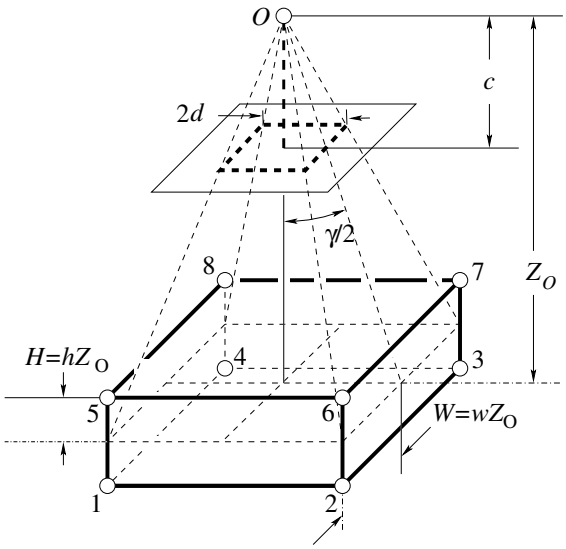


Fig. 12.23 Eight point configuration for the orientation of a camera

centre of the block and the two factors h and w defining the height and the width of the block. Referring to the centre of the block, the height of the projection centre is Z_O, the principal distance of the camera is c, and the width and depth of the block is $2W = 2wZ_O$, related to the angle γ under which the block is visible by $W = Z_O \tan(\gamma/2)$. The height of the block is $2H = 2hZ_O$. Observe that $h < 1$, as otherwise the projection centre is inside the block. The image coordinates are assumed to be measured with a standard deviation of $\sigma_0 = \sigma_{x'} = \sigma_{y'}$.

12.2.5.1 Theoretical Precision Using an Uncalibrated Camera

We start with the pose determination using an uncalibrated camera. From the observation equations (12.120), p. 495 we can algebraically derive the normal equation matrix and its inverse using an algebra package such as MAPLE. We obtain the theoretical standard deviations for the exterior orientation, for scale difference m, and for shear s between the sensor coordinate axes,

$$\begin{aligned}
\sigma_{X_O} = \sigma_{Y_O} &= \frac{\sqrt{2}}{4}\frac{|1-h^2|}{h}\sqrt{\frac{(1+h^2)^2+8h^2}{(1+h^2)^2+4h^2}}\frac{Z_O}{c}\sigma_0 \\
\sigma_{Z_O} &= \frac{1}{4}\frac{|1-h^2|}{h}\frac{\sqrt{1+h^2}}{h}\frac{Z_O}{c}\sigma_0 \\
\sigma_\omega = \sigma_\phi &= \frac{\sqrt{2}}{4}\frac{(1-h^2)^2}{\sqrt{(1+h^2)^2+4h^2}}\frac{1}{c}\sigma_0 \\
\sigma_\kappa = \frac{1}{2}\sigma_m = \frac{1}{2}\sigma_s &= \frac{\sqrt{2}}{4}\frac{1}{w}\frac{|1-h^2|}{\sqrt{1+h^2}}\frac{1}{c}\sigma_0\,.
\end{aligned}$$

The expressions for the other parameters of the interior orientation are lengthy and therefore not given. The height factor $h > 0$ of the block needs to be nonzero for the projection centre to be determinable, confirming the critical configuration when all points are coplanar. The precision of the coordinates of the projection centre is given in Fig. 12.2.5.1. Obviously, the precision of the Z_O coordinate is always worse than the precision of the X_O and Y_O coordinates.

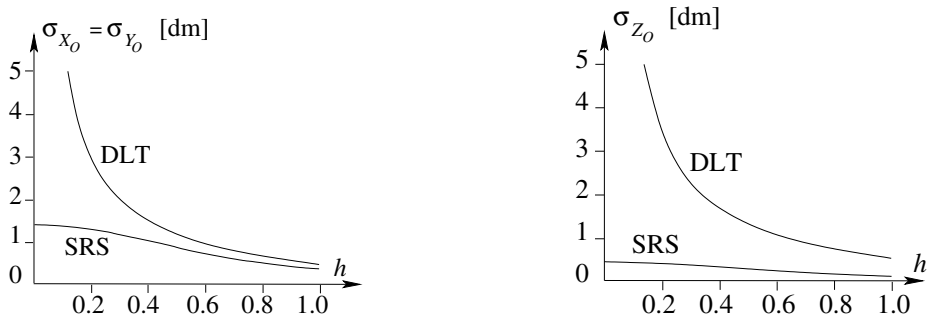


Fig. 12.24 Theoretical precision of the projection centre from the orientation of a camera given the observations of the eight corners of a block. Flying height over ground 1500 m, principal distance $c = 1500$ pixel, $\sigma_0 = 0.1$ pixel, half the width of the image of the block $d = 9.2$ cm. **Left:** precision in XY. **Right:** precision in Z. DLT: direct estimation of the projection matrix with DLT, i.e., 11 parameters for a straight line-preserving camera, SRS: spatial resection, i.e., six parameters for a calibrated camera. The factor h is the ratio H/Z_O of the height of the block in units to the distance Z_O

The large standard deviations for the projection centre result from large correlations with the rotation angles and the principal distance. We have

$$\rho_{X_O\phi} = -\rho_{Y_O\omega} = \frac{-2h}{\sqrt{(1+h^2)^2+8h^4}}\,, \quad \rho_{Z_Oc} = \frac{1+3h^2}{\sqrt{1+8h^2+5h^4+2h^6}}\,. \tag{12.228}$$

The maximal correlation $|\rho_{X_O\phi}|$ between X_O and ϕ is 0.5, which is acceptable. For heights H of the block larger than half of the flying height Z_O, the correlation between the position X_O of the projection centre and the principal point x'_H is less than 0.9. However, only for values $h > 1.5$, which is rarely used, is the correlation $|\rho_{Z_Oc}|$ of the distance Z_O between object and camera and the principal distance c better than 0.95.

12.2.5.2 Theoretical Precision Using a Calibrated Camera

We now give the theoretical precision for the exterior orientation of a vertical view using a calibrated camera. Unfortunately, the expressions for the standard deviations are lengthy. For simplification, we therefore assume all points to lie in one plane; thus, the height of the block is assumed to be 0. This corresponds to measuring *double points in the four corners of a square*. Its size in the image is $2d \times 2d$ (see Fig. 12.23, p. 521). We obtain the following expressions for the standard deviations:

$$\sigma_{X_O} = \sigma_{Y_O} = \frac{\sqrt{2}}{4} \frac{Z_O}{c} \sqrt{1 + \frac{1}{\sin^4 \frac{\gamma}{2}}} \quad \sigma_0 \tag{12.229}$$

$$\sigma_{Z_O} = \frac{Z_O}{4d} \quad \sigma_0 \tag{12.230}$$

$$\sigma_\omega = \sigma_\phi = \frac{\sqrt{2}}{4} \frac{c}{d^2} \quad \sigma_0 \tag{12.231}$$

$$\sigma_\kappa = \frac{1}{4d} \quad \sigma_0 \,. \tag{12.232}$$

We now just have correlations between the lateral position (X_O, Y_O) and the angles ω and ϕ, namely

$$\rho_{X_O\phi} = -\rho_{Y_O\omega} = \frac{1}{\sqrt{1 + \sin^4 \frac{\gamma}{2}}} \,. \tag{12.233}$$

For viewing angles of $\gamma = 90°$ the correlations approach 0.9, while for viewing angles below 45^o the correlations are nearly 0.99. The standard deviation of the lateral position of the camera determined with a spatial resection is always larger than the standard deviation of the distance between object and camera by a factor larger than 3.1.

precision of DLT is worse than that of spatial resection

precision of DLT increases with relative height variations

Compared to the precision obtainable with the DLT, the superiority of the spatial resection is obvious: In order to obtain the same height accuracy σ_{Z_O} as with the spatial resection solution, we need to exploit the full depth, namely using a cube with side length Z_O. In order to obtain the same lateral precision $\sigma_{X_O} = \sigma_{Y_O}$ we need to have a spatial object with a height of at least $H > 0.6\, Z_O$ when using the DLT. On the other hand, with the DLT, already small height differences H lead to standard deviations which are only approximately larger, by a factor Z_0/H, than those of the spatial resection. This might be sufficient in certain applications.

12.3 Inverse Perspective and 3D Information from a Single Image

12.3.1 Uncertainty of Projection Rays 524
12.3.2 Reconstructing Points on Planes 524
12.3.3 Position of a 3D Line Segment Using Triangulation 528
12.3.4 Using Vanishing Points 529
12.3.5 3D Circle Reconstructed from Its Image 534

Inverse perspective covers all aspects of recovering partial information of the scene or of the camera from a single image. Methods of inverse perspective have been extensively studied by Criminisi (2001), including all types of 3D measurements. We provide methods of inverse perspective based on image points, lines or conics, illustrate the use of the cross ratio, and finally present a method for recovering the pose of a 3D circle from its image. For the reconstruction of 3D shapes from line drawings, cf. Sugihara (1986); Cooper (1993); Varley and Martin (2002).

12.3.1 Uncertainty of Projection Rays

Given an image point $\mathcal{X}'$ and the camera parameters $\boldsymbol{Z}$, R, and K, the projection ray $\mathcal{L}_{x'}$ in the camera coordinate system is given by (12.110), p. 492, guaranteeing the correct sign of the ray direction. In the scene coordinate system using the motion matrix M_L in (12.75), p. 481 it therefore is

$$\mathbf{L}_{x'} = \begin{bmatrix} R^{\mathsf{T}} & 0 \\ S(\boldsymbol{Z})R^{\mathsf{T}} & R^{\mathsf{T}} \end{bmatrix} {}^{c}\mathbf{L}_{x'} \,. \tag{12.234}$$

Given the covariance matrix Σ_{pp} of the elements $\mathbf{p} = \mathrm{vec}\mathsf{P}$ of the projection matrix and the covariance matrix $\Sigma_{\mathrm{x'x'}}$ of the image point, assuming that they are statistically independent, the covariance matrix of the projection ray results from

$$\Sigma_{\mathrm{L}_{x'}\mathrm{L}_{x'}} = \overline{\mathsf{Q}}^{\mathsf{T}} \Sigma_{\mathrm{x'x'}} \overline{\mathsf{Q}} + J_{\mathrm{L}_{x'}\mathrm{p}} \Sigma_{\mathrm{pp}} J_{\mathrm{L}_{x'}\mathrm{p}}^{\mathsf{T}} \,, \tag{12.235}$$

with the Jacobian

$$J_{\mathrm{L}_{x'}\mathrm{p}} = [v'\mathsf{\Pi}(\mathbf{C}) - w'\mathsf{\Pi}(\mathbf{B}) \mid w'\mathsf{\Pi}(\mathbf{A}) - u'\mathsf{\Pi}(\mathbf{C}) \mid u'\mathsf{\Pi}(\mathbf{B}) - v'\mathsf{\Pi}(\mathbf{A})] \,, \tag{12.236}$$

where the transposed projection matrix is $\mathsf{P}^{\mathsf{T}} = [\mathbf{A}, \mathbf{B}, \mathbf{C}]$. This easily follows from (12.85), p. 483 using (12.71), p. 480.

uncertainty of direction of projection ray

When deriving the ray directions from their coordinates $\boldsymbol{x}'_i$ in the sensor plane we generally assume the calibration matrix to have zero variance. Thus the covariance matrix of the ray direction in the camera system is $\Sigma_{c_{\mathrm{x'}}c_{\mathrm{x'}}} = \mathsf{K}^{-1}\Sigma_{\mathrm{x'x'}}\mathsf{K}^{-\mathsf{T}}$, cf. (12.111), p. 492, which is independent of the choice of the sign.

12.3.2 Reconstructing Points on Planes

Given the image point $\mathcal{X}'$ of a 3D point $\mathcal{X}$ on a given plane $\mathcal{A}$, knowing the projection matrix P, i.e., exterior and interior orientation, we can derive the coordinates of the 3D point $\mathcal{X}$ together with its covariance matrix. The planes may be given in various ways.

12.3.2.1 3D Point on an Arbitrary Plane Given with Homogeneous Coordinates

If the 3D plane $\mathcal{A}$ is known by its homogeneous plane parameters (Fig. 12.25) $\{\mathbf{A}, \Sigma_{\mathrm{AA}}\}$,

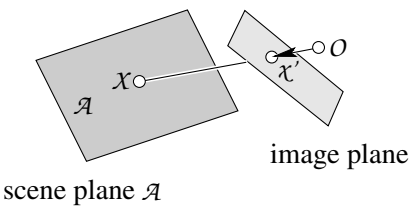


Fig. 12.25 Back projection of an image to an arbitrary scene plane. If the camera is straight line-preserving the mapping is a homography

due to $\mathbf{L}_{x'} \cap \mathbf{A} = \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\mathbf{L}_{x'} = \mathsf{\Gamma}^{\mathsf{T}}(\mathbf{L}_{x'})\mathbf{A}$, cf. (7.45), p. 301, the back projected 3D point is given by

$$\underset{4\times 3}{\mathbf{X}} = \mathsf{H}_A \, \mathbf{x}'_i \,, \tag{12.237}$$

with the homography

$$\underset{4\times3}{\mathsf{H}_A} = \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\overline{\mathsf{Q}}^\mathsf{T} . \tag{12.238}$$

This is a straight line-preserving mapping if we have a perspective camera. The corresponding homography $\mathcal{H}_A : \mathbb{P}^2 \mapsto \mathbb{P}^3$ is singular. The resulting 3D point is uncertain with covariance matrix,

$$\Sigma_{\mathrm{XX}} = \mathsf{\Gamma}^\mathsf{T}(\mathbf{L}_{x'})\Sigma_{\mathrm{AA}}\mathsf{\Gamma}^\mathsf{T}(\mathbf{L}_{x'}) + \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\Sigma_{\mathrm{L}_{x'}\mathrm{L}_{x'}}\mathsf{\Pi}^\mathsf{T}(\mathbf{A}) . \tag{12.239}$$

12.3.2.2 3D Point on a Parametrized Plane

If the points in the plane are given in a local plane coordinate system S_p with coordinates ${}^p\boldsymbol{x}$, we can make the projective mapping from the image to the plane coordinates explicit.

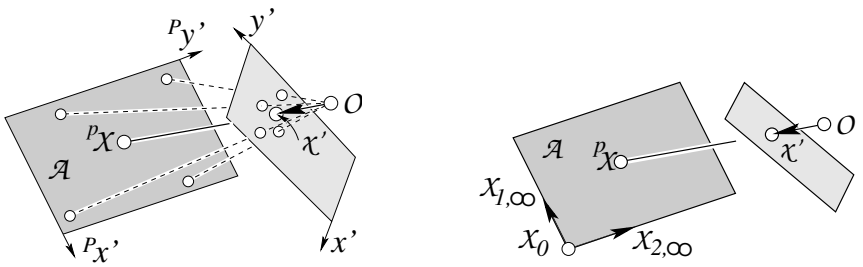


Fig. 12.26 Back projection of an image to a scene plane. **Left:** The two-dimensional plane coordinate system is given by four points $\mathcal{X}_i({}^p\mathbf{x}_i)$. **Right:** The plane coordinate system is given by its origin and two spatial directions

Plane with ≥ 4 Known Points. If four or more points $\mathcal{X}_i({}^p\mathbf{x}_i)$ are given in the local coordinate system of the plane and we have observed their images $\mathcal{X}'_i$, we can derive the homography

$${}^p\mathbf{x}_i = {}^p\mathsf{H}\,\mathbf{x}'_i \tag{12.240}$$

using the method discussed in Sects. 10.3.1.3, p. 387, or 10.6.3, p. 424, for determining a homography from four or more corresponding points. The uncertainty of a mapped point can be derived from (10.118), p. 387 if the point is not identical to those used for determining the homography, as otherwise the point and the homography cannot be assumed to be independent.

Plane with Local 3D Coordinate System. Often the plane coordinate system is given by its origin in 3D and the 3D directions of its two axes, see Fig. 12.26. An example is a roof with its corner and the two directions, the one of the gable and the one of the eave. Let the coordinate system in the plane be specified by its origin $\mathcal{X}_0$ and two mutually orthogonal 3D directions $\mathcal{X}_{1,\infty}$ and $\mathcal{X}_{2,\infty}$. Then the 3D point with plane coordinates ${}^p\boldsymbol{x} = [{}^px, {}^py]^\mathsf{T}$ has 3D coordinates

3D point on plane with local coordinate system

$$\boldsymbol{X} = \boldsymbol{X}_0 + {}^px\boldsymbol{X}_{1,\infty} + {}^py\boldsymbol{X}_{2,\infty} , \tag{12.241}$$

or, in homogeneous coordinates,

$$\mathbf{X} = \mathsf{G}_p\,{}^p\mathbf{x} = \mathbf{X}_0 + {}^px\mathbf{X}_{1,\infty} + {}^py\mathbf{X}_{2,\infty} = \underbrace{\begin{bmatrix} \boldsymbol{X}_{1\infty} & \boldsymbol{X}_{2,\infty} & \boldsymbol{X}_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathsf{G}_p} \begin{bmatrix} {}^px \\ {}^py \\ 1 \end{bmatrix} . \tag{12.242}$$

With the projection from the plane to the image we obtain $\mathbf{x}' = \mathsf{P}\mathbf{X} = \mathsf{P}\mathsf{G}_p\,{}^p\mathbf{x}$. Thus he homography from the image coordinates to the plane coordinates is given by

$$ {}^p\mathbf{x} = {}^p\mathsf{H}\,\mathbf{x}' \quad \text{with} \quad {}^p\mathsf{H} = (\mathsf{P}\mathsf{G}_p)^{-1}\,. \tag{12.243}$$

mono-plotting

Point on Map. In mapping applications a task could be to directly relate the image coordinates to the map coordinates, a procedure called *mono-plotting* in photogrammetry. Here we can assume the scene plane $\mathcal{A}$ to be parametrized as a function of the map coordinates $\boldsymbol{x}_M = [X, Y]^\mathsf{T}$, see Fig. 12.27,

$$ Z = a\,X + b\,Y + c =: \mathbf{l}^\mathsf{T}\,\mathbf{x}_M^e\,. \tag{12.244}$$

Then the scene point on the plane has homogeneous coordinates,

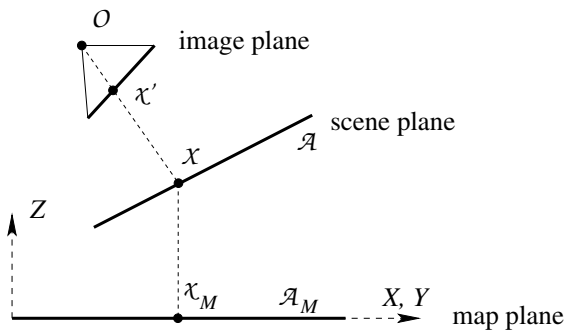


Fig. 12.27 Mono-plotting $\mathcal{X}' \rightarrow \mathcal{X}_M$ for given plane $\mathcal{A}$ parametrized by the coordinates $\boldsymbol{x}_M$ of the horizontal mapping plane $\mathcal{A}_M$. If the scene plane $\mathcal{A}$ is identical to the map plane $\mathcal{A}_M$ then the points $\mathcal{X}_M$ and $\mathcal{X}$ are identical

$$ \mathbf{X} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & c \\ 0 & 0 & 1 \end{bmatrix}}_{\mathsf{T}^M} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathsf{T}^M\,\mathbf{x}_M\,. \tag{12.245}$$

With the projection matrix $\mathsf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ of the image and the homography from the map to the image coordinate system,

$$ \mathsf{H}^M = \mathsf{P}\mathsf{T}^M = [\mathbf{p}_1 + a\mathbf{p}_3,\ \mathbf{p}_2 + b\mathbf{p}_3,\ \mathbf{p}_4 + c\mathbf{p}_3]\,. \tag{12.246}$$

We therefore have the projective mapping from the image point $\mathcal{X}'(\mathbf{x}')$ to the map point $\mathcal{X}_M(\mathbf{x}_M)$,

$$ \mathbf{x}_M = \left(\mathsf{H}^M\right)^{-1}\mathbf{x}'\,. \tag{12.247}$$

After Euclidean normalization of $\mathbf{x}_M$, we obtain the height Z from (12.244).

12.3.2.3 Quality of 3D Points

Theoretical Precision of New Points. The theoretical precision of a 3D point lying on a given plane can be based on (12.237): $\mathbf{X} = \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\mathsf{Q}^\mathsf{T}(\mathsf{P}(\boldsymbol{p}))\mathbf{x}'$. Thus the precision Σ_{XX} of the homogeneous coordinates of the 3D point $\underline{\mathbf{X}}$ depends on

1. the precision $\Sigma_{\mathrm{x'x'}}$ of the observed image point $\underline{\mathbf{x}}'$ possibly depending on its position in the image,

2. the precision Σ_{pp} of the parameters $\underline{\boldsymbol{p}}$ of the image orientation, and
3. the precision Σ_{AA} of the given plane $\underline{\mathbf{A}}$.

Determining Σ_{XX} requires the corresponding Jacobians,

$$J_{Xx'} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}'} \qquad J_{Xp} = \frac{\partial \mathbf{X}}{\partial \boldsymbol{p}} \qquad J_{XA} = \frac{\partial \mathbf{X}}{\partial \mathbf{A}}\,, \tag{12.248}$$

of the 3D point with respect to these three elements. These Jacobians reflect the geometry of the situation. The angles between the projection ray and the plane and to some extent between the projection ray and the viewing direction have an effect on the precision of the 3D point.

Some of the explicit expressions of these Jacobians are lengthy. Practically, it is easier to approximate the differential quotients by difference quotients and determine the Jacobian by numerical differentiation. If the three elements are stochastically independent we obtain by variance propagation,

$$\Sigma_{XX} = J_{Xx'}\Sigma_{x'x'}J_{Xx'}^{\mathsf{T}} + J_{Xp}\Sigma_{pp}J_{Xp}^{\mathsf{T}} + J_{XA}\Sigma_{AA}J_{XA}^{\mathsf{T}}\,. \tag{12.249}$$

Theoretical Precision of a 3D Point Derived from an Aerial Image in Standard Position. The theoretical precision of new points can easily be given explicitly for standard situations.

We refer to the reconstruction of a 3D point in the XY plane from its aerial image in a nadir view of a calibrated camera. The results do not change much if the slope of the terrain is below 15°. The orientation is assumed to be determined by a spatial resection with four control points, which are arranged in a square in the horizontal plane (Sect. 12.2.5.2).

The variance of the planar coordinates is composed of the following three components (Fig. 12.28):

1. We first assume that the uncertainty of the orientation is the only error source. We obtain the average variance in X and Y within the square formed by the four control points:

$$\overline{\sigma_X^2} = \overline{\sigma_Y^2} \approx \frac{1}{2}\left(\frac{H_g}{c}\right)^2 \sigma_{x'_{CP}}^2\,. \tag{12.250}$$

   Here $\sigma_{x'_{CP}}$ is the standard deviation of the image coordinate measurements of the control points. The factor H_g/c, the ratio of the camera height over ground ($= Z_O$ here) to the principal distance c, is the image scale number, which transfers the precision to the object space. The factor $1/2$ is plausible: The centroid of the control points has a variance of $1/4\ \sigma_{x'_{CP}}^2$; a similar uncertainty is caused by rotation and scale, doubling the variance. The uncertainty in ω and ϕ has no effect, due to the high correlations with the coordinates X_0 and Y_0 of the camera centre.
2. In the second step we take into account the additional influence of random errors in the image coordinates of the new point, measured with standard deviation $\sigma_{x'_{NP}}$, but still assume an error-free Z of the scene point. We then obtain

$$\overline{\sigma_X^2} = \overline{\sigma_Y^2} \approx \left(\frac{H_g}{c}\right)^2 \left(\frac{1}{2}\sigma_{x'_{CP}}^2 + \sigma_{x'_{NP}}^2\right)\,. \tag{12.251}$$

3. Finally we also take the uncertainty of the given height Z into account, which influences only the radial component. Therefore, the standard deviation in the tangential direction (12.251) is not influenced. In the radial direction the total variance is

$$\overline{\sigma_R^2} \approx \left(\frac{H_g}{c}\right)^2 \left(\frac{1}{2}\sigma_{x'_{CP}}^2 + \sigma_{x'_{NP}}^2\right) + \sigma_Z^2\ \tan^2\beta\,, \tag{12.252}$$

obviously depending on the slope of the spatial ray, the angle β between the optical ray, and the nadir direction.

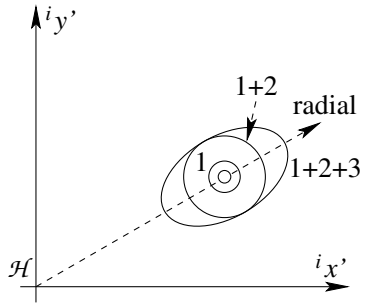


Fig. 12.28 Uncertainty of a new 3D point. Cumulative effect of (1) uncertainty of measured image point of control point, (1+2) uncertainty of measured new point, (1+2+3) uncertainty of given height only influencing the radial component

If the new image point is $({}^i x', {}^i y')$, we obtain the variances

$$\overline{\sigma_X^2} \approx \left(\frac{H_g}{c}\right)^2 \left(\frac{1}{2}\sigma^2_{x'_{CP}} + \sigma^2_{x'_{NP}}\right) + \left(\frac{{}^i x'^2}{{}^i x'^2 + {}^i y'^2} \tan^2\beta\right) \sigma_Z^2$$
$$\overline{\sigma_Y^2} \approx \left(\frac{H_g}{c}\right)^2 \left(\frac{1}{2}\sigma^2_{x'_{CP}} + \sigma^2_{x'_{NP}}\right) + \left(\frac{{}^i y'^2}{{}^i x'^2 + {}^i y'^2} \tan^2\beta\right) \sigma_Z^2 .$$

The Z-coordinate of the new point is given and has standard deviation σ_Z.

12.3.3 Position of a 3D Line Segment Using Triangulation

Under certain conditions, the 3D pose of a 3D line segment can be derived from its image. Let the start and end points $\mathcal{X}_1$ and $\mathcal{X}_2$ of a 3D line segment be observed in a camera whose orientation is known, leading to image points $\mathcal{X}'_1$ and $\mathcal{X}'_2$, respectively. Let us further assume we know the unnormalized vector $\boldsymbol{B} = {}^c\boldsymbol{X}_2 - {}^c\boldsymbol{X}_1$ of the line segment in the camera system, but not its position. Then we can determine the 3D coordinates of the start and end points in the camera system by analysing the spatial triangle $O\mathcal{X}_1\mathcal{X}_2$. The situation may arise when we know the camera pose w.r.t. the coordinate system of a Legoland scene, where only 3D lines with mutually orthogonal directions occur.[13]

First we determine the not necessarily normalized directions $\mathbf{u} := {}^c\mathbf{x}'_1$ and $\mathbf{v} := {}^c\mathbf{x}'_2$ using (12.109), p. 492 and assume they are coplanar with the vector $\boldsymbol{B}$. Then we have

$$ {}^c\boldsymbol{X}_1 = r\mathbf{u}\,, \qquad {}^c\boldsymbol{X}_2 = s\mathbf{v}\,. \tag{12.253}$$

We use the vector

$$\mathbf{m} = (\boldsymbol{B} \times \mathbf{u}) \times \boldsymbol{B}\,, \tag{12.254}$$

which lies in the plane spanned by $\boldsymbol{B}$ and $\mathbf{u}$ and points perpendicular to $\boldsymbol{B}$. From the three equations $r\mathbf{u} - s\mathbf{v} = \boldsymbol{B}$ we obtain the distances r and s,

Exercise 12.24

$$\begin{bmatrix} r \\ s \end{bmatrix} = \frac{1}{|\boldsymbol{B}, \mathbf{m}, \mathbf{u} \times \mathbf{v}|} \begin{bmatrix} \mathbf{m}^\mathsf{T}\mathbf{v} \\ \mathbf{m}^\mathsf{T}\mathbf{u} \end{bmatrix}\,, \tag{12.255}$$

[13] The geometric situation is identical to that of triangulation, cf. Sect. 13.4, p. 596. In a spatial triangle we know the spatial directions of the three sides in a local coordinate system and the length of one side and derive the relative positions of all three points of the triangle.

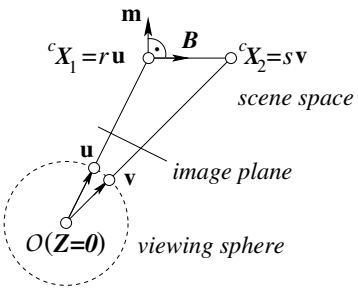


Fig. 12.29 Coordinates of observed line segment. Given the direction and length $\boldsymbol{B} = {}^c\boldsymbol{X}_2 - {}^c\boldsymbol{X}_1$ of a 3D line segment (X_1, X_2) in the camera system and the observations $\mathbf{x}'_1$ and $\mathbf{x}'_2$ of its end points, we can derive the 3D coordinates ${}^c\boldsymbol{X}_1$ and ${}^c\boldsymbol{X}_2$ in the camera system from direction vectors $\mathbf{u} = {}^c\mathbf{x}'_1$ and $\mathbf{v} = {}^c\mathbf{x}'_2$. The vector $\mathbf{m}$ lies in the plane spanned by the two projection rays, is perpendicular to $\boldsymbol{B}$, and points away from $O(\boldsymbol{Z})$

and with these the 3D coordinates in the camera coordinate system from (12.253).

The solution allows us to determine the coordinates in the scene coordinate system if all directions are given in that system. We will use the determination of the distances r and s from a vector and two directions when we derive a 3D scene point from its image points in two images in Sect. 13.4, p. 596.

12.3.4 Using Vanishing Points

Vanishing points play a central role when inverting the perspective projection. They regularly occur in images of Legoland scenes or in Manhattan scenes, which consist of multiple mutually rotated Legoland scene parts. They can be used for both partial scene reconstruction and for partial camera orientation and calibration.

Legoland and Manhattan scenes

A *vanishing point* v' is the image of the point at infinity, $\mathcal{V} = \mathcal{X}_\infty$, of a 3D line $\mathcal{L}$. Similarly, a *vanishing line* ℓ' is the image of a line at infinity, $\mathcal{L}_\infty$, of a plane.

vanishing point

vanishing line

A vanishing point can be observed in an image if at least two parallel 3D lines $\mathcal{L}_i$ are seen in the image, since the point at infinity $\mathcal{V}$ of two parallel lines $\mathcal{L}_i$ can be interpreted as the intersection $\mathcal{V} = \mathcal{L}_1 \cap \mathcal{L}_2$ of the two 3D lines, which is mapped to the vanishing point $v' = \ell'_1 \cap \ell'_2$ of the two image lines, see Fig. 12.30. Thus images of points at infinity of 3D lines provide directional information of this set.

direction of 3D lines from vanishing point v'

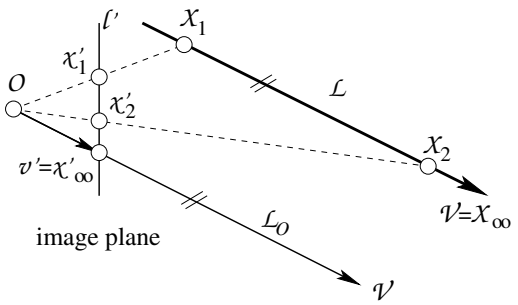


Fig. 12.30 Direction of parallel 3D lines from vanishing point. Points on the 3D line $\mathcal{L}$, say $\mathcal{X}_1$ and $\mathcal{X}_2$, are mapped to the image points on ℓ', say $\mathcal{X}'_1$ and $\mathcal{X}'_2$; thus, also the point at infinity $\mathcal{V} = \mathcal{X}_\infty$ of the 3D lines is mapped to $\mathcal{X}'_\infty \in \ell'$. The vanishing point of all 3D lines parallel to $\mathcal{L}$, and also that of $\mathcal{L}_O$ passing through the projection centre O, have the same image $v' = \mathcal{X}'_\infty$. Therefore the direction Ov' in the camera can be used to infer the direction of the 3D lines

Given a 3D line $\mathcal{L}(\mathbf{L})$ with its homogeneous and its Euclidean parts $\mathbf{L} = [\boldsymbol{L}_h^\mathsf{T}, \boldsymbol{L}_0^\mathsf{T}]^\mathsf{T}$, the point at infinity has homogeneous coordinates $\mathbf{V} = [\boldsymbol{L}_h^\mathsf{T}, 0]^\mathsf{T}$, thus only depends on the direction $\boldsymbol{L}_h$ of the 3D line. Its image is $\mathbf{v}' = \mathsf{P}\mathbf{V} = \mathsf{H}_\infty \boldsymbol{L}_h$, with the infinite homography $\mathsf{H}_\infty = \mathsf{A}$ from the left part of the projection matrix $\mathsf{P} = [\mathsf{A}|\mathbf{a}]$.

When observing a vanishing point with homogeneous coordinates, $\mathbf{v}' = \mathbf{l}'_1 \times \mathbf{l}'_2$, as the intersection of two or possibly more lines, we can infer the direction of the projection ray in the camera system and in the scene system.

We start with situations where the orientation and the calibration of the camera, thus the projection matrix P, is known and the camera is perspective, thus free of nonlinear distortions.

12.3.4.1 Direction of Parallel 3D Lines and of the Normal of a Plane

Given the vanishing point $v'(\mathbf{v}')$ as image of the point at infinity $\mathcal{V}$ of a set of parallel 3D lines, their direction $\boldsymbol{L}_h$ in the scene coordinate system can be inferred from (12.41), p. 473. For simplifying the equations, we assume the image coordinates refer to an image in viewing direction, thus $c < 0$; otherwise, the ray directions need to be changed by $-\text{sign}(c)$, cf. (12.108), p. 492. Thus we have

$$\boldsymbol{L}_h = \boldsymbol{V} = {}^n\mathbf{v}' = \mathsf{H}_\infty^{-1}\mathbf{v}' \qquad \text{or} \qquad {}^c\boldsymbol{L}_h = {}^c\boldsymbol{V} = {}^c\mathbf{v}' = -\text{sign}(c)\mathsf{K}^{-1}\mathbf{v}' \tag{12.256}$$

in the normalized camera system S_n, which is parallel to the scene system. The semantics of the expressions refer to the direction $\boldsymbol{L}_h$ of the 3D lines, the direction $\boldsymbol{V}$ to the point at infinity, and the direction $\mathbf{v}'$ of the projection ray. The sign of the directions to the vanishing points needs to be specified by the user, as the directions are given by the scene coordinate system.

If we observe the image of a parallelogram with four corners $\mathcal{X}_i, i = 1, 2, 3, 4$, in consecutive order, we can infer its normal from the two points at infinity derived from opposite sides, see Fig. 12.31, left. From the four observed corner points $\mathcal{X}'_1(\mathbf{x}'_1)$ to $\mathcal{X}'_4(\mathbf{x}'_4)$, we derive the image coordinates of the two vanishing points

$$\mathbf{v}'_j = (\mathbf{x}'_j \times \mathbf{x}'_{j+1}) \times (\mathbf{x}'_{j+2} \times \mathbf{x}'_{j+3}), \quad j = 1, 2, \tag{12.257}$$

taking the indices cyclically. This yields the normal in the scene coordinate system,

$$\mathbf{n} = \mathsf{N}(\mathsf{H}_\infty^\mathsf{T}(\mathbf{v}'_1 \times \mathbf{v}'_2)), \tag{12.258}$$

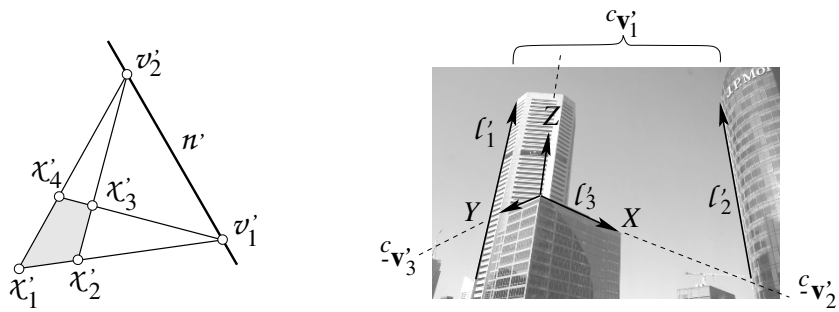


Fig. 12.31 Reconstruction of the normal of a plane and of the rotation matrix of the camera. **Left:** Given the image of a parallelogram in an oriented camera, the normal of the scene plane can be derived. The line n' joining the images v'_1 and v'_2 of two points at infinity is the image of the line at infinity of the parallelogram's plane. **Right:** The rotation matrix of the camera w.r.t. the local scene coordinate system can be derived from the three lines l'_i. We follow (12.259). The first direction ${}^c\mathbf{v}'_1$ we obtain is the point at infinity in the Z-direction. It is defined by l'_1 and l'_2, thus ${}^c\mathbf{v}'_1 = \mathsf{N}({}^c\mathbf{l}'_1 \times {}^c\mathbf{l}'_2)$. The line l'_3 is parallel to the X-axis. Therefore the second direction ${}^c\mathbf{v}'_2 = \mathsf{N}({}^c\mathbf{l}'_3 \times {}^c\mathbf{v}'_1)$ points in the negative X-direction and hence, in contrast to (12.260), the rotation matrix here is $\boldsymbol{R} = [-{}^c\mathbf{v}'_2, -{}^c\mathbf{v}'_3, {}^c\mathbf{v}'_1]^\mathsf{T}$, with ${}^c\mathbf{v}'_3 = \mathsf{N}({}^c\mathbf{v}'_1 \times {}^c\mathbf{v}'_2)$

which results from $\mathbf{n} = \mathsf{N}(\boldsymbol{V}_1 \times \boldsymbol{V}_2)$. Here we used $\mathbf{V} = \mathsf{H}_\infty^{-1}\mathbf{v}'$ from (12.256), left, the relation $M\boldsymbol{a} \times M\boldsymbol{b} = M^\circ(\boldsymbol{a} \times \boldsymbol{b})$ for general M and 3-vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and the regularity of H_∞, cf. App. (A.46), p. 772.

12.3.4.2 Rotation Matrix Obtained from Three Image Lines in a Legoland Scene

The rotation matrix of the camera w.r.t. the scene coordinate system can be determined if we have observed two lines, ℓ_1' and ℓ_2', belonging to one point at infinity, say in the X-direction, and a third line ℓ_3' belonging to the direction to a second point at infinity, say in the Z-direction in a Legoland scene, see Fig. 12.31 right, where the coordinate axes are chosen differently.

The three vanishing points are

$$ {}^c\mathbf{v}_1' = \mathsf{N}({}^c\mathbf{l}_1' \times {}^c\mathbf{l}_2')) \quad {}^c\mathbf{v}_2' = \mathsf{N}({}^c\mathbf{l}_3' \times {}^c\mathbf{v}_1') \quad {}^c\mathbf{v}_3' = \mathsf{N}({}^c\mathbf{v}_1' \times {}^c\mathbf{v}_2')\,, \tag{12.259} $$

using the line coordinates

$$ {}^c\mathbf{l}_i' = \mathsf{K}^\mathsf{T}\,\mathbf{l}_i' $$

in the camera coordinate system, cf. (12.256) and Sect. 6.2.4, p. 258. The rotation matrix is given approximately by

$$ R^{(0)} = [{}^c\mathbf{v}_1', {}^c\mathbf{v}_2', {}^c\mathbf{v}_3']^\mathsf{T}\,; \tag{12.260} $$

cf. (8.11), p. 327. The first direction v_1' results from the intersection of the first two lines, ℓ_1' and ℓ_2', represented in the camera coordinate system. The second direction, v_2', results from the intersection of the third line, ℓ_3', with the image of the line at infinity $v_2' \wedge v_3'$, the coordinates of which are identical of those of v_1' by the duality principle. The last direction, v_3', is perpendicular to the first two directions v_1' and v_2'.

Observe, the definition of the rotation matrix is ambiguous if the correspondence between the lines and the axes is not given, see the example in Fig. 12.31, right.

Due to measuring deviations, the matrix $R^{(0)}$ is not a proper rotation matrix. Following Sect. 8.4.3, p. 340 we can use an SVD to correct this approximate rotation matrix, obtaining an orthonormal matrix $R = UV^\mathsf{T}$ from $R^{(0)} = UDV^\mathsf{T}$.

12.3.4.3 Principal Distance and Principal Point Derived from Vanishing Points

Given two vanishing points v_i' in a Legoland scene, we are able to determine the principal distance. If we have the third vanishing point, we can also determine the principal point. In both cases, we then are able to derive the rotation matrix in the same way as in the previous section.

principal point is the intersection of the perpendiculars of the vanishing point triangle

The geometric configuration of the images of the three vanishing points and the projection centre is shown in Fig. 12.32. The four points, the projection centre O together with the three vanishing points v_i', span a tetrahedron with rectangular triangles as faces. The perpendiculars $O\mathcal{F}_i$ from the projection centre onto the sides of the vanishing point triangle intersect in the principal point.

Principal Distance Derived from Two Vanishing Points. We first assume the image is taken with a Euclidean camera having coordinate system S_e, thus without shear and scale difference, but with unknown principal distance (cf. Sect. 12.1.2.4, p. 464). We assume the principal point $\mathcal{H}$ is known to be $[{}^e x_H', {}^e y_H']$ and the coordinates of the vanishing points v_i' are $[{}^e x_{i,\infty}' - {}^e x_H', {}^e y_{i,\infty} - {}^e y_H']^\mathsf{T}$. Given two vanishing points, we therefore can determine the cosine of the angle $\alpha = \angle(v_1' O v_2')$ from

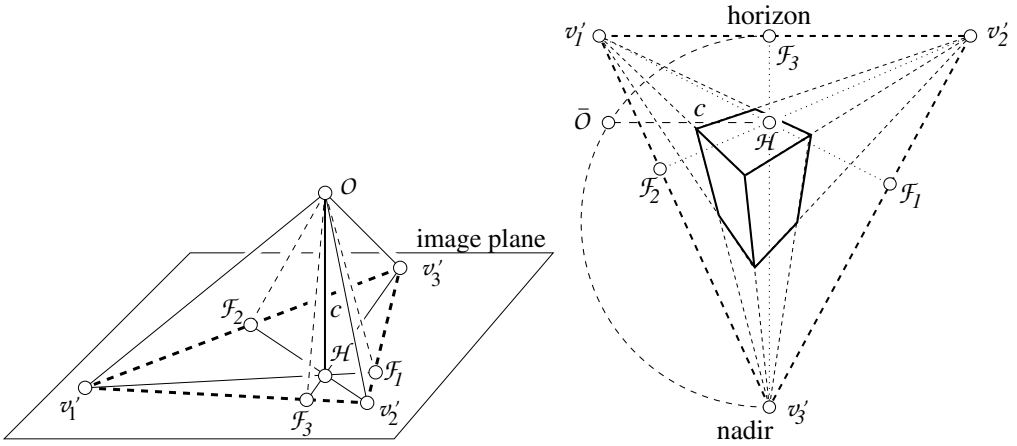


Fig. 12.32 Interior orientation from a Legoland image. In the image of a Legoland scene, the principal point $\mathcal{H}$ is the intersection of the lines $v_i'\mathcal{F}_i$ perpendicular to the sides $v_{i-1}'v_{i+1}'$ of the vanishing point triangle (v_1', v_2', v_3'), taking the indices cyclically. The three directions Ov_i' and the three planes $(v_i'Ov_j')$ are mutually orthogonal. The feet $\mathcal{F}_i$ of the perpendiculars can be used to determine the principal distance since each direction Ov_i' is perpendicular to the plane $v_{i-2}'Ov_{i+1}'$: the Thales circle over $\mathcal{F}_i v_i'$, here $\mathcal{F}_3 v_3'$ (right), contains the projection centre

$$\cos(\alpha) = \frac{{}^c\mathbf{v}_1{}'^{\mathsf{T}}\,{}^c\mathbf{v}_2'}{|{}^c\mathbf{v}_1'|\,|{}^c\mathbf{v}_2'|} \tag{12.261}$$

as a function of the unknown principal distance c since ${}^c\mathbf{v}_i' = [{}^e x_{i,\infty}' - {}^e x_H', {}^e y_{i,\infty} - {}^e y_H', c]^{\mathsf{T}}$. We know $\alpha = 90°$; thus, the cosine needs to be zero, and we obtain the principal distance

$$c^2 = -({}^e x_{1,\infty}' - {}^e x_H')({}^e x_{2,\infty}' - {}^e x_H') - ({}^e y_{1,\infty}' - {}^e y_H')({}^e y_{2,\infty}' - {}^e y_H')\,. \tag{12.262}$$

The sign of the principal distance has to be specified by the user.

If the image centre is taken as principal point two vanishing points are sufficient to derive the principal distance and the rotation matrix; this information may be used for determining the transformation of the image such that the principal planes appear undistorted, see Fig. 12.33.

Principal Point and Principal Distance Determined from Three Vanishing Points. As we can see from Fig. 12.32, the coordinates ${}^e\boldsymbol{Z} = [x_H', y_H', c]^{\mathsf{T}}$ of the projection centre O in a Euclidean camera system S_e can be determined by a spatial resection of three mutually orthogonal rays. Therefore we have three constraints

$${}^c\mathbf{v}_i{}'^{\mathsf{T}}\,{}^c\mathbf{v}_{i+1}' = 0\,, \qquad i = 1, 2, 3, \tag{12.263}$$

or, explicitly, using the coordinates $[x_i', y_i', c]^{\mathsf{T}} := [{}^e x_i', {}^e y_i', c]^{\mathsf{T}} = {}^c\mathbf{v}_i'$ in the following and omitting the superscript ^{e} for simplicity,

$$(x_i' - x_H')(x_{i+1}' - x_H') + (y_i' - y_H')(y_{i+1}' - y_H') + c^2 = 0\,. \tag{12.264}$$

Using the substitute variable for c,

$$z = c^2 + x_H'^2 + y_H'^2\,, \qquad c = \pm\sqrt{z - (x_H'^2 + y_H'^2)}\,, \tag{12.265}$$

Fig. 12.33 Image (upper left) with rectifications to principal planes of object

we obtain the equation system

projection centre by spatial resection

$$\begin{bmatrix} x'_2 + x'_3 & y'_2 + y'_3 & -1 \\ x'_3 + x'_1 & y'_3 + y'_1 & -1 \\ x'_1 + x'_2 & y'_1 + y'_2 & -1 \end{bmatrix} \begin{bmatrix} x'_H \\ y'_H \\ z \end{bmatrix} = \begin{bmatrix} x'_2 x'_3 + y'_2 y'_3 \\ x'_3 x'_1 + y'_3 y'_1 \\ x'_1 x'_2 + y'_1 y'_2 \end{bmatrix} . \tag{12.266}$$

Finally we find the principal distance from (12.265), again choosing the sign appropriate for the application.

If the three vanishing points do not belong to mutually orthogonal sets of 3D lines, but correspond to known spatial directions, we can determine the rotation matrix, the principal distance, and the principal point (the coordinates of the projection centre in the coordinate system of the image) with the general solution for the spatial resection (Sect. 12.2.4.1), using the vanishing points as three-dimensional points with Z-coordinate 0 (Sect. 5).

The following example demonstrates how to reconstruct the 3D structure of a building from a perspective line drawing by exploiting vanishing points if one length of the building is known.

Example 12.3.45: ***3D reconstruction of a building.*** We are given a perspective line drawing of a building as in Fig. 12.34, left, consisting of two intersecting gable roofs. We assume that the camera's interior orientation and the width s of the building front is known.

We reconstruct the 3D form of the building using the rules for inverse perspective discussed so far.

1. Assuming the left and the right roof of the front building to be of equal height, we can determine the spatial direction of the lines $(1, 2)$ and $(11, 3)$ from their vanishing points.
2. With the assumed width s of the building front we can determine the 3D coordinates of points 1 and 2.
3. Assuming the lines $(1, 11)$ and $(2, 3)$ to be parallel and using rule 1, we can determine their direction in space and thus the pose of the front plane. This allows us to determine the 3D points 3, 4, and 11 in that plane.
4. Assuming the lines $(1, 9)$ and $(4, 5)$ to be parallel we can determine their 3D direction. This specifies the side plane $(1, 11, 10, 9)$ and the roof plane $(4, 5, 10, 11)$ and allows us to determine the 3D points 5, 9, and 10. Observe, we alternatively could have chosen the lines $(1, 9)$ and $(10, 11)$, which would have led to a slightly different result, unless the line drawing were noise-free.

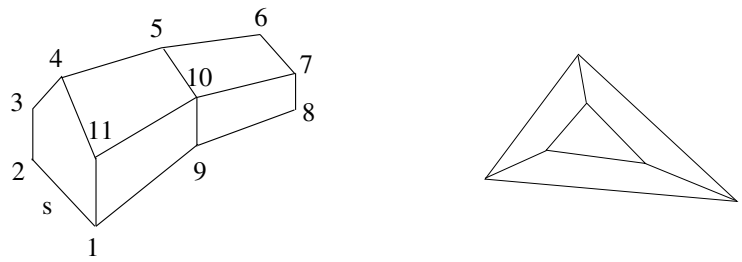


Fig. 12.34 Left: Perspective image of a building with given width s of the facade. **Right:** Inconsistent line drawing of a truncated pyramid, after Sugihara (1986)

5. Similarly, assuming the lines $(8, 9)$ and $(6, 5)$ to be parallel we can determine their 3D direction. This specifies the side plane $(7, 8, 9, 10)$ and the roof plane $(5, 6, 7, 10)$ and finally allows us to determine the 3D points 6, 7, and 8.

Observe, if we did not know the width of the building, we could completely reconstruct it from a single view up to scale. ◇

The sequential procedure used in the example assumes that the line drawing is consistent. Otherwise the result depends on the sequence and the choice of lines used for reconstruction, or even may lead to contradictions (see Fig. 12.34, right), where the three side planes of the truncated pyramid, and thus also the three side lines meet at a single point.

If we have four or more collinear points or four or more concurrent lines, we may take advantage of the invariance of their cross ratio when observing the image points as already discussed in Sect. 6.4.3.1, p. 268.

12.3.5 3D Circle Reconstructed from Its Image

Circular structures often occur in man made-scenes, either as bows on doors or windows, as the base line of a cylinder, or as circular targets. 3D circles are mapped to conics in general. They map to ellipses if they are completely in front of the camera, which always is the case when observing circular targets.

We address the following task. Given is the image $\mathcal{C}'(\mathsf{C}')$. We assume the camera is normalized with $\mathsf{K} = I_3$ and the radius r of the 3D circle $\mathcal{C}$ is known. The task is to determine the position ${}^c\boldsymbol{X}_0$ of the circle's centre and its normal ${}^c\boldsymbol{N}$. As we will see, we can also infer the image of the centre of a circle with an arbitrarily chosen radius r.

When using circular targets, the centre of the ellipse cannot be used for measuring the 3D position of the 3D circle, as generally the centre of the 3D circle is not mapped to the centre of the ellipse. However, once the 3D position of the centre ${}^c\boldsymbol{X}$ of the 3D circle is known in the camera system, its image has homogeneous coordinates ${}^c\mathbf{x}' = {}^c\boldsymbol{X}$, which can be used as an observation for the centre of the 3D circle. This also holds if the radius of the 3D circle is not known, as the coordinates ${}^c\boldsymbol{X}$ are proportional to the radius of the circle. For small 3D circles, the difference between the projected circle centre and the centre of the projected circle is small; thus, the uncertainty of the measured centre of the ellipse $\mathcal{C}'$ transfers to the projected circle centre.

There are two solutions which are symmetric w.r.t. the elliptic projection cone spanned by the projection centre and the ellipse $\mathcal{C}'$, see Fig. 12.35.

There are several solutions to this problem, e.g., by Dhome et al. (1990) or Kanatani (1993) (cf. Wrobel, 2012). We sketch the procedure proposed by Philip (1997). The centre and the normal of the 3D circle can be derived in three steps:

1. We first rotate the 3D circle such that the image $\mathcal{C}'_r$ of the rotated circle $\mathcal{C}_r$ is in the centre of the image. Then its conic matrix C'_r is diagonal. This is achieved by an

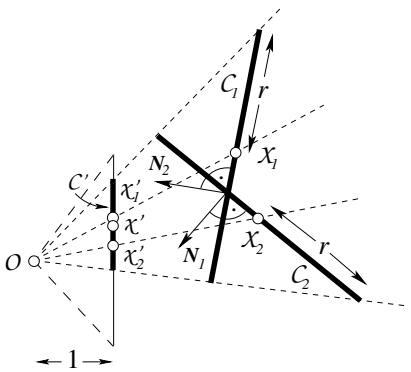
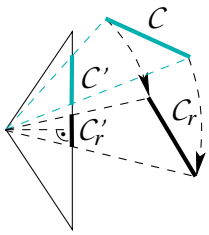


Fig. 12.35 3D circle from its image. Given is the ellipse $\mathcal{C}'$, which is the image of a 3D circle with a given radius r in a camera with principal distance $c = 1$. There are two possible positions $\mathcal{C}_1$ and $\mathcal{C}_2$ with centres $\mathcal{X}_1$ and $\mathcal{X}_2$ and normals $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$. The images $\mathcal{X}_1'$ and $\mathcal{X}_2'$ differ from the centre $\mathcal{X}'$ of the given ellipse. Adapted from Wrobel (2012)

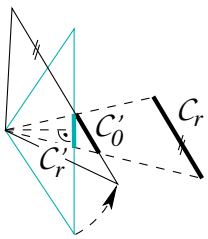
eigenvalue decomposition of the conic C',

$$\mathsf{C}' = R_1 \mathsf{C}_r' R_1^\mathsf{T}\,, \qquad \mathsf{C}_r' = \mathrm{Diag}([a, d, -f])\,. \tag{12.267}$$



We can always assume the eigenvalues to be sorted such that $a \geq d$ if we require the determinant of C_r' to be negative. This yields an ellipse with the major axis $\sqrt{-f/d}$ in the y'-direction and the minor axis $\sqrt{-f/a}$ in x' direction. If the 3D circle has radius 1 and is assumed to lie in front of the camera, the major axis is smaller than 1 and therefore $|f| < d$. This fact can be used to replace the eigenvalue decomposition by a singular value decomposition to achieve

$$\mathsf{C}' = U \mathsf{D} R_1^\mathsf{T}\,, \qquad D = \mathrm{Diag}([a, d, f])\,, \tag{12.268}$$



with $a \geq d > f$, which is guaranteed by classical programs for determining the SVD. Choosing $R_1 = U$ does not change the result.

2. We now rotate the camera such that the 3D circle lies parallel to the image plane. Then the image of the 3D circle is a circle $\mathcal{C}_0$, and the upper left 2×2 matrix $\mathsf{C}_{0,hh}$ of

$$\mathsf{C}_0' = R_2 \mathsf{C}_r' R_2^\mathsf{T} \tag{12.269}$$

is a multiple of the unit matrix. The rotation needs to be around the ${}^c y$-axis of the camera, thus of the form

$$R_2 = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix}\,. \tag{12.270}$$

From the constraint $\mathsf{C}_{0,hh}' = d I_2$ we obtain the constraint $\cos^2\phi = (d+f)/(a+f)$ and therefore two solutions for $\cos\phi$ and two solutions for $\sin\phi$,

$$\cos\phi = \pm\sqrt{\frac{d+f}{a+f}}\,, \qquad \sin\phi = \pm\sqrt{1-\cos^2\phi} = \pm\sqrt{\frac{a-d}{a+f}}\,, \tag{12.271}$$

and thus four solutions for R_2. The resulting circle then has the form

$$\mathsf{C}_0' = \pm \begin{bmatrix} d & 0 & \pm\sqrt{(a-d)(d+f)} \\ 0 & d & 0 \\ \pm\sqrt{(a-d)(d+f)} & 0 & a-d-f \end{bmatrix}\,. \tag{12.272}$$

Since the conic matrix is homogeneous, its sign has no influence on the position of the conic and we end up with two solutions. We can arbitrarily choose the sign of $\cos\phi$ in

(12.271). If we choose the positive sign of $\cos\phi$, the image $\boldsymbol{x}'_0$ of the centre $\boldsymbol{X}_0$ and image radius r' are

$$\boldsymbol{x}'_0 = \begin{bmatrix} \pm\sqrt{(a-d)(d+f)} \\ 0 \end{bmatrix}, \qquad r' = \frac{af}{d^2}; \tag{12.273}$$

cf. (5.148), p. 237. Therefore the centre of the 3D circle after these two rotations is

$$\boldsymbol{X}_0 = \frac{r}{r'} \begin{bmatrix} \pm\sqrt{(a-d)(d+f)} \\ 0 \\ 1 \end{bmatrix}. \tag{12.274}$$

The normal of the 3D circle now is $\boldsymbol{N}_0 = [0, 0, 1]^\mathsf{T}$.

3. We now undo the two rotations resulting from (12.267) and (12.269), which resulted in $\mathsf{C}'_0 = \mathsf{R}_2\mathsf{R}_1^\mathsf{T}\mathsf{C}'\mathsf{R}_1\mathsf{R}_2^\mathsf{T}$. We obtain the image of the centre of the 3D circle,

$${}^c\boldsymbol{X} = r\,\mathsf{R}_1\mathsf{R}_2^\mathsf{T}\boldsymbol{X}_0 = r\,\mathsf{R}_1 \begin{bmatrix} \pm\sqrt{f/a}\sin\phi \\ 0 \\ \sqrt{a/f}\cos\phi \end{bmatrix} \tag{12.275}$$

in the camera system. We choose the sign of the vector ${}^c\boldsymbol{X}$ such that the centre of the circle lies in front of the camera, thus ${}^cX_3 < 0$. Similarly, we can find the normal of the 3D circle by back-rotating the normal vector $\boldsymbol{N}_0 = [0, 0, 1]^\mathsf{T}$, yielding the normal

$${}^c\boldsymbol{N} = \mathsf{R}_1\mathsf{R}_2^\mathsf{T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathsf{R}_1 \begin{bmatrix} \mp\sin\phi \\ 0 \\ \cos\phi \end{bmatrix} \tag{12.276}$$

in the camera system. Finally, we choose the sign of the normal such that it points towards the camera, thus ${}^cN_3 > 0$.

Algorithm 19 describes the process. If the radius of the 3D circle is unknown, the algorithm can be called with $r = 1$, still yielding the direction ${}^c\boldsymbol{X}$ to the centre of the circle, which is identical to the homogeneous coordinates ${}^c\mathbf{x}'$ of its image. If the algorithm is only meant to determine the image of the centre of the circle, line 4 can be omitted. The algorithm assumes the singular values are sorted in decreasing order.

Algorithm 19: 3D circle $\mathcal{C}$ with given radius r determined from its image $\mathcal{C}'$ in a normalized camera with $\mathsf{K} = \mathsf{I}_3$;
$[{}^c\boldsymbol{X}_1, {}^c\boldsymbol{N}_1, {}^c\boldsymbol{X}_2, {}^c\boldsymbol{N}_2]$=`3D_Circle_from_its_Image`$(\mathsf{C}, r)$
Input: conic matrix C' of image $\mathcal{C}'$, radius r of 3D circle $\mathcal{C}$.
Output: centres and normals $({}^c\boldsymbol{X}_i, {}^c\boldsymbol{N}_i), i = 1, 2$ of the 3D circle $\mathcal{C}$.

1 SVD: $[U, \mathrm{Diag}([a, d, f]), \mathsf{R}_1] =$ `svd`$(-\mathsf{C}' \,|\mathsf{C}'|)$, with $a \geq d \geq f$;
2 Cosine of angle ϕ: $\cos\phi = \sqrt{(d+f)/(a+f)}$;
3 Two values for sine of angle ϕ: $\sin\phi_i = \pm\sqrt{1-\cos^2\phi}, i = 1, 2$;
4 Normals: ${}^c\boldsymbol{N}_i = \mathsf{R}_1[\sin\phi_i\,, 0\,, \cos\phi]^\mathsf{T}$, ${}^c\boldsymbol{N}_i := \boldsymbol{N}_i\,\mathrm{sign}(N_{i,3}), i = 1, 2$;
5 Centres: ${}^c\boldsymbol{X}_i = r\,\mathsf{R}_1[-\sqrt{f/a}\,\sin\phi_i\,, 0\,, \sqrt{a/f}\,\cos\phi]^\mathsf{T}$, ${}^c\boldsymbol{X}_i := -{}^c\boldsymbol{X}_i\,\mathrm{sign}({}^cX_{i,3}), i = 1, 2$.

The derivation of the method by Philip (1997) assumes the image of the circle to be an ellipse. It is an open question how the method needs to be modified if the image of the circle is a hyperbola.

12.4 Exercises

Basics

1. (1) Given a camera with 1200×800 pixels and a principal distance of $c = 320$ pixels. What is the maximum tilt angle τ according to Fig. 12.2, p. 458 such that the nadir point is visible in the image?
2. (1) Determine the viewing angle of the camera in your mobile phone by taking an image. Refer the viewing angle to the diagonal of the image.
3. (1) Refer to Fig. 12.2, p. 458 and show that the image scale at the isocentre for an infinitesimal distance along the line $\mathcal{N}'\mathcal{H}$ is given by the ratio $\overline{OI}/\overline{OJ}$, where $\mathcal{J}$ is the intersection of $O \wedge I$ and the horizontal ground plane. Give an argument for why the image scale for infinitesimal distances at this point is independent of the direction of this distance. *Hint:* Draw the geometric situation in a vertical plane through $\mathcal{N}'\mathcal{H}$ and add the bisector of the horizontal line through $\mathcal{N}$ and the line $\mathcal{N}'\mathcal{H}$.
4. (1) A camera lies on a table such that the viewing direction is approximately horizontal, namely pointing $\alpha = 5°$ upwards. Assume you move the camera on the table. How many parameters, and which ones, do you need to describe the exterior orientation? Give the projection matrix P as a function of α and these parameters.
5. (1) Refer to Fig. 12.4, p. 461 and explain why the distance between the two points $\mathcal{K}_1$ and $\mathcal{K}_2$ in the two principal planes of the lens is not used when modelling a camera.
6. (1) Explain the differences between the following pairs of camera models. Give the number and names of the parameters which are different. Give essential differences in their properties. Give reasons why you would choose one or the other model:
   a. the perspective camera and the spherical camera model,
   b. the ideal and the normalized camera model, and
   c. the Euclidean camera and the unit camera.
7. (1) What units do the nine entries of the calibration matrix K_3 have if the scene coordinates are given in meters and the image coordinates are given in pixels? Observe: The matrix is homogeneous.
8. (1) What are the units of the 12 entries in the projection matrix P if the scene coordinates are given in meters and the image coordinates are given in pixels?
9. (3) An image has size 3000×2000 pixels, the scene coordinates lie in the range $[400..600, 1300..1700, 100..200]$ m. Given is the projection matrix (cf. `HOME/P_matrix`.[14])

$$\mathsf{P} = \begin{bmatrix} 0.0012682807017 & -0.0006478859649 & 0.0003109824561 & -0.1611793859649 \\ 0.0008165263157 & 0.0010670263157 & 0.0001048421052 & -2.1127144736842 \\ 0.0000002017543 & 0.0000000350877 & 0.0000004561403 & -0.0008359649122 \end{bmatrix} . \tag{12.277}$$

   a. If you apply conditioning to the scene coordinates and the image coordinates, leading to conditioned coordinates $\breve{\boldsymbol{X}}$ and $\breve{\boldsymbol{x}}'$, what units do the 12 entries of the corresponding conditioned projection matrix $\breve{\mathsf{P}}$ have? Refer to Sect. 6.9, p. 286.
   b. Condition the matrix P.
   c. Determine the condition numbers κ and $\breve{\kappa}$ using the maximum and minimum nonzero singular values analogously to (4.248), p. 118 of the two matrices P and $\breve{\mathsf{P}}$, respectively. Comment on the result.
   d. What would the condition number be if you conditioned only the image or only the scene coordinates?
   e. What effect does conditioning have on the directions ${}^{c}\mathbf{x}'$ of projection rays?

[14] cf. Sect. 1.3.2.4, p. 16.

f. Determine the condition number of $\mathsf{P}\mathsf{P}^\mathsf{T}$ for an ideal camera with $\mathsf{R} = \mathsf{I}_3$ and a projection centre H_g above the origin of the scene coordinate system. How does the condition number depend on c and H_g? Is it possible to choose c and H_g such that the condition number is $\kappa = 1$? Sketch the situation and interpret the result assuming scene and image coordinates are conditioned.

10. (1) Prove that in Eq. (12.88), p. 483 the relative scaling is $\lambda = 1$ if the projection centre is taken from (12.45), p. 475. *Hint:* Use (12.44), p. 474 and (12.77), p. 481, and the dual of (5.103), p. 225.
11. (2) You install a camera on a mast (camera with affine sensor coordinate system, principal point at $[320, 240]$ pixel, principal distance $c = 600$ pixel, shear $s = 0.003$, scale difference $m = 0$). The local coordinate system is at the foot of the mast, the ground plane is the XY-plane, the Z-direction points towards the zenith. The projection centre has a height of 12 m above the ground and is 30 cm away from the centre of the mast in the X-direction. The camera can only rotate around the horizontal Y-axis with a tilt angle τ. You use a camera model with positive principal distance. Given are two scene points, $\boldsymbol{X}_1 = [50, 0, 2]^\mathsf{T}$ m and $\boldsymbol{X}_2 = [48, 3, 0]^\mathsf{T}$ m.

    a. Make a sketch of the situation and in particular show the tilt angle τ.
    b. Explain the parameters of the exterior and interior orientations. Give the projection matrix $\mathsf{P}(\tau)$ as a function of the tilt angle τ.
    c. Determine the image coordinates of the two points in case the camera is tilted downwards by 5°.

12. (1) Give the redundancy when estimating the pose of a camera using a spatial resection with 24 points.
13. (1) Show that when performing a spatial resection, a point $\mathcal{X}_i$ at infinity does not influence the position $\boldsymbol{Z}$ of the projection centre, and that the position of the projection centre has no influence on the rotation.
14. (2) Show that if the bounding box of a set of points $\boldsymbol{X}_i$ is centred at $\boldsymbol{X}_0$ and has size $A \times B \times C$ taking the four points of the tetrahedron

$$\boldsymbol{X}_0 + \begin{bmatrix} \pm uA \\ 0 \\ -vC \end{bmatrix}, \quad \boldsymbol{X}_0 + \begin{bmatrix} 0 \\ \pm uB \\ vC \end{bmatrix} \quad \text{with} \quad u = \frac{2+\sqrt{2}}{6}, v = \frac{u}{\sqrt{2}} \tag{12.278}$$

    as reference points lead to barycentric coordinates $|\alpha_i| \le 1$ (cf. Kaseorg, 2014, second solution).
15. (1) The DLT often is given in the form

$$x' = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + 1}, \quad y' = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + 1}$$

    in order to estimate only 11 parameters from p_{11} to p_{33}.

    a. Why is this model not able to handle the case where actually $p_{34} = 0$?
    b. Assume an ideal camera with $\mathsf{R} = \mathsf{R}_y(90°)$ and generate a situation where we have $p_{34} = 0$.
    c. Describe a realistic scenario where this situation occurs.

16. (2) Theoretical accuracy of spatial resection: Given is an aerial image taken with an aerial camera, DMC, of Zeiss. The interior orientation is known. The exterior orientation is to be determined from a spatial resection using a set of I points. Assume the following situation:

    - Flying height above ground $H_g = 1000$ m.
    - Image size 7.690 × 13.824 pixel.
    - Principal distance 120 mm. Pixel size 12×12 μm^2.

- Standard deviation of measured image points $\sigma_{x'} = 0.1$ pixel.

Provide the equations for the standard deviations of the six parameters of the exterior orientation as well as their numerical values. Assume 12 points have been measured at the border of the image, with four points at each side having a common distance.

a. What accuracy can you expect for the XY- and the Z-coordinates of the projection centre?
b. What accuracy do you expect for the three rotation angles?
c. Assume the image coordinates of a scene point are measured with an accuracy of 0.5 pixel. The height of the point is taken from a map with an uncertainty of 0.5 m. The image point lies at the border of the image. You can derive the XY-coordinates for the point. What accuracy do you expect? Which uncertainty predominantly influences the variance of the points? Which has the least influence?
d. Explain why the standard deviation of the XY-position of the projection centre w.r.t. the control points is much larger than the standard deviation of a 3D point derived from its image coordinates and a given mean height μ_Z.

Proofs and Problems

17. (1) Prove (12.76), p. 481 assuming the 3D line is given by $\mathcal{L} = \mathcal{X} \wedge \mathcal{Y}$.
18. (2) Prove that (12.61), p. 478 is a sufficiently good approximation of (12.60), p. 478, if the shear and the scale difference are small.
19. (2) Confirm the standard deviations and correlations of the pose using a spatial resection with four double points in the corners of the square in Sect. 12.2.5.2, p. 523.
20. (3) Assume an ideal camera with projection centre $\boldsymbol{Z} = \mathbf{0}$, see Fig. 12.36

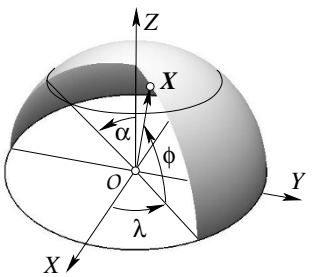


Fig. 12.36 Spatial resection with scene point $\boldsymbol{X}$ on a unit sphere. The calotte has its centre in $[0, 0, 1]^\mathsf{T}$ and radius α

It is observing I control points $\boldsymbol{X}_i = \boldsymbol{X}(\lambda_i, \phi_i)$ on the unit sphere assumed to be fixed values. The angles λ_i and ϕ_i denote longitude and latitude. Assume the points are evenly distributed in the calotte around $[0, 0, 1]$ with a radial angle of α. Determine the covariance matrix of the six parameters of the pose assuming isotropic uncertainty σ of the ray directions. Show that the structure of the normal equation system is the same as in the example in Sect. 12.2.5.2, p. 523. How does the correlation $\rho_{y\omega}$ change in the range $\alpha = 0, ..., \pi$, where $\alpha = \pi$ means observing the full sphere? Give the covariance matrix in case the full sphere is observed.
Hint: Replace the sums in the normal equation matrix by integrals, e.g., substitute $\sum_{i=1}^{I} f(\boldsymbol{X}_i) = \sum_{i=1}^{I} f(\lambda_i, \phi_i)$ by $m_f I$ where the mean m_f is determined using a uniform distribution in the observed calotte of the sphere, thus

$$m_f = \frac{1}{A} \int_{\lambda=0}^{2\pi} \int_{\phi=\pi/2-\alpha}^{\pi/2} f(\lambda, \phi)\ \cos(\phi)\ \mathrm{d}\lambda\, \mathrm{d}\phi \quad \text{with} \quad A = 2\pi(1 - \cos(\alpha))\,. \tag{12.279}$$

Use an algebra package.

21. (2) Show that a 3D unit circle in the XY-plane can be represented as dual quadric $\mathsf{Q}^* = \mathrm{Diag}([1,1,0,-1])$. Give the representation of a general 3D circle $\mathcal{C}$ with centre $\boldsymbol{X}_0$, normal $\boldsymbol{N}$ and radius R. Generate a point $\mathcal{X}$ on a general 3D circle, project it into a camera with projection matrix $\mathsf{P}_0 = [\boldsymbol{I}_3 \mid \boldsymbol{0}]$ and show that it lies on the image $\mathcal{C}'$ of the 3D circle $\mathcal{C}$.
22. (1) Well-designed optics of consumer cameras still show significant radial distortion, easily up to ten pixels at the border of the image. For small enough values K_1 the two distortion models (12.184), p. 507 and (12.185), p. 508 are similar. How large can K_1 be such that for an image with 3000×2000 pixels the difference between both models is below 0.5 pixel? How large is the maximal distortion in this case?
23. (2) Show the correlation of $\rho = -\sqrt{21}/5 \approx 0.92$ for the parameters for the linear and the cubic distortion in the polynomial distortion model, cf. Sect. 12.2.3.3, p. 508. Assume the image distortions can be modelled by $z = \Delta x' = a_1 x + a_3 x^3$, $\Delta y' = 0$. Assume you have a regular grid of $(2N+1) \times (2N+1)$ points with a spacing of $1/(2N)$ symmetrically filling the square $[-1,+1]^2$ where $z(x)$ is observed. Assume the observed values have standard deviation σ. Use the Gauss–Markov model to estimate the two parameters a_1 and a_3 from distortions $\Delta \boldsymbol{x}'$ at the grid points:
    a. Give an explicit expression for the normal equation matrix and its inverse.
    b. Take the limit $N \to \infty$ to derive the correlation coefficient $\rho_{\widehat{a}_1 \widehat{a}_3}$ of the estimated parameters $\widehat{a}_1$ and $\widehat{a}_3$. *Hint:* You alternatively may take the limit on the individual elements of the normal equation system, except for a common factor N, and replace the sums by integrals.
24. (2) Prove Eq. (12.255), p. 528. *Hint:* Rotate the coordinate system such that the plane spanned by the two viewing rays is the (x,y)-plane. Use the rotation matrix $[\mathsf{N}(\boldsymbol{B}) \mid \mathsf{N}((\boldsymbol{B} \times \mathbf{u}) \times \boldsymbol{B}) \mid \mathsf{N}(\boldsymbol{B} \times \mathbf{u})]$, neglect a possible skew between the two 3D lines, and determine r and s from a planar triangulation.
25. (2) Prove (12.54), p. 476.

Computer Experiments

26. (3) Program for the spatial resection
    - Write an algorithm `SRS` for the iterative solution of the spatial resection following Sect. 12.2.4, p. 513. Take as input a set of corresponding image and scene coordinates, $({}^c\mathbf{x}'_i, \Sigma_{{}^c x'_i {}^c x'_i}, \mathbf{X}_i), i = 1, ..., I$. In a first step, provide as output the estimated parameters of the exterior orientation and their covariance matrix, the residuals, and the estimated variance factor.
    - Write a program for simulating scene and image points for checking the algorithm `SRS`. Take as input the spatial region of the scene points (e.g., a box, a ball or a calotte of a sphere, cf. Exerc. 20), the number I of scene points, the pose $(R, \boldsymbol{Z})$ of the central camera and the uncertainty of the ray directions ${}^c\mathbf{x}'_i$ (e.g., homogeneous or inhomogeneous, isotropic or anisotropic).
    - Check the implementation following Sect. 4.6.8, p. 139.
    - Empirically verify the findings of Exerc. 20.
    - Now extend the situation by assuming outliers in the observations. Extend the estimation program by an ML-type estimation to eliminate outliers following the algorithm (4), p. 168. Provide an option to use the rigorous statistical test with the statistic X_i^2, the normalized residuals $\widehat{\boldsymbol{v}}_i^\mathsf{T} \Sigma_{l_i l_i} \widehat{\boldsymbol{v}}_i$, and the squared length $|\widehat{\boldsymbol{v}}_i|^2$ of the residuals as argument for the ρ-function (4.364), p. 144, possibly taking into account a robust estimate for σ_0. For the rigorous test statistic w.r.t. outliers in the image coordinates follow Sect. 4.6.4, p. 124. Take each image point as an

observational group. What are the degrees of freedom of the optimal test statistic (4.302), p. 129?
Analyse the behaviour of the estimation procedure w.r.t.
- its ability to find outliers,
- the rejection rate based on the estimated variance factor,
- the achievable precision based on $\widehat{\sigma}_0^2 \Sigma_{\widehat{x}\widehat{x}}$, and
- the consistency of this covariance matrix with ground truth.

Vary the configuration w.r.t. the geometric distribution of the image rays, the stochastical model of the ray directions and the rigour of the argument of the ρ-function.

- What is the maximum outlier percentage the algorithm can handle? How does it depend on the number of observations, the configuration, and the rigour of the outlier rejection?
- Invent an indicator for the success of the estimation in the sense of a traffic light program, see Fig. 3.2, p. 64. Using simulated data, determine how good this indicator is.

Exercises 27 to 36 use meters and pixels as units for the scene and the image coordinates, respectively. The projection centre is $\boldsymbol{Z} = [0, 0, 1200]^\mathsf{T}$ m. The image has 600 rows and 800 columns. The exercises use one of the calibration matrices:

$$\mathsf{K}_0 = \mathsf{I}_3\,, \quad \mathsf{K}_1 = \begin{bmatrix} 300 & & \\ & 300 & \\ & & 1 \end{bmatrix}, \quad \mathsf{K}_2 = \begin{bmatrix} 300 & & 300 \\ & 300 & 400 \\ & & 1 \end{bmatrix}, \quad \mathsf{K}_3 = \begin{bmatrix} 300 & 0 & 300 \\ & 305 & 400 \\ & & 1 \end{bmatrix}. \tag{12.280}$$

They also use the following rotation matrices (R_Q is the representation with quaternions (8.55), p. 335),

$$R_0 = \mathsf{I}_3\,, \quad R_1 = R_Q\left(\begin{bmatrix} 1 \\ +0.01 \\ -0.02 \\ +0.03 \end{bmatrix}\right), \quad R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{12.281}$$

and the 3D points

$$\mathbf{X}_1 = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 1 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 100 \\ 1000 \\ 100 \\ 1 \end{bmatrix}, \quad \mathbf{X}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{X}_4 = \begin{bmatrix} 1000 \\ 100 \\ 100 \\ 1 \end{bmatrix}. \tag{12.282}$$

27. (1) Determine the normalized ray direction ${}^c\mathbf{x}_1'^s$ using $\mathsf{K}_k, k = 0, 1, 2, 3$, and R_0. Discuss the difference between using K_2 and K_3, particularly using the angle between the two directions.
28. (1) Determine the image coordinates ${}^i\mathbf{x}_2$ using $\mathsf{K}_k, k = 0, 1, 2, 3$, and R_1. Why are there differences between the coordinates, when using K_0 and K_1, large. Give a geometric explanation without referring to equations or numbers.
29. (2) Make a sketch of the camera $(\mathsf{K}_2, \mathsf{R}_3)$ and the points $\mathcal{X}_1$ and $\mathcal{X}_2$. Indicate the three camera planes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Visually verify their homogeneous coordinates as derived from the projection matrix P. Are the points in front or behind the camera? Numerically verify that the projection centre is the intersection of all three camera planes.
30. (1) Determine the image $\mathcal{h}'$ of the horizon, i.e., the image of the line at infinity of the XY-plane. Use the camera with K_2 and R_3. Determine the image point $\mathcal{x}_3'$ of $\mathcal{X}_3$ in (12.282). Show that it lies on the image horizon.
31. (1) Give the viewing direction for the cameras with $\mathsf{K}_k, k = 0, 1, 2, 3$, and R_2. Draw a unit sphere and the normalized viewing direction vectors.

32. (1) Determine the 3D line $\mathcal{L} = \mathcal{X}_1 \wedge \mathcal{X}_2$. Determine the image points $\mathcal{x}_1'$ and $\mathcal{x}_2'$ and the image line $\mathcal{l}'$ by direct projection, using the camera with K_1 and R_1, cf. (12.280)ff. Numerically verify that $\mathcal{l}' = \mathcal{x}' \wedge \mathcal{y}'$.
33. (1) Use the image line $\mathcal{l}'$ from Exerc. 32 and determine the projection plane $\mathcal{A}_{l'}$. Numerically verify that it passes through the points $\mathcal{Z}$, $\mathcal{X}_1$ and $\mathcal{X}_2$.
34. (1) Use the image point $\mathcal{x}_2'$ from Exerc. 32. Determine the projection line $\mathcal{L}_{x_2'}$ and numerically verify that it passes through $\mathcal{Z}$ and $\mathcal{X}_2$.
35. (2) Determine the dual quadric of the circle $\mathcal{K}$ through $\mathcal{X}_i, i = 1, 2, 4$ (*Hint:* Shift the three points to the XY-plane, determine the dual quadric of this circle, and shift back.). Determine the image $\mathcal{k}'$ of the 3D circle. Numerically verify that it passes through the three image points $\mathcal{x}_i, i = 1, 2, 4$.
36. (1) Which of the four points is in front of the cameras with $(\mathsf{K}_t, \mathsf{R}_t), t = 1, 2$?

37. (2) You have an image of a building with a flat roof. You can measure six points $\mathcal{x}_i'$ in the image for which you have the 3D coordinates $\boldsymbol{X}_i$ from a map (`HOME/DLT_data.txt`).

point	X [m]	Y [m]	Z [m]
$\mathcal{X}_1$	10.0	10.0	3.0
$\mathcal{X}_2$	10.0	32.0	3.0
$\mathcal{X}_3$	30.0	10.0	3.0
$\mathcal{X}_4$	10.0	10.0	23.0
$\mathcal{X}_5$	10.0	32.0	23.0
$\mathcal{X}_6$	30.0	10.0	23.0

point	x' [pixel]	y' [pixel]
$\mathcal{x}_1'$	264	44
$\mathcal{x}_2'$	390	92
$\mathcal{x}_3'$	312	10
$\mathcal{x}_4'$	247	120
$\mathcal{x}_5'$	359	191
$\mathcal{x}_6'$	293	95

   a. Provide an algorithm for determining the projection matrix from the scene to the image using point correspondences. Program the algorithm and determine P. Scale P such that $P(3, 4) = 1$.
   b. Decompose the projection matrix such that the principal distance is negative. Determine K, R, and $\boldsymbol{Z}$.
   c. Provide pseudocode of an algorithm for determining the projection matrix from the scene to the image using point correspondences. Program the algorithm and determine P using a set of six lines connecting the given points. Discuss the result w.r.t. the selection of the six lines.

38. (2) Given is a camera with Euclidean sensor with $c = 543$ pixel and a principal point $[256, 192]$ pixel. Use an algorithm for the direct solution (e.g., the MATLAB-code `HOME/rrs_3point.m`) of the spatial resection and determine the pose of the camera from the scene and image coordinates of four points, cf. `HOME/SRS_data.txt`:

i	$\boldsymbol{X}_i$ [m]	$\boldsymbol{x}_i'$ [pixel]
1	$[-0.1580, +0.1988, -1.8461]$	$[188.38, 138.27]$
2	$[+0.4830, +0.4154, -1.8252]$	$[377.97, 95.37]$
3	$[+0.3321, -0.2304, -1.6413]$	$[261.50, 295.72]$
4	$[-0.3991, +0.0896, -1.7235]$	$[106.02, 151.23]$

39. (2) RANSAC for DLT: Given are the manually measured sensor points $[x_i', y_i']$ of 3D points of Rubik's cube, cf. `HOME/RANSAC_cube.jpg`. The data are given in the data set `HOME/RANSAC_points.txt`: Each row contains the sensor coordinates and the nominal 3D coordinates. The observations contain outliers. Write a RANSAC routine for a DLT and find the erroneous observations. Explain which criteria you use and why.
40. (1) Decompose the projection matrix from Exerc. 9. Give its projection centre and its rotation matrix.
41. (2) Given is the image `HOME/building.jpg`. Determine the horizon $\mathcal{h}'$ from the vanishing point. Measure points manually.
42. (3) Fig. 12.37, left, shows the image of a barn taken with a Euclidean camera. The ground plan of the barn has two doors and is rectangular.
    Given are the image coordinates of the points $\mathcal{C}$ to $\mathcal{J}$ and of the vanishing points $\mathcal{V}_k, k = 1, 2, 3$, where $\mathcal{V}_3$ is the nadir point, see Table 12.7.

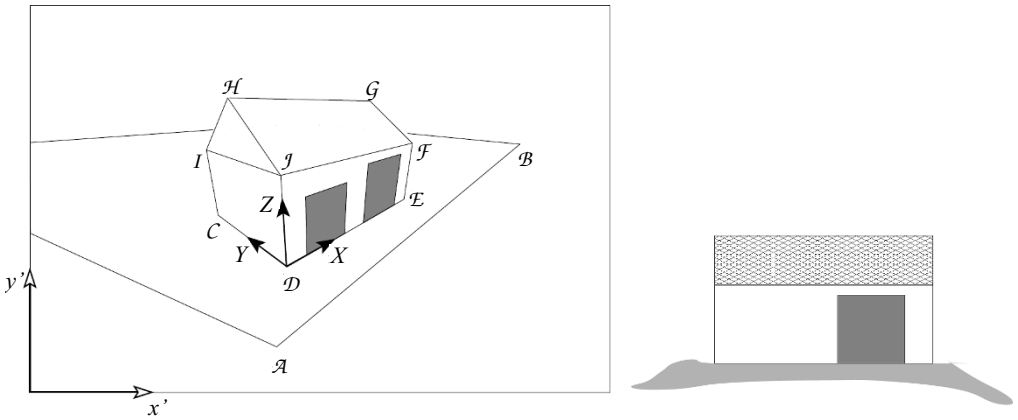


Fig. 12.37 Image and plan of a barn and coordinates of image and vanishing points, cf. HOME/barn-image.png and HOME/barn-plan.png

No.	x' [cm]	y' [cm]
C	4.540	4.278
D	6.195	3.043
G	5.920	7.034
H	4.773	7.103
I	4.260	5.852
J	6.053	5.234
V_1	12.601	6.632
V_2	0.861	7.024
V_3	7.018	-9.650

Table 12.7 Coordinates taken from a printout

a. According to the plan (see Fig. 12.37, right), originally the barn had only one door. What method is useful for transferring the shape of the second door into the plan? What invariant properties are used with this method? Construct the second door in the plan using printouts of the image and the plan.
b. Check numerically the coordinates of the nadir point $\mathcal{V}_3$ using the coordinates of the corner points of the barn.
c. Check numerically the principal distance $c = 56$ mm of the camera.
d. Determine the principal point.
e. Verify the rotation matrix R of the transformation ${}^c\boldsymbol{X} = {}^cR(\boldsymbol{X} - \boldsymbol{Z})$

$$ {}^cR = \begin{bmatrix} 0.82352 & 0.09535 & -0.53242 \\ 0.43343 & 0.77833 & +0.73212 \\ 0.36598 & 0.62056 & -0.42486 \end{bmatrix} . \tag{12.283} $$

43. (3) The task is to draw a traffic sign onto the right lane of a road shortly before a junction, see Fig. 12.38 upper left. The traffic sign should be drawn such that a driver, who has a distance of 30 m to the sign, perceives it as a circular sign.
Instead of the eye take as sensor system an ideal camera with $c = 400$ pixel whose projection centre is 1.33 m above the road and has viewing direction towards the centre of the traffic sign. The scene coordinate system is centred in O with the Y-axis along the road and the X-axes to the right. The Z-axis points upwards, see Fig. 12.38.
The goal is to develop a stencil which allows us to correctly draw the sign onto the road.

a. Determination of the projection matrix
   i. Give the calibration matrix including the units of their entries, where appropriate.

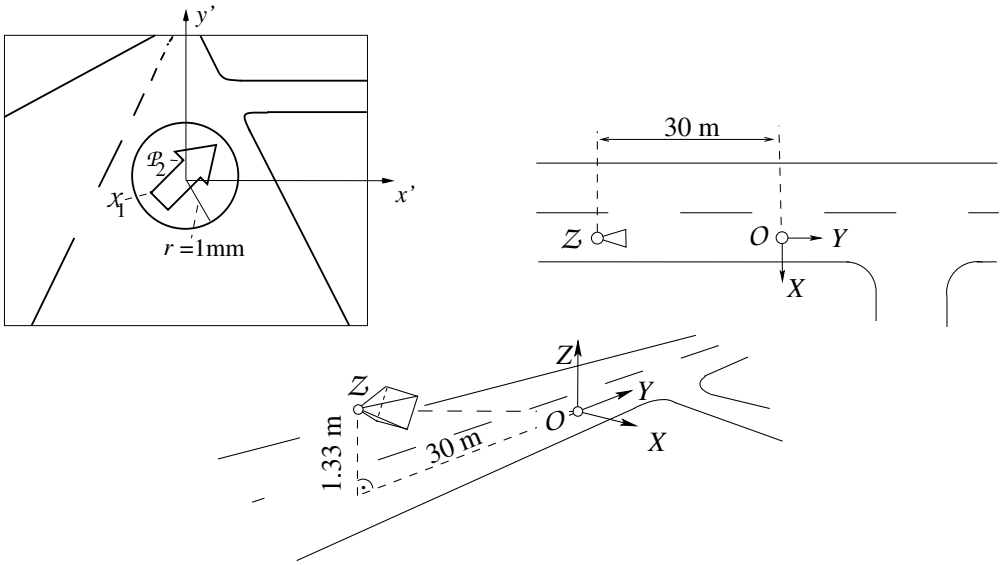


Fig. 12.38 Upper left: Image of a road sign on a lane. **Top right and bottom** Road scene

ii. Give the coordinates of the projection centre in the scene coordinate system.
iii. Verify the rotation and the projection matrices following the configuration in Fig. 12.38

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.04429 & 0.9990 \\ 0 & -0.9990 & 0.04429 \end{bmatrix} \quad \mathsf{P} = \begin{bmatrix} -400.000 & 0 & 0 & 0 \\ 0 & -17.716 & -399.6 & -0.0000 \\ 0 & -0.9990 & 0.0443 & -30.026 \end{bmatrix}$$

Give the unit of each nonzero element in P.

b. Mapping from the road into the image coordinate system.
i. Express the homogeneous coordinates $\mathbf{x}'$ of an image point for a general projection matrix of the form $\mathsf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ with its column vectors $\mathbf{p}_i$ for a general point $\mathcal{X}$ on the road.
ii. What type of projection is the mapping from the road into the image? Give it in the form $\mathbf{x}' = \mathsf{H}\mathbf{x}$, where $\mathbf{x} = [X, Y, 1]^\mathsf{T}$.

c. Mapping from image onto road
i. Confirm the matrix

$$\mathsf{B} = \begin{bmatrix} -0.0025 & 0 & 0 \\ 0 & -0.0564 & 0.0000 \\ 0 & 0.0019 & -0.0333 \end{bmatrix}$$

for the backprojection from the image onto the road.
ii. Where is the origin of the image coordinate system back projected to?
iii. What are the scene coordinates of the point $p'_1(-4, -2)$?
iv. Give the parameters $\mathbf{l}'_{12}$ of the image line through the points p'_1 and $p'_2(0, 2)$.
v. Backproject the image line $\mathbf{l}'_{12}$ and give $\mathbf{l}_{12}$ in the scene coordinate system.
vi. Conics in the image can be represented by $\mathbf{x}'^\mathsf{T}\mathsf{C}'\mathbf{x}' = 0$.
vii. Give the matrix C' for the circle in the image, which is supposed to have radius 8 pixels.
viii. Give the transformation matrix for the conic $C = f(C')$ such that $\mathbf{x} = \mathsf{B}\mathbf{x}'$.
ix. Verify the conic matrix

$$\mathsf{M} = \begin{bmatrix} -2.7723 & 0 & 0 \\ 0 & -0.0043 & 0.0333 \\ 0 & 0.0333 & 1.0000 \end{bmatrix}$$

for the boundary of the road sign.

d. Size of road sign
   i. Determine the semiaxes of the boundary of the road sign.
   ii. In Fig. 12.38 upper left you see the image of the sign. Draw the smaller semiaxis into the image.

44. (3) A projection is called a *parallel projection* if all projection rays are parallel. Thus the projection centre is at infinity.

a. Specialize the general form of the perspective projection $\mathbf{x}' = \mathsf{P}\mathbf{X}$ such that the mapping is a parallel projection. *Hint:* (1) start with a normal case where $\mathsf{R} = \mathsf{I}_3$, $\mathsf{K} = \mathrm{Diag}([1, 1, 1/c]$, and $\boldsymbol{Z} = [0, 0, c]$ and determine the limit of this projection matrix for $c \to \infty$. (2) Apply two transformations: a spatial motion of the camera and an affine transformation in the image. How many parameters do you have for these two transformations?

b. Show that a general straight line-preserving parallel projection can be represented as

$$\underset{2\times 1}{\boldsymbol{x}'} = \underset{2\times 4}{\mathsf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{12.284}$$

c. How many degrees of freedom has this mapping? Give a geometric explanation for the number of d.o.f.

d. What is the minimum number of corresponding points in the scene and in the image in order to determine P?

Chapter 13
Geometry and Orientation of the Image Pair

13.1 Motivation 547
13.2 The Geometry of the Image Pair 549
13.3 Relative Orientation of the Image Pair 568
13.4 Triangulation 596
13.5 Absolute Orientation and Spatial Similarity Transformation 607
13.6 Orientation of the Image Pair and Its Quality 608
13.7 Exercises 615

A pair of perspective images showing the scene from different directions is sufficient to reconstruct it without having pre-knowledge about it. This is not surprising, since it represents an analogy to our visual system, which also recovers the 3D structure of the scene based on the two retina images in our eyes.

Given two images, their mutual pose is constrained, as viewing rays from both cameras referring to the same scene points need to intersect, or – equivalently – be coplanar. We will algebraically describe the geometry of the image pair, especially the coplanarity constraint (Sect. 13.2). Here, depending on whether the cameras are calibrated or uncalibrated, the essential or the fundamental matrix describe the geometry of a pair of images in the same way that the projection matrix describes the geometry of a single camera. Knowing enough corresponding points in two perspective images allows us to determine the relative pose of the two cameras up to scale. We assume these correspondences to be provided, either manually or by some automatic matching procedure. We provide algorithms for recovering the orientation of the image pair (Sect. 13.3), especially various minimal solutions particularly for outlier detection and the determination of approximate values, but also statistically rigorous methods which allow an evaluation of the quality of the orientation. We then determine the 3D coordinates of scene points by triangulation (Sect. 13.4), i.e., by intersecting corresponding rays. The 3D points form a sparse description or model of the scene up to an unknown spatial transformation. Using known scene points, this transformation can be determined so that the 3D points of the scene in its coordinate system can be recovered (Sect. 13.5). The various algorithms give rise to several methods for orienting the image pair and determining 3D coordinates of scene points. We compare the quality of the different methods in Sect. 13.6 at the end of the chapter.

13.1 Motivation

If nothing is known about the object, we are not able to infer its three-dimensional structure from one image alone. This is because the depth information is lost when taking an image. We either need partial information about the object, e.g., its surface as discussed in the previous chapter, or additional images.

Therefore, we now consider two images. They need to be taken from different positions such that the scene points are seen from different directions, see Fig. 13.1; otherwise,

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_13

their geometry, i.e., the bundles of rays derived from the two images do not differ. If the scene is static, we might use the same camera and take the two images one after the other. With a scene that changes over time we need to take the two images simultaneously, so we need to synchronize two different cameras. Generally the images have different interior orientations IO′ and IO″ and different exterior orientations EO′ and EO″.

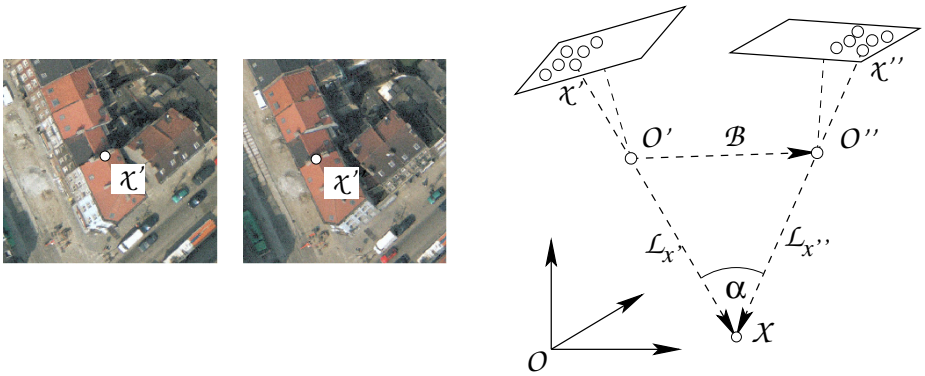


Fig. 13.1 Principle of two-view analysis. **Left:** Corresponding points x' and x'' on two perspective images referring to the same scene point X. **Right:** Geometry of an image pair. The image rays $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ from the projection centres O' and O'' forming the base line $\mathcal{B}$ intersect in the scene point with the *parallactic angle* α

corresponding points

A scene point X when projected into the two images leads to image points x' and x''. Such image points x' and x'' which refer to the same scene point X are called *corresponding image points* or *homologous image points*. Finding such points in two or more images is the correspondence problem, which can be solved manually or automatically, as discussed in the second volume. In the following we assume the correspondences are given.

control points and lines

We thus assume that we have measured several corresponding points x'_i and x''_i, and possibly corresponding straight lines l'_i and l''_i, in the two images. Observe the notation: The index i refers to the scene point X_i or line $\mathcal{L}_i$, and the prime and double prime indicate the first and the second image. Some of these scene points or lines may be known in object space, thus are control points or control lines, respectively.

We now have two tasks:

1. The determination of the *orientation of the image pair*.
2. The *reconstruction* of the coordinates of the scene features observed in the two images.

coplanarity constraint,

When enough control points or lines are visible in each image the first task can easily be solved using a direct linear transformation or a spatial resection for each image separately. This solution is suboptimal if there exist corresponding points (x', x''), visible in both images, whose scene coordinates are not known. These points impose powerful constraints on the mutual pose of the two cameras, as their projection rays $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ need to intersect or, equivalently, be coplanar. Based on this *coplanarity constraint*, interestingly, we can obtain what is called the *relative orientation of the images* (RO), even if no scene features are available.[1] It describes the geometric relation between the two images taken with calibrated or uncalibrated cameras, respectively. In both cases we are able to recover the relative pose of the two cameras up to an unknown transformation, which at least contains a scale parameter.

When performing the orientation tasks we therefore need to distinguish between situations where some scene features are known and situations where we do not have any scene

[1] Sometimes this is called the relative orientation of the two cameras. We will not use this term and always will refer to the relative pose, when addressing the relative motion between the cameras, which requires six parameters.

information. Furthermore, we need to analyse the geometry of the image pair without referring to known scene information.

The second task, the reconstruction of unknown scene points, scene lines, or other scene features visible in two images, can be solved by *intersection*, also called *triangulation*. For two corresponding points $(\mathcal{X}', \mathcal{X}'')$, the determination of the scene point $\mathcal{X}$ already is an overconstrained problem, as we have four image coordinates for determining the three spatial coordinates. The accuracy will depend on what is called the *parallactic angle* between the two projection rays.

intersection, triangulation

parallactic angle

The situation is different for straight 3D lines $\mathcal{L}$, not shown in the figure. The two projection planes $\mathcal{A}_{l'}$ and $\mathcal{A}_{l''}$ spanned by the projection centres and the image lines always intersect. There is *no constraint based on corresponding image lines* on the pose of the cameras as we have four coordinates (parameters) for the observed two image lines, which are necessary for determining the four parameters of the 3D line. If the 3D line is curved, we again will have constraints, but their number depends on the form of the 3D line, e.g., it will be different depending on whether we have a 3D circle, a 3D ellipse, or a free form line in 3D.

there is no constraint based on corresponding image lines

The complete procedure for orienting two images is shown in Fig. 13.2. It consists of two

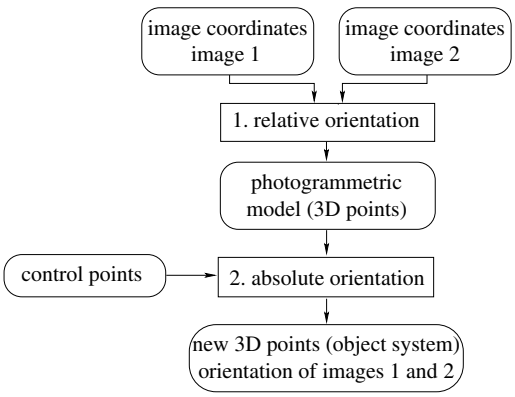


Fig. 13.2 Two-step procedure for the orientation of the image pair

steps. The relative orientation uses only image information, namely corresponding points, and yields a photogrammetric model. It consists of all derived 3D points and the relative pose of the two cameras up to a common similarity or projective transformation into the scene coordinate system. Since the algebraic relations between the image coordinates and the pose parameters are nonlinear and we have to handle possible outliers, we need direct as well as statistically optimal solutions for the relative orientation. The absolute orientation transforms the photogrammetric model into the scene system using control points. The result is statistically suboptimal and can be seen as an approximate solution which may be refined using bundle adjustment as discussed in Chap. 15.

photogrammetric model

13.2 The Geometry of the Image Pair

13.2.1 Number of Parameters 550
13.2.2 Coplanarity Constraints for Images of Uncalibrated Cameras 552
13.2.3 Coplanarity Constraint for Images of Calibrated Cameras 555
13.2.4 The Normal Case of the Image Pair 561
13.2.5 Epipolar Geometry 562
13.2.6 Generating Normalized Stereo Pairs 565
13.2.7 Homography and Homology Induced by a Plane 567

The geometry of the image pair describes the geometric relations between the scene points, two cameras and the image points. This is achieved explicitly by the orientation parameters and the scene coordinates or implicitly by constraints, which do not use the coordinates of the scene points. Depending on whether the cameras are calibrated or not, we use the spherical or the perspective camera model (see Sect. 12.7, p. 464) in order to exploit the generality of the algebraic expressions. Thus we use the spherical camera model for calibrated perspective cameras. This simplifies the algebraic description of the geometric relations, collected in what is called epipolar geometry of the image pair.

13.2.1 Number of Parameters for Orientation and Reconstruction

The orientation of an image pair can be described easily by the orientation of the two individual cameras, specifically their exterior and interior orientation, see Sect. 12.1.2.1, p. 460. For calibrated cameras where the IO is known, we need six parameters for the exterior orientation of each image and thus 12 parameters for the image pair.

In the case of uncalibrated cameras we have many options, depending on whether the IOs of the two cameras are the same or not, or whether we only assume the cameras to be straight line-preserving or not. In this chapter, we assume the nonlinear distortions of the two cameras to be sufficiently small enough to be neglected in the application. Furthermore we assume they have different interior orientations. Thus the two cameras follow a straight line-preserving perspective camera model and are characterized by their projection matrices. Then we need 11 parameters for each camera, five for the IO and six for the EO, so we need a total of 22 parameters to describe the orientation of a pair of straight line-preserving cameras.

in this chapter we assume cameras to be straight line-preserving

In both cases, control points or lines, i.e., points or lines known in object space and visible in the images, are required in order to determine the exterior orientation of the cameras.

To find the minimum number of control points required to determine the exterior orientation, we investigate how many and which parameters are already determinable if we have only two images and *no 3D information*.

13.2.1.1 Two Calibrated Cameras

We start with two calibrated cameras. If we had a perfect orientation of the cameras and if the image measurements were perfect, the two corresponding camera rays,[2] $\mathcal{L}_{x'} = \mathcal{O}' \wedge \mathcal{X}'$ and $\mathcal{L}_{x''} = \mathcal{O}'' \wedge \mathcal{X}''$, would intersect in the scene point $\mathcal{X}$. This is the *coplanarity constraint*,

$$\mathcal{L}_{x'}(\mathcal{X}', \mathcal{O}') \cap \mathcal{L}_{x''}(\mathcal{X}'', \mathcal{O}'') = \mathcal{X}\,, \tag{13.1}$$

since corresponding rays $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ of an oriented image pair are coplanar (see Fig. 13.1). Given the orientation parameters of the images, the 3D point $\mathcal{X}$ generally can be derived from its images $\mathcal{X}'$ and $\mathcal{X}''$ by triangulation. The angle between the two rays $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ is the *parallactic angle* α. The two rays are parallel, i.e., the parallactic angle is zero if the scene *point is at infinity*, if the two projection centres are identical, or if the ray passes through the two projection centres. In all three cases only the direction to the scene point can be derived. In the following we assume that the two projection centres are distinct if not stated otherwise.

coplanarity constraint

Observe, we did not refer to a specific camera model. For the spherical model the rays, e.g., $\mathcal{L}_{x'} = \mathcal{O}' \wedge \mathcal{X}'$, point from the projection centre to the 3D point. For perspective cameras, the direction of $\mathcal{L}_{x'}$ depends on whether the image is in viewing or taking position (see Sect. 12.1.1, p. 456).

[2] Interpreting the image points as points in 3D.

The two bundles of corresponding rays $\{\mathcal{L}_{x'_i}, \mathcal{L}_{x''_i}\}$ intersect in a set $\mathcal{X}_i$ of 3D points. They represent an angle-preserving model of the scene, a photogrammetric model. Nothing more can be derived from these bundles. Especially, the scale of the photogrammetric model cannot be determined, as there is no possibility of deriving length information from the angles encoded in the bundle of imaging rays.

Therefore, the orientation of the two cameras and hence the object can be reconstructed up to a spatial similarity transformation, which is specified by seven parameters (three translation parameters, three rotation parameters, one scale parameter), which means that only $12 - 7 = 5$ parameters out of the 12 total parameters of the exterior orientation of the calibrated image pair are determinable if no object information is available.

We can identify these five parameters: Assume two cameras in arbitrary pose in space which observe a 3D object represented by a set of points. Corresponding image rays intersect in the corresponding space points. What motions of the two cameras are possible if we fix the image points $\mathcal{X}'_i$ and $\mathcal{X}''_i$ and still require corresponding rays to intersect? The pose in space of the camera pair as a whole may be chosen arbitrarily if the cameras remain in the same relative position. This allows us to fix the pose of the camera pair by fixing the pose of one camera. Corresponding rays will no longer intersect if the second camera is rotated or shifted across the line joining the two cameras.

Only if the second camera is shifted in the direction towards or away from the first camera will corresponding rays remain coplanar. This leads to a geometric scaling, reduction or enlargement, of the point set produced by the intersecting rays (see Fig. 13.3). There is no constraint for corresponding image lines $\mathcal{l}'$ and $\mathcal{l}''$, as discussed above.

relative orientation

Therefore the following parameters fix the *relative orientation* of the two images in terms of the camera parameters:

- The rotation R_{12} of the second camera relative to the first, which involves three parameters.
- The direction $\boldsymbol{B}$ of the base line $O'O''$ connecting the two projection centres O' and O''. This involves additional two parameters, since the distance of the two projection centres cannot be determined from the coplanarity constraint.

Given enough corresponding image points, the relative orientation is only unique if we assume that the projection rays are oriented, thus if they are half-rays. Otherwise we could reverse the direction of the basis or rotate one camera around the basis by 180° without violating the coplanarity constraint. However, in that case the projection rays would not intersect *in front of the cameras* anymore, a notion which needs to be clarified for spherical cameras. We will discuss this situation in more detail when estimating the relative orientation of two images.

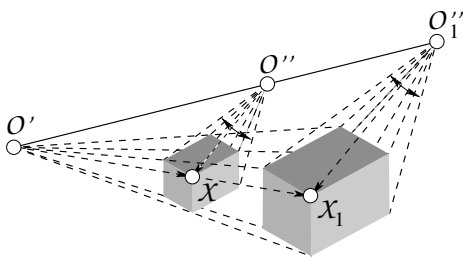


Fig. 13.3 The coplanarity constraint is invariant with respect to a translation of the camera along the base line $O'O''$, e.g., from O'' to O''_1. The two bundles at O'' and O''_1 have rays with the same spatial directions. This visualizes the fact that the scale of the photogrammetric model cannot be determined from image information alone

Table 13.1 sums up the results so far: The relative orientation of two images taken with calibrated cameras is characterized by five independent parameters. An object can

be reconstructed from two images of calibrated cameras only up to a spatial similarity transformation. The result is a photogrammetric model.

absolute orientation

The orientation of the photogrammetric model in space is called the *absolute orientation*. The notion again refers to the process of orientation as well as to the result of the orientation. To determine the absolute orientation of a photogrammetric model derived from images taken with calibrated cameras, at least seven constraints are required.

Table 13.1 Number of free parameters of the orientation O=IO+EO, the relative orientation RO of the images, the absolute orientation AO of the photogrammetric model

cameras	# O/image	# O/image pair	# RO	# AO
calibrated spherical perspective	6	12	5	7
uncalibrated perspective	11	22	7	15

13.2.1.2 Two Uncalibrated Cameras

For uncalibrated cameras, which we assume to be straight line-preserving perspective cameras, we make a similar argument. Starting from an oriented pair of uncalibrated cameras we can again reconstruct the scene points from their corresponding image points by triangulation. Since the mapping from scene to image space is straight line-preserving, straight lines, not angles, are the most general invariants in object space. From the information in an image pair we can therefore reconstruct the object up to a straight line preserving transformation (Faugeras, 1992). This is a spatial homography. It is described by a homogeneous 4×4 matrix, thus by a minimum of 15 parameters in the general case. Therefore, only $22 - 15 = 7$ parameters of the interior and the exterior orientation of the image pair can be reconstructed from two images alone. Here the interpretation is less intuitive. However, it can be shown that, along with the rotation and the direction of the base line, we can determine the two principal distances (Hartley, 1992) in a general situation.

Table 13.1 again summarizes the result: The relative orientation of two images of uncalibrated cameras is characterized by seven independent parameters. An object can be reconstructed only up to a spatial homography. The result is also called a photogrammetric model.

The *absolute orientation* of a photogrammetric model of *uncalibrated* straight line-preserving perspective *cameras* needs at least 15 constraints.

13.2.2 Coplanarity Constraints for Images of Uncalibrated Cameras

We now give explicit expressions for the coplanarity constraint for the relative orientation of two images taken with uncalibrated straight line-preserving cameras which for reasons of generality are assumed to have different interior orientation.

13.2.2.1 The Coplanarity Constraint

Let the two cameras be characterized by the two projection matrices P' and P''. Thus we have the following mappings:

$$\mathbf{x}' = \mathsf{P}'\mathbf{X} \qquad \mathbf{x}'' = \mathsf{P}''\mathbf{X}\,, \tag{13.2}$$

with the projection matrices, see (12.44), p. 474

$$\mathsf{P}' = [\mathsf{A}'|\mathbf{a}'] = \mathsf{K}'R'[I_3| - \boldsymbol{Z}'] \qquad \mathsf{P}'' = [\mathsf{A}''|\mathbf{a}''] = \mathsf{K}''R''[I_3| - \boldsymbol{Z}''] \,. \tag{13.3}$$

The coplanarity constraint of the three space vectors $\mathcal{O}'\mathcal{X}'$, $\mathcal{O}''\mathcal{X}''$ and $\mathcal{O}'\mathcal{O}''$ can be expressed as

$$\det[\mathcal{O}'\mathcal{X}' \quad \mathcal{O}'\mathcal{O}'' \quad \mathcal{O}''\mathcal{X}''] = 0 \,, \tag{13.4}$$

where $\det[\cdot,\cdot,\cdot]$ is the volume of the parallelepiped of three vectors.

The directions of the vectors $\mathcal{O}'\mathcal{X}'$ and $\mathcal{O}''\mathcal{X}''$ can be derived from the image coordinates $\mathbf{x}'$ and $\mathbf{x}''$ using the parameters of the interior and the exterior orientation, while the *base vector* $\mathcal{O}'\mathcal{O}''$ directly results from the coordinates of the projection centres. The determination of the volume requires that the three vectors are given in mutually parallel coordinate systems. We express the base vector in the scene coordinate system $S := S_o$ and the directions in the normalized camera systems S_n (see the normalized camera in Table 12.1, p. 479),

base vector

$$\mathbf{b} = \boldsymbol{B} = \boldsymbol{Z}'' - \boldsymbol{Z}' \,, \quad {}^n\mathbf{x}' = R'^{\mathsf{T}}(\mathsf{K}')^{-1}\mathbf{x}' \,, \quad {}^n\mathbf{x}' = R'^{\mathsf{T}}(\mathsf{K}'')^{-1}\mathbf{x}'' \,; \tag{13.5}$$

the coplanarity constraint (13.4) is

$$|{}^n\mathbf{x}' \,, \mathbf{b} \,, {}^n\mathbf{x}''| = \begin{vmatrix} {}^nx' & B_x & {}^nx'' \\ {}^ny' & B_y & {}^ny'' \\ 1 & B_z & 1 \end{vmatrix} = {}^n\mathbf{x}' \cdot (\mathbf{b} \times {}^n\mathbf{x}'') = {}^n\mathbf{x}'^{\mathsf{T}}\mathsf{S}_{\mathrm{b}}\,{}^n\mathbf{x}'' = 0 \,. \tag{13.6}$$

Using (13.5) we have, explicitly,

$$\mathbf{x}'^{\mathsf{T}}(\mathsf{K}')^{-\mathsf{T}}R'\mathsf{S}_{\mathrm{b}}R''^{\mathsf{T}}(\mathsf{K}'')^{-1}\mathbf{x}'' = 0 \,. \tag{13.7}$$

The constraint is a bilinear form in the vectors $\mathbf{x}'$ and $\mathbf{x}''$ and depends on the ten parameters of the two calibration matrices, the six rotation parameters and the two parameters of the direction of the basis. However, the relative orientation of two images of uncalibrated cameras has only seven degrees of freedom, a difference which will be relevant when deriving the projection matrices from a given fundamental matrix in Sect. 13.3.7.1, p. 594.

13.2.2.2 The Fundamental Matrix

We therefore analyse the 3×3 matrix

$$\mathsf{F} = (\mathsf{K}')^{-\mathsf{T}}R'\mathsf{S}_{\mathrm{b}}R''^{\mathsf{T}}(\mathsf{K}'')^{-1} \tag{13.8}$$

of the bilinear form (13.7). It is called the *fundamental matrix* of the relative orientation of a pair of images of uncalibrated cameras. The general form of the coplanarity constraint then reads as

coplanarity constraint for uncalibrated cameras

$$\mathbf{x}'^{\mathsf{T}}\mathsf{F}\mathbf{x}'' = 0 \,. \tag{13.9}$$

Several remarks are useful here:

- The coplanarity constraint is bilinear in the homogeneous image coordinates $\mathbf{x}'$ and $\mathbf{x}''$ and linear in the elements of the fundamental matrix. This is the basis for a simple determination of the fundamental matrix from corresponding points.
- The fundamental F matrix has seven degrees of freedom. This is because F is homogeneous and singular, since the skew symmetric matrix $\mathsf{S}_{\mathrm{b}'}$ is singular with rank 2. Therefore any matrix of the form

$$\mathsf{F} = U\,\mathrm{Diag}([s_1, s_2, 0])\,V^{\mathsf{T}} \,, \qquad s_i > 0 \,, \tag{13.10}$$

with orthogonal matrices U and V, is a fundamental matrix. As shown in Sect. 13.2.1.2, p. 552, seven parameters are *necessary* for describing the relative orientation of two images taken with uncalibrated cameras. Since the fundamental matrix is *sufficient* for describing the relative orientation in terms of the epipolar geometry and has seven degrees of freedom, the fundamental matrix contains the complete information about the relative orientation of two images of uncalibrated cameras. Since we assume a central projection, the seven parameters refer to five parameters for the relative pose of the two cameras and to two parameters of the interior orientation of the two cameras.

fundamental matrix from projection matrices

- If the projection matrices are given, we can derive the fundamental matrix without the complete partitioning of P' and P''. Let the projection matrices be partitioned into a left 3×3 matrix and a 3-vector as in (13.3), then the fundamental matrix is given by

$$\mathsf{F} = \mathsf{A}'^{-\mathsf{T}} \mathsf{S}_{\mathbf{b}_{12}} \mathsf{A}''^{-1} \quad \text{with} \quad \mathbf{b}_{12} = \mathsf{A}''^{-1}\mathbf{a}'' - \mathsf{A}'^{-1}\mathbf{a}' \,. \tag{13.11}$$

This is because the projection centres are $\boldsymbol{Z}' = -\mathsf{A}'^{-1}\mathbf{a}'$ and $\boldsymbol{Z}'' = -\mathsf{A}''^{-1}\mathbf{a}''$, $\mathsf{A}' = \mathsf{K}'R'$ and $\mathsf{A}'' = \mathsf{K}''R''$. Equation (13.11) can also be written as

$$\mathsf{F} = \mathsf{A}'^{\circ} \mathsf{S}_{\mathbf{b}_{12}} \mathsf{A}''^{\circ\mathsf{T}} \,, \tag{13.12}$$

since for any square matrix A we have $\mathsf{A}^{-1} = \mathsf{A}^{\circ\mathsf{T}}/|\mathsf{A}|$ with the cofactor matrix A°, see (A.19), p. 769. If the first projection matrix is fixed to be $\mathsf{P}' = [I_3|\mathbf{0}]$, and $\mathsf{P}'' = [\mathsf{A}''|\mathbf{a}'']$, the fundamental matrix has the form

$$\mathsf{F} = \mathsf{A}''\mathsf{S}(\mathbf{a}'') \,, \tag{13.13}$$

since $\mathsf{F} = \mathsf{S}_{\mathbf{b}_{12}}\mathsf{A}''^{\circ\mathsf{T}} = \mathsf{S}_{\mathsf{A}''^{\circ\mathsf{T}}\mathbf{a}''}\mathsf{A}''^{\circ\mathsf{T}} = \mathsf{A}''\mathsf{S}(\mathbf{a}'')$ due to $\mathbf{b}_{12} = \mathsf{A}''^{-1}\mathbf{a} = \mathsf{A}''^{\circ\mathsf{T}}\mathbf{a}$ and (A.47), p. 772 and (A.25), p. 770.

fundamental matrix from camera planes

- An alternative expression for the fundamental matrix F exploits the tools from algebraic projective geometry, which will be used when describing the relations between three images.

The derivation starts from the projection matrices of the two cameras,

$$\mathsf{P}_1 := \mathsf{P}' = \begin{bmatrix} \mathbf{A}_1^{\mathsf{T}} \\ \mathbf{B}_1^{\mathsf{T}} \\ \mathbf{C}_1^{\mathsf{T}} \end{bmatrix} \qquad \mathsf{P}_2 := \mathsf{P}'' = \begin{bmatrix} \mathbf{A}_2^{\mathsf{T}} \\ \mathbf{B}_2^{\mathsf{T}} \\ \mathbf{C}_2^{\mathsf{T}} \end{bmatrix} \,, \tag{13.14}$$

the vectors $\mathbf{A}_i$, etc., which are 4-vectors representing the camera planes of each camera, i.e., the planes passing through the projection centre and the axes of the sensor coordinate system (see Sect. 12.1.3.8, p. 473).

The fundamental matrix is given by the 3×3 matrix

$$\mathsf{F} = - \begin{bmatrix} |\mathbf{B}_1, \mathbf{C}_1, \mathbf{B}_2, \mathbf{C}_2| & |\mathbf{B}_1, \mathbf{C}_1, \mathbf{C}_2, \mathbf{A}_2| & |\mathbf{B}_1, \mathbf{C}_1, \mathbf{A}_2, \mathbf{B}_2| \\ |\mathbf{C}_1, \mathbf{A}_1, \mathbf{B}_2, \mathbf{C}_2| & |\mathbf{C}_1, \mathbf{A}_1, \mathbf{C}_2, \mathbf{A}_2| & |\mathbf{C}_1, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_2| \\ |\mathbf{A}_1, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_2| & |\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_2, \mathbf{A}_2| & |\mathbf{A}_1, \mathbf{B}_1, \mathbf{A}_2, \mathbf{B}_2| \end{bmatrix} \,, \tag{13.15}$$

composed of 4×4 determinants. It results from the coplanarity constraint that the two projection lines $\mathbf{L}_{x'}$ and $\mathbf{L}_{x''}$ intersect,

$$\mathbf{L}_{x'}^{\mathsf{T}} \overline{\mathbf{L}}_{x''} = 0 \,. \tag{13.16}$$

We can see this when using (12.85), p. 483

$$\mathbf{L}_{x'} = u'(\mathbf{B}_1 \cap \mathbf{C}_1) + v'(\mathbf{C}_1 \cap \mathbf{A}_1) + w'(\mathbf{A}_1 \cap \mathbf{B}_1) \tag{13.17}$$

$$\mathbf{L}_{x''} = u''(\mathbf{B}_2 \cap \mathbf{C}_2) + v''(\mathbf{C}_2 \cap \mathbf{A}_2) + w''(\mathbf{A}_2 \cap \mathbf{B}_2) \tag{13.18}$$

and collecting the coefficients for products $u'u''$, etc., e.g., $F_{11} = (\mathbf{B}_1 \cap \mathbf{C}_1)^\mathsf{T}\overline{(\mathbf{B}_2 \cap \mathbf{C}_2)} = -|\mathbf{B}_1, \mathbf{C}_1, \mathbf{B}_2, \mathbf{C}_2|$, due to (7.61), p. 304. With the rows $\overline{\mathbf{L}}_{1i}$ and $\overline{\mathbf{L}}_{2j}$ of the projection matrices Q' and Q'', see (12.77), p. 481, this is equal to $F_{11} = \mathbf{L}_{11}^\mathsf{T}\overline{\mathbf{L}}_{21}$. Therefore the fundamental matrix can also be written as

$$\mathsf{F} = [F_{ij}] = [\mathbf{L}_{1i}^\mathsf{T}\overline{\mathbf{L}}_{2j}] = \mathsf{Q}'\,\overline{\mathsf{Q}}''^\mathsf{T}\,, \tag{13.19}$$

a form we will use when discussing the geometry of the image pair in Sect. 13.2.5, p. 562.

- If the fundamental matrix is known together with the covariance matrix of its elements (see Sect. 13.3.2.2, p. 572) we can check the correspondence of two points $\mathbf{x}'$ and $\mathbf{x}''$. Using the Kronecker product (see Sect. A.7, p. 775) the left side of the coplanarity constraint (13.9) will yield a residual

$$w = (\mathbf{x}'' \otimes \mathbf{x}')^\mathsf{T}\mathbf{f}\,, \tag{13.20}$$

with the vector $\mathbf{f} = \mathrm{vec}\mathsf{F}$ collecting the columns of F. The variance of w can be given explicitly as

$$\sigma_w^2 = [\mathbf{l}'^\mathsf{T} \mid \mathbf{l}''^\mathsf{T}] \begin{bmatrix} \Sigma_{\mathbf{x}'\mathbf{x}'} & \Sigma_{\mathbf{x}'\mathbf{x}''} \\ \Sigma_{\mathbf{x}''\mathbf{x}'} & \Sigma_{\mathbf{x}''\mathbf{x}''} \end{bmatrix} \begin{bmatrix} \mathbf{l}' \\ \mathbf{l}'' \end{bmatrix} \tag{13.21}$$

$$+(\mathbf{x}'' \otimes \mathbf{x}')^\mathsf{T}\Sigma_{\mathrm{ff}}(\mathbf{x}'' \otimes \mathbf{x}')\,, \tag{13.22}$$

since we have the Jacobians

$$\frac{\partial w}{\partial \mathbf{x}'} = \mathbf{x}''^\mathsf{T}\mathsf{F}^\mathsf{T} =: \mathbf{l}'^\mathsf{T}\,,\quad \frac{\partial w}{\partial \mathbf{x}''} = \mathbf{x}'^\mathsf{T}\mathsf{F} =: \mathbf{l}''^\mathsf{T}\,,\quad \frac{\partial w}{\partial \mathbf{f}} = (\mathbf{x}'' \otimes \mathbf{x}')^\mathsf{T}\,. \tag{13.23}$$

statistical test for correspondence

Here we assumed that the observed coordinates $\mathbf{x}'$ and $\mathbf{x}''$ are correlated, see the discussion in Sect. 10.6.3.1, p. 425 and exercise 21, p. 435. In addition we assumed the coordinates are independent of the elements of the fundamental matrix. This will certainly be the case if the point pair has not taken part in the determination of the fundamental matrix. The test statistic

$$z = \frac{w}{\sigma_w} \sim \mathcal{N}(0, 1) \tag{13.24}$$

can be tested for significance. It is normally distributed if the perspective model holds and the two points are corresponding. Thus we will reject the hypothesis that the two points $\mathcal{X}'$ and $\mathcal{X}''$ correspond if

$$|z| > k_\alpha\,, \tag{13.25}$$

where k_α is the $(1-\alpha)$ percentile of the normal distribution, e.g., $k_\alpha = 1.96$ for $\alpha = 5\%$.

- The definition (13.8), p. 553 of the fundamental matrix is not the same as in Hartley and Zisserman (2000) and Faugeras and Luong (2001). It generally differs just by a transposition. However, in the context of many images we will call F_{ij} that fundamental matrix which yields the constraint $\mathbf{x}'^\mathsf{T}_i\mathsf{F}_{ij}\mathbf{x}'_j = 0$. Thus in our case, for images 1 and 2, we have $\mathsf{F} \doteq \mathsf{F}_{12}$.[3]

13.2.3 Coplanarity Constraint for Images of Calibrated Cameras

For images of calibrated cameras the coplanarity constraint can be simplified by using the directions ${}^c\mathbf{x}'$ and ${}^c\mathbf{x}''$ of the camera rays of a *spherical camera*. If we start from sensor coordinates $\mathbf{x}'$ and $\mathbf{x}''$ of a calibrated perspective camera, the directions ${}^c\mathbf{x}'$ and ${}^c\mathbf{x}''$ from

[3] The definition in Hartley and Zisserman (2000) is $\mathsf{F} \doteq \mathsf{F}_{21} = \mathsf{F}_{12}^\mathsf{T}$, due to $\mathbf{x}_i''^\mathsf{T}\mathsf{F}\mathbf{x}'_i = 0$.

the projection centres to the image points in the camera coordinate systems $S_{c'}$ and $S_{c''}$ are

$$ {}^c\mathbf{x}' = -\text{sign}(c')\mathsf{K}'^{-1}\mathbf{x}' = -\text{sign}(c') \begin{bmatrix} {}^i x' \\ {}^i y' \\ c' \end{bmatrix} \tag{13.26} $$

(see (12.22), p. 469 and (12.109), p. 492) and similarly for ${}^c\mathbf{x}''$.[4]

Hence, in the case of calibrated perspective cameras with principal distances c' and c'', which need not be identical, we obtain the elements of the direction vectors explicitly from the image coordinates ${}^i x'$ and ${}^i x''$ in the ideal cameras.

relative orientation of images of spherical cameras

In the following we allow for bundles of rays of two *calibrated perspective cameras* when determining the relative orientation. The cameras then also may be *calibrated spherical cameras*, e.g., if the camera rays are derived from the image coordinates of a fish-eye or a catadioptric camera.

13.2.3.1 The Essential Matrix

From the coplanarity constraint (13.7) for uncalibrated cameras we immediately obtain

$$ {}^c\mathbf{x}'^{\mathsf{T}}\, R'\, \mathsf{S}_{\mathrm{b}}\, R''^{\mathsf{T}}\, {}^c\mathbf{x}'' = 0\,, \tag{13.27} $$

which is equivalent to requiring the determinant

$$ |R'^{\mathsf{T}}\, {}^c\mathbf{x}',\ \mathbf{b},\ R''^{\mathsf{T}}\, {}^c\mathbf{x}''| = 0 $$

to vanish. We define the *essential matrix*,

$$ \mathsf{E} \doteq R'\, \mathsf{S}_{\mathrm{b}}\, R''^{\mathsf{T}}\,, \tag{13.28} $$

and obtain the coplanarity constraint for calibrated cameras,

$$ {}^c\mathbf{x}'^{\mathsf{T}}\, \mathsf{E}\, {}^c\mathbf{x}'' = 0\,. \tag{13.29} $$

With observed image coordinates ${}^i\boldsymbol{x}'$ and ${}^i\boldsymbol{x}''$ in an ideal camera, see (13.26), p. 556, we explicitly have

$$ [{}^i x',\ {}^i y',\ c'] \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix} \begin{bmatrix} {}^i x'' \\ {}^i y'' \\ c'' \end{bmatrix} = 0 \tag{13.30} $$

independent of the sign of the principal distances. Some remarks are helpful:

- The coplanarity constraint for calibrated cameras is a bilinear form in the direction vectors ${}^c\mathbf{x}'$ and ${}^c\mathbf{x}''$ and linear in the elements of the essential matrix E . This again gives rise to direct solutions for the essential matrix.
- The essential matrix E has five degrees of freedom. Therefore it has to fulfil $9-5=4$ constraints which can be expressed as a function of the columns of E and its cofactor matrix $\mathsf{E}^\circ = \text{cof}(\mathsf{E})$ (A.19), p. 769,

$$ \mathsf{E} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3] \qquad \mathsf{E}^\circ = [\boldsymbol{a}_2 \times \boldsymbol{a}_3, \boldsymbol{a}_3 \times \boldsymbol{a}_1, \boldsymbol{a}_1 \times \boldsymbol{a}_2] = [\boldsymbol{a}_1^\circ, \boldsymbol{a}_2^\circ, \boldsymbol{a}_3^\circ] \tag{13.31} $$

constraints on the essential matrix

It is homogeneous. Therefore we may fix its squared Frobenius norm to 2,

$$ ||\mathsf{E}||^2 = |\boldsymbol{a}_1|^2 + |\boldsymbol{a}_2|^2 + |\boldsymbol{a}_3|^2 = 2\,, \tag{13.32} $$

and this fixes the length of the base vector to $|\mathbf{b}| = 1$. Moreover we have the constraints (Rinner, 1963)

[4] Assuming $|\mathsf{K}| > 0$ and $\mathbf{x}'$ positive, thus $x_h > 0$.

$$\begin{aligned} |\boldsymbol{a}_1|^2 + |\boldsymbol{a}_1^\circ|^2 &= 1 \\ |\boldsymbol{a}_2|^2 + |\boldsymbol{a}_2^\circ|^2 &= 1 \\ |\boldsymbol{a}_3|^2 + |\boldsymbol{a}_3^\circ|^2 &= 1\,. \end{aligned} \tag{13.33}$$

Proof: Without loss of generality we may choose $\mathsf{E} = \mathsf{S}(\boldsymbol{b})$, since we may replace $R^\mathsf{T}\boldsymbol{a}_i = \boldsymbol{c}_i$, $R^\mathsf{T}(\boldsymbol{a}_i \times \boldsymbol{a}_j) = \boldsymbol{c}_i \times \boldsymbol{c}_j$ and $\boldsymbol{c}_i^\circ = R^\mathsf{T}\boldsymbol{a}_i^\circ$. If we assume $|\boldsymbol{b}| = |(a,b,c)^\mathsf{T}| = 1$, we get the proof for index 1:

$$|\boldsymbol{c}_1|^2 + |\boldsymbol{c}_1^\circ|^2 = \left|\begin{bmatrix} 0 \\ c \\ -b \end{bmatrix}\right|^2 + \left|\begin{bmatrix} -c \\ 0 \\ a \end{bmatrix} \times \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix}\right|^2 = (b^2 + c^2) + a^2 \left|\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right|^2 = 1\,.$$

Rinner (1963) shows that the four constraints (13.32) and (13.33) are independent. ◇

- The essential matrix is singular

$$|\mathsf{E}| = 0\,. \tag{13.34}$$

- We have the nine constraints (see Stefanovic, 1973),

$$\mathsf{E}\mathsf{E}^\mathsf{T}\,\mathsf{E} - \frac{1}{2}\mathrm{tr}(\mathsf{E}\mathsf{E}^\mathsf{T})\,\mathsf{E} = \mathbf{0} \tag{13.35}$$

  which result from the cubic of a skew 3×3 matrix (A.4.2), p. 771, and will be used for a direct solution for relative orientation with five points in Sect. 13.3.2.4, p. 575.

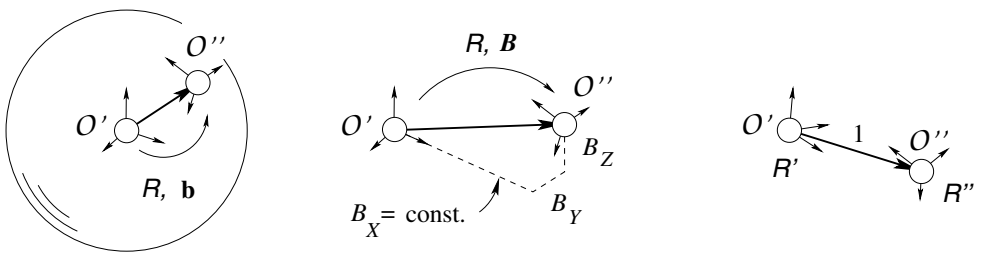
- Finally we have the SVD of an essential matrix, *SVD of essential matrix*

$$\mathsf{E} = U\,\mathrm{Diag}([s,s,0])\,V^\mathsf{T}\,, \qquad s > 0\,, \tag{13.36}$$

  where the first two singular values are identical. The SVD is not unique since with any rotation matrix $R = R_3(\alpha)$ we have $\mathsf{E} = (UR)\mathrm{Diag}([s,s,0])(R^\mathsf{T}V^\mathsf{T}) = \overline{U}\mathrm{Diag}([s,s,0])\overline{V}^\mathsf{T}$. This freedom can be used to specify one of the two orthogonal matrices, say U, by only two parameters. Due to the homogeneity of the essential matrix it therefore has five degrees of freedom.

13.2.3.2 Parametrizations of Relative Orientation

We now take into account that the relative orientation of two images is determined by only five independent parameters and give three classical parametrizations which have an intuitive geometric interpretation. We refer you to Fig. 13.4, p. 557.



(1) (2) (3)

Fig. 13.4 Parametrization of relative orientation of two images taken with calibrated cameras. **(1)** General parametrization of dependent images with the normalized direction vector $\mathbf{b}$, with $|\mathbf{b}| = 1$ and the rotation matrix R. Thus the second projection centre O'' is located on the unit sphere around the first projection centre O. **(2)** Photogrammetric parametrization of dependent images with two components B_Y and B_Z of the base vector $\boldsymbol{B}$ and the rotation matrix R. The component B_X of the base vector is fixed, either to 1 or to another value close to the length of the basis in the scene coordinate system. **(3)** Parametrization with independent images using two rotation matrices, $R'(\phi', \kappa')$ and $R''(\omega'', \phi'', \kappa'')$. Here the base vector defines the X-axis of the image pair

Parametrization for Dependent Images. Starting from (13.28), p. 556 we refer to the coordinate system of the first camera and describe the relative orientation via the direction of the base vector and the rotation parameters of the second camera: Thus we assume $R' = I_3$. The mutual rotation $R := R''$ and the basis $\mathbf{b} = \boldsymbol{B}$ refer to the coordinate system of the first camera. The coordinate systems of the base vector and the rotation matrices are not indicated.

With the rotation matrix R and the skew symmetric matrix of the base vector $\mathbf{b} = \boldsymbol{B}$, the essential matrix is

$$\mathsf{E} = \mathsf{S}_\mathrm{b} R^\mathsf{T} \tag{13.37}$$

and the coplanarity constraint reads

$$^c\mathbf{x}'^\mathsf{T} \mathsf{S}_\mathrm{b} R^\mathsf{T} {}^c\mathbf{x}'' = 0\,. \tag{13.38}$$

Since the base vector contains three elements but has only two degrees of freedom, since the length is arbitrary, we need to impose one constraint.

This can be done in two ways:

general parametrization with dependent images

1. General parametrization: We describe the relative pose of the second camera using the direction $\mathbf{b}/|\mathbf{b}|$ of the base vector $\mathbf{b} = [B_X, B_Y, B_Z]^\mathsf{T}$ and the rotation matrix R. We therefore obtain

$$^c\mathbf{x}'^\mathsf{T} \mathsf{S}_\mathrm{b} R^\mathsf{T} {}^c\mathbf{x}'' = 0 \qquad \text{with} \qquad |\mathbf{b}| = 1\,. \tag{13.39}$$

   Thus the relative orientation is described by six parameters and one additional constraint on the length of the base vector $\mathbf{b}$:

$$(B_X, B_Y, B_Z, \omega, \phi, \kappa) \qquad \text{with} \qquad B_X^2 + B_Y^2 + B_Z^2 = 1\,, \tag{13.40}$$

   all elements referring to the coordinate system of the first camera.

photogrammetric parametrization with dependent images

2. Classical photogrammetric parametrization: Here we assume the direction of the base vector to be approximately in the X-direction; for aerial images very often this is the flight direction. We may then arbitrarily fix the length of the base vector in this direction and obtain

$$^c\mathbf{x}'^\mathsf{T} \mathsf{S}_\mathrm{b} R^\mathsf{T} {}^c\mathbf{x}'' = 0 \qquad \text{with} \qquad B_X = \text{constant}\,. \tag{13.41}$$

   Thus we parametrize the relative orientation with the three parameters of the rotation matrix R and the two components B_Y and B_Z of the base vector,

$$(B_Y, B_Z, \omega, \phi, \kappa)\,, \tag{13.42}$$

   again all elements referring to the coordinate system of the first camera.

In all cases the basis $\mathbf{b}$ is identical to the left epipole $^c\mathbf{e}'$, measured in the first camera system,

$$\mathbf{b} = {}^c\mathbf{e}'\,, \tag{13.43}$$

since $^c\mathbf{e}'^\mathsf{T}\mathsf{E} = \mathbf{b}^\mathsf{T}\mathsf{S}(\mathbf{b})R^\mathsf{T} = \mathbf{0}$.

Both parametrizations for dependent images are asymmetric with respect to the two images. Therefore we sometimes use the following symmetric parametrization, e.g., when rectifying images into what is called the normal case, which is useful for stereo analysis, see Sect. 13.2.6, p. 565.

parametrization with independent images

Parameterization with Independent Images. Here we fix the base vector $\boldsymbol{B}$ and describe the relative orientation with parameters of independent rotations, R_i' and R_i'', for the cameras, the index standing for independent images.

The base vector is assumed to be $\mathbf{b} = [B_X, 0, 0]^\mathsf{T}$ with a priori fixed length B_X =const. The rotations of both cameras together need six parameters. A common rotation of the two cameras around the X-axis is not determinable from the two image bundles; it would lead to a rotation of the camera pair as a whole, which is part of the absolute orientation. Therefore the rotation angles are constrained such that only the difference $\Delta\omega = \omega'' - \omega'$ of the two cameras, or just one ω-rotation around the X axis, usually ω'', is a free parameter.

We therefore obtain from (13.28), p. 556,

$$ {}^c\mathbf{x}'^\mathsf{T} R' \mathsf{S}_\mathrm{b} R''^\mathsf{T} {}^c\mathbf{x}'' = 0 \quad \text{with constant} \quad \mathsf{S}_\mathrm{b}\,, \qquad \omega' = -\omega'' = -\frac{1}{2}\,\Delta\omega\,, \tag{13.44} $$

with the five parameters

$$ (\Delta\omega, \phi', \kappa', \phi'', \kappa''), \tag{13.45} $$

which are symmetric w.r.t. the two cameras.

Discussion of the Parametrizations:

- All of the mentioned parametrizations demonstrate that the relative orientation can be *parametrized with angles only*, since even in the parametrization (2), see (13.41), p. 558, the direction of the base vector is parametrized via the direction cosines B_Y/B_X and B_Z/B_X. This is to be expected since the points of an image together with the projection centre geometrically represent two *bundles of rays* from which only angles can be derived, not distances.
- While these two parametrizations are general and thus can represent all two-camera configurations, the classical photogrammetric parametrization (2), see (13.41), p. 558, has a singularity in the following situation: If the base vector is directed orthogonal to the X axis, the base components B_Y and B_Z generally will be infinitely large. This parametrization therefore leads to instabilities close to such a two-camera configuration.

singularity of the classical photogrammetric parametrization

- In all cases one base component is fixed to be nonzero, defining the scale of the photogrammetric model.
- The coordinate system chosen for relative orientation is identical to the coordinate system of the photogrammetric model.
- For normalized cameras (see Fig. 12.7, p. 464) with $R' = R'' = I_3$ we obtain the essential matrix

$$ \mathsf{E} = \mathsf{S}_\mathrm{b}\,. \tag{13.46} $$

As the general parametrization for dependent images (13.39) and (13.40), p. 558 is the simplest one, we will use it in the following and establish the relations to the classical photogrammetric parametrization where appropriate.

Camera Poses Consistent with the Relative Orientation The projection matrices for both images can easily be given in the local coordinate system chosen for the specific parametrizations, namely

$$ \mathsf{P}'_d = \mathsf{K}'[I_3|\mathbf{0}] \quad \text{or} \quad \mathsf{P}'_i = \mathsf{K}' R'_i [I_3|\mathbf{0}] \tag{13.47} $$

$$ \mathsf{P}''_d = \mathsf{K}'' R_d [I_3| - \boldsymbol{B}_d] \quad \text{or} \quad \mathsf{P}''_i = \mathsf{K}'' R''_i [I_3| - \boldsymbol{B}_i]\,, \tag{13.48} $$

where the indices d and i of the rotation matrices and the base vectors refer to the parametrization with *d*ependent and *i*ndependent images.

If we know the projection matrix

$$ \mathsf{P}' = \mathsf{K}' R' [I_3| - \boldsymbol{Z}'] \tag{13.49} $$

of the first camera referring to the scene coordinate system, we can derive the projection matrix $\mathsf{P}'' = \mathsf{K}''R''[I_3|-\boldsymbol{Z}'']$ for the second camera referring to the scene coordinate system if the relative orientation is given, provided we fix the scale of the base vector in the scene coordinate system and assume the second calibration matrix K'' to be known. Thus we want to determine R'' and $\boldsymbol{Z}''$ from the parameters of the relative orientation. In both cases we perform an adequate motion of the image pair. In the following we assume the base vectors $\boldsymbol{B}_d$ and $\boldsymbol{B}_i$ have the desired scene scale.

1. For the representation with dependent images we use the rotation matrix R_d and the base vector $\boldsymbol{B}_d$ from the relative orientation.
   The projection matrix P' referring to the scene coordinate system is related to the projection matrix P'_d referring to the first image by

$$\mathsf{P}' = \mathsf{K}'R'[I_3| - \boldsymbol{Z}'] = \underbrace{\mathsf{K}'[I_3|\mathbf{0}]}_{\mathsf{P}'_d} \begin{bmatrix} R'^{\mathsf{T}} & \boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}^{-1} = \mathsf{P}'_d \begin{bmatrix} R' & -R'\boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} . \tag{13.50}$$

   Applying the same motion, we obtain the projection matrix for the second image,

$$\mathsf{P}'' = \underbrace{\mathsf{K}''R_d[I_3| - \boldsymbol{B}_d]}_{\mathsf{P}''_d} \begin{bmatrix} R' & -R'\boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \mathsf{K}''\underbrace{R_dR'}_{R''}[I_3 \mid -\underbrace{(\boldsymbol{Z}' + R'^{\mathsf{T}}\boldsymbol{B}_d)}_{\boldsymbol{Z}''}] . \tag{13.51}$$

   Thus we have the relations

$$R'' = R_dR' \quad \text{and} \quad \boldsymbol{Z}'' = \boldsymbol{Z}' + R'^{\mathsf{T}}\boldsymbol{B}_d , \tag{13.52}$$

   which are plausible as $R_d = R''R'^{\mathsf{T}}$ and $\boldsymbol{B}_d = R'^{\mathsf{T}}(\boldsymbol{Z}'' - \boldsymbol{Z}')$.
2. For the representation with independent images we use the rotation matrices R'_i and R''_i and the base vector $\boldsymbol{B}_i$ from the relative orientation.
   Here the projection matrix P' referring to the scene coordinate system is related to the projection matrix P'_i in the system of the photogrammetric model by a motion where the rotation is the difference $R'^{\mathsf{T}}R'_i$ of the two rotations R' and R'_i. Thus we have

$$\mathsf{P}' = \mathsf{K}'R'[I_3| - \boldsymbol{Z}'] = \underbrace{\mathsf{K}'R'_i[I_3|\mathbf{0}]}_{\mathsf{P}'_i} \begin{bmatrix} R'^{\mathsf{T}}R'_i & \boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}^{-1} = \mathsf{P}'_i \begin{bmatrix} R_i'^{\mathsf{T}}R' & -R_i'^{\mathsf{T}}R'\boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} . \tag{13.53}$$

   Applying this motion to the second projection, we obtain

$$\mathsf{P}'' = \underbrace{\mathsf{K}''R''_i[I_3| - \boldsymbol{B}_i]}_{\mathsf{P}''_i} \begin{bmatrix} R_i'^{\mathsf{T}}R' & -R_i'^{\mathsf{T}}R'\boldsymbol{Z}' \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \mathsf{K}''R''_iR_i'^{\mathsf{T}}R'[I_3 \mid -(\boldsymbol{Z}' + R'^{\mathsf{T}}R'_i\boldsymbol{B}_i)] . \tag{13.54}$$

   Thus we have the relations

$$R'' = R''_iR_i'^{\mathsf{T}}R' \quad \text{and} \quad \boldsymbol{Z}'' = \boldsymbol{Z}' + R'^{\mathsf{T}}R'_i\boldsymbol{B}_i , \tag{13.55}$$

   which are plausible as the rotation differences in the scene system and the system of the photogrammetric model are the same, thus $R''_iR_i'^{\mathsf{T}} = R''R'^{\mathsf{T}}$, and the base vector in the photogrammetric system $\boldsymbol{B}_i = R_i'^{\mathsf{T}}R'(\boldsymbol{Z}'' - \boldsymbol{Z}')$ is obtained from the base vector $\boldsymbol{B} = \boldsymbol{Z}'' - \boldsymbol{Z}'$ in the scene system by the rotation $R_i'^{\mathsf{T}}R'$.

13.2.4 The Normal Case of the Image Pair

Similarly to a single image, we have a *normal case* for the image pair: Both cameras are ideal and identical, the viewing directions are parallel and orthogonal to the base vector, and the x'- and x''-axes are parallel to the basis.

The normal case is close to the geometric configuration of the human stereo vision system. In photogrammetry, the normal case of the image pair is used for analysing the geometric configuration based on the theoretical accuracy. It can often be used as an approximation for the configuration of a pair of aerial vertical view images.

We first give a geometric derivation of the coplanarity constraint before we derive it algebraically (Fig. 13.5). If the image coordinates of the points $\mathcal{x}'$ and $\mathcal{x}''$ are error-free,

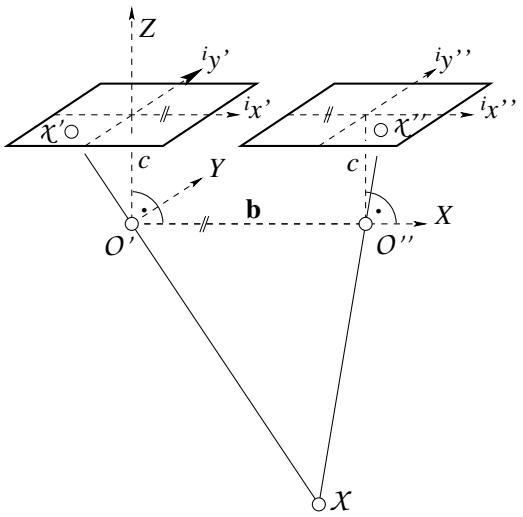


Fig. 13.5 Geometry of the normal case of the image pair. The image points $\mathcal{x}'$ and $\mathcal{x}''$, the projection centres O' and O'' and the scene point $\mathcal{X}$ are coplanar. The ${}^{i}y$-coordinates of the image points should be the same

the corresponding rays intersect in the object point $\mathcal{X}$. The constraint for the rays to intersect depends only on the ${}^{i}y'$ and ${}^{i}y''$ coordinates since changes in the ${}^{i}x'$ or the ${}^{i}x''$ coordinates lead to a change of the distance of the scene point from the base line, while the rays still intersect. The difference

$$p_y = {}^{i}y'' - {}^{i}y' \tag{13.56}$$

is called the *y-parallax*. Obviously the y-parallax needs to be zero, meaning that the ${}^{i}y$ coordinates of corresponding points have to be identical. This leads to the coplanarity constraint ,

y-parallax

$$p_y = {}^{i}y'' - {}^{i}y' = 0\,. \tag{13.57}$$

When referring to the human visual system, the difference p_y of the y coordinates is often called *vertical parallax*.

The algebraic derivation of the coplanarity constraint starts from the special configuration shown in Fig. 13.5,

$$R' = R'' = I_3 \qquad \mathbf{b} = \begin{bmatrix} B_X \\ 0 \\ 0 \end{bmatrix} \qquad \mathsf{K} \doteq \mathsf{K}' = \mathsf{K}'' = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{13.58}$$

i.e., $c = c' = c''$. With

$$\mathsf{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -B_X \\ 0 & B_X & 0 \end{bmatrix}, \tag{13.59}$$

we obtain the following coplanarity constraint:

$$[{}^i x' \; {}^i y' \; c] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -B_X \\ 0 & B_X & 0 \end{bmatrix} \begin{bmatrix} {}^i x' \\ {}^i y'' \\ c \end{bmatrix} = c\, B_X \, ({}^i y'' - {}^i y') = 0\,. \tag{13.60}$$

The residual of the constraint, if it is not fulfilled, can therefore be used to derive the y-parallax by dividing it by c and B_X. Since neither of them can be zero we finally obtain the coplanarity constraint for the normal case (13.57).

The different versions of the essential matrix are collected in Table 13.2. In addition the table contains the two special cases, namely the normal case and relative orientation with two normalized cameras. In both cases, the number of parameters is less than five, so that they are real specializations. The normal case is relevant for automatic image analysis since all matching procedures can take advantage of the much more simplified equations. The relative orientation of images of normalized cameras occurs when the rotation matrices of two calibrated cameras are measured or determined, e.g., from vanishing points.

Table 13.2 Parametrizations of the coplanarity constraint with the fundamental matrix F for straight line-preserving uncalibrated cameras and the essential matrix E for calibrated cameras. Lines 2 and 3: rotation matrix R and angles ω, ϕ and κ refer to the second camera; $R \doteq R''$

camera/parametrization a priori constraints	matrix F or E	Eqn.	free parameters (#) add. constraints
uncalibrated perspective camera	$\mathsf{F} = \mathsf{K}'^{-\mathsf{T}} \mathsf{R}' \mathsf{S}_\mathrm{b} \mathsf{R}''^{\mathsf{T}} \mathsf{K}''^{-1}$	(13.8)	F (9) $\|\|\mathsf{F}\|\| = 1$, $\|\mathsf{F}\| = 0$
spherical camera, calibrated perspective camera			
indep. images general $R' = I_3$	$\mathsf{E} = \mathsf{S}_\mathrm{b} R^{\mathsf{T}}$	(13.37)	$\mathbf{b}, \omega, \phi, \kappa$, (6) $\|\mathbf{b}\|$=1
indep. images special $R' = I_3$, B_X=const.	$\mathsf{E} = \mathsf{S}_\mathrm{b} R^{\mathsf{T}}$	(13.37)	$B_Y, B_Z, \omega, \phi, \kappa$ (5)
dependent images $\mathbf{b} = (B_X, 0, 0)^{\mathsf{T}}$, $\omega' = -\omega'' = -\frac{1}{2}\Delta\omega$	$\mathsf{E} = R' \mathsf{S}_\mathrm{b} R''^{\mathsf{T}}$	(13.59)	$\Delta\omega, \phi', \kappa', \phi'', \kappa''$ (5)
normalized cameras $R' = R'' = I_3$ $\mathsf{K}' = \mathsf{K}'' = I_3$	$\mathsf{E} = \mathsf{S}_\mathrm{b}$	(13.46)	$\mathbf{b}$ (3) $\|\mathbf{b}\| = 1$
normal case $R' = R'' = I_3$ $\mathsf{K}' = \mathsf{K}'' = \mathrm{Diag}([c, c, 1])$ $\mathbf{b} = (B_X, 0, 0)^{\mathsf{T}}$	$\mathsf{E} = \mathsf{S}_\mathrm{b}$	(13.59)	–

13.2.5 Epipolar Geometry

We now address the problem of predicting the position of a point $\mathbf{x}''$ in the second image if the point $\mathbf{x}'$ is given in the first image and the relative orientation is known. We will find that the point $\mathbf{x}''$ lies on a straight line for a perspective camera and on a circle for

a spherical camera. The underlying geometry is the *epipolar geometry* of the image pair. This knowledge can be used (1) for guiding an operator when measuring a point which has already been measured in one other image, and (2) for reducing the search space for finding corresponding image points in automatic image matching.

We define the following entities (Fig. 13.6), which can be used for the perspective and the spherical camera models:

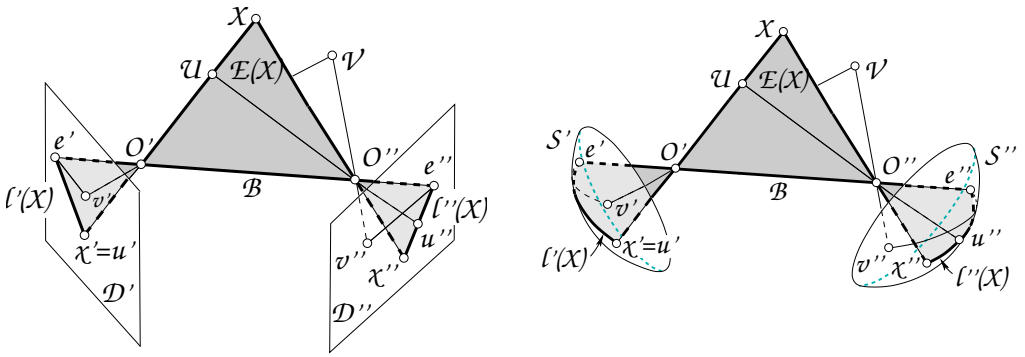


Fig. 13.6 Elements of the epipolar geometry. **Left:** perspective images. **Right:** spherical images. Both in taking position (see Fig. 12.8, p. 469). Epipolar plane $\mathcal{E}(\mathcal{X})$ through $\mathcal{O}'\mathcal{O}''\mathcal{X}$, with the epipoles e' and e'' as images of the other projection centre; the straight or circular epipolar lines $l'(\mathcal{X})$ and $l''(\mathcal{X})$ which are the intersections of the epipolar plane $\mathcal{E}(\mathcal{X})$ and the image planes $\mathcal{D}'$ and $\mathcal{D}''$ or image spheres $\mathcal{S}'$ and $\mathcal{S}''$. Additional scene points, e.g., point $\mathcal{V}$, induce additional epipolar planes, building a pencil of planes with the base line $\mathcal{B} = \mathcal{O}' \wedge \mathcal{O}''$ as axis. Therefore, the epipolar lines also form a pencil of lines with the epipoles as carrier. Observe, $\mathcal{X}'$ does not allow the inference of the position of $\mathcal{X}$ on the projecting line. Point $\mathcal{U}$ mapped to $\mathcal{X}'$, however, has a different image u''. Both the point $\mathcal{X}$ and the point $\mathcal{X}'$ induce the epipolar line $l''(\mathcal{X}) = l''(\mathcal{X}') = e'' \wedge \mathcal{X}''$ in the other image

1. The *epipolar axis*
$$\mathcal{B} = \mathcal{O}' \wedge \mathcal{O}'' \tag{13.61}$$
is identical to the line through the two projection centres and contains the base line.[5]
2. The *epipolar plane*
$$\mathcal{E}(\mathcal{X}) = \mathcal{O}' \wedge \mathcal{O}'' \wedge \mathcal{X} \tag{13.62}$$
depends on the fixed projection centres $\mathcal{O}'$ and $\mathcal{O}''$ and on the object point $\mathcal{X}$. The epipolar planes build a pencil of planes with the epipolar axis as the common line. The epipolar planes therefore pass through the epipoles.
3. The *epipoles*
$$e' = \mathcal{P}'(\mathcal{O}'') \qquad e'' = \mathcal{P}''(\mathcal{O}') \tag{13.63}$$
are the images of the other projection centres using the projections $\mathcal{P}'$ and $\mathcal{P}''$. The vectors $\mathcal{O}'e'$ and $\mathcal{O}''e''$ thus provide the direction to the other projection centres.
For perspective cameras the epipoles are the intersection points of the epipolar axis and the image planes $\mathcal{D}'$ and $\mathcal{D}''$
$$e' = (\mathcal{O}' \wedge \mathcal{O}'') \cap \mathcal{D}' \qquad e'' = (\mathcal{O}' \wedge \mathcal{O}'') \cap \mathcal{D}'' \,. \tag{13.64}$$
For spherical cameras the epipoles are the intersection of the basis with the two image spheres $\mathcal{S}'$ and $\mathcal{S}''$.
4. The *epipolar lines*
$$l'(\mathcal{X}) = \mathcal{P}'(\mathcal{O}'' \wedge \mathcal{X}) \qquad l''(\mathcal{X}) = \mathcal{P}''(\mathcal{O}' \wedge \mathcal{X}) \tag{13.65}$$

[5] The base line $\mathcal{B}$ is not to be confused with the plane represented by the second row of a projection matrix (see Sect. 12.1.3.8, p. 473).

are the images of the rays $\mathcal{O}' \wedge \mathcal{X}$ and $\mathcal{O}'' \wedge \mathcal{X}$ in the other image, respectively.
For perspective and for spherical cameras they are the intersections of the epipolar plane with the image planes and spheres. They depend on the point $\mathcal{X}$, e.g.,

$$\mathcal{l}'(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \mathcal{D}' \qquad \mathcal{l}''(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \mathcal{D}'' \,. \tag{13.66}$$

The projection centres $\mathcal{O}'$ and $\mathcal{O}''$, the object point $\mathcal{X}$, the epipolar lines $\mathcal{l}'(\mathcal{X})$ and $\mathcal{l}''(\mathcal{X})$, and the two image points $\mathcal{x}'$ and $\mathcal{x}''$ lie in the same epipolar plane.

In this case the prediction of $\mathcal{x}''$ can be solved easily: The epipolar plane $\mathcal{E}(\mathcal{x}')$ is spanned by the three given points, the two projection centres $\mathcal{O}'$ and $\mathcal{O}''$, and the image point $\mathcal{x}'$; its intersection with the other image plane yields the epipolar line $\mathcal{l}''(\mathcal{x}')$ on which the predicted point $\mathcal{x}''$ must lie.

These entities can easily be determined algebraically using the projection matrices or the fundamental matrix:

1. The epipolar axis has the direction of

$$\mathbf{b} = \boldsymbol{B} = \boldsymbol{Z}'' - \boldsymbol{Z}' \,. \tag{13.67}$$

   We interpret it either as the vector $\boldsymbol{B}$ between the two projection centres or as its direction $\mathbf{b}$.
2. The epipolar lines are the projections of the projection lines $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ into the other images,

$$\mathbf{l}'(\mathbf{x}'') = \mathsf{Q}'\overline{\mathsf{Q}}''^{\mathsf{T}}\mathbf{x}'' \,, \qquad \mathbf{l}''(\mathbf{x}') = \mathsf{Q}''\overline{\mathsf{Q}}'^{\mathsf{T}}\mathbf{x}' \,, \tag{13.68}$$

   using the equation $\mathbf{L}_{x'} = \overline{\mathsf{Q}}'^{\mathsf{T}}\mathbf{x}'$ for the projection line (12.85), p. 483 with the projection $\mathbf{l}'' = \mathsf{Q}''\mathbf{L}$ for object lines (12.72), p. 480. The epipolar lines $\mathcal{l}'$ and $\mathcal{l}''$ are characterized by the constraint that corresponding points $\mathcal{x}'$ and $\mathcal{x}''$ have to fulfil the coplanarity constraint $\mathbf{x}'^{\mathsf{T}}\mathsf{F}\mathbf{x}'' = 0$ and the points have to lie on the lines, thus $\mathcal{x}' \in \mathcal{l}'$ and $\mathcal{x}'' \in \mathcal{l}''$. This is valid if

epipolar lines $\mathcal{l}'$ and $\mathcal{l}''$

$$\mathbf{l}'(\mathbf{x}') = \mathsf{F}\mathbf{x}'' \,, \qquad \mathbf{l}''(\mathbf{x}') = \mathsf{F}^{\mathsf{T}}\mathbf{x}' \,, \tag{13.69}$$

   since then $\mathbf{x}'^{\mathsf{T}}\mathbf{l}' = 0$ and $\mathbf{x}''^{\mathsf{T}}\mathbf{l}'' = 0$. The second equation is obviously the prediction line for $\mathcal{x}''$ in the second image if $\mathcal{x}'$ is given in the first image.
   Equation (13.69) is a remarkable transformation: points of one image are transformed into lines in the other image. This is one example of a broader group of dualizing transformations in projective geometry, known as projective correlation (see Semple and Kneebone (1998); Tschupik and Hohenberg (1972); Niini (2000) and Sect. 6.6, p. 282). Since the matrix F is singular, the mapping is also called a *singular correlation*. The notion has nothing in common with a statistical correlation, however. The relation

fundamental matrix as singular correlation

$$\mathsf{F} = \mathsf{Q}'\overline{\mathsf{Q}''}^{\mathsf{T}} \tag{13.70}$$

   can be used to prove (13.15) in a different way.
   Equation (13.69) yields *oriented* epipolar lines if the fundamental matrix is determined from proper projection matrices from one of the equations in Sect. 13.2.2.2, p. 553. If the fundamental matrix is estimated from corresponding points, the sign needs to be adapted. A similar line of thought is valid also for the essential matrix. We discuss this in the context of deriving the base vector and the rotation from an estimated essential matrix (see Sect. 13.3.3, p. 581).
   If the projection is *not distortion-free*, thus not straight line-preserving, we can first determine the projection line and project it into the other image by sampling the space line by a few points, yielding a *polygonal approximation of the curved epipolar line.*

3. The epipoles can be determined either directly from

$$\mathbf{e}' = \mathsf{P}'\mathbf{Z}'' \qquad \text{and} \qquad \mathbf{e}'' = \mathsf{P}''\mathbf{Z}' \tag{13.71}$$

or using the camera planes in the projection matrices, e.g., $\mathsf{P}' = [\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1]^\mathsf{T}$ from

$$\mathbf{e}' = \begin{bmatrix} |\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{A}_1| \\ |\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{B}_1| \\ |\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{C}_1| \end{bmatrix} \quad \mathbf{e}'' = \begin{bmatrix} |\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{A}_2| \\ |\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{B}_2| \\ |\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{C}_2| \end{bmatrix} . \tag{13.72}$$

The second expressions result from the fact that the projection centres are the intersections of the camera planes, thus $\mathbf{Z}'' = -\mathbf{A}_2 \cap \mathbf{B}_2 \cap \mathbf{C}_2$ and $\mathbf{A}_1^\mathsf{T}\mathbf{Z}'' = -|\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2| = |\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{A}_1|$, using the dual of (5.117), p. 227. If the projection matrices, P' and P'', are proper, both equations (13.71) and (13.72) yield oriented vectors for the epipoles. The epipole of an image, however, is incident with all its epipolar lines. Therefore, the epipoles e' and e'' are the left and right eigenvectors of F, respectively,

epipoles e' and e''

$$\mathbf{e}'^\mathsf{T}\mathsf{F} = \mathbf{0}^\mathsf{T} \quad \text{and} \quad \mathsf{F}\mathbf{e}'' = \mathbf{0}\,, \tag{13.73}$$

since for all $\mathbf{l}'$ we have $\mathbf{e}'^\mathsf{T}\mathbf{l}' = 0$ and for all $\mathbf{l}''$ we have $\mathbf{e}''^\mathsf{T}\mathbf{l}'' = 0$. Observe, calculating the epipoles from (13.73) does *not* necessarily yield oriented vectors $\mathbf{e}'$ and $\mathbf{e}''$.

epipolar geometry for calibrated cameras

The relations for *calibrated cameras* are similar, namely replacing the entities for perspective cameras by those for spherical cameras, i.e., the homogeneous vectors of the image points are replaced by the ray directions

$$\mathbf{x}' \to {}^c\mathbf{x}' = -\text{sign}(c')(\mathsf{K}')^{-1}\mathbf{x}'\,, \qquad \mathbf{e}' \to {}^c\mathbf{e}' = -\text{sign}(c')(\mathsf{K}')^{-1}\mathbf{e}'\,. \tag{13.74}$$

Observe, in each camera the vectors ${}^c\mathbf{e}'$ and ${}^c\mathbf{e}''$ provide the directions to the other projection centre.

Similarly, we need to replace the expressions for image lines, due to $((\mathsf{K}')^{-1})^\flat = \mathsf{K}'^\mathsf{T}/|\mathsf{K}'|$ and assuming $|\mathsf{K}'| > 0$,

$$\mathbf{l}' \to {}^c\mathbf{l}' = \mathsf{K}'^\mathsf{T}\,\mathbf{l}'\,, \tag{13.75}$$

and for projection matrices,

$$\mathsf{P}' \to {}^c\mathsf{P}' = R'[I_3| - \boldsymbol{Z}']\,, \quad \mathsf{Q}' \to {}^c\mathsf{Q}' = R'[-S(\boldsymbol{Z}') \mid I_3]\,; \tag{13.76}$$

see (12.17), p. 468 and (12.76), p. 481. For example, we obtain the expression for the essential matrix from (13.70), p. 564,

$$\mathsf{E} = {}^c\mathsf{Q}'\,\overline{{}^c\mathsf{Q}'}^\mathsf{T} = R'S(\boldsymbol{Z}'' - \boldsymbol{Z}')R'^\mathsf{T}\,, \tag{13.77}$$

replacing the fundamental matrix.

Remark: If the signs are not of interest, the factor $1/|\mathsf{K}'|$ can be omitted in (13.75) and (13.76), see Sect. 6.2.4, p. 258. ◇

13.2.6 Generating Normalized Stereo Pairs

Image pairs can be viewed stereoscopically if certain preconditions are fulfilled; one of them is that the y-parallaxes are zero for all points. In addition, exploiting the epipolar geometry of an image pair is computationally more efficient if the geometry follows the normal case, see (Sect. 13.2.4). This is usually approximately fulfilled by aerial images for a larger image area but never for arbitrary image pairs.

However, one may transform a given image pair into the normal case. A general solution uses as new epipolar lines the intersection of a cylinder (where the basis is its axis) with a pencil of planes around the basis and exploits the orientation of corresponding epipolar lines (Pollefeys et al., 1999). If the two viewing directions are not too close to the basis, an alternative is to keep the two bundles of rays and intersect them with a common plane parallel to the base vector such that the normal of the plane is as close as possible to the two viewing directions.

normalized image pair

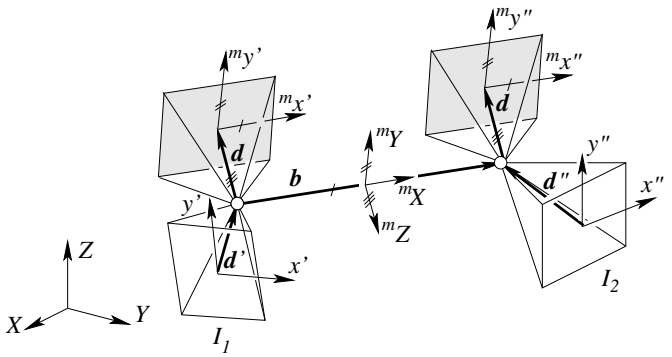


Fig. 13.7 Normalizing an image pair I_1, I_2: the normalized stereo images, shaded gray, are chosen such that corresponding image points have no y-parallax. Thus the normalized images have a common calibration and rotation matrix, the common viewing direction $\boldsymbol{d}$, which is the average of the two viewing directions $\boldsymbol{d}_i$, rotated such that it is perpendicular to the basis. The x axes of the normalized images are parallel to the base vector $\boldsymbol{b}$. The common principal distance c is chosen to be negative, leading to normalized stereo images in viewing position

Now, the two bundles are kept and just the two image planes are replaced. The relations between the given image coordinates $\mathbf{x}'$ and $\mathbf{x}''$ and the transformed image coordinates ${}^m\mathbf{x}'$ and ${}^m\mathbf{x}''$ are two homographies,

$$\mathbf{x}' = \mathsf{H}'_m\, {}^m\mathbf{x}' \qquad \mathbf{x}'' = \mathsf{H}''_m\, {}^m\mathbf{x}'' \,. \tag{13.78}$$

They may be used to digitally transform the two given images by indirect resampling. In the simplest case of ideal cameras we just have to rotate the image planes around the two projection centres.

The problem can be formalized as follows: Given are the two projection matrices P' and P'', with the possibly different calibration matrices K' and K'', the rotation matrices R' and R'', and the two projection centres $\boldsymbol{Z}'$ and $\boldsymbol{Z}''$, leading to the projections

$$\mathbf{x}' = \mathsf{P}'\mathbf{X} = \mathsf{K}'R'[I| - \boldsymbol{Z}']\mathbf{X} \qquad \mathbf{x}'' = \mathsf{P}''\mathbf{X} = \mathsf{K}''R''[I| - \boldsymbol{Z}'']\mathbf{X} \,. \tag{13.79}$$

The goal is to find the two homographies H'_m and H''_m.

The idea is to achieve two new projections,

$${}^m\mathbf{x}' = {}^m\mathsf{P}'\mathbf{X} = \mathsf{K}R[I| - \boldsymbol{Z}']\mathbf{X} \qquad {}^m\mathbf{x}'' = {}^m\mathsf{P}''\mathbf{X} = \mathsf{K}R[I| - \boldsymbol{Z}'']\mathbf{X} \,, \tag{13.80}$$

with the common calibration matrix K and common rotation matrix R. The common calibration and rotation matrices guarantee the image content is different only with respect to x-parallaxes; the y-parallaxes are zero.

From (13.79) and (13.80) we obtain the homography matrices

$$\mathsf{H}'_m = \mathsf{K}'R'R^{\mathsf{T}}\mathsf{K}^{-1} \qquad \mathsf{H}''_m = \mathsf{K}''R''R^{\mathsf{T}}\mathsf{K}^{-1} \tag{13.81}$$

or the inverse homographies

$${}^m\mathbf{x}' = \mathsf{K}RR'^{\mathsf{T}}\mathsf{K}'^{-1}\mathbf{x}' \qquad {}^m\mathbf{x}'' = \mathsf{K}RR''^{\mathsf{T}}\mathsf{K}''^{-1}\mathbf{x}'' \,, \tag{13.82}$$

useful for digital image rectification. The homography matrices H'_m and H''_m are conjugate rotations, as each is similar to a rotation matrix (see Sect. 6.5.2, p. 281).

The common calibration matrix can be chosen to be $\mathsf{K} = \mathrm{Diag}([c, c, 1])$. The sign of c should be chosen negatively in order to obtain the two transformed images in viewing position.

The rotation matrix R is chosen such that the two following conditions are fulfilled:

1. The ${}^m x'$ and the ${}^m x''$ axes of the transformed image coordinates are parallel to the base vector. This is identical to requiring the X axes of the two camera coordinate systems to be parallel to the base vector. Therefore the first row $\boldsymbol{r}_1^\mathsf{T}$ of the rotation matrix is the normalized base vector $\boldsymbol{r}_1 = \mathsf{N}(\boldsymbol{B})$.
2. The viewing directions (the Z-axes of the camera coordinate systems) are orthogonal to the base vector and are as close as possible to the original viewing directions. One choice is to use the average viewing direction

$$\mathbf{d}^* = \mathsf{N}\left(\frac{\mathbf{d}'}{|\mathbf{d}'|} + \frac{\mathbf{d}''}{|\mathbf{d}''|}\right), \tag{13.83}$$

   with (12.47), p. 475

$$\mathbf{d}' = -[p'_{31}, p'_{32}, p'_{33}]^\mathsf{T} \, |\mathsf{A}'| \qquad \mathbf{d}'' = -[p''_{31}, p''_{32}, p''_{33}]^\mathsf{T} \, |\mathsf{A}''| \,. \tag{13.84}$$

   and require the common viewing direction $\boldsymbol{d}$ to lie in the plane spanned by $\mathbf{b}$ and $\mathbf{d}^*$, cf. Fig. 13.7.
3. the ${}^m y'$ and the ${}^m y''$ axes of the transformed image coordinates are perpendicular to the base vector, thus the y-axes of the two camera coordinate systems are completing the Cartesian coordinate systems.

The rotation matrix therefore is

$$R = [\mathsf{N}(\mathbf{b}),\ \mathsf{N}(\mathbf{b} \times \mathbf{d}^*),\ \mathsf{N}(\mathbf{b} \times (\mathbf{b} \times \mathbf{d}^*))]^\mathsf{T} \,. \tag{13.85}$$

The common viewing direction then is $-\mathsf{N}(\mathbf{b} \times (\mathbf{b} \times \mathbf{d}^*))$.

13.2.7 Homography and Homology Induced by a Plane in Object Space

As a straight line-preserving mapping between two planes is a homography and the concatenation of two homographies is again a homography, the mapping between two images of the same plane in object space is a homography. The eight parameters of this homography can be described as a function of the five parameters of the relative orientation of two images of calibrated cameras and the three parameters of the plane in object space.

We start from the two projection matrices

$${}^c\mathsf{P}' := [I_3|0] = \mathsf{K}'^{-1} \, \mathsf{P}' \,, \qquad {}^c\mathsf{P}'' := R[I| - \boldsymbol{B}] = \mathsf{K}''^{-1} \, \mathsf{P}'' \tag{13.86}$$

and a plane $\mathbf{A} = [\boldsymbol{A}_h^\mathsf{T}, A_0]^\mathsf{T}$. Then the mapping from the first to the second image is given by the homography, see Fig. 13.8,

plane induced homography

$${}^c\mathbf{x}' = \mathsf{H} \, {}^c\mathbf{x}'' \quad \text{with} \quad \mathsf{H} = R + \frac{\boldsymbol{T}\boldsymbol{A}_h^\mathsf{T}}{A_0} \tag{13.87}$$

and

$$\boldsymbol{T} = R\boldsymbol{B} \qquad {}^c\mathbf{x}' = \mathsf{K}'^{-1} \, \mathbf{x}' \qquad {}^c\mathbf{x}'' = \mathsf{K}''^{-1} \, \mathbf{x}'' \,. \tag{13.88}$$

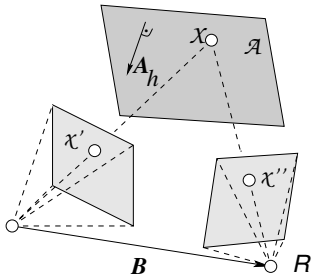


Fig. 13.8 Plane induced homography H between two images, which can be used for relative orientation with ≥ 4 points (see Sect. 13.3.2.5, p. 577). The homography ${}^{c}\mathbf{x}' = \mathsf{H}\,{}^{c}\mathbf{x}''$ may be determined from ≥ 4 corresponding points $(\mathbf{x}'_i, \mathbf{x}''_i)$. It depends on the normal $\boldsymbol{A}_h$ of plane $\mathbf{A}$, the rotation matrix R of the second image with respect to the coordinate system of the first, and the base vector $\boldsymbol{B}$ via $\boldsymbol{T} = R\boldsymbol{B}$

Proof: The 3D coordinates of the 3D point $\mathbf{X} = [\boldsymbol{X}^\mathsf{T}, 1]^\mathsf{T}$ in the two camera coordinate systems are

$$ {}^{1}\boldsymbol{X} = \boldsymbol{X} \qquad \text{and} \qquad {}^{2}\boldsymbol{X} = R(\boldsymbol{X} - \boldsymbol{B}) = R\boldsymbol{X} - \boldsymbol{T}\,. \tag{13.89} $$

If we use the relation $\mathbf{A}^\mathsf{T}\mathbf{X} = \boldsymbol{A}_h^\mathsf{T}\boldsymbol{X} + A_0 = 0$, thus $-\boldsymbol{A}_h^\mathsf{T}\boldsymbol{X}/A_0 = 1$ for all points $\boldsymbol{X}$ on the plane, we obtain

$$ {}^{2}\boldsymbol{X} = R\boldsymbol{X} - \frac{-\boldsymbol{T}\boldsymbol{A}_h^\mathsf{T}\boldsymbol{X}}{A_0} = \left(R + \frac{\boldsymbol{T}\boldsymbol{A}_h^\mathsf{T}}{A_0}\right) {}^{1}\boldsymbol{X}\,, $$

from which (13.87) follows due to ${}^{c}\mathbf{x}' = {}^{1}\boldsymbol{X}$ and ${}^{c}\mathbf{x}'' = {}^{2}\boldsymbol{X}$. ◇

plan induced homology for pure translation

If the rotation matrix is a unit matrix, the two images are related by a translation only, and the mapping between the images is a *homology*. It is given by

$$ \mathsf{H} = I_3 + \frac{\boldsymbol{T}\boldsymbol{A}_h^\mathsf{T}}{A_0} \tag{13.90} $$

and only depends on five parameters, namely the direction vectors $\boldsymbol{T}$ and $\boldsymbol{A}_h$ and the parameter A_0. Then two singular values of the matrix H are identical.

13.3 Relative Orientation of the Image Pair

13.3.1 Uncertainty of Corresponding Image Points 569
13.3.2 Direct Solutions for the Relative Orientation 570
13.3.3 Orientation Parameters from a Given Essential Matrix 581
13.3.4 Iterative Solution for Uncalibrated Cameras 583
13.3.5 Iterative Solution for Calibrated Cameras 585
13.3.6 Orientation and the Normal Case 588
13.3.7 Projection Matrices from Relative Orientation 594

In the following section we discuss the determination of the relative orientation of two images from image point measurements, without control points or lines. This problem occurs

1. when observing a possibly moving object simultaneously with two cameras,
2. when observing a static object at two time instances with a moving camera (called structure from motion in the computer vision literature), and
3. when observing a moving object from a static camera, neglecting the stable background.

Only corresponding points can be used, as corresponding straight lines do not give a constraint on the relative orientation. The determination of the relative orientation of two images is based here on the coplanarity equation. The procedure is nonlinear in

the unknown orientation parameters, so we discuss direct solution procedures useful for obtaining approximate values for a subsequent statistically optimal solution.

The results of relative orientation are ambiguous or indefinite for critical configurations, and unstable when close to critical configurations. Therefore, they require some attention.

Once the relative orientation is determined we may derive pairs of projection matrices consistent with the derived relative orientation. As no 3D information is used, these projection matrices are not unique. They are, however, sufficient to determine the 3D coordinates of corresponding points and straight lines in the local coordinate system of the images. Together with the cameras we obtain a photogrammetric model, related to the 3D scene by a similarity or a projectivity.

We also discuss the precision and reliability of relative orientation and especially the precision of 3D points derived from relatively oriented images.

13.3.1 Uncertainty of Corresponding Image Points

In all cases we assume N pairs of corresponding image points $\boldsymbol{x}'_n$ and $\boldsymbol{x}''_n$ to be measured, manually or automatically. For the statistically optimal solution we also assume the precision of these measurements to be approximately known in the form of a covariance matrix $\mathbb{D}([\boldsymbol{x}_n{}'^{\mathsf{T}}, \boldsymbol{x}_n{}''^{\mathsf{T}}]^{\mathsf{T}})$, allowing for correlated measurements. Here we anticipate results discussed in the second volume, specifically on image matching.

The stochastic properties of the coordinates of image points depend on the observational procedure. There are two alternatives:

- The points are detected and located independently. Then their joint 4×4 covariance matrix is a block diagonal matrix,

$$\mathbb{D}\left(\begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{x}''\end{bmatrix}\right) = \begin{bmatrix}\Sigma_{x'x'} & \boldsymbol{0}\\ \boldsymbol{0} & \Sigma_{x''x''}\end{bmatrix}, \tag{13.91}$$

  with the covariance matrices for the individual points following Sect. 12.2.1.1, p. 491.
- Alternatively, first the point in one image is detected and located with an uncertainty represented by $\Sigma_{x'x'}$. Then the coordinate difference, i.e., the parallax $\boldsymbol{p} = \boldsymbol{x}'' - \boldsymbol{x}'$, is measured using some image processing tool, e.g., by image correlation. The measured parallax $\boldsymbol{p}$ is stochastically independent of the location of the first point. Therefore, the joint covariance matrix of the four image coordinates is no longer block diagonal but the coordinates $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are correlated, Exercise 13.21

$$\mathbb{D}\left(\begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{x}''\end{bmatrix}\right) = \begin{bmatrix}\Sigma_{x'x'} & \Sigma_{x'x'}\\ \Sigma_{x'x'} & \Sigma_{x'x'} + \Sigma_{pp}\end{bmatrix}. \tag{13.92}$$

  The induced correlations depend on the relative accuracy of the detected point and the measured parallax. The smaller the standard deviation, the larger the correlation. The *accuracy of the parallax measurement* depends highly on the local image intensity function. High gradients will lead to better localization. Here the inverse of the structure tensor scaled with the image noise variance is a good approximation for the covariance matrix of the measured parallax (see Förstner (1984); Haralick and Shapiro (1992, Sec. 9)):

covariance matrix of parallaxes

$$\Sigma_{pp} = 2\sigma_n^2 \begin{bmatrix}\sum_k f_{x_k}^2 & \sum_k f_{x_k} f_{y_k}\\ \sum_k f_{y_k} f_{x_k} & \sum_k f_{y_k}^2\end{bmatrix}^{-1}, \tag{13.93}$$

  where σ_n is the standard deviation of the image noise and f_{x_k} and f_{y_k} are estimates for the first derivatives of the image function at the kth pixel in the window used for the parallax measurement.

  For colour images, the weight matrix $W_{pp} = \Sigma_{pp}^{-1}$ is the sum of the weight matrices of the colour channels $\mathcal{C}$; for a three-channel RGB image we therefore have the covariance

covariance matrix of of parallaxes in colour images

matrix

$$\Sigma_{pp} = \left(\sum_{c=1}^{3} W_{pp}^{(c)} \right)^{-1}, \tag{13.94}$$

where (13.93) is used for each of the three channels with their individual noise variance $\sigma_n^{(c)2}$ and their individual gradients $[f_{x_k}, f_{y_k}]^{(c)}$.

The result can be generalized to sets of points which are obtained from a tracking procedure in video sequences.

13.3.2 Direct Solutions for the Relative Orientation

This section collects direct, closed form solutions for the fundamental matrix F and the essential matrix E under various conditions. The direct solutions are used

1. to obtain approximate values for a subsequent optimal estimation and
2. within a RANSAC-type procedure (see Sect. 4.7.7, p. 153) for finding good observations in the presence of outliers.

We provide solutions for the following cases:

1. Determination of F or E with $I = 8$ or more points (see p. 570). This is the most simple solution for two uncalibrated cameras which can also be used in the overconstrained case. It does not work for planar scenes and is unstable for nearly planar scenes.
2. Determination of F with $I = 7$ points (see p. 571). This is the minimal solution for two uncalibrated cameras, obtainable with little additional effort. It can also be used for the determination of E (see p. 575)
3. Determination of E with $I = 5$ points (see p. 575). This is the minimal solution for two calibrated cameras. For general camera poses the object may be planar.
4. Determination of S_B for $\mathsf{E} = \mathsf{S}_B R^\mathsf{T}$ with $I = 2$ points in the case of calibrated cameras with given rotation matrix R (see p. 578). This is useful for close-by images in a video stream, where the rotation differences are small.
5. If a single calibrated image shows an object and its mirror image, determination of the relative pose of the two images from $I = 2$ corresponding points and reconstruction of the visible parts of the object (see p. 579).

Useful particular solutions are included as well , such as the four-point solution for coplanar scene points. Finding the relative orientation based on corresponding conics is discussed by Kahl and Heyden (1998). Using correspondences of curved lines can be found in Forkert (1994) and Nurutdinova and Fitzgibbon (2015).

13.3.2.1 Estimation of the Fundamental Matrix F from $I \geq 8$ Points

The coplanarity constraint for uncalibrated, straight line-preserving cameras is

$$\mathbf{x}'^\mathsf{T}_i \mathsf{F} \mathbf{x}''_i = w_i \stackrel{!}{=} 0 \qquad n = 1, ..., I\,, \tag{13.95}$$

see (13.9), p. 553. Remember, the number of free parameters of the 3×3 fundamental matrix F is seven, as it is homogeneous and singular. The constraints are linear in the elements of F. For this we use the vector of the fundamental matrix

$$\mathbf{f} = \mathrm{vec}\mathsf{F} = [F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23}, F_{33}]^\mathsf{T} \tag{13.96}$$

and the coefficients

$$\mathbf{a}_i^\mathsf{T} = \mathbf{x}_i''^\mathsf{T} \otimes \mathbf{x}_i'^\mathsf{T} = [x''x', x''y', x'', y''x', y''y', y'', x', y', 1]_i\,, \tag{13.97}$$

so that we can write the constraints as

$$\mathbf{a}_i^\mathsf{T}\mathbf{f} = w_i \stackrel{!}{=} 0 \qquad i = 1, ..., I\,. \tag{13.98}$$

We can directly estimate F. Similarly to algebraically estimating the parameters of the projection matrix (Sect. 12.2.2.1), minimizing $\boldsymbol{w}^\mathsf{T}\boldsymbol{w}$ we obtain as a solution the right singular vector of the matrix with their singular value decomposition,

$$\underset{I\times 9}{\mathsf{A}} = [\mathbf{a}_i^\mathsf{T}] = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\mathsf{T}\,, \tag{13.99}$$

namely from

$$\widehat{\mathbf{f}}^{(1)} = \boldsymbol{v}_9 \quad \text{with} \quad \boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_9]\,, \tag{13.100}$$

assuming decreasing singular values. The solution is unique if the rank of the matrix is 8 or larger. Thus we need at least $I \geq 8$ corresponding image points where their object points are in general 3D positions. The solution is unique only if the points do not lie on a critical surface, which in this case is a 3D plane or, more generally, a ruled quadric (hyperboloid of one sheet), including its degeneracies, which also contain the projection centres, see Krames (1941, p. 334) and Faugeras and Luong (2001, p. 301).

critical surface

The solution $\widehat{\mathbf{f}}^{(1)}$ for $\mathbf{f}$ obtained in this way will generally not yield a valid fundamental matrix, since $|\widehat{\mathsf{F}}| = 0$ is not enforced. This can be achieved by setting its smallest singular value λ_3 to zero, thus

$$\widehat{\mathsf{F}} = \boldsymbol{U}\widehat{\boldsymbol{\Lambda}}\boldsymbol{V}^\mathsf{T} \qquad \widehat{\boldsymbol{\Lambda}} = \frac{1}{\sqrt{\lambda_1^2 + \lambda_2^2}}\mathrm{Diag}([\lambda_1, \lambda_2, 0])\,. \tag{13.101}$$

The factor in $\widehat{\boldsymbol{\Lambda}}$ forces the fundamental matrix to have Frobenius norm 1.

For stability reasons it is necessary to use the conditioning technique when determining the essential or fundamental matrix (see Sect. 6.9, p. 286).

13.3.2.2 Estimating the Fundamental Matrix F from Seven Points

We have a direct minimal solution of the fundamental matrix, if we observe seven corresponding points in general position. The result is not necessarily unique, but has one or three solutions, in general (see von Sanden, 1908; Tschupik and Hohenberg, 1972; Hartley and Zisserman, 2000). The matrix A in (13.99), p. 571 then has a two-dimensional null space, say $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2]$. Any linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$,

$$\boldsymbol{f}(\mu) = \mu\mathbf{v}_1 + (1-\mu)\mathbf{v}_2\,, \tag{13.102}$$

fulfils $\mathsf{A}\mathbf{f} = \mathbf{0}$, where μ is an arbitrary scalar.

Forcing the 3×3 determinant to fulfil $|\mathsf{F}(\mu)| = 0$ leads to a cubic equation in μ whose one or three roots yield fundamental matrices satisfying the singularity constraint. At least one additional point is necessary in order to select the correct solution.

Uncertainty of Fundamental Matrix and Epipoles We now derive the uncertainty of the fundamental matrix, epipoles, and epipolar lines, which are needed in a RANSAC procedure for consistency checking. In spite of having applied an algebraically optimal solution not exploiting the uncertainty, we now need the covariance matrix of the I point correspondences $\{(\mathbf{x}', \mathbf{x}'')_i\}, i = 1, ..., I$, collected in the $6I \times 6I$ block diagonal matrix

$$\boldsymbol{\Sigma}_{ll} = \mathrm{Diag}\left(\left\{\mathbb{D}\left(\begin{bmatrix}\mathbf{x}'\\ \mathbf{x}''\end{bmatrix}_i\right)\right\}\right)\,; \tag{13.103}$$

see (13.91), p. 569 and (13.92), p. 569. In the simplest case it may be assumed to be $\boldsymbol{\Sigma}_{ll} = \sigma_{x'}^2 \boldsymbol{I}_{6I}$.

Uncertainty of Fundamental Matrix We use the vector $\mathbf{f} := \widehat{\text{vec}\mathsf{F}}$ from (13.101) and impose the two constraints: The Frobenius norm needs to be 1 and the matrix needs to be singular. This is done by interpreting $\mathbf{f}$ as observations in the estimation model C with constraints on the observations only, see Table 4.10, p. 172, row C. Using the covariance matrix of the algebraic solution from (4.521), p. 181, this leads to the final covariance matrix of $\widehat{\mathbf{f}}$ including the constraints.

The covariance matrix Σ_{ff} derived from the eight-point solution is given by

$$\Sigma_{\text{ff}} = A^{+} B \Sigma_{ll} B^{\mathsf{T}} A^{+\mathsf{T}} , \tag{13.104}$$

with the $I \times 9$ matrix A from (13.99) and the $I \times 6I$ matrix

$$B = \text{Diag}([\boldsymbol{b}_i^{\mathsf{T}}]) , \qquad \boldsymbol{b}_i^{\mathsf{T}} = \frac{\partial w_i}{\partial \mathbf{f}} = [\mathbf{x}_i''^{\mathsf{T}} \mathsf{F}^{\mathsf{T}} , \mathbf{x}_i'^{\mathsf{T}} \mathsf{F}] \tag{13.105}$$

see (13.23), p. 555. The Jacobians A and B need to be evaluated at the vector from the algebraical optimization, including the singularity constraint. This constraint, however, is not taken into account in (13.104).

We therefore derive an expression for the covariance matrix of the consistent estimate $\widehat{\mathbf{f}} = \widehat{\text{vec}\mathsf{F}}$. For this, we apply the two constraints on $\mathbf{f}$. They can be written as

$$\boldsymbol{g}(\tilde{\mathbf{f}}) = \begin{bmatrix} \frac{1}{2}\left(\tilde{\mathbf{f}}^{\mathsf{T}}\tilde{\mathbf{f}} - 1\right) \\ 3|\tilde{\mathsf{F}}| \end{bmatrix} = \mathbf{0} . \tag{13.106}$$

The determinant can be written as the inner product of the vector $\mathbf{f}^{\circ}$ of the cofactor matrix F° of the matrix $\mathsf{F} = [\boldsymbol{f}_1 , \boldsymbol{f}_2 , \boldsymbol{f}_3]$,

$$\mathsf{F}^{\circ} = [\boldsymbol{f}_2 \times \boldsymbol{f}_3 , \boldsymbol{f}_3 \times \boldsymbol{f}_1 , \boldsymbol{f}_1 \times \boldsymbol{f}_2] , \tag{13.107}$$

and the vector $\mathbf{f}$. Therefore we have the constraint

$$0 = |\tilde{\mathsf{F}}| = \frac{1}{3}\text{tr}(|\tilde{\mathsf{F}}|\, I_3) = \frac{1}{3}\text{tr}(\tilde{\mathsf{F}}^{\circ\mathsf{T}} \tilde{\mathsf{F}}) = \frac{1}{3}\tilde{\mathbf{f}}^{\circ\mathsf{T}}\tilde{\mathbf{f}} . \tag{13.108}$$

The last step results from the property of the Kronecker product $\text{tr}(A^{\mathsf{T}}B) = \text{vec}^{\mathsf{T}}A\, \text{vec}B$, see (A.94), p. 775.

We now impose the nonlinear constraints

$$\boldsymbol{g}(\mathbf{f} + \widehat{\boldsymbol{v}}_f) = \begin{bmatrix} \frac{1}{2}\left((\mathbf{f} + \widehat{\boldsymbol{v}}_f)^{\mathsf{T}}(\mathbf{f} + \widehat{\boldsymbol{v}}_f) - 1\right) \\ (\mathbf{f} + \widehat{\boldsymbol{v}}_f)^{\circ\mathsf{T}}(\mathbf{f} + \widehat{\boldsymbol{v}}_f) \end{bmatrix} = \mathbf{0} \tag{13.109}$$

on the stochastic vector $\underline{\mathbf{f}} \sim \mathcal{M}(\mathbf{f}, \Sigma_{\text{ff}})$ taken as vector $\underline{\boldsymbol{l}}$ of observations in estimation model C (see Table 4.10, p. 172). With the Jacobian

$$B = \frac{\partial \boldsymbol{g}}{\partial \mathbf{f}} = \begin{bmatrix} \widehat{\mathbf{f}}^{\mathsf{T}} \\ \widehat{\mathbf{f}}^{\circ\mathsf{T}} \end{bmatrix}^{a} \tag{13.110}$$

to be evaluated at the approximate values $\widehat{\mathbf{f}}^{a}$ from the algebraic solution, the estimated vector can be written as

$$\widehat{\underline{\mathbf{f}}} = \underline{\mathbf{f}} + \widehat{\underline{\boldsymbol{v}}}_f = \widehat{\mathbf{f}}^{a} + \widehat{\underline{\Delta\mathbf{f}}} = \underline{\mathbf{f}} + B(B^{\mathsf{T}}B)^{-1}(-\boldsymbol{g}(\widehat{\mathbf{f}}^{a}) + B^{\mathsf{T}}(\widehat{\mathbf{f}}^{a} - \underline{\mathbf{f}})) ; \tag{13.111}$$

see Table 4.10, p. 172, row C. Here we used $\Sigma_{ll} = I_9$, since enforcing the constraints using the SVD weights all nine elements equally. Of course, the algebraically optimal estimate in (13.101), p. 571 already has been forced to have Frobenius norm 1. Therefore, imposing the constraints does not change the estimate, whatever its covariance matrix.

The covariance matrix of the final estimate thus is obtained by variance propagation,

$$\Sigma_{\widehat{\mathrm{ff}}} = J\,\Sigma_{\mathrm{ff}}\,J^{\mathsf{T}}\,, \qquad J = \frac{\partial \widehat{\mathbf{f}}}{\partial \mathbf{f}} = I_9 - B(B^{\mathsf{T}}B)^{-1}B\,. \tag{13.112}$$

Uncertainty of the Epipoles. The uncertainty of the epipoles can easily be determined. We show it for the left epipole $\mathbf{e}'$, where it follows from $\mathbf{e}'^{\mathsf{T}}\mathsf{F} = \mathbf{0}$. With the column partitioning of the fundamental matrix $\mathsf{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3]$, this can be written as

$$\mathbf{e}'^{\mathsf{T}}\boldsymbol{f}_j = 0\,, \quad j = 1, 2, 3, \tag{13.113}$$

from which we obtain

$$\mathbf{e}' = \boldsymbol{f}_j \times \boldsymbol{f}_k\,, \quad j \neq k. \tag{13.114}$$

With $(j, k) = (1, 2)$, this yields the covariance matrix for

$$\mathbf{e}' = \boldsymbol{f}_1 \times \boldsymbol{f}_2\,, \quad \Sigma_{\mathbf{e}'\mathbf{e}'} = [-S(\boldsymbol{f}_2) \mid S(\boldsymbol{f}_1)] \begin{bmatrix} \Sigma_{f_1 f_1} & \Sigma_{f_1 f_2} \\ \Sigma_{f_2 f_1} & \Sigma_{f_2 f_2} \end{bmatrix} \begin{bmatrix} -S^{\mathsf{T}}(\boldsymbol{f}_2) \\ S^{\mathsf{T}}(\boldsymbol{f}_1) \end{bmatrix}. \tag{13.115}$$

Uncertainty of an Epipolar Line. The uncertainty of an epipolar line depends on both the uncertainty of the given point and the uncertainty of the relative orientation. For example, take the epipolar line $\mathbf{l}'' = \widehat{\mathsf{F}}^{\mathsf{T}}\mathbf{x}'$. Here we have the covariance matrix

$$\Sigma_{\mathrm{ll}} = (I_3 \otimes \mathbf{x}'^{\mathsf{T}})\Sigma_{\widehat{\mathrm{ff}}}(I_3 \otimes \mathbf{x}) + \widehat{\mathsf{F}}^{\mathsf{T}}\Sigma_{\mathbf{x}'\mathbf{x}'}\widehat{\mathsf{F}}\,. \tag{13.116}$$

Example 13.3.46: ***Uncertainty of epipolar lines.*** The example demonstrates uncertainty propagation through the measuring chain, see Fig. 13.9.

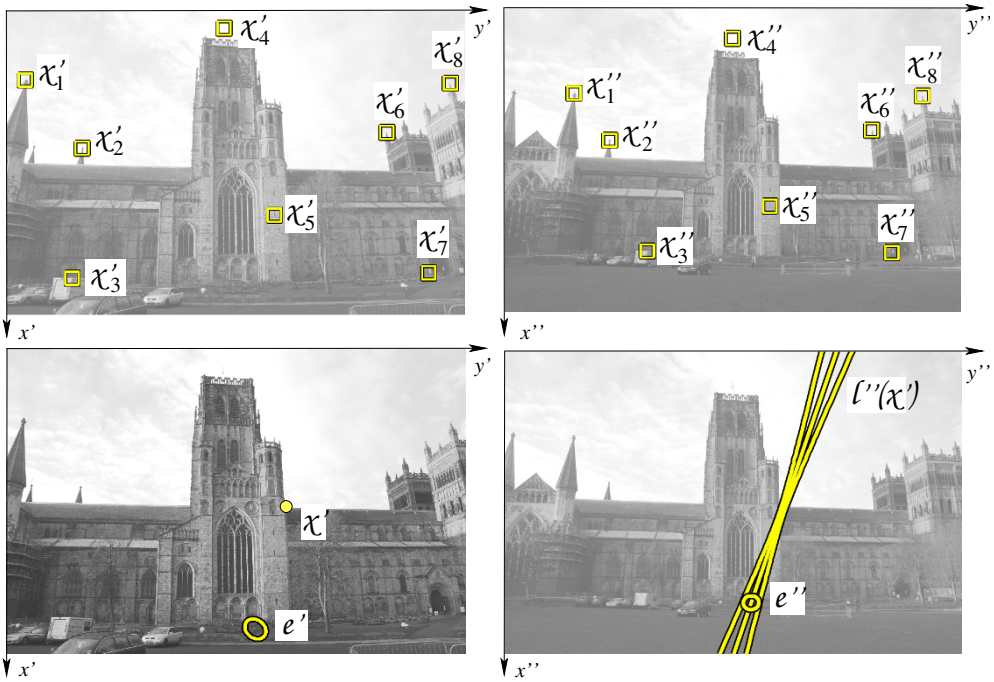


Fig. 13.9 Uncertainty of epipolar lines: The uncertainty of an epipolar line depends on both the uncertainty of the observed point $\mathbf{x}'$ and the uncertainty of the relative orientation $\widehat{\mathsf{F}}$. **Upper row:** an image pair of size 3456×2304 pixels with eight corresponding points $(\mathbf{x}', \mathbf{x}'')_i$. **Lower left:** Epipole $\boldsymbol{e}'$ and given image point $\mathbf{x}'$. **Lower right:** Epipole $\boldsymbol{e}''$ and epipolar line $\boldsymbol{l}''(\mathbf{x}')$ with hyperbolic uncertainty region. The uncertainty regions are magnified by a factor of 10

Eight corresponding points are measured manually in the left and right images (top row), see Table 13.3. Their uncertainty is assumed to be due to rounding errors: $\sigma_{x'} = \sigma_{y'} = \sigma_{x''} = \sigma_{y''} = \sqrt{1/12} \approx 0.287$

Table 13.3 Coordinates of corresponding points in pixels

i	x'_i	y'_i	x''_i	y''_i
1	533.4	140.2	650.6	524.9
2	1047.3	570.0	1008.2	795.4
3	2033.0	491.8	1849.7	1080.9
4	139.7	1639.8	235.9	1730.0
5	1555.2	2021.5	1510.1	2012.5
6	927.1	2869.0	933.1	2784.9
7	557.5	3346.8	671.7	3169.5
8	1991.0	3184.6	1858.7	2938.1

pixels. The fundamental matrix is estimated using the eight-point algorithm, internally conditioning the image coordinates and unconditioning the resulting fundamental matrix following the procedure in Sect. 6.9, p. 286. The zero-determinant constraint is applied as in (13.101), but without the normalizing factor. The resulting fundamental matrix is

$$\widehat{\mathsf{F}} = 10^{-2} \begin{bmatrix} +0.0528 \cdot 10^{-6} & -2.2435 \cdot 10^{-6} & +4.1001 \cdot 10^{-3} \\ +2.1966 \cdot 10^{-6} & +0.0484 \cdot 10^{-6} & -4.0426 \cdot 10^{-3} \\ -4.5645 \cdot 10^{-3} & +3.8409 \cdot 10^{-3} & +1.03 \end{bmatrix} . \tag{13.117}$$

Observe the large differences in magnitude of the entries. This results from the different units, e.g., $\widehat{F}(1,1)$ having unit [1/pixel^2] and $\widehat{F}(3,3)$ having unit [1]. With the standard deviations $\boldsymbol{\sigma}$ of $\widehat{\mathbf{f}} = \mathrm{vec}\widehat{\mathsf{F}}$,

$$\boldsymbol{\sigma} = 10^{-5} \, \mathrm{vec} \begin{bmatrix} 0.7260 \cdot 10^{-6} & 0.6443 \cdot 10^{-6} & 13.54 \cdot 10^{-3} \\ 1.7859 \cdot 10^{-6} & 2.4994 \cdot 10^{-6} & 16.01 \cdot 10^{-3} \\ 11.26 \cdot 10^{-3} & 15.03 \cdot 10^{-3} & 3.6557 \end{bmatrix} , \tag{13.118}$$

and the correlation matrix[6] $R = [\rho_{ij}]$,

$$\begin{bmatrix} 1.0000 & -0.2985 & -0.6894 & -0.6238 & 0.3808 & 0.2853 & 0.5697 & -0.2139 & 0.4862 \\ & 1.0000 & 0.1323 & 0.2245 & -0.4292 & -0.7726 & -0.1510 & 0.7269 & 0.4479 \\ & & 1.0000 & \textit{0.9623} & -0.6811 & -0.0572 & \textit{-0.9837} & 0.0437 & -0.5435 \\ & & & 1.0000 & -0.6356 & 0.0169 & \textit{-0.9783} & -0.0526 & -0.4037 \\ & & & & 1.0000 & 0.3331 & 0.6980 & -0.4629 & 0.2726 \\ & & & & & 1.0000 & 0.0158 & \textit{-0.9690} & -0.3031 \\ & & & \text{sym.} & & & 1.0000 & -0.0084 & 0.4552 \\ & & & & & & & 1.0000 & 0.2279 \\ & & & & & & & & 1.0000 \end{bmatrix} , \tag{13.119}$$

the covariance matrix $\Sigma_{\widehat{\mathrm{ff}}}$ is

$$\Sigma_{\widehat{\mathrm{ff}}} = \mathrm{Diag}(\boldsymbol{\sigma}) \, R \, \mathrm{Diag}(\boldsymbol{\sigma}) . \tag{13.120}$$

It has rank 7. Observe, the relative accuracy of the entries in the fundamental matrix is worse than 1:1000, in spite of the high resolution of the image with directional accuracies of approximately 1:4000. The maximal and minimal eigenvalues of the correlation matrix are 4.35 and 0.0009, leading to a condition number of 4885 and high correlations (italics in (13.119)). Both observations indicate a low stability of the fundamental matrix, which is due to the low depth differences of the underlying 3D points.

In Fig. 13.9 the epipoles are shown together with their standard ellipse, magnified by a factor 10. The epipoles are

$$\boldsymbol{e}' = \begin{bmatrix} 1755.9 \pm 8.3 \\ 2035.8 \pm 5.5 \end{bmatrix} , \ \rho_{x'_e y'_e} = +0.120, \quad \boldsymbol{e}'' = \begin{bmatrix} 1799.2 \pm 6.6 \\ 1869.9 \pm 3.7 \end{bmatrix} , \ \rho_{x''_e y''_e} = -0.137 . \tag{13.121}$$

The standard ellipses of the epipoles have maximum and minimum semiaxes of 8.1 pixels and 3.8 pixels, which indicate a high uncertainty of the relative orientation in the direction of the basis.

An additional point $\mathcal{X}'([2156.0, 1226.0])$ is measured in the left image, leading to the epipolar line $\mathcal{l}''(\mathcal{X}')$ in the other image. It is given by its homogeneous coordinates and in its centroid representation:

$$\mathbf{l}'' = \begin{bmatrix} -0.9404 \\ -0.3400 \\ 2410.9 \end{bmatrix} , \quad \boldsymbol{x}_0 = \begin{bmatrix} 2099.0 \\ 1285.1 \end{bmatrix} , \ \phi = -160.12^\circ , \ \sigma_\phi = 0.485^\circ , \ \sigma_q = 2.12 . \tag{13.122}$$

As the uncertainty of the epipolar line is also shown with a magnification factor of 10 in Fig. 13.9, p. 573, the angle between the two branches of the standard hyperbola is $2\sigma_\phi \cdot 10 \approx 9.7^\circ$, see Sect. 10.2.2.3, p. 373. ◇

[6] Not to be confused with a rotation matrix.

13.3.2.3 Estimating the Essential Matrix E from ≥ 7 Points

If the cameras are calibrated we can compute the direction vectors ${}^c\mathbf{x}'_i = \mathsf{K}_1^{-1}\mathbf{x}'_i$ and ${}^c\mathbf{x}''_i = \mathsf{K}_2^{-1}\mathbf{x}''_i$. The coplanarity constraints are

$$ {}^c\mathbf{x}'^{\mathsf{T}}_i \, \mathsf{E} \, {}^c\mathbf{x}''_i = w_i \stackrel{!}{=} 0 \qquad i = 1, ..., I \,, \tag{13.123} $$

in full analogy to (13.95). The resulting homogeneous equation system for the vector $\mathbf{e} = \mathrm{vec}\mathsf{E}$ of the essential matrix, not to be confused with the epipoles, is

$$ \mathsf{A} \, \mathrm{vec}\mathsf{E} \stackrel{!}{=} \mathbf{0} \qquad \text{with} \qquad \mathsf{A} = [{}^c\mathbf{x}_i''^{\mathsf{T}} \otimes {}^c\mathbf{x}_i'^{\mathsf{T}}] \,. \tag{13.124} $$

Thus the solution procedure for the fundamental matrix can also be used here, leading generally to one or three solutions. Also here the scene points should not be coplanar.

The algebraically optimal solution does not lead to a valid essential matrix. Since it has two identical eigenvalues, Sect. 13.2.3, p. 555, the final estimate for E results from the SVD of some initial estimate $\mathsf{E} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathsf{T}}$ by

$$ \widehat{\mathsf{E}} = \boldsymbol{U}\widehat{\boldsymbol{\Lambda}}\boldsymbol{V}^{\mathsf{T}} \qquad \text{with} \qquad \widehat{\boldsymbol{\Lambda}} = \mathrm{Diag}([1, 1, 0]) \,. \tag{13.125} $$

Though $\widehat{\mathsf{E}}$ is a valid essential matrix, the four constraints from (13.32) and (13.33), p. 557 need to be imposed on the covariance matrix, which can be done as for the fundamental matrix in (13.112), p. 573.

13.3.2.4 Direct Solution for E from Five Points

The following direct solution uses the minimum number of corresponding point pairs for determining the essential matrix and therefore is recommended for a RANSAC scheme for detecting outliers. It is known that for any configuration there are up to ten real solutions, see the references in Faugeras and Maybank (1990). The proposed algorithm yields exactly these solutions. It also works if the scene is planar, except for the critical configuration where the base line is perpendicular to the plane, e.g., when a space craft is landing.

We give a more detailed derivation, which requires manipulation of polynomials following Nistér (2003) and Stewénius et al. (2006). A more general and therefore systematic approach using *Gröbner bases*, which allows handling systems of polynomial equations, can be used to solve more complex direct solutions, see Stewénius (2005) and Kúkelová (2013).

In a first step, we start with the 5×9 matrix,

$$ \underset{5\times 9}{\mathsf{A}} = \begin{bmatrix} {}^c\mathbf{x}_1''^{\mathsf{T}} \otimes {}^c\mathbf{x}_1'^{\mathsf{T}} \\ \dots \\ {}^c\mathbf{x}_5''^{\mathsf{T}} \otimes {}^c\mathbf{x}_5'^{\mathsf{T}} \end{bmatrix} \tag{13.126} $$

(see (13.124), p. 575) for determining the nine coefficients of $\mathrm{vec}\mathsf{E}$ from $I = 5$ point pairs. The rank of matrix A in general is 5. Let the four-dimensional null space (see Sect. A.11, p. 777), be $[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4]$, then the vector of the unknown essential matrix can be expressed as

$$ \mathrm{vec}\mathsf{E} = u\boldsymbol{v}_1 + v\boldsymbol{v}_2 + w\boldsymbol{v}_3 + t\boldsymbol{v}_4 \,, \tag{13.127} $$

This leads to an essential matrix parametrized by the four unknown parameters (u, v, w, t),

$$ \mathsf{E}(u, v, w, t) = \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix} = u\mathsf{E}_1 + v\mathsf{E}_2 + w\mathsf{E}_3 + t\mathsf{E}_4 \,, \tag{13.128} $$

where the elements e_i of the essential matrix are linear in the unknown factors and the matrices E_j are related to the base vectors of the null space by

$$\text{vec}\mathsf{E}_1 = \boldsymbol{v}_1\,, \quad \text{vec}\mathsf{E}_2 = \boldsymbol{v}_2\,, \quad \text{vec}\mathsf{E}_3 = \boldsymbol{v}_3\,, \quad \text{vec}\mathsf{E}_4 = \boldsymbol{v}_4\,. \tag{13.129}$$

The goal is to determine the factors (u, v, w, t) such that E has the properties of an essential matrix.

We now observe that due to the homogeneity of the essential matrix only the ratios of these four parameters are needed, e.g.,

$$x = \frac{u}{t}\,, \qquad y = \frac{v}{t}\,, \qquad z = \frac{w}{t}\,. \tag{13.130}$$

Therefore the essential matrix $\mathsf{E}(x, y, z)$ depends on these three unknown parameters. Additionally, due to $\mathsf{E} = \mathsf{S}_B R^\mathsf{T}$ and the property of the cube of a skew matrix, see A.4.2, p. 771, the following (dependent) ten conditions must hold:

$$|\mathsf{E}| \stackrel{!}{=} 0\,, \tag{13.131}$$

$$\mathsf{E}\mathsf{E}^\mathsf{T}\mathsf{E} - \frac{1}{2}\text{tr}(\mathsf{E}\mathsf{E}^\mathsf{T})\,\mathsf{E} \stackrel{!}{=} \underset{3\times 3}{\mathsf{0}}\,. \tag{13.132}$$

These constraints are cubic in the factors (x, y, z). They can be reorganized as a set of ten polynomials of third degree in the three variables x, y, and z,

$$\underset{10\times 20}{\mathsf{M}}\;\underset{20\times 1}{\boldsymbol{p}} = \underset{10\times 1}{\mathbf{0}}\,, \tag{13.133}$$

with the monomial vector $\boldsymbol{p}$ containing the monomials in graded lexicographic order,

$$\boldsymbol{p} = [x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3; x^2, xy, xz, y^2, yz, z^2, x, y, z, 1]^\mathsf{T}\,, \tag{13.134}$$

where the elements of the real matrix M depend on the entries of A. This equation system has ten solutions in $\mathbb{C}$, thus up to ten solutions in $\mathbb{R}$. The goal now is to arrive at a system of ten equations in the ten monomials up to order 2 whose values then can be solved for using an eigenvalue decomposition of the coefficients. We first split the monomial vector into two parts:

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix} = \begin{bmatrix} [x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3]^\mathsf{T} \\ [x^2, xy, xz, y^2, yz, z^2, x, y, z, 1]^\mathsf{T} \end{bmatrix}\,. \tag{13.135}$$

The vector $\boldsymbol{q}$ contains all third-order monomials. Now observe, the first six monomials in $\boldsymbol{q}$ can be obtained from the first six monomials in $\boldsymbol{r}$ by multiplication with x, thus

$$q_i = x\, r_i \qquad i = 1, ..., 6\,. \tag{13.136}$$

Using the partitioning of the polynomials, (13.135) allows us to write (13.133) as a partitioned system of equations,

$$\mathsf{M}\boldsymbol{p} = [\mathsf{C}|\mathsf{D}] \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix} = \mathsf{C}\boldsymbol{q} + \mathsf{D}\boldsymbol{r} = \mathbf{0}\,. \tag{13.137}$$

It may be solved for the third-order monomials,

$$\boldsymbol{q} = -\mathsf{C}^{-1}\mathsf{D}\boldsymbol{r} = \mathsf{B}\boldsymbol{r}\,. \tag{13.138}$$

We now derive a set of ten equations, which relate the ten elements of $\boldsymbol{r}$ to themselves, i.e., they yield an eigenvalue problem. For this we take the rows $\boldsymbol{b}_i^\mathsf{T}, i = 1, ..., 10$, of B so that we may explicitly write this system as

$$q_i = \boldsymbol{b}_i^\mathsf{T}\boldsymbol{r} \qquad i = 1, ..., 10\,. \tag{13.139}$$

Especially, with (13.138) and (13.136) for the first six rows, we have

$$x\, r_i = \boldsymbol{b}_i^\mathsf{T} r_i \qquad i = 1, ..., 6\,. \tag{13.140}$$

In addition, we obviously also have the relations between the polynomials in $\boldsymbol{r}$,

$$x\, r_7 = r_1\,, \quad x\, r_8 = r_2\,, \quad x\, r_9 = r_3\,, \quad x\, r_{10} = r_7\,. \tag{13.141}$$

Observe, Eqs. (13.140) and (13.141) relate elements of $\boldsymbol{r}$ to themselves, which can be written as

$$x\, \boldsymbol{r} = \mathsf{A}_f^\mathsf{T} \boldsymbol{r}\,, \tag{13.142}$$

with the matrix

$$\mathsf{A}_f^\mathsf{T} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{b}_1^\mathsf{T} \\ \dots \\ \boldsymbol{b}_6^\mathsf{T} \end{bmatrix} \\ \begin{bmatrix} \mathsf{I}_3 & \mathsf{0}_{3\times 3} & \mathbf{0}_3 & \mathsf{0}_{3\times 3} \\ \mathbf{0}_3^\mathsf{T} & \mathbf{0}_3^\mathsf{T} & 1 & \mathbf{0}_3^\mathsf{T} \end{bmatrix} \end{bmatrix}\,, \tag{13.143}$$

which is called the action matrix in the context of Gröbner bases. The right eigenvectors of the 10×10 matrix A_f therefore are the sought solutions for $\boldsymbol{r}$, one for each of the ten eigenvalues x_k.

These eigenvalues must be either real or pairwise complex. The $K \leq 10$ eigenvectors $\boldsymbol{r}_k, k = 1, ..., K$, belonging to the real eigenvalues x_k can be used to determine the k solutions for the unknown factors,

$$x_k = \frac{r_{k,7}}{r_{k,10}}\,, \qquad y_k = \frac{r_{k,8}}{r_{k,10}}\,, \qquad z_k = \frac{r_{k,9}}{r_{k,10}}\,, \qquad k = 1, ..., K\,, \tag{13.144}$$

or directly the original factors in (13.128),

$$u_k = r_{k,7}\,, \quad v_k = r_{k,8}\,, \quad w_k = r_{k,9}\,. \quad t_k = r_{k,10}\,, \quad k = 1, ..., K\,. \tag{13.145}$$

In general, we thus obtain an even number $K \leq 10$ of solutions for the essential matrix. Observe, the method allows us to capture the maximum of possible solutions (see Faugeras and Maybank, 1990). In a RANSAC scheme they can be checked for consensus with other than the used five pairs.

This direct solution can also be applied with more than five points. Then the matrix A from (13.124), p. 575 will have a rank larger than 5, and possibly be regular, due to the noise of the given image coordinates. The algorithm uses the decomposition (13.128) derived from the eigenvectors belonging to the four smallest eigenvalues values of $\mathsf{A}^\mathsf{T}\mathsf{A}$. However, we will only obtain approximate essential matrices, and therefore need to perform adequate normalization (13.125), p. 575. With seven or more points the algorithm in Sect. 13.3.2.3, p. 575 is simpler, at the expense of not being able to handle planar scenes.

Demazure (1988) showed that for correspondences ${}^c\mathbf{x}'_i = {}^c\mathbf{x}''_i$, $i = 1, 2, 3$, ${}^c\mathbf{x}'_4 = {}^c\mathbf{x}''_5$ and ${}^c\mathbf{x}'_5 = {}^c\mathbf{x}''_4$ in general position, ten real and distinct solutions are obtained. This also holds if the vectors are slightly perturbed, as demonstrated by Faugeras and Maybank (1990). Simulations by Stewénius et al. (2006) revealed that on the average only two or four real solutions are to be expected. As mentioned above, the most important property of this five point procedure is that it also works if the points are coplanar, except for some special degenerate configurations, see 13.3.6.2, p. 593. This is of high practical value. When coplanar points are used, two valid solutions are generally obtained.

13.3.2.5 Determination of the Essential Matrix E from ≥ 4 Points

All solution procedures discussed so far, except the five-point solution, fail if the 3D points are coplanar. However, there exist solutions with four coplanar corresponding points (see

Wunderlich, 1982; Tsai et al., 1982; Kager et al., 1985; Faugeras, 1993). This is possible since for a planar object the relation between corresponding points in the two images is a 2D homography with eight independent parameters, see Sect. 13.2.7, p. 567. This allows the determination of five parameters of the relative orientation together with the three of the object plane. These procedures work in most practical cases, but may have two solutions in some rare cases. The solution is lengthy (see Faugeras and Lustman, 1988; Faugeras, 1993).

13.3.2.6 Direct Solution of the Essential Matrix E from ≥ 2 Corresponding Points

The essential matrix can be determined from less than four points if certain pre-knowledge about the configuration is available. We discuss two relevant cases:

1. The rotation between the two images is known, e.g., zero. Then only the two parameters of the direction of the base line have to be determined. This is relevant for neighbouring images in a video, where the rotational component of the motion is small.
2. The object is symmetric with respect to a plane in 3D. Then only the two parameters of the relative rotation between the viewing direction and the plane normal have to be determined.

Essential Matrix for Pure Translation. If the rotation matrix of the relative orientation is known, it is possible to directly estimate the direction of the base vector $\mathbf{b}$ from a minimum of two corresponding points $\mathcal{X}$ and $\mathcal{Y}$, as the essential matrix $\mathsf{E} = \mathsf{S}(\mathbf{b})R^{\mathsf{T}}$ only depends on the direction of the base vector in this case.

Let the observed directions to the points in object space be ${}^{n}\mathbf{x}' = (\mathsf{K}'R')^{-1}\mathbf{x}'$, and ${}^{n}\mathbf{x}''$, ${}^{n}\mathbf{y}'$, and ${}^{n}\mathbf{y}''$ be determined similarly, see (12.41), p. 473. Then the base vector is given by

$$\mathbf{b} = ({}^{n}\mathbf{x}' \times {}^{n}\mathbf{x}'') \times ({}^{n}\mathbf{y}' \times {}^{n}\mathbf{y}'')\,, \tag{13.146}$$

since its direction is perpendicular to the normals $\mathbf{n}_X$ and $\mathbf{n}_Y$ of the two epipolar planes, which themselves are spanned by the directions to the observed points in object space. Again, there are two solutions for $\mathbf{b}$, induced by the sign of the base vector. Of these, the one where the rays intersect in front of the cameras (Fig. 13.10) needs to be chosen, see (13.164), p. 582.

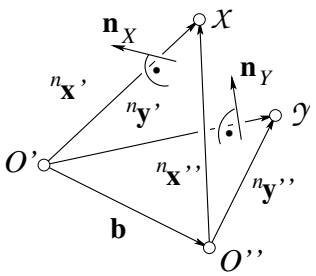


Fig. 13.10 Determination of the base vector from two corresponding points $\mathcal{X}$ and $\mathcal{Y}$ for given rotation matrices

If we have $I \geq 2$ corresponding points $(\mathbf{x}'_i, \mathbf{x}''_i), i = 1, ..., I$, we can determine the base vector from the I constraints,

$$({}^{n}\mathbf{x}'_i \times {}^{n}\mathbf{x}''_i)^{\mathsf{T}}\,\mathbf{b} = 0 \qquad i = 1, ..., I\,, \tag{13.147}$$

or $\mathsf{A}\mathbf{b} = \mathbf{0}$, minimizing the algebraic error. The solution $\mathbf{b}$ is the right eigenvector of the matrix

$$\mathsf{A} = \begin{bmatrix} {}^{n}\mathbf{x}_1{}'^{\mathsf{T}} \times {}^{n}\mathbf{x}_1{}''^{\mathsf{T}} \\ \dots \\ {}^{n}\mathbf{x}_i{}'^{\mathsf{T}} \times {}^{n}\mathbf{x}_i{}''^{\mathsf{T}} \\ \dots \\ {}^{n}\mathbf{x}_I{}'^{\mathsf{T}} \times {}^{n}\mathbf{x}_I{}''^{\mathsf{T}} \end{bmatrix} \tag{13.148}$$

belonging to the smallest eigenvalue. It is identical to the corresponding eigenvector of the matrix $\mathsf{A}^{\mathsf{T}}\mathsf{A}$ belonging to the smallest eigenvalue, which is identical to the last column of V of the SVD $\mathsf{A} = U\Lambda V^{\mathsf{T}}$.

Essential Matrix for Symmetric Object. If a symmetric object is seen in a single image, see e.g., Fig. 13.11, left, the 3D structure of the object can be recovered as if it were shown in two images. Thus we assume the object to be symmetric with respect to a plane $\mathcal{A}$. We have one image of the object showing enough details to identify points at the object which are symmetric w.r.t. $\mathcal{A}$, say x_i' and y_i', as shown in the figure.

We now can generate a second image by mirroring the given image at one of the two axes of the sensor coordinate system – in the case of Fig. 13.11, centre – at the y'-axis. This mirroring of the image maps the points (x_i', y_i') to the points (y_i'', x_i'') in the second image. Therefore, we can treat the pairs (x_i', x_i'') and (y_i', y_i'') as corresponding point pairs in the image pair.

We can then imagine that the second image was taken with a virtual camera at O'', which is the point O' mirrored at the plane of symmetry $\mathcal{A}$ of the object. The pose of the two cameras, the real and the virtual one, are closely related, see Fig. 13.11, right. Using the representation of the relative orientation with independent images, see Sect. 13.2.3.2, p. 558, the two rotation matrices depend on the same two angles, $\phi' = -\phi''$ and $\kappa' = -\kappa''$. Thus a minimum of two corresponding rays, which refer to an image point of the object and the corresponding image point of the symmetric object, are generally necessary for determining these two parameters of the relative orientation.

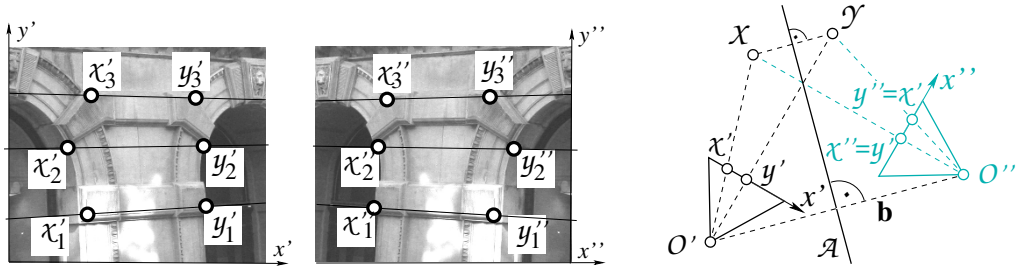


Fig. 13.11 Relative orientation of a single image of a symmetric object and its mirrored image. **Left/centre:** points $\mathcal{Y}_i$ are mirror points w.r.t. a plane of the points $\mathcal{X}_i$. Image points (x_i', y_i') of symmetric scene points $(\mathcal{X}_i, \mathcal{Y}_i)$ are corresponding. **Right:** the mirror image can be imagined as having been taken by a camera at O'' mirrored at the mirror plane $\mathcal{A}$

Let the base line be the vector $\mathbf{b} = [1, 0, 0]^{\mathsf{T}}$ representing the normal of the symmetry plane. Then the rotation around the basis, i.e., the X-axis, cannot be determined; thus, the rotation matrix can be specified by a rotation around the Y- and the Z-axes, the Y-axis, which is freely defined parallel to the plane of symmetry.

$$R' = R_Q([1, 0, q_2, q_3]) = \frac{1}{1 + q_2^2 + q_3^2} \begin{bmatrix} 1 - {q_2}^2 - {q_3}^2 & -2\,q_3 & 2\,q_2 \\ 2\,q_3 & 1 + {q_2}^2 - {q_3}^2 & 2\,q_3 q_2 \\ -2\,q_2 & 2\,q_3 q_2 & 1 - {q_2}^2 + {q_3}^2 \end{bmatrix}. \tag{13.149}$$

The rotation of the other camera is specified similarly by

$$R'' = R_Q([1, 0, -q_2, -q_3]) = \frac{1}{1+q_2^2+q_3^2} \begin{bmatrix} 1-{q_2}^2-{q_3}^2 & 2\,q_3 & -2\,q_2 \\ -2\,q_3 & 1+{q_2}^2-{q_3}^2 & 2\,q_2q_3 \\ 2\,q_2 & 2\,q_2q_3 & 1-{q_2}^2+{q_3}^2 \end{bmatrix} . \tag{13.150}$$

Therefore only two corresponding point pairs $(\mathbf{x}_i', \mathbf{x}_i''), i = 1, 2$, are necessary to determine the essential matrix,

$$\mathsf{E} = R'\mathsf{S}(\mathbf{b})R''^{\mathsf{T}} = \frac{1}{1+q_2^2+q_3^2} \begin{bmatrix} 0 & 2\,q_2 & 2\,q_3 \\ 2\,q_2 & 0 & {q_2}^2+{q_3}^2-1 \\ 2\,q_3 & 1-{q_2}^2-{q_3}^2 & 0 \end{bmatrix} . \tag{13.151}$$

The epipole of the first image is the left eigenvector of the essential matrix and therefore related to the two parameters of the essential matrix by

$$\mathbf{e}' = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \sim \begin{bmatrix} 1-q_2^2-q_3^2 \\ 2q_3 \\ -2q_2 \end{bmatrix} . \tag{13.152}$$

Its coordinates can be determined as the vanishing point of two lines joining image points of symmetric scene points, e.g., $l_i' = \mathbf{x}_i' \wedge \mathbf{y}_i', i = 1, 2$, with the normalized vector

$$\mathbf{e}' = \mathsf{N}((\mathbf{x}_1' \times \mathbf{y}_1') \times (\mathbf{x}_2' \times \mathbf{y}_2')) . \tag{13.153}$$

From (13.152) the values q_2 and q_3 can be derived,

$$q_{2,(1,2)} = -w\ s_{1,2} , \quad q_{3,(1,2)} = v\ s_{1,2} , \quad \text{with} \quad s_{1,2} = \frac{\pm 1 - u}{v^2+w^2} , \tag{13.154}$$

leading to two solutions. They differ by a rotation θ of 180° around the axis $[0, w, -v]^{\mathsf{T}} \perp \mathbf{e}'$ as the lengths $\tan(\theta_1/2)$ and $\tan(\theta_2/2)$ of the vector parts of the two quaternions multiply to -1. The solution with the points in front of the two cameras is the correct one, see the next section. This allows us to derive the two rotation matrices, R' and R'', from (13.149) and (13.150).

Exercise 13.19

In case there are more correspondences, we may use the best estimate for the vanishing point $\mathbf{e}'$.

Example 13.3.47: ***Essential matrix from an image of a symmetric object.*** Given are three points with their coordinates $\boldsymbol{X}_i$ and a plane of symmetry $\mathcal{A}([0.9988, 0.500, 0, -2.0000]^{\mathsf{T}})$, which allows the determination of the mirror points $\boldsymbol{Y}_i$, see Table 13.4 and (6.118), p. 281. Though we only need two points, the third one can be used to check the result of the relative orientation.

Table 13.4 3D coordinates of scene points

No	$\boldsymbol{X}_i^{\mathsf{T}}$	$\boldsymbol{Y}_i^{\mathsf{T}}$
1	[0.9488 , 1.0487 , 1.2000]	[2.9463 , 1.1487 , 1.2000]
2	[1.1485 , 1.0587 , 1.5000]	[2.7465 , 1.1387 , 1.5000]
3	[0.9487 , 3.0512 , 2.0000]	[2.7465 , 3.1412 , 2.0000]

They are projected with the projection matrix

$$\mathsf{P} = \begin{bmatrix} 0.7719 & -0.4912 & -0.4035 \\ 0.5614 & 0.8246 & 0.0702 \\ 0.2982 & -0.2807 & 0.9123 \end{bmatrix} \left[I_3 | - \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix} \right] , \tag{13.155}$$

leading to the image points in Table 13.5.

The first normalized epipole $\mathbf{e}' = [0.7464, 0.6019, 0.2838]^{\mathsf{T}}$ leads to the parameter values $s_1 = 0.5726$ and $s_2 = -3.9435$, the first value leading to the correct solution,

Table 13.5 Homogeneous 2D coordinates of image points

No	$\mathbf{x}_i'^\mathsf{T}$	$\mathbf{y}_i'^\mathsf{T}$
1	[0.6979 4.1833 0.2061]	[2.1908 5.3872 0.7738]
2	[0.7262 4.3248 0.5366]	[1.9204 5.2878 0.9907]
3	[-0.6086 5.8906 0.3738]	[0.7350 6.9741 0.8848]

$$s_1 = 0.5726\,, \quad \text{thus} \quad q_2 = -0.1625\,, \quad q_3 = 0.3447\,, \tag{13.156}$$

and the essential matrix,

$$\mathsf{E} = \begin{bmatrix} 0 & -0.2838 & 0.6019 \\ -0.2838 & 0 & -0.7464 \\ 0.6019 & 0.7464 & 0 \end{bmatrix}. \tag{13.157}$$

As the data are not contaminated by noise, all constraints,

$$w_i = [x_{1i}, x_{2i}, x_{3i}]\mathsf{E}\begin{bmatrix} -y_{1i} \\ y_{2i} \\ y_{3i} \end{bmatrix} = 0\,, \tag{13.158}$$

are zero. Observe, we used the negative coordinate $-y_{1i}$ for the mirror points. The second solution ($q_2 = 1.1194$, $q_3 = -2.3736$) is not valid, as the triangulated points lie behind the camera. ◇

relative orientation with given plumb line

An important situation arises if the scene is a Legoland scene and the direction of the plumb line can be observed. Then the rotation axis between the two images is the plumb line direction. It can be inferred from vanishing points (see Sect. 12.3.4, p. 529): The relative orientation of two images of calibrated cameras with the rotation axis of the second camera given requires three corresponding points for determining the direction of the base vector and the rotation angle (see Fraundorfer et al., 2010).

relative orientation for images of generic cameras

Stewénius et al. (2005) provide a solution for the relative orientation of two images of generic cameras, see Fig. 11.8, p. 446. As the projection rays in a generic camera do not intersect in a common point, also the distances of the rays in the two cameras need to be consistent. This is why – at least conceptually – also the length of the basis between the two cameras; thus, all six parameters of the relative pose of the two cameras can be determined. The problem has up to 64 solutions.

13.3.3 Determining the Orientation Parameters from a Given Essential Matrix

With a known essential matrix E, the parameters of the relative orientation can be derived explicitly. The essential matrix (13.37) can be written as

$$\mathsf{E} = \mathsf{S}_\mathrm{b} R^\mathsf{T}\,, \tag{13.159}$$

where S_b is the skew matrix of the base vector and $R = R''$ is the rotation matrix of the second image with respect to the first image and $R' = I_3$, i.e., we assume the case of dependent images.

Following Hartley (1992) the decomposition relies on the SVD $\mathsf{E} = U\boldsymbol{\Lambda}V^\mathsf{T}$, assuming U and V to be proper orthogonal matrices, and is given by

$$\mathsf{S}_\mathrm{b} = \mu U Z U^\mathsf{T} \quad \text{or} \quad \mathsf{S}_\mathrm{b} = \mu U Z^\mathsf{T} U^\mathsf{T}\,, \tag{13.160}$$

with arbitrary scale factor $\mu \neq 0$ and

$$R^\mathsf{T} = UWV^\mathsf{T} \quad \text{or} \quad R^\mathsf{T} = UW^\mathsf{T}V^\mathsf{T}\,, \tag{13.161}$$

with the two matrices

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{13.162}$$

We can easily verify that $ZW \cong \mathrm{Diag}([1, 1, 0])$, etc., for any combination of transposition of W or Z. Therefore we have four possible solutions. Two alternatives result from changing the sign of the base vector, i.e., using Z^T instead of Z, and two other alternatives result from rotating the second camera around the base vector by 180°, i.e., using W^T instead of W. Only one of these four solutions will be admissible.

Instead of deriving the basis $\mathbf{b}$ from (13.160) it is simpler to take it directly as the third column $\boldsymbol{u}_3$ of the matrix $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]$, since $\mathbf{b}$ is the left singular vector of E belonging to the singular value 0, and $\mathbf{b}^\mathsf{T} U \boldsymbol{\Lambda} = \mathbf{0}^\mathsf{T}$ independently of V. Thus instead of choosing the sign of Z in (13.160) we rather choose the sign of the basis.

Now we need to select the correct solution out of the four. If we have perspective cameras, we select the solution where all or the majority of all 3D points determined by intersection are in front of both cameras (Fig. 13.12): the ${}^{\mathrm{c}}Z$-coordinates of the 3D points in the two camera systems need to be negative.

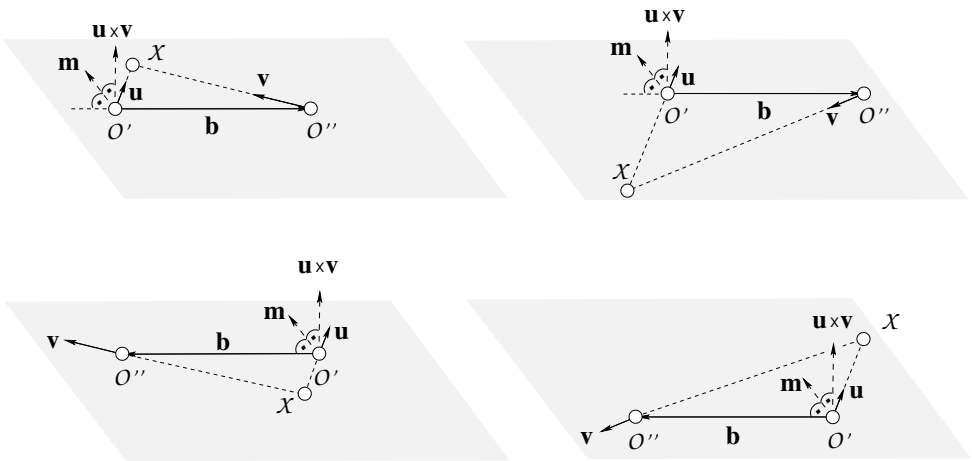


Fig. 13.12 The decomposition of the essential matrix E leads to four solutions for the relative orientation of which only one is admissible: the one where the camera rays $\mathbf{u}$ and $\mathbf{v}$ point towards the intersected 3D point $\mathcal{X}$, shown at top left. The vector $\mathbf{m}$ is the binormal (the normal to the normal $\mathbf{u} \times \mathbf{v}$); i.e., it lies in the epipolar plane (grey). It is perpendicular to $\mathbf{b}$ and on the same side of the base line as $\mathbf{u}$

If we have a spherical camera, however, the notion *'in front'* of the cameras does not make sense. Here we require that the normalized rays $\mathbf{u}$ and $\mathbf{v}$ from the camera centres O' and O'' to the intersected point $\mathcal{X}$ have the same direction as the camera rays, i.e., $\mathbf{u} \stackrel{+}{=} {}^{\mathrm{c}}\mathbf{x}'$ and $\mathbf{v} \stackrel{+}{=} {}^{\mathrm{c}}\mathbf{x}''$.

We obtain the distances r and s to the intersection point as follows: Let the 3D point $\mathcal{X}$ be given by

$$\boldsymbol{X} = r\mathbf{u} , \qquad \boldsymbol{X} = \mathbf{b} + sR^\mathsf{T}\mathbf{v} . \tag{13.163}$$

If the base vector $\mathbf{b}$ and the rotation matrix R are properly selected, the two scalars r and s should be positive. The geometric problem is mathematically equivalent to the determination of the distances to the end points of a given 3D line segment; see Fig. 12.29, p. 529. Therefore we obtain the values r and s, see (12.255), p. 528,

$$\begin{bmatrix} r \\ s \end{bmatrix} = \frac{1}{\det\left(\mathbf{b} \mid \mathbf{m} \mid \mathbf{u} \times R^\mathsf{T}\mathbf{v}\right)} \begin{bmatrix} \mathbf{m}^\mathsf{T} R^\mathsf{T} \mathbf{v} \\ \mathbf{m}^\mathsf{T} \mathbf{u} \end{bmatrix} , \tag{13.164}$$

using the binormal (see Fig. 13.12)

$$\mathbf{m} = (\mathbf{b} \times \mathbf{u}) \times \mathbf{b}\,. \tag{13.165}$$

The determinant in the denominator of (13.164) is positive if the three vectors form a right-handed system. The bilinear form $\mathbf{m}^\mathsf{T} R^\mathsf{T} \mathbf{v}$ is positive if the two vectors $\mathbf{u}$ and $\mathbf{v}$ are on the same side of the base line; otherwise, it is negative. Finally, the sign of $\mathbf{m}^\mathsf{T}\mathbf{u}$ is always positive. This allows a simple determination of the signs of r and s which only works if the two camera rays are not parallel to the base line.

Remark: We can use one of the expressions in (13.163) for triangulating the point $\mathcal{X}$. If the two rays do not intersect, this is an approximate method, since then the two expressions for $\boldsymbol{X}$ in (13.163) do not yield the same point. We will discuss methods for triangulation in Sect. 13.4, p. 596, where one method enforces the coplanarity of the two rays prior to ray intersection. ◇

Algorithm 20 selects the proper base direction and rotation matrix using a set of corresponding points. Only if the signs of the majority of the points are consistent will there be a proper base and rotation. Generally, one point would be sufficient for deciding on the correct signs. Taking a set prevents accidental errors if we choose a point close to infinity, where due to random noise the sign could be determined incorrectly. The algorithm also provides the code for the configuration (base vector $\mathbf{b}$ and rotation matrix W) and the normalized essential matrix consistent with the resulting base vector $\mathbf{b}$ and rotation matrix R.

Algorithm 20: Base direction and rotation from essential matrix;
$[\boldsymbol{b}, R, \text{type}]$=`b_and_R_from_E`$(\mathsf{E}, \{\mathbf{u}, \mathbf{v}\}_i)$
Input: essential matrix E, set $(\mathbf{u}, \mathbf{v})_i, i = 1, ...I$ of corresponding camera rays.
Output: base vector $\mathbf{b}$, rotation matrix R, code type $\in \{+\mathtt{W}, -\mathtt{W}, +\mathtt{W}', -\mathtt{W}'\}$ and normalized E.

Set matrix W;
SVD of E: $USV^\mathsf{T} =$ `svd`(E);
Enforce proper signs: $U = U\,|U|$, $V = V\,|V|$, $\mathsf{E} = U\,\mathrm{Diag}([1,1,0])\,V^\mathsf{T}$, $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]$;
for type $\in \{+\mathtt{W}, -\mathtt{W}, +\mathtt{W}', -\mathtt{W}'\}$ **do**
  case type $= +\mathtt{W}$ $\mathbf{b} = +\boldsymbol{u}_3$, $R = VWU^\mathsf{T}$;
  case type $= -\mathtt{W}$ $\mathbf{b} = -\boldsymbol{u}_3$, $R = VWU^\mathsf{T}$;
  case type $= +\mathtt{W}'$ $\mathbf{b} = +\boldsymbol{u}_3$, $R = VW^\mathsf{T}U^\mathsf{T}$;
  case type $= -\mathtt{W}'$ $\mathbf{b} = -\boldsymbol{u}_3$, $R = VW^\mathsf{T}U^\mathsf{T}$;
  skew symmetric matrix S_b;
  for *i=1,...,I* **do**
    Binormal: $\mathbf{m}_i = \mathsf{N}((\mathbf{b} \times \mathbf{u}_i) \times \mathbf{b})$;
    Sign for $\mathbf{v}_i$: $s_{s_i} = \mathrm{sign}(\det[\mathbf{b} \mid \mathbf{m}_i \mid \mathbf{u}_i \times R^\mathsf{T}\mathbf{v}_i])$;
    Sign for $\mathbf{u}_i$: $s_{r_i} = s_{s_i}\,\mathrm{sign}(\mathbf{m}_i^\mathsf{T} R^\mathsf{T}\mathbf{v}_i)$;
  end
  if (`mean`$(\boldsymbol{s}_s) > 0$ & `mean`$(\boldsymbol{s}_r) > 0$) **then** return **else** type $= -1$.
end

All solutions for the relative orientation discussed so far are suboptimal in a statistical sense, as they do not use any information about the uncertainty of the measured image coordinates. However, in general they lead to approximate values for the relative orientation parameters which are good enough for an optimal iterative solution to converge quickly.

13.3.4 Iterative Solution for Uncalibrated Cameras

Statistically optimal estimates for the relative orientation only can be achieved using iterative algorithms. As a side effect, they yield the covariance matrix of the estimated

parameters and estimate for the noise level, namely the estimated variance factor. We will discuss an iterative solution for the fundamental matrix using the norm and the determinant constraint.

The maximum likelihood solution for the fundamental matrix F from a set of $I \geq 7$ corresponding points $(\mathcal{X}', \mathcal{X}'')_i, i = 1, ..., I$, starts from the equivalent representations of the coplanarity condition

$$0 = \tilde{\mathbf{x}}'^{\mathsf{T}}_i \tilde{\mathsf{F}} \tilde{\mathbf{x}}''_i = \tilde{\mathbf{l}}''^{\mathsf{T}}_i \tilde{\mathbf{x}}''_i = \tilde{\mathbf{l}}'^{\mathsf{T}}_i \tilde{\mathbf{x}}'_i = (\tilde{\mathbf{x}}''^{\mathsf{T}}_i \otimes \tilde{\mathbf{x}}'^{\mathsf{T}}_i)\tilde{\mathbf{f}}\,, \tag{13.166}$$

with the epipolar lines

$$\tilde{\mathbf{l}}'_i = \tilde{\mathsf{F}} \tilde{\mathbf{x}}''_i\,, \qquad \text{and} \qquad \tilde{\mathbf{l}}''_i = \tilde{\mathsf{F}}^{\mathsf{T}} \tilde{\mathbf{x}}'_i\,. \tag{13.167}$$

The stochastical model assumes the observed corresponding points $(\mathbf{x}'_i, \mathbf{x}''_i)$ to have the joint covariance matrix

$$\mathbb{D}\left(\begin{bmatrix} \underline{\mathbf{x}}'_i \\ \underline{\mathbf{x}}''_i \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{\mathbf{x}'_i\mathbf{x}'_i} & \Sigma_{\mathbf{x}'_i\mathbf{x}''_i} \\ \Sigma_{\mathbf{x}''_i\mathbf{x}'_i} & \Sigma_{\mathbf{x}''_i\mathbf{x}''_i} \end{bmatrix}; \tag{13.168}$$

see the discussion in Sect. 10.6.3.1, p. 425 and exercise 21, p. 435.

13.3.4.1 The linearized Model for Estimating the Fundamental Matrix

The total differential of the coplanarity constraint (13.166) yields the linearized substitute model. We linearize at approximate values for all estimates indicated with ^{a}.

Here we use the minimal representations for the homogeneous image coordinates of the observed points:

$$\boldsymbol{x}'_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}'^a_i)\,\mathbf{x}'_i \quad \text{and} \quad \boldsymbol{x}''_{ri} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}''^a_i)\,\mathbf{x}''_i\,, \tag{13.169}$$

with

$$J_r(\mathbf{x}) = \mathsf{null}\left(\mathbf{x}^{\mathsf{T}}\right)\,. \tag{13.170}$$

Observe, the argument of the Jacobian $J_r^{\mathsf{T}}(\widehat{\mathbf{x}}'^a_i)$ is the approximate value $\widehat{\mathbf{x}}'^a_i$ for the final estimate $\widehat{\mathbf{x}}'_i$, and therefore different from the observed value $\mathbf{x}'_i$, except for the first iteration. Therefore (except for the first iteration) the reduced coordinates $\boldsymbol{x}'_{ri}$ are not zero, see the discussion on the iteration sequence in the Gauss–Helmert model in Sect. 4.8.2, p. 163 and the example in Sect. 10.6.2, p. 417.

We also use a minimal representation of the uncertain fundamental matrix for estimating the seven parameters $\boldsymbol{f}_r$,

$$\boldsymbol{f}_r = J_r(\mathbf{f})|^{\mathsf{T}}_{f=\widehat{f}^a}\,\mathbf{f} \quad \text{with} \quad J_r(\mathbf{f}) = \mathsf{null}\left(H^{\mathsf{T}}(\mathbf{f})\right) \quad \text{and} \quad H(\mathbf{f}) = [\mathbf{f}, \mathbf{f}^\circlearrowright]\,. \tag{13.171}$$

The matrix H results from linearizing the nonlinear constraints for the fundamental matrix, see (13.106), p. 572 and (13.110), p. 572.

With the assignments of the geometric entities to the elements within the estimation procedure

$$\boldsymbol{a}_i^{\mathsf{T}} := \left(\widehat{\mathbf{x}}''^{a\mathsf{T}}_i \otimes \widehat{\mathbf{x}}'^{a\mathsf{T}}_i\right) J_r(\widehat{\mathbf{f}}^a)\,, \tag{13.172}$$

$$\widehat{\Delta\boldsymbol{x}} := \widehat{\Delta\boldsymbol{f}}_r\,, \tag{13.173}$$

$$\boldsymbol{b}_i^{\mathsf{T}} := \left[\widehat{\mathbf{l}}'^{a\mathsf{T}}_i J_r(\widehat{\mathbf{x}}'^a_i) \mid \widehat{\mathbf{l}}''^{a\mathsf{T}}_i J_r(\widehat{\mathbf{x}}''^a_i)\right]\,, \tag{13.174}$$

$$\widehat{\Delta\boldsymbol{l}}_i := \begin{bmatrix} \Delta\widehat{\boldsymbol{x}}'_{ri} \\ \Delta\widehat{\boldsymbol{x}}'_{ri} \end{bmatrix}\,, \tag{13.175}$$

$$\boldsymbol{l}_i := \begin{bmatrix} J_r^{\mathsf{T}}(\widehat{\mathbf{x}}'^a_i)\,\mathbf{x}'_i \\ J_r^{\mathsf{T}}(\widehat{\mathbf{x}}''^a_i)\,\mathbf{x}''_i \end{bmatrix}\,, \tag{13.176}$$

$$\boldsymbol{c}_{g_i} := -\boldsymbol{g}(\widehat{\boldsymbol{x}}^a, \widehat{\boldsymbol{l}}^a_i) + B_i^{\mathsf{T}} \boldsymbol{l}_i\,, \tag{13.177}$$

and

$$\Sigma_{x'_{ri}x''_{ri}} = J_r^{\mathsf{T}}(\widehat{\mathbf{x}}_i'^a)\,\Sigma_{\mathbf{x}'_i\mathbf{x}''_i}\,J_r(\widehat{\mathbf{x}}_i''^a)\,, \tag{13.178}$$

the linearized Gauss–Helmert model reads

$$\boldsymbol{a}_i^{\mathsf{T}}\widehat{\Delta\boldsymbol{x}} + \boldsymbol{b}_i^{\mathsf{T}}\widehat{\Delta\boldsymbol{l}}_i = \boldsymbol{c}_{g_i}\,, \quad \Sigma_{l_il_i} := \begin{bmatrix} \Sigma_{x'_{ri}x'_{ri}} & \Sigma_{x'_{ri}x''_{ri}} \\ \Sigma_{x''_{ri}x'_{ri}} & \Sigma_{x''_{ri}x''_{ri}} \end{bmatrix}\,, \quad i = 1, ..., I\,. \tag{13.179}$$

13.3.4.2 The Normal Equations

The setup of the normal equations uses

$$A = [\boldsymbol{a}_i^{\mathsf{T}}]\,, \qquad B^{\mathsf{T}} = \mathrm{Diag}([\boldsymbol{b}_i^{\mathsf{T}}])\,, \qquad \Sigma_{ll} = \mathrm{Diag}(\{\Sigma_{l_il_i}\})\,, \qquad \boldsymbol{c}_g = [\boldsymbol{c}_{g_i}] \tag{13.180}$$

and leads to

$$\underbrace{A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}A}_{N}\,\widehat{\Delta\boldsymbol{x}} = A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}\,\boldsymbol{c}_g\,. \tag{13.181}$$

Using $\widehat{\Delta\boldsymbol{f}}_r := \widehat{\Delta\boldsymbol{x}}$, the improved estimate of fundamental matrix is

$$\widehat{\mathsf{F}} = \mathsf{N}(\widehat{\mathsf{F}}^a + \widehat{\Delta\mathsf{F}}) \quad \text{with} \quad \mathrm{vec}(\widehat{\Delta\mathsf{F}}) = \widehat{\Delta\mathbf{f}} = J_r(\widehat{\mathbf{f}}^a)\,\widehat{\Delta\boldsymbol{f}}_r\,, \tag{13.182}$$

where $\mathsf{N}(\cdot)$ with matrix argument enforces the Frobenius norm to be 1.

The normal equation matrix can be simplified if the correspondences are uncorrelated. We use the partitioning $A = [A_i^{\mathsf{T}}]$ from (13.99), p. 571 into a column of 1×9-vectors A_i^{T} and obtain

$$A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{xx}B)^{-1}A = \sum_{i=1}^{I} \boldsymbol{a}_i \left(\begin{bmatrix} \widehat{\mathbf{l}}_i'^{a\mathsf{T}} \mid \widehat{\mathbf{l}}_i''^{a\mathsf{T}} \end{bmatrix} \begin{bmatrix} \Sigma_{\mathbf{x}'_i\mathbf{x}'_i} & \Sigma_{\mathbf{x}'_i\mathbf{x}''_i} \\ \Sigma_{\mathbf{x}''_i\mathbf{x}'_i} & \Sigma_{\mathbf{x}''_i\mathbf{x}''_i} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{l}}_i'^a \\ \widehat{\mathbf{l}}_i''^a \end{bmatrix} \right)^{-1} \boldsymbol{a}_i^{\mathsf{T}}\,. \tag{13.183}$$

Observe, we assume all homogeneous image coordinate vectors to be spherically normalized and the covariance matrices $\Sigma_{\mathbf{x}'_i\mathbf{x}'_i}$, etc., refer to these spherically normalized image coordinates, see (10.19), p. 368.

A similar simplification can be used for the right-hand sides. Both simplifications are even more effective if corresponding coordinates $\mathbf{x}'_i$ and $\mathbf{x}''_i$ are independent.

Finally, we determine the covariance matrix of the estimated vector $\widehat{\boldsymbol{f}}$. The inverse of N in (13.181) immediately provides the theoretical covariance matrix of the reduced vector $\widehat{\boldsymbol{f}}_r$. With (13.171) similar to (10.28), p. 371, we have the covariance matrix of $\widehat{\mathbf{f}}$:

$$\Sigma_{\widehat{\mathbf{f}}^s\widehat{\mathbf{f}}^s} = J_r(\widehat{\mathbf{f}})\,\Sigma_{\widehat{f}_r\widehat{f}_r}\,J_r^{\mathsf{T}}(\widehat{\mathbf{f}})\,. \tag{13.184}$$

13.3.5 Iterative Solution for Calibrated Cameras

Estimation of the essential matrix starts from $I \geq 5$ corresponding points $(\mathcal{X}'_i, \mathcal{X}''_i)$ whose coordinates may be correlated. An optimal solution for the essential matrix can be achieved by maximum likelihood estimation. The solution is equivalent to a bundle adjustment for the two calibrated images when we fix one camera and a scale parameter, i.e., seven of the 12 exterior orientation parameters.

The iterative solution for the essential matrix yields expressions for the covariance matrix of the parameters, which can also be used in the case of a minimal solution. It is useful for obtaining approximate values when checking the consensus in a RANSAC procedure. We also discuss the iterative solution for the normal case in order to obtain insight into the theoretical accuracy of the calibrated image pair.

13.3.5.1 Iterative Solution for General Configurations

We will first give an iterative solution for the general parametrization of dependent images, namely the normalized base vector and the rotation matrix of the second image, all with general values. Thus we assume that approximate values for $\boldsymbol{B}$ with $|\boldsymbol{B}| = 1$ and for the rotation matrix R are given. We set up the constraints for the Gauss–Helmert model.

Using the essential matrix in the form

$$\mathsf{E} = \mathsf{S}(\mathbf{b})R^{\mathsf{T}}\,, \tag{13.185}$$

its estimation starts from the model

$$g_i(\tilde{\mathbf{x}}_i', \tilde{\mathbf{x}}_i'', \tilde{\mathbf{b}}, \tilde{R}) = \tilde{\mathbf{x}}_i{}'^{\mathsf{T}}\mathsf{S}(\tilde{\mathbf{b}})\tilde{R}^{\mathsf{T}}\tilde{\mathbf{x}}_i'' = 0\,, \quad i = 1, ..., I \geq 5\,, \tag{13.186}$$

with the spherically normalized left epipole $\mathbf{b}$ (omitting the ^{s} for convenience), see (13.43), p. 558. Starting from approximate values $\widehat{\mathbf{b}}^a$ for the base vector and $\widehat{R}^a$ for the rotation, we develop the linear Gauss–Helmert model. We apply the multiplicative update of the rotation matrix,

$$R = \exp(\mathsf{S}(\mathrm{d}\boldsymbol{p}))\,\widehat{R}^a \approx (I_3 + \mathsf{S}(\mathrm{d}\boldsymbol{p}))\,\widehat{R}^a\,, \tag{13.187}$$

using the differential rotation vector $\mathrm{d}\boldsymbol{p}$ and reduced homogeneous coordinates for the image points and the base vector.

The total differential of the essential matrix is

$$\begin{aligned}
\mathrm{d}\mathsf{E} &= \mathrm{d}\mathsf{S}(\mathbf{b})R^{\mathsf{T}} + \mathsf{S}(\mathbf{b})\,\mathrm{d}\left(R^{\mathsf{T}}\right) && (13.188)\\
&= \mathrm{d}\mathsf{S}(\mathbf{b})R^{\mathsf{T}} + \mathsf{S}(\mathbf{b})\,(\mathsf{S}(\mathrm{d}\boldsymbol{p})R)^{\mathsf{T}} && (13.189)\\
&= \mathrm{d}\mathsf{S}(\mathbf{b})R^{\mathsf{T}} + \mathsf{S}(\mathbf{b})R^{\mathsf{T}}\mathsf{S}^{\mathsf{T}}(\mathrm{d}\boldsymbol{p}) && (13.190)\\
&= \mathrm{d}\mathsf{S}(\mathbf{b})R^{\mathsf{T}} + \mathsf{E}\,\mathsf{S}^{\mathsf{T}}(\mathrm{d}\boldsymbol{p})\,. && (13.191)
\end{aligned}$$

We now write the constraints in the required multiple differential forms evaluated at the approximate values for the estimates using $\mathrm{d}\mathsf{S}(\mathbf{b}) = \mathsf{S}(\mathrm{d}\mathbf{b})$,

$$\begin{aligned}
0 = \mathrm{d}g_i(\mathbf{x}_i', \mathbf{x}_i'', \mathbf{b}, R) &+ \mathrm{d}\mathbf{x}_i{}'^{\mathsf{T}}\mathsf{E}\mathbf{x}_i''\\
&+ \mathbf{x}_i'^{\mathsf{T}}\mathsf{S}(\mathrm{d}\mathbf{b})R^{\mathsf{T}}\mathbf{x}_i''\\
&+ \mathbf{x}_i'^{\mathsf{T}}\mathsf{E}\mathsf{S}^{\mathsf{T}}(\mathrm{d}\mathbf{p})\mathbf{x}_i''\\
&+ \mathbf{x}_i'^{\mathsf{T}}\mathsf{E}\,\mathrm{d}\mathbf{x}_i''\,. \qquad (13.192)
\end{aligned}$$

Using the transposed epipolar line coordinates and the point of the second image $\mathbf{x}_i''$ rotated into the first camera system,

$$\mathbf{l}_i{}'^{\mathsf{T}} = \mathbf{x}_i{}''^{\mathsf{T}}\mathsf{E}^{\mathsf{T}} \qquad \mathbf{l}_i{}''^{\mathsf{T}} = \mathbf{x}_i{}'^{\mathsf{T}}\mathsf{E}\,, \qquad {}^1\mathbf{x}_i'' = R^{\mathsf{T}}\mathbf{x}_i''\,, \tag{13.193}$$

we can write (13.192) as

$$\mathbf{l}_i{}''^{\mathsf{T}}\mathsf{S}^{\mathsf{T}}(\mathrm{d}\mathbf{p})\mathbf{x}_i'' = \mathbf{l}_i{}''^{\mathsf{T}}\mathsf{S}(\mathbf{x}_i'')\mathrm{d}\boldsymbol{p} = (\mathsf{S}^{\mathsf{T}}(\mathbf{x}_i'')\mathbf{l}_i'')^{\mathsf{T}}\mathrm{d}\boldsymbol{p} = (\mathbf{l}_i'' \times \mathbf{x}_i'')^{\mathsf{T}}\mathrm{d}\boldsymbol{p}\,. \tag{13.194}$$

We therefore obtain the differential for the constraints, reordering the terms, first observations, then unknown parameters,

$$\begin{aligned}
0 = \mathrm{d}g_i(\mathbf{x}_i', \mathbf{x}_i'', \mathbf{b}, R) &+ \mathbf{l}_i{}''^{\mathsf{T}}\,\mathrm{d}\mathbf{x}_i''\\
&+ \mathbf{l}_i{}'^{\mathsf{T}}\,\mathrm{d}\mathbf{x}_i'\\
&+ ({}^1\mathbf{x}_i'' \times \mathbf{x}_i')^{\mathsf{T}}\,\mathrm{d}\mathbf{b}\\
&+ (\mathbf{l}_i'' \times \mathbf{x}_i'')^{\mathsf{T}}\,\mathrm{d}\boldsymbol{p}\,. \qquad (13.195)
\end{aligned}$$

After reducing the homogeneous vectors $\Delta\mathbf{x}'_i$, $\Delta\mathbf{x}''_i$ and $\Delta\mathbf{b}$ using the general form of the reduction, see (10.26), p. 370 in Sect. 10.2.2.1, p. 369, the linearized model now can be written as

$$\begin{aligned} 0 = g_i(\widehat{\mathbf{x}}^a_i, \widehat{\mathbf{x}}'^a_i, \widehat{\mathbf{b}}^a, \widehat{R}^{a\mathsf{T}}) &+ \widehat{\mathbf{l}}'^{a\mathsf{T}}_i J_r(\widehat{\boldsymbol{x}}'^a_i)\ \ \Delta\boldsymbol{x}'_{ri} \\ &+ \widehat{\mathbf{l}}''^{a\mathsf{T}}_i J_r(\widehat{\boldsymbol{x}}''^a_i)\ \ \Delta\boldsymbol{x}''_{ri} \\ &+ ({}^1\widehat{\mathbf{x}}''^a_i \times \widehat{\mathbf{x}}'^a_i)^\mathsf{T}\ J_r(\widehat{\mathbf{b}}^a)\ \ \Delta\boldsymbol{b}_r \\ &+ (\widehat{\mathbf{l}}''^a_i \times \widehat{\mathbf{x}}''^a_i)^\mathsf{T}\ \ \Delta\boldsymbol{p}\,. \end{aligned} \tag{13.196}$$

We are now prepared to set up the estimation model by assigning the necessary entities, the unknown parameters, the observations with their covariance matrix, and the Jacobians. The reduced coordinates $\Delta\boldsymbol{b}_r$ of the epipole and the rotation parameters $\Delta\boldsymbol{p}$ form the vector of unknown parameters, leading to the assignment for the unknown parameters,

$$\Delta\boldsymbol{x} := \begin{bmatrix} \Delta\boldsymbol{b}_r \\ \Delta\boldsymbol{p} \end{bmatrix}. \tag{13.197}$$

As in the case of the homography, the four reduced coordinates of the two corresponding points $(\mathcal{x}'_i, \mathcal{x}'')$ form the ith observational group,

$$\Delta\boldsymbol{l}_i := \begin{bmatrix} \Delta\boldsymbol{x}'_{ri} \\ \Delta\boldsymbol{x}''_{ri} \end{bmatrix}, \tag{13.198}$$

in order to allow for correlated points. Therefore the Jacobians for the model are

$$\underset{1\times5}{\boldsymbol{a}^\mathsf{T}_i} := [({}^1\widehat{\mathbf{x}}''^a_i \times \widehat{\mathbf{x}}'^a_i)^\mathsf{T}\ J_r(\widehat{\mathbf{b}}^a) \mid (\widehat{\mathbf{l}}''^a_i \times \widehat{\mathbf{x}}''^a_i)^\mathsf{T}] \tag{13.199}$$

and

$$\underset{1\times4}{\boldsymbol{b}^\mathsf{T}_i} := [\widehat{\mathbf{l}}'^{a\mathsf{T}}_i J_r(\boldsymbol{x}'^a_i) \mid \widehat{\mathbf{l}}''^{a\mathsf{T}}_i J_r(\boldsymbol{x}''^a_i)]\,. \tag{13.200}$$

The linearized Gauss–Helmert model reads

$$\boldsymbol{a}^\mathsf{T}_i \widehat{\Delta\boldsymbol{x}} + \boldsymbol{b}^\mathsf{T}_i \widehat{\Delta\boldsymbol{l}}_i = \boldsymbol{c}_{g_i}\,, \quad \mathbb{D}(\widehat{\Delta\boldsymbol{l}}_i) = \begin{bmatrix} \Sigma_{x'_{ri}x'_{ri}} & \Sigma_{x'_{ri}x''_{ri}} \\ \Sigma_{x''_{ri}x'_{ri}} & \Sigma_{x''_{ri}x''_{ri}} \end{bmatrix} \qquad i = 1, ..., I\,, \tag{13.201}$$

with

$$\widehat{\Delta\boldsymbol{l}}_i := \begin{bmatrix} \widehat{\Delta\boldsymbol{x}'_{ri}} \\ \widehat{\Delta\boldsymbol{x}''_{ri}} \end{bmatrix} \tag{13.202}$$

$$\boldsymbol{c}_{g_i} := -\boldsymbol{g}(\widehat{\boldsymbol{x}}^a, \widehat{\boldsymbol{l}}^a_i) + \boldsymbol{b}^\mathsf{T}_i \widehat{\boldsymbol{l}}_i \tag{13.203}$$

$$\boldsymbol{l}_i := \begin{bmatrix} J^\mathsf{T}_r(\widehat{\mathbf{x}}'^a_i)\mathbf{x}'_i \\ J^\mathsf{T}_r(\widehat{\mathbf{x}}''^a_i)\mathbf{x}''_i \end{bmatrix} \tag{13.204}$$

and

$$\Sigma_{x'_{ri}x''_{ri}} = J^\mathsf{T}_r(\widehat{\mathbf{x}}'^a_i)\, \Sigma_{\mathbf{x}'_i\mathbf{x}''_i}\, J_r(\widehat{\mathbf{x}}''^a_i)\,. \tag{13.205}$$

The update for the five parameters results from the normal equations

$$A^\mathsf{T}(B^\mathsf{T}\Sigma_{ll}B)^{-1}A\,\widehat{\Delta\boldsymbol{x}} = A^\mathsf{T}(B^\mathsf{T}\Sigma_{ll}B)^{-1}\,\boldsymbol{c}_g\,. \tag{13.206}$$

The block structure of the covariance matrix Σ_{ll} can again be used to simplify the normal equation matrix and the right-hand sides, see (13.183), p. 585. We now obtain the updated estimated base vector and the updated rotation matrix using

$$\begin{bmatrix} \Delta\widehat{\boldsymbol{b}}_r \\ \Delta\widehat{\boldsymbol{p}} \end{bmatrix} := \widehat{\Delta\boldsymbol{x}}\,, \tag{13.207}$$

and obtain

$$\widehat{\mathbf{b}}^{(\nu+1)} = \mathsf{N}\left(\widehat{\mathbf{b}}^{(\nu)} + J_r\left(\widehat{\mathbf{b}}^{(\nu)}\right)\widehat{\Delta \boldsymbol{b}}_r\right) \quad \text{and} \quad \widehat{R}^{(\nu+1)} = \exp\left(S\left(\widehat{\boldsymbol{p}}^{(\nu)}\right)\right)\widehat{R}^{(\nu)}. \tag{13.208}$$

Checking the Consistency of Correspondences. The covariance matrix of the five parameters of the relative orientation derived from a minimum of five correspondences $(\mathcal{x}_i', \mathcal{x}_i''), i = 1, ..., 5$, can be used for checking other correspondences $\{\mathcal{x}_j', \mathcal{x}_j''\}, j \neq 1, ..., 5$, for consistency in a RANSAC procedure, similarly to (13.21), p. 555 (see Raguram et al., 2009). Here this procedure reduces to checking the constraint

$$g_j = \mathbf{x}_j'^{\mathsf{T}} \mathsf{S}_{\widehat{b}} \widehat{R} \mathbf{x}_j'', \tag{13.209}$$

which under null hypothesis should be zero. For this we use the covariance matrix $\Sigma_{l_j l_j}$ of the four observed reduced image coordinates $\boldsymbol{x}'_{r_j}$ and $\boldsymbol{x}''_{r_j}$ reflecting the uncertainty of the correspondence $(\mathcal{x}_j', \mathcal{x}_j'')$. With the covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ of the five estimated unknown parameters, the variance of the residual g_j of the constraint is

$$\sigma_{g_j}^2 = \boldsymbol{a}_j^{\mathsf{T}} \Sigma_{\widehat{x}\widehat{x}} \boldsymbol{a}_j + \boldsymbol{b}_j^{\mathsf{T}} \Sigma_{l_j l_j} \boldsymbol{b}_j, \tag{13.210}$$

with the Jacobians $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ from (13.199) and (13.200), p. 587, evaluated at the fitted parameters and the given observations.

The covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ of the parameters, which is required in (13.210), can be calculated directly if we have the final result from a direct solution. Then we can determine the Jacobians A and B,

$$\underset{5\times 5}{A} = \begin{bmatrix} \boldsymbol{a}_1^{\mathsf{T}} \\ \dots \\ \boldsymbol{a}_5^{\mathsf{T}} \end{bmatrix}, \qquad \underset{5\times 20}{B^{\mathsf{T}}} = \mathrm{Diag}([\boldsymbol{b}_i^{\mathsf{T}}]), \; i = 1, ..., 5, \tag{13.211}$$

covariance matrix for the five-point solution

using the coefficients $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ from (13.199) and (13.200) and, as A is invertible in a general configuration, arrive at the theoretical covariance matrix for the parameters of the minimal configuration,

$$\Sigma_{\widehat{x}\widehat{x}} = (A^{\mathsf{T}}(B^{\mathsf{T}}\Sigma_{ll}B)^{-1}A)^{-1} = A^{-1}B^{\mathsf{T}}\Sigma_{ll}BA^{-\mathsf{T}}. \tag{13.212}$$

Critical Configurations The iterative solution for the relative orientation of images of calibrated cameras does not work if the object points *and* the two projection centres lie on a critical surface. This is an orthogonal hyperboloid of one sheet or one of its degeneracies. It includes the case where the scene is planar and the base line is perpendicular to the plane (Horn, 1990). A critical situation can be identified a posteriori using the covariance matrix of the estimated parameters, an argument we used in the context of the spatial resection (Sect. 12.2.4.1, p. 516) and that we will revisit when discussing the relative orientation of images of two calibrated cameras (see Sect. 13.3.6.2, p. 593).

13.3.6 Iterative Relative Orientation Close to the Normal Case of the Image Pair

sideward motion

This section provides the iterative solution for the relative orientation of two images of calibrated cameras for a simplified scenario, namely for two cameras in a sideward motion, where the basis is perpendicular to the viewing direction. This is similar to the classical setup of two consecutive aerial images, or to the observation of a facade from two neighbouring positions. The linearized model allows an algebraic investigation of the achievable precision as a function of relevant design parameters, such as measurement precision, principal distance, length of base line, and distance from the object, and an analysis of the

sensitivity of the solution w.r.t. outliers, see Sect. 13.3.6.1. We use the same procedure for analysing the precision of forward motion, where the basis is in the viewing direction, which is close to the setup of stereo systems used in cars for driver assistance, see Sect. 13.3.6.2.

forward motion

We will give the solution using the classical photogrammetric parametrization of dependent images, namely via the two elements B_Y and B_Z of the base vector $\boldsymbol{B}$ and the three parameters $\boldsymbol{r} = [\omega, \phi, \kappa]^\mathsf{T}$ of the rotation matrix, fixing B_X a priori. We assume the approximate values of all five elements to be 0, so $\boldsymbol{B}^{(0)} = [B_X, 0, 0]^\mathsf{T}$, $R^{(0)} = I_3$, and $c' = c'' = c$. The initial geometry of the stereo pair is the normal case, approximating the final geometry. We may therefore start with image coordinates $x' \doteq {}^i x'$, etc., related to the principal point.

The linearized observation equations can be determined from

$$ {}^c\mathbf{x}'^\mathsf{T} S_B R^\mathsf{T} {}^c\mathbf{x}'' = 0\,. \tag{13.213}$$

When using differential matrices, especially with $R^\mathsf{T} = I_3 + \mathrm{d}R^\mathsf{T} \approx I_3 + S(\mathrm{d}\boldsymbol{r})$, see (12.9), p. 467,

$$ [x', y', c] \begin{bmatrix} 0 & -\mathrm{d}B_Z & \mathrm{d}B_Y \\ \mathrm{d}B_Z & 0 & -B_X \\ -\mathrm{d}B_Y & B_X & 0 \end{bmatrix} \begin{bmatrix} 1 & -\mathrm{d}\kappa & \mathrm{d}\phi \\ \mathrm{d}\kappa & 1 & -\mathrm{d}\omega \\ -\mathrm{d}\phi & \mathrm{d}\omega & 1 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ c \end{bmatrix} = 0\,. \tag{13.214}$$

After multiplying out, omitting second-order terms, and setting $y' = y''$ where appropriate, we obtain the following linearized observation equation for each corresponding point pair,

$$ p_y + v_{p_y} = -\frac{p_x}{B_X}\mathrm{d}B_Y + \frac{p_x}{B_X}\frac{y'}{c}\mathrm{d}B_Z - \left(c + \frac{y'y''}{c}\right)\mathrm{d}\omega + \frac{y'x''}{c}\mathrm{d}\phi + x''\mathrm{d}\kappa\,, \tag{13.215}$$

with the x'- and y'-parallaxes

$$ p_x = x'' - x', \qquad p_y = y'' - y'\,. \tag{13.216}$$

Collecting all observation equations for $p_{y_n}, n = 1, ..., N$, we obtain

$$ \Delta\boldsymbol{l} + \boldsymbol{v} = A\Delta\boldsymbol{x}\,, \tag{13.217}$$

with

$$ \Delta\boldsymbol{l} := \begin{bmatrix} p_{y_1} \\ .. \\ p_{y_n} \\ .. \\ p_{y_N} \end{bmatrix}, \quad \boldsymbol{v} := \begin{bmatrix} v_{p_{y_1}} \\ .. \\ v_{p_{y_n}} \\ .. \\ v_{p_{y_1}} \end{bmatrix}, \quad \Delta\boldsymbol{x} := \begin{bmatrix} \mathrm{d}B_Y \\ \mathrm{d}B_Z \\ \mathrm{d}\omega \\ \mathrm{d}\phi \\ \mathrm{d}\kappa \end{bmatrix} \tag{13.218}$$

– specifically the corrections $v_{p_{y_n}}$ of the nth y-parallaxes – and

$$ A := \begin{bmatrix} -\frac{p_{x_1}}{B_X} & +\frac{p_{x_1}}{B_X}\frac{y'_1}{c} & -\left(c + \frac{y'_1 y''_1}{c}\right) & +\frac{y'_1 x''_1}{c} & +x''_1 \\ .. & .. & .. & .. & .. \\ -\frac{p_{x_n}}{B_X} & +\frac{p_{x_n}}{B_X}\frac{y'_n}{c} & -\left(c + \frac{y'_n y''_n}{c}\right) & +\frac{y'_n x''_n}{c} & +x''_n \\ .. & .. & .. & .. & .. \\ -\frac{p_{x_N}}{B_X} & +\frac{p_{x_N}}{B_X}\frac{y'_N}{c} & -\left(c + \frac{y'_N y''_N}{c}\right) & +\frac{y'_N x''_N}{c} & +x''_N \end{bmatrix}. \tag{13.219}$$

Assuming independent measurements with

$$ \Sigma_{ll} = \mathrm{Diag}([\sigma^2_{p_{y_n}}])\,, \tag{13.220}$$

we obtain the normal equations

$$A^{\mathsf{T}}\Sigma_{ll}^{-1}A\,\Delta\boldsymbol{x} = A^{\mathsf{T}}\Sigma_{ll}^{-1}\Delta\boldsymbol{l}\,, \tag{13.221}$$

which can be solved for corrections $\Delta\boldsymbol{x}$ of the unknown parameters $\boldsymbol{x}$, initiating an iteration scheme.

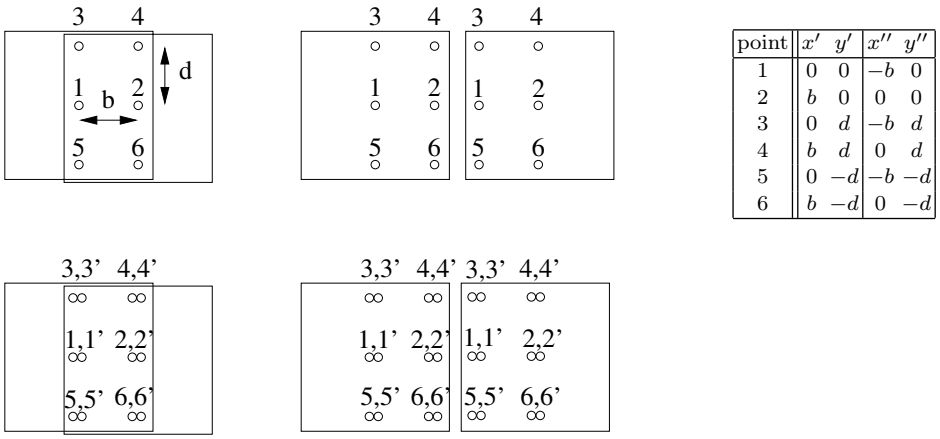
A similar derivation can be done for the case of independent images.

13.3.6.1 Quality of Ideal Sideward Motion

The quality of the relative orientation will be given for standard configurations. We will give the theoretical precision and reliability, especially the lower bound for detectable errors, see (4.6.2), p. 117, and Sect. 4.6.3, p. 122. We start with the normal case of the image pair, corresponding to sideward motion.

We assume 60% overlap (Fig. 13.13). For a standard aerial image size of 10 000 × 10 000 pixels we therefore have the base length $b_x = 0.4 \times 10\,000$ pixel $= 4\,000$ pixel at image scale. We assume the points are only in the rectangular area of size $b \times 2d$, leading to $p_x = b = b_x$ for all points. The principal distance is c, measured in pixels. The base length in object space is $B_X = b_x \times S$, where $S = H_g/c$ is the scale number in units m/pixel (equivalent to the ground sampling distance), and H_g is the flying height above the ground.

ground sampling distance H_g/c



point	x'	y'	x''	y''
1	0	0	$-b$	0
2	b	0	0	0
3	0	d	$-b$	d
4	b	d	0	d
5	0	$-d$	$-b$	$-d$
6	b	$-d$	0	$-d$

Fig. 13.13 Classical relative orientation of two images with points in Gruber positions. **Top row:** single points, **bottom row:** double points. Stereo images. **Left column:** shown overlapped, **right column:** shown separately. Double points ∞ practically are close to each other; in our simulation they are assumed to be identical

We first assume that the y-parallax is measured at six corresponding points in the configuration proposed by von Gruber for relative orientation of photogrammetric analog instruments (von Gruber, 1938; McGlone, 2013). These points are often called Gruber points. Their coordinates are given in the table on the right of Fig. 13.13. If we observe the y'-parallaxes with the same precision, $\sigma_{p_y} = \sqrt{2}\sigma_{y'}$, we obtain the coefficient matrix

$$A = \begin{bmatrix} \frac{b}{B_X} & 0 & -c & 0 & -b \\ \frac{b}{B_X} & 0 & -c & 0 & 0 \\ \frac{b}{B_X} & -\frac{d\,b}{c\,B_X} & -c+\frac{d^2}{c} & -\frac{d\,b}{c} & -b \\ \frac{b}{B_X} & -\frac{d\,b}{c\,B_X} & -c+\frac{d^2}{c} & 0 & 0 \\ \frac{b}{B_X} & \frac{d\,b}{c\,B_X} & -c+\frac{d^2}{c} & \frac{d\,b}{c} & -b \\ \frac{b}{B_X} & \frac{d\,b}{c\,B_X} & -c+\frac{d^2}{c} & 0 & 0 \end{bmatrix} \tag{13.222}$$

and the covariance matrix $\Sigma_{\widehat{x}\widehat{x}} = (A^\mathsf{T}\Sigma_{ll}^{-1}A)^{-1}$ of the parameters

$$\sigma_{p_y}^2 \begin{bmatrix} \frac{1}{12}\frac{{B_X}^2\left(9\,c^4+8\,d^4+12\,d^2c^2\right)}{b^2d^4} & 0 & \frac{1}{4}\frac{\left(3\,c^2+2\,d^2\right)B_X\,c}{bd^4} & 0 & -\frac{1}{3}\frac{B_X}{b^2} \\ 0 & \frac{1}{2}\frac{{B_X}^2c^2}{b^2d^2} & 0 & \frac{1}{2}\frac{B_X\,c^2}{b^2d^2} & 0 \\ \frac{1}{4}\frac{\left(3\,c^2+2\,d^2\right)B_X\,c}{bd^4} & 0 & \frac{3}{4}\frac{c^2}{d^4} & 0 & 0 \\ 0 & \frac{1}{2}\frac{B_X\,c^2}{b^2d^2} & 0 & \frac{c^2}{b^2d^2} & 0 \\ -\frac{1}{3}\frac{B_X}{b^2} & 0 & 0 & 0 & \frac{2}{3}\frac{1}{b^2} \end{bmatrix}. \tag{13.223}$$

We therefore have the following *standard deviations* for the orientation parameters:

$$\sigma_{B_Y} = \frac{1}{\sqrt{6}}\frac{H_g}{c}\frac{\sqrt{9c^4+8d^4+12d^2c^2}}{d^2}\,\sigma_{y'} \tag{13.224}$$

$$\sigma_{B_Z} = \frac{H_g}{c}\frac{c}{d}\,\sigma_{y'} = \frac{H_g}{d}\,\sigma_{y'} \tag{13.225}$$

$$\sigma_\omega = \sqrt{\frac{3}{2}}\frac{c}{d^2}\,\sigma_{y'} \tag{13.226}$$

$$\sigma_\phi = \sqrt{2}\frac{c}{bd}\,\sigma_{y'} \tag{13.227}$$

$$\sigma_\kappa = \frac{2}{\sqrt{3}}\frac{1}{b}\,\sigma_{y'} \tag{13.228}$$

Examination of the results shows:

- The standard deviations depend directly on the measuring precision $\sigma_{y'} = \sigma_{p_y}/\sqrt{2}$.
- The uncertainty of the base components B_Y and B_Z increases with the scale number $S = H_g/c$.
- The precision of the angles ω and ϕ highly depends on the extension d of the rectangular overlapping area (see Fig. 13.13). Since the x-coordinates of the points (differentially) do not have an influence on the parallax, see (13.215), the standard deviation σ_ω also does not depend on the basis b.
- If $d = b$, and the full area of the model is exploited, all standard deviations decrease with the length of the base b.
- If the basis is zero, the rotation angles still can be determined. Exercise 13.5

To analyse the *reliability*, and specifically the detectability of gross errors in the observations, see Sect. 4.6.4.1, p. 125, we need the covariance matrix of the residuals $\widehat{v} \doteq v_{p_y}$ of the parallaxes p_y. When six points are measured, this matrix is

$$\Sigma_{\widehat{v}\widehat{v}} = \Sigma^{(6)}_{\widehat{v}\widehat{v}} = \sigma^2_{p_y}(I_6 - A(A^\mathsf{T}A)^{-1}A^\mathsf{T}) = \frac{\sigma^2_{p_y}}{12}\left[\begin{array}{cc|cc|cc} +4 & -4 & -2 & +2 & -2 & +2 \\ -4 & +4 & +2 & -2 & +2 & -2 \\ \hline -2 & +2 & +1 & -1 & +1 & -1 \\ +2 & -2 & -1 & +1 & -1 & +1 \\ \hline -2 & +2 & +1 & -1 & +1 & -1 \\ +2 & -2 & -1 & +1 & -1 & +1 \end{array}\right], \tag{13.229}$$

independent of the principal distance c, the base length b, and the extension of the model area $2d$. The matrix has rank 1 as the redundancy of the estimation is $R = N - U = 6 - 5 = 1$, see (4.60), p. 87.

The testability of the observations p_y can be characterized using the redundancy numbers r_i, which indicate how the redundancy, here equal to 1, is distributed over the observations, here the y-parallaxes. They are the diagonal elements of $\Sigma_{\widehat{v}\widehat{v}}$, except for the factor $\sigma^2_{p_y}$, see (4.69), p. 88. For the six points, we have

$$r_1 = r_2 = \frac{1}{3} \qquad r_3 = r_4 = r_5 = r_6 = \frac{1}{12}\,. \tag{13.230}$$

Because $\Delta\widehat{v}_i = -r_i\Delta l_i$, after an adjustment we only see a small fraction of original errors Δl_i in the residual parallaxes $\widehat{v}_i$. The minimum size $\nabla_0 l_i$ of detectable gross errors, $\delta_0(\sqrt{1/r_i})\sigma_{p_y}$, with $\delta_0 = 4.13$, see Sect. 4.6.4.1, p. 125, is

$$\nabla_0 l_1 = \nabla_0 l_2 = 7.2\ \sigma_{p_y} \qquad \nabla_0 l_3 = \nabla_0 l_4 = \nabla_0 l_5 = \nabla_0 l_6 = 14.3\ \sigma_{p_y}\,. \tag{13.231}$$

Thus gross errors in the y-parallaxes must be quite large compared to the standard deviation σ_{p_y} of the parallaxes in order to be detectable.

For that reason it is better to measure pairs of points, with the points in each pair selected *close to each other*. When measuring such double points, the design matrix A for the second group is almost the same as for the first group. So, from

$$\Sigma^{(12)}_{\widehat{v}\widehat{v}} = \sigma^2_{p_y}\left\{\begin{bmatrix} I_6 & \mathbf{0} \\ \mathbf{0} & I_6 \end{bmatrix} - \begin{bmatrix} A \\ A \end{bmatrix}\left[2\,A^\mathsf{T}A\right]^{-1}[A^\mathsf{T}\ A^\mathsf{T}]\right\} \tag{13.232}$$

we obtain the redundancy numbers,

$$r_n = \frac{2}{3} \quad \text{for} \quad n = 1, 1', 2, 2' \qquad r_n = \frac{7}{12} \quad \text{for} \quad n = 3, 3', 4, 4', 5, 5', 6, 6'\,. \tag{13.233}$$

Measuring double points results in a much more reliable situation: in all corresponding points more than half of the magnitude of outliers is visible. This is confirmed by the minimum size of detectable errors, which in all cases is at least $5.4\ \sigma_{p_y}$. When measuring double points, the precision of the orientation parameters increases by a factor of $\sqrt{2}$, thus

$$\Sigma^{(12)}_{\widehat{x}\widehat{x}} = \frac{1}{2}\Sigma^{(6)}_{\widehat{x}\widehat{x}}\,. \tag{13.234}$$

If automatically selected key points are used for relative orientation, a large number of corresponding points usually is obtained. Due to the high redundancy, the redundancy numbers then are all close to 1, indicating that *individual* outliers can easily be detected. If we have to face *groups of outliers*, the sensitivity analysis needs to refer to groups of observations, see Sect. 4.6.4.2, p. 128, which only is practicable if we have a hypothesis about the group of outlying observations. However, this type of sensitivity analysis will be very useful for coordinates of control points within block adjustment, see Sect. 15.3.5, p. 670.

13.3.6.2 Quality of Ideal Forward Motion

In a similar manner, we can analyse the quality of the estimated relative orientation for the case of forward motion. Only the parameters of the base line are different. We assume the forward motion to be ideal, i.e., $\mathbf{b}'^{(0)} = [0, 0, B_Z]^\mathsf{T}$. This is the idealized configuration of two consecutive images taken with a camera from a moving car when looking in the direction of the motion.

The residual of the constraint from (13.213), p. 589 is scaled such that we obtain the linearized model

$$l_i + v_i = +\frac{c}{s}\frac{p_{y_i}}{B_Z}\mathrm{d}B_X - \frac{c}{s}\frac{p_{x_i}}{B_Z}\mathrm{d}B_Y - \frac{c}{s}x'_i\mathrm{d}\omega - \frac{c}{s}y'_i\mathrm{d}\phi + \frac{x'_i x''_i + y'_i y''_i}{s}\mathrm{d}\kappa\,, \tag{13.235}$$

with

$$l_i := \frac{x'_i y''_i - x''_i y'_i}{s}\,, \quad s = \sqrt{x'^2_i + y'^2_i + x''^2_i + y''^2_i}\,, \quad p_{x_i} = x''_i - x'_i\,, \quad p_{y_i} = y''_i - y'_i\,,$$

with the variance

$$\sigma^2_{l_i} = \sigma^2_{x'}\,. \tag{13.236}$$

Observe, as the epipoles are in the centre of the image, epipolar lines are radial lines from the origin, and only tangential parallaxes l_i can be observed. We normalized the parallaxes such that the observation has the standard deviation of the measured image points. Points at or close to the epipoles cannot be used, as they lead to indefinite coefficients.

For an analysis of the theoretical precision in a first step we therefore assume eight points on a planar square at a distance Z in front of both cameras, see Fig. 13.14.

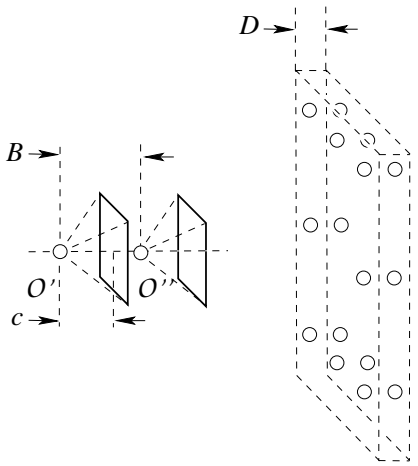


Fig. 13.14 Ideal forward motion

The corresponding design matrix A turns out to have rank 3: we encounter a degenerate geometric configuration for the relative orientation of images with calibrated cameras (see Horn, 1990): The scene points lie on a plane which is perpendicular to the basis. Therefore only three of the five parameters of the relative orientation can be determined. The reason is that a small rotation ω around the X-axis leads to the same change in the image coordinates as a small change in the direction B_Y of the base line, an effect which also holds for the two parameters ϕ and B_X.

Assuming the scene to have double the number of points, see Fig. 13.14, one at distance Z, one at distance $Z + D$, resolves the singularity. The expressions for the resulting theoretical standard deviations are somewhat convoluted, but can be written as

$$\sigma_{\widehat{B}_X} = \sigma_{\widehat{B}_Y} = f(B,Z,D)\,\frac{1}{cD}\,\sigma_{x'}\,, \quad \sigma_{\widehat{\omega}} = \sigma_{\widehat{\phi}} = g(B,Z,D)\,\frac{1}{cD}\,\sigma_{x'}\,, \tag{13.237}$$

$$\sigma_{\widehat{\kappa}} = h(B,Z,D)\,\frac{1}{c}\,\sigma_{x'}\,, \tag{13.238}$$

where f, g and h are bounded positive functions of the base length B, the distance Z, and the depth D of the scene. The correlations between the parameters are

$$\rho_{\widehat{B}_X\widehat{\phi}} = -\rho_{\widehat{B}_Y\widehat{\omega}} = 1 - k(B,Z,D)\,D^2\,, \tag{13.239}$$

with some bounded positive function $k(B,Z,D)$.

The theoretical standard deviations increase inversely with the depth D, and correlations approach 100% in case D approaches 0. This confirms the degeneracy of the configuration with a planar scene. An experiment with the five-point algorithm shows that none of the resulting essential matrices fulfils the property of having two identical large singular values and one singular value equal to 0, since the action matrix A_f has identical eigenvalues, which hinders a definite determination of the corresponding eigenvectors, thus of the parameters (x,y,z) in (13.144), p. 577.

13.3.7 Projection Matrices from Fundamental Matrix and Essential Matrix

After having determined the relative orientation, i.e., estimated the fundamental or the essential matrix, we need to derive a pair of projection matrices for the two cameras which is consistent with the estimated matrix, F or E, in order to be able to derive the coordinates of scene points by triangulation.

13.3.7.1 Projection Matrices from F

Given the fundamental matrix F of an image pair, we can determine projection matrices P' and P'' which are consistent with F; i.e., the fundamental matrix derived from P' and P'', e.g., using (13.11), p. 554, will be identical to F.

If we fix the first camera to be $\mathsf{P}'_d = [I_3 \mid \mathbf{0}]$, indexed with d as we adopt the case of dependent images, we are left with four degrees of freedom, since the fundamental matrix has only seven degrees of freedom, compared to the second projection matrix P''_d, which has 11 degrees of freedom. Therefore the solution for P'' is not unique but depends on four parameters. Unfortunately, there is no simple geometric interpretation of these parameters as in the case of calibrated cameras, discussed below.

We have a classical solution (Hartley and Zisserman, 2000, Result 9.15) for the choice of two projection matrices,

$$\mathsf{P}'_d = [I_3|\mathbf{0}] \qquad \mathsf{P}''_d = [\mathsf{S}_{e''}\mathsf{F}^\mathsf{T} + \mathbf{e}''\boldsymbol{D}^\mathsf{T}|\alpha\mathbf{e}''] = [\mathsf{A}''|\mathbf{a}'']\,, \tag{13.240}$$

which are a four-parameter family of valid pairs of projection matrices depending on $\alpha \neq 0$ and arbitrary $\boldsymbol{D}$. Here, $\boldsymbol{e}''$ is the epipole in the second image.

Proof: With these projection matrices we obtain for the coplanarity constraint applied to an arbitrary point $\mathbf{X}^\mathsf{T} = [\boldsymbol{X}_0^\mathsf{T}, X_h]$, the image points $\mathbf{x}' = \mathsf{P}'_d\mathbf{X} = \boldsymbol{X}_0$ and $\mathbf{x}'' = \mathsf{P}''_d\mathbf{X}$, and therefore the coplanarity constraint

$$\begin{aligned}\mathbf{x}'^\mathsf{T}\,\mathsf{F}\,\mathbf{x}'' &= \boldsymbol{X}_0^\mathsf{T}\mathsf{F}\,\left((\mathsf{S}_{e''}\mathsf{F}^\mathsf{T} + \mathbf{e}''\boldsymbol{D}^\mathsf{T})\boldsymbol{X}_0 + \alpha\mathbf{e}''X_h\right)\\ &= \boldsymbol{X}_0^\mathsf{T}\mathsf{F}\,\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}\boldsymbol{X}_0 + \boldsymbol{X}_0^\mathsf{T}\mathsf{F}\mathbf{e}''\boldsymbol{D}^\mathsf{T}\boldsymbol{X}_0 + \alpha\boldsymbol{X}_0^\mathsf{T}\mathsf{F}\mathbf{e}''X_h = 0\,.\end{aligned}$$

The first expression vanishes since $\mathsf{F}\,\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}$ is skew symmetric, say $\mathsf{S}_{c'}$, and $\boldsymbol{X}_0^\mathsf{T}\mathsf{S}_{c'}\boldsymbol{X}_0 = \boldsymbol{X}_0\cdot(\mathbf{c}'\times\boldsymbol{X}_0) = 0$ for any $\boldsymbol{X}_0$. The other two expressions vanish as $\mathsf{F}\mathbf{e}'' = \mathbf{0}$. ◇

The parameter α fixes the distance of the second projection centre from the origin and thus can be chosen arbitrarily from $\mathbb{R} \setminus \{0\}$.

If $\boldsymbol{D} = \mathbf{0}$, the left 3×3 matrix A of the projection matrix is singular, and the projection centre therefore lies at infinity. Although this is not a disadvantage in theory, as the resulting 3D model must be projectively transformed based on 3D points in object space, it might be undesirable in practice.

The free vector $\boldsymbol{D}$ can be chosen such that the left 3×3 matrix A of P''_d is close to a rotation matrix, as opposed to $\boldsymbol{D} = \mathbf{0}$, where $|A| = 0$. This can easily be achieved by inspecting the SVD of $\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}$,

$$\mathsf{S}_{e''}\mathsf{F}^\mathsf{T} = U\Lambda V^\mathsf{T} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\mathsf{T} + \lambda_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\mathsf{T} + \lambda_3 \boldsymbol{u}_3 \boldsymbol{v}_3^\mathsf{T} \,. \tag{13.241}$$

We know $\lambda_3 = 0$. However, the last dyad is $\boldsymbol{u}_3 \boldsymbol{v}_3^\mathsf{T} = \boldsymbol{e}'' \boldsymbol{e}'^\mathsf{T}$ as the left eigenvector of $\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}$ is $\mathbf{e}''$ and its right eigenvector is $\mathbf{e}'$. We therefore choose

$$\boldsymbol{D} = \beta \mathbf{e}' \tag{13.242}$$

to guarantee a regular matrix A and determine β such that the singular value λ_3 of A belonging to the dyad $\mathbf{e}''\boldsymbol{D}^\mathsf{T} = \beta \mathbf{e}'' \mathbf{e}'^\mathsf{T}$ lies between the singular values λ_1 and λ_2 of $\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}$. From the requirement $\frac{1}{2}||\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}|| = \frac{1}{2}\sqrt{\lambda_1^2 + \lambda_2^2} = ||\mathbf{e}''\boldsymbol{D}^\mathsf{T}|| = \beta||\mathbf{e}''\mathbf{e}'^\mathsf{T}||$, we obtain

$$\beta = \frac{||\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}||}{2||\mathbf{e}''\mathbf{e}'^\mathsf{T}||} \,. \tag{13.243}$$

Then the matrix

$$\mathsf{A} = \mathsf{S}_{e''}\mathsf{F}^\mathsf{T} + \mathbf{e}''\boldsymbol{D}^\mathsf{T} = \mathsf{S}_{e''}\mathsf{F}^\mathsf{T} + \beta \mathbf{e}''\mathbf{e}'^\mathsf{T} \tag{13.244}$$

is regular with $\lambda_1(\mathsf{A}) = \lambda_1$, $\lambda_2(\mathsf{A}) = \frac{1}{2}\sqrt{\lambda_1^2 + \lambda_2^2}$ and $\lambda_3(\mathsf{A}) = \lambda_2$, where λ_1 and λ_2 are the first two singular values of $\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}$ (13.241). If its determinant is negative, A can be replaced by $-\mathsf{A}$. Thus one could choose the projection matrix

$$\mathsf{P}''_d = [\mathsf{A}''_d \mid \mathbf{a}''_d] = \left[\left(2\,||\mathbf{e}''\mathbf{e}'^\mathsf{T}||\,\mathsf{S}_{e''}\mathsf{F}^\mathsf{T} + ||\mathsf{S}_{e''}\mathsf{F}^\mathsf{T}||\,\mathbf{e}''\mathbf{e}'^\mathsf{T} \right) \mid \alpha \mathbf{e}'' \right] \tag{13.245}$$

with some $\alpha \neq 0$, which is a projection matrix with the left 3×3 matrix of P'' close to a rotation matrix.

Remark: The algebraic solution for determining the projection matrix P''_d from F specializes to the projection matrix $R[I_3| - \boldsymbol{B}]$ if F is actually an essential matrix, and has two equal singular values, as then $\mathsf{A} = \mathsf{S}_{e''}\mathsf{F} + \mathbf{e}''\boldsymbol{D}^\mathsf{T}$ is a rotation matrix.

13.3.7.2 Projection Matrices from E

If the calibration is known, the determination procedure of the projection matrices for both cameras can directly use the estimated base vector and rotation matrix in the coordinate system of the first camera (Sect. 13.3.3), indexed d for 'dependent images',

$$\mathsf{P}'_d = \mathsf{K}'[I_3|\mathbf{0}] \qquad \mathsf{P}''_d = \mathsf{K}''R[I_3| - \boldsymbol{B}] \,. \tag{13.246}$$

If necessary, we may determine the covariance matrix of all 24 parameters of the two projection matrices. It has rank 5, as it depends only on the five independent parameters of the relative orientation.

In all cases, the ambiguity in the signs, especially of the base vector, needs to be resolved by analysing whether the 3D points determined by triangulation are in front of the camera. For the calibrated case, see (13.164), p. 582.

13.4 Triangulation

13.4.1 Reconstruction of 3D Points 596
13.4.2 Reconstruction of 3D Lines 605

Given the image coordinates of a 3D point in two or more images, the determination of the coordinates of the 3D point is called *triangulation*. Triangulation can be done in several ways. We discuss the following cases:

- Triangulation of two projection rays yielding a 3D point. Since this creates a redundancy, it can be used for checking the observations but not for localizing gross errors.
- Triangulation of multiple projection rays, which generally allows gross error localization.
- Intersection of two projection planes yielding a 3D line.[7]
- Triangulation of multiple projection planes, which allows checking of the observations but not necessarily localization of gross errors.

We also give the theoretical precision of 3D points derived from two images.

The determination of 3D points exploiting the stochastic properties of all participating entities, including the uncertainty of the camera parameters, is achieved with a *bundle adjustment*, which we discuss in Chap. 15. Triangulation is based on a simplified mathematical model, since *the cameras are assumed to be fixed*. This model is useful for determining approximate values, or may yield acceptable results if the camera poses are very precise.

cameras are assumed to be fixed

13.4.1 Reconstruction of 3D Points

Triangulation is an overconstrained problem even with only two images, as we have four observations, two coordinates per image point, and three unknown spatial coordinates. Several solutions are known (see Hartley and Sturm, 1997):

1. An optimal solution. It first corrects the image coordinates by exploiting their stochastic properties. Based on the fitted observations, the 3D point is then determined by intersecting the two rays. Though this solution cannot easily be generalized to multiple rays, we will present it below due to its simplicity, generality, and speed.
2. The approximate solution from Sect. 13.3.3, p. 581, in (13.163). We will use this as the second step of the previous optimal procedure, since the optimal solution guarantees that the two rays actually intersect.
3. A purely geometric solution which determines the point closest to both projection rays using the solution in Sect. 8, p. 402. As shown in Sect. 10.6.2.1, p. 419, confirming the analysis in Hartley and Sturm (1997), this solution has a relatively large bias if the parallactic angle is small, and in spite of its simplicity it is not recommended as a general procedure.
4. An approximate solution useful for the normal case. It has the advantage of allowing easy prediction of the precision of the 3D point, but may show bias for large y' values.
5. An approximate solution minimizing the algebraic error, which can be generalized to three or more images.

13.4.1.1 Optimal Triangulation of Points

The optimal solution for the triangulation with two cameras which are assumed to be of superior quality, and therefore taken as nonstochastic, consists of two steps:

[7] We generally use the term triangulation. We only use the term intersection if the context is unique.

1. Correcting the image coordinates or image rays, guaranteeing that the resulting rays are coplanar.
2. Intersecting the two corrected image rays in 3D.

We discuss the first step for calibrated and uncalibrated cameras without lens distortion. In both cases, we perform an optimal estimation of the fitted image observations, enforcing the coplanarity constraint following Kanatani et al. (2008). In all cases we assume the relative orientation to be fixed, i.e., nonstochastic.

Triangulation for Calibrated Cameras Using the Spherical Camera Model. We first discuss the triangulation for calibrated cameras using the spherical camera model. This includes calibrated perspective cameras. We start from spherically normalized camera rays $\mathbf{u} := {}^c\mathbf{x}'^s$ and $\mathbf{v} := {}^c\mathbf{x}''^s$ using (12.109), p. 492, and in the first step we enforce the epipolar constraint. Using camera rays, we can deal with a 3D point which may lie in all directions of the viewing sphere and possibly be at infinity.

We use the nonlinear constraint in estimation model C of constraints between observations only, see Sect. 4.8.1, p. 162,

$$g([\tilde{\mathbf{u}};\tilde{\mathbf{v}}]) = \tilde{\mathbf{u}}^\mathsf{T}\tilde{\mathsf{E}}\tilde{\mathbf{v}} = 0\,. \tag{13.247}$$

This is a constraint nonlinear in the observations. We start from approximate values $\widehat{\mathbf{u}}^a$ and $\widehat{\mathbf{v}}^a$ for the fitted observations, which in the first iteration are the observed directions themselves. With the corrections of the camera rays using reduced coordinates, see Sect. 10.2.2.1, p. 369,

$$\Delta\widehat{\boldsymbol{u}}_r = J_r^\mathsf{T}(\widehat{\mathbf{u}}^a)\widehat{\mathbf{u}}\,, \qquad \Delta\widehat{\boldsymbol{v}}_r = J_r^\mathsf{T}(\widehat{\mathbf{v}}^a)\widehat{\mathbf{v}} \tag{13.248}$$

$$\Delta\widehat{\mathbf{u}} = J_r(\widehat{\mathbf{u}}^a)\,\Delta\widehat{\boldsymbol{u}}_r\,, \qquad \Delta\widehat{\mathbf{v}} = J_r(\widehat{\mathbf{v}}^a)\,\Delta\widehat{\boldsymbol{v}}_r\,; \tag{13.249}$$

using $J_r(\boldsymbol{a}) = \mathsf{null}(\boldsymbol{a}^\mathsf{T})$, we have the linearized model

$$g([\widehat{\mathbf{u}};\widehat{\mathbf{v}}]) = \widehat{\mathbf{u}}^{a\mathsf{T}}\mathsf{E}\,\widehat{\mathbf{v}}^a + \widehat{\mathbf{v}}^{a\mathsf{T}}\mathsf{E}^\mathsf{T}\,J_r(\widehat{\mathbf{u}}^a)\,\Delta\widehat{\boldsymbol{u}}_r + \widehat{\mathbf{u}}^{a\mathsf{T}}\mathsf{E}\,J_r(\widehat{\mathbf{v}}^a)\,\Delta\widehat{\boldsymbol{v}}_r\,. \tag{13.250}$$

Referring to estimation model C and including the covariance matrix, we obtain

$$g(\widehat{\Delta\boldsymbol{l}}) = g(\widehat{\boldsymbol{l}}) + B^\mathsf{T}\widehat{\Delta\boldsymbol{l}}\,, \qquad \mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll}\,, \tag{13.251}$$

and with the reduced coordinates of the camera rays and their covariance matrix, see Sect. 10.2.2.1, p. 369,

$$\boldsymbol{l} := \begin{bmatrix}\boldsymbol{u}_r\\ \boldsymbol{v}_r\end{bmatrix} = \begin{bmatrix}J_r^\mathsf{T}(\widehat{\mathbf{u}}^a)\mathbf{u}\\ J_r^\mathsf{T}(\widehat{\mathbf{v}}^a)\mathbf{v}\end{bmatrix}\,, \quad \Sigma_{ll} := \begin{bmatrix}J_r^\mathsf{T}(\widehat{\boldsymbol{u}}^a)\Sigma_{\mathrm{uu}}J_r(\widehat{\mathbf{u}}^a) & \mathsf{0}\\ \mathsf{0} & J_r^\mathsf{T}(\widehat{\boldsymbol{v}}^a)\Sigma_{\mathrm{vv}}J_r(\widehat{\mathbf{v}}^a)\end{bmatrix}\,. \tag{13.252}$$

The Jacobian B^T of the constraint w.r.t. the observations is

$$B^\mathsf{T} = \left[\widehat{\mathbf{v}}^{a\mathsf{T}}\mathsf{E}^\mathsf{T}J_r(\widehat{\mathbf{u}}^a)\,, \widehat{\mathbf{u}}^{a\mathsf{T}}\mathsf{E}J_r(\widehat{\mathbf{v}}^a)\right]\,. \tag{13.253}$$

The general update of the observations is

$$\widehat{\Delta\boldsymbol{l}} = \boldsymbol{l} + \Sigma_{ll}B(B^\mathsf{T}\Sigma_{ll}B)^{-1}c_g \tag{13.254}$$

(see (4.448), p. 165), with the residual constraint

$$c_g = -g(\boldsymbol{l}^a) + B^\mathsf{T}(\widehat{\boldsymbol{l}}^a - \boldsymbol{l}) = -g(\boldsymbol{l})\,. \tag{13.255}$$

Here we have

$$c_g = -\mathbf{u}^\mathsf{T}\mathsf{E}\,\mathbf{v}\,. \tag{13.256}$$

For isotropic and equal uncertainty of the two rays we can assume $\Sigma_{uu} = \Sigma_{vv} = \sigma^2 I_3$, thus $\Sigma_{ll} = \sigma^2 I_6$. This simplifies the normal equation matrix to the scalar

$$n = B^T B \,. \tag{13.257}$$

The corrections for the reduced coordinates are

$$\Delta\hat{\boldsymbol{u}}_r = \boldsymbol{u}_r + \frac{c_g}{n} J_r^T(\hat{\mathbf{u}}^{aT}) \mathsf{E} \hat{\mathbf{v}}^a \tag{13.258}$$

$$\Delta\hat{\boldsymbol{v}}_r = \boldsymbol{v}_r + \frac{c_g}{n} J_r^T(\hat{\mathbf{v}}^{aT}) \mathsf{E}^T \hat{\mathbf{u}}^a \,. \tag{13.259}$$

This finally yields the updates in an iterative scheme; substituting $a = (\nu)$,

$$\hat{\mathbf{u}}^{(\nu+1)} = \mathsf{N}\left(\hat{\mathbf{u}}^{(\nu)} + J_r(\hat{\mathbf{u}}^{(\nu)}) \Delta\hat{\boldsymbol{u}}_r\right) \tag{13.260}$$

$$\hat{\mathbf{v}}^{(\nu+1)} = \mathsf{N}\left(\hat{\mathbf{v}}^{(\nu)} + J_r(\hat{\mathbf{v}}^{(\nu)}) \Delta\hat{\boldsymbol{v}}_r\right) \,. \tag{13.261}$$

Generally, not more than three iterations are required. In practice, one iteration is sufficient since it uses an approximate model, since it neglects the uncertainty of the camera poses.

Using the final estimates for the observed image coordinates $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$, the 3D point is given by

$$\hat{\mathbf{X}} = \begin{bmatrix} \bar{r}\,\hat{\mathbf{u}} \\ D \end{bmatrix} \tag{13.262}$$

(see (13.163), p. 582 and (13.164), p. 582), with

$$\begin{bmatrix} \bar{r} \\ \bar{s} \end{bmatrix} = D \begin{bmatrix} r \\ s \end{bmatrix} = D \begin{bmatrix} \mathbf{m}^T \hat{\mathbf{w}} \\ \mathbf{m}^T \hat{\mathbf{u}} \end{bmatrix} \,, \quad \mathbf{m} = (\mathbf{b} \times \hat{\mathbf{u}}) \times \mathbf{b} \tag{13.263}$$

and

$$D = \det(\mathbf{b} \mid \mathbf{m} \mid \hat{\mathbf{u}} \times \hat{\mathbf{w}}) \quad \text{and} \quad \hat{\mathbf{w}} = R^T \hat{\mathbf{v}} \,. \tag{13.264}$$

The point is indefinite if it lies in the base line, as then $\mathbf{m} = \mathbf{0}$, and thus $\bar{r} = D = 0$. (We will need the variable $\bar{s}$ later.)

The *covariance matrix* of the 3D points should be given for their homogeneous coordinates in order to allow for points at infinity.

We first give the covariance matrix for points close to the cameras, thus with a parallactic angle significantly larger than zero. The uncertainty of the 3D point can be derived as the uncertainty of three intersecting planes $\mathcal{A}_i$ with fixed normals $\mathbf{n}_i$, and only uncertain along their normals, by using the weight matrices, see (10.180), p. 403,

$$W_i = \frac{\mathbf{n}_i \mathbf{n}_i^T}{\sigma_i^2} \,. \tag{13.265}$$

The normals are related to the normal $\mathbf{n} = \mathsf{N}(\hat{\mathbf{u}} \times \hat{\mathbf{v}})$ of the epipolar plane, and the standard deviations depend on the distances:

1. The first plane lies across the first ray in the epipolar plane. Its standard deviation at the estimated point is $\sigma_u = r\sigma$. Its direction is $\mathbf{r} = \mathbf{n} \times \hat{\mathbf{u}}$.
2. The second plane lies across the second ray in the epipolar plane. Its standard deviation is $\sigma_v = s\sigma$. Its direction is $\mathbf{s} = \mathbf{n} \times \hat{\mathbf{v}}$.
3. The third plane lies parallel to the epipolar plane with normal $\mathbf{n}$. Its variance results from the weighted average of the two distances from the epipolar plane:

$$\sigma_w^2 = \frac{1}{\dfrac{1}{\sigma_u^2} + \dfrac{1}{\sigma_v^2}} = \frac{\sigma_u^2 \sigma_v^2}{\sigma_u^2 + \sigma_v^2} \,. \tag{13.266}$$

Adding the weight matrices we therefore obtain the covariance of the 3D point,

$$\Sigma_{\widehat{X}\widehat{X}} = (\mathbf{r}\mathbf{r}^\mathsf{T}/\sigma_u^2 + \mathbf{s}\mathbf{s}^\mathsf{T}/\sigma_v^2 + \mathbf{n}\mathbf{n}^\mathsf{T}/\sigma_w^2)^{-1} \,. \tag{13.267}$$

In order to allow for points at infinity, we use the covariance matrix for the homogeneous coordinates $\widehat{\mathbf{X}} = [\widehat{\boldsymbol{X}}_0^\mathsf{T}, \widehat{X}_h]^\mathsf{T}$ (13.262), specifically that of $\widehat{\boldsymbol{X}}_0$,

$$\Sigma_{\widehat{X}_0\widehat{X}_0} = D^2\,\Sigma_{\widehat{X}\widehat{X}} = \frac{1}{\bar{r}^2}\left(\mathbf{r}\mathbf{r}^\mathsf{T} + \frac{\bar{r}^2}{\bar{s}^2}\mathbf{s}\mathbf{s}^\mathsf{T} + \mathbf{n}\mathbf{n}^\mathsf{T}\frac{\bar{r}^2+\bar{s}^2}{\bar{s}^2}\right)^{-1}\sigma^2 \,, \tag{13.268}$$

with values $\bar{r}$ and $\bar{s}$, which generally are nonzero. This finally yields the covariance matrix of the spherically normalized coordinate vector,

$$\widehat{\mathbf{X}}^s = \frac{\widehat{\mathbf{X}}}{|\widehat{\mathbf{X}}|} \,, \qquad \Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s} = \boldsymbol{J}_s \begin{bmatrix} \Sigma_{\widehat{X}_0\widehat{X}_0} & \mathbf{0}^\mathsf{T} \\ \mathbf{0} & 0 \end{bmatrix} \boldsymbol{J}_s^\mathsf{T} \,, \tag{13.269}$$

with

$$\boldsymbol{J}_s = \frac{1}{|\widehat{\mathbf{X}}|}(\boldsymbol{I}_4 - \widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^{s\mathsf{T}}) \,. \tag{13.270}$$

The solution is only valid for points not lying on the base line.

The procedure is given in Algorithm 21. It incorporates a check on whether the two rays do not intersect, or intersect in backward direction. The algorithm assumes the rays $\mathbf{u}$ and $\mathbf{v}$ to have isotropic uncertainty with a standard deviation of σ [rad] in all directions. The number of iterations usually is two or less, so the variable `maxiter` can safely be set to 2. Besides the fitted rays, the algorithm provides the spherically normalized 3D point, which may be at infinity. Some lines require explanation:

(2) The critical value k_c for the residual $c_g = \mathbf{u}^\mathsf{T}\mathsf{E}\mathbf{v}$ of the epipolar constraint is determined using variance propagation. From

$$\mathrm{d}c_g = \mathrm{d}(\mathbf{u}^\mathsf{T}\mathsf{E}\mathbf{v}) = \mathbf{u}^\mathsf{T}\mathsf{E}\mathrm{d}\mathbf{v} + \mathbf{v}^\mathsf{T}\mathsf{E}^\mathsf{T}\mathrm{d}\mathbf{u} \,, \tag{13.271}$$

assuming E to be fixed, due to $\Sigma_{uu} = \Sigma_{vv} = \sigma^2 I_3$, we obtain

$$\sigma_{c_g}^2 = \mathbf{u}^\mathsf{T}\mathsf{E}\Sigma_{vv}\mathsf{E}^\mathsf{T}\mathbf{u} + \mathbf{v}^\mathsf{T}\mathsf{E}^\mathsf{T}\Sigma_{vv}\mathsf{E}\mathbf{v} = \sigma^2(\mathbf{u}^\mathsf{T}\mathsf{E}\mathsf{E}^\mathsf{T}\mathbf{u} + \mathbf{v}^\mathsf{T}\mathsf{E}^\mathsf{T}\mathsf{E}\mathbf{v}) \,. \tag{13.272}$$

(9) Here we check the iterative procedure for convergence. The tolerance T usually is chosen to be < 1, e.g., $T = 0.1$.

(26) If at least one of the two signed distances d_1 and d_2 to the 3D point is negative, the point is either invalid, or behind the camera, or at infinity, but only if $|D|$ is small enough.

(24,27) If the determinant is close to zero, the point is at infinity. Therefore we set $D = 0$ in order to guarantee that the point is in the direction of $\mathbf{u}$ and $\mathbf{v}$.

A MATLAB implementation of the algorithm requires approximately 0.7 milliseconds for deriving $\widehat{\mathbf{X}}^s$ from $(\mathbf{u}, \mathbf{v})$.

Figure 13.15 shows the standard ellipses for a region around the base line $\mathcal{O}'\mathcal{O}''$ in one epipolar plane. The uncertainty of the points varies greatly.

The uncertainty is best for points between the two projection centres at a distance between 50% and 100% of the base length from the base line. The uncertainty decreases with the distance from the basis. The ratio of the maximum to the minimum standard deviation, in this example is approximately 11. The parallactic angle α is decisive for the ratio of the major axes of the ellipses, which in this example has a maximum of 15. The orientation of the ellipses is dominated by the closest projection centre. Points closer to the base line than 20% of the base line, not shown in the figure, have highly uncertain distances from the projection centres. The inhomogeneity of the uncertainty field generally should not be neglected when using scene points for further analysis.

Algorithm 21: Optimal triangulation of a 3D point from two images and spherical camera model $[\widehat{\mathbf{X}}^s, \Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s}, f]$=`triangulation`$(\mathbf{u}, \mathbf{v}, \sigma)$
Input: relative orientation $[\mathbf{b}, R]$, $|\mathbf{b}| = 1$,
corresponding camera rays $\{\mathbf{u}, \mathbf{v}\}$, $|\mathbf{u}| = |\mathbf{v}| = 1$,
directional standard deviation σ [rad] assuming isotropic uncertainty,
maximum number `maxiter` for iterations, tolerance T, critical value k for testing.
Output: type of result f,
(f=0: success, f=1: point invalid or backwards, f=2: rays not coplanar),
fitted camera rays $\{\widehat{\mathbf{u}}, \widehat{\mathbf{v}}\}$,
triangulated 3D point $\{\widehat{\mathbf{X}}^s, \Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s}\}$, if $f = 0$ else $\{\widehat{\mathbf{X}}^s, \Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s}\}\} = \{\mathbf{0}, \mathit{0}\}$.

Essential matrix $\mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T}$;
Critical value $k_c = k\sigma\sqrt{\mathbf{u}^\mathsf{T}\mathsf{E}\mathsf{E}^\mathsf{T}\mathbf{u} + \mathbf{v}^\mathsf{T}\mathsf{E}^\mathsf{T}\mathsf{E}\mathbf{v}}$;
if $|\mathbf{u}^\mathsf{T}\mathsf{E}\mathbf{v}| > k_c$ **then**
  rays not coplanar, failure $f = 2$;
  $\mathbf{X}^s = \mathbf{0}$, $\Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s} = \mathit{0}$, $\widehat{\mathbf{u}} = \mathbf{u}$, $\widehat{\mathbf{v}} = \mathbf{v}$, **return**
Initialize: $f = 0$, $\nu = 0$, $\widehat{\mathbf{u}}^a = \mathbf{u}$, $\widehat{\mathbf{v}}^a = \mathbf{v}$;
for $\nu = 1, ...,$ `maxiter` **do**
  Constraint: $c_g = \widehat{\mathbf{u}}^{a\mathsf{T}}\mathsf{E}\widehat{\mathbf{v}}^a$;
  if $|c_g| < T\sigma$ **then exit** ν-loop;
  Jacobians: $J_1 = \mathsf{null}(\widehat{\mathbf{u}}^{a\mathsf{T}})$, $J_2 = \mathsf{null}(\widehat{\mathbf{v}}^{a\mathsf{T}})$;
  Observations: $\boldsymbol{l} = [J_1^\mathsf{T}\mathbf{u}; J_2^\mathsf{T}\mathbf{v}]$;
  Jacobian: $B = [J_1^\mathsf{T}\mathsf{E}\widehat{\mathbf{v}}^a; J_2^\mathsf{T}\mathsf{E}^\mathsf{T}\widehat{\mathbf{u}}^a]$;
  Corrections of reduced observations: $\widehat{\Delta\boldsymbol{l}} = \boldsymbol{l} + c_g/(B^\mathsf{T}B)\,B$;
  Update rays: $\widehat{\mathbf{u}}^a = \mathsf{N}(\widehat{\mathbf{u}}^a + J_1\widehat{\Delta\boldsymbol{l}}(1:2))$, $\widehat{\mathbf{v}}^a = \mathsf{N}(\widehat{\mathbf{v}}^a + J_2\widehat{\Delta\boldsymbol{l}}(3:4))$;
end
Fitted camera rays: $\widehat{\mathbf{u}} = \widehat{\mathbf{u}}^a$, $\widehat{\mathbf{v}} = \widehat{\mathbf{v}}^a$, in model system $\widehat{\mathbf{w}} = R^\mathsf{T}\widehat{\mathbf{v}}$;
Ancillary vector: $\mathbf{m} = \mathsf{N}((\mathbf{b} \times \widehat{\mathbf{u}}) \times \mathbf{b})$;
Relative distances: $[\bar{r}, \bar{s}] = \mathbf{m}^\mathsf{T}[\widehat{\mathbf{w}}, \widehat{\mathbf{u}}]$;
determinant for triangulation: $D = |\mathbf{b}, \mathbf{m}, \widehat{\mathbf{u}} \times \widehat{\mathbf{w}}|$;
Estimated point $\widehat{\mathbf{X}}^s = \mathsf{N}([\bar{r}\widehat{\mathbf{u}}; D])$;
Normal of epipolar plane $\widehat{\mathbf{n}} = \mathsf{N}(\widehat{\mathbf{u}} \times \widehat{\mathbf{v}})$;
Normals $\mathbf{r} = \widehat{\mathbf{n}} \times \widehat{\mathbf{u}}$, $\mathbf{s} = \widehat{\mathbf{n}} \times \widehat{\mathbf{v}}$;
Covariance matrix $\Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s}$ using (13.268) and (13.269);
if $|D| \le 10^{-15}$ **then** 3D point at infinity, $D = 0$ **return**;
Distances $\boldsymbol{d} = [r; s]/D$;
if (sign$(d_1) < 0$ or sign$(d_2) < 0$) **then**
  if $|D| < k\sqrt{2}\,\sigma$ **then** 3D point at infinity, $D = 0$ **return**;
  else point invalid or in backward direction, failure $f = 1$;
  $\mathbf{X}^s = \mathbf{0}$, $\Sigma_{\widehat{\mathbf{X}}^s\widehat{\mathbf{X}}^s} = \mathit{0}$, **return**.
end

Triangulation with Perspective Cameras. The perspective camera model must be used for uncalibrated cameras but may be used also for calibrated cameras. With the perspective camera model, we proceed in the same manner as with the spherical camera model, first optimally correcting the image coordinates such that the epipolar constraint is fulfilled (see Kanatani et al., 2008), and then determining the 3D coordinates of the scene point by intersection.

The procedure is fully analogous to the one for spherical cameras, setting $\mathbf{u} = [\boldsymbol{x}'^\mathsf{T}, 1]^\mathsf{T}$ and $\mathbf{v} = [\boldsymbol{x}''^\mathsf{T}, 1]^\mathsf{T}$, replacing E by F, and selecting the first two components of the homogeneous vector by replacing

$$J_r(\widehat{\mathbf{u}}^a) = J_r(\mathbf{v}^a) \to \begin{bmatrix} I_2 \\ \mathbf{0}^\mathsf{T} \end{bmatrix} ; \tag{13.273}$$

thus the reduced coordinates are identical to the Cartesian ones, e.g., $\boldsymbol{u}_r \equiv \boldsymbol{u}$. Then the covariance matrices of the reduced coordinates are the ones provided for the Cartesian

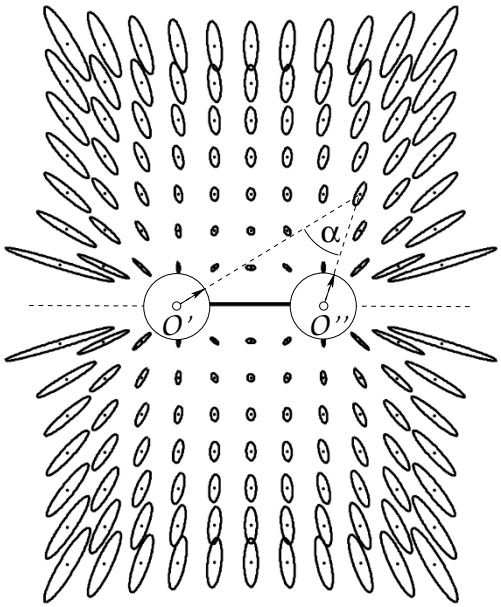


Fig. 13.15 Uncertainty field in an epipolar plane close to the basis for a pair of spherical cameras and homogeneous directional uncertainty. The size, shape, and direction of the standard ellipses of the points depend on the distances from the projection centres O' and O'' and the parallactic angle α

coordinates $\Sigma_{x'x'}$ and $\Sigma_{x''x''}$, possibly simplified to $\sigma^2 I_2$. The update of the homogeneous vectors $\mathbf{u}$ and $\mathbf{v}$ is simply the update of their Euclidean parts.

Given the fitted image coordinates $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{v}}$, (13.262), p. 598, they can be used for the intersection, replacing R by A''_d and $\mathbf{b}$ by $\mathsf{A}''^{-1}_d \mathbf{a}''_d$ from (13.245), p. 595.

The derivation of the covariance matrix of the 3D point is more complex.

13.4.1.2 Approximate Triangulation for the Normal Case of the Image Pair

For the normal case, the canonical photogrammetric solution determines the coordinates of the 3D point $\mathcal{X}$ separately, at first the X and the Z coordinates as the intersection of the projection of the two rays on a view perpendicular to the plane through the coordinate axes x', z' and x'', z'' (Fig. 13.16, top). The Y coordinate then results from the midpoint of two points on the rays (Fig. 13.16, bottom). This simple procedure was tailored for the first photogrammetric stereo instruments in the 1920s (see McGlone, 2013, Sect. 1).

With the length of the base line $B \doteq B_X$ and the parallax $p_x = x'' - x'$, the coordinates are

$$X = x' \frac{B}{-p_x}, \quad Y = \frac{y' + y''}{2} \frac{B}{-p_x}, \quad Z = c \frac{B}{-p_x} \tag{13.274}$$

or

$$X = x' \frac{Z}{c}, \quad Y = \frac{y' + y''}{2} \frac{Z}{c}, \quad Z = c \frac{B}{-p_x} = c \frac{Z}{c} = Z\,. \tag{13.275}$$

They are fully symmetric, namely proportional to the three coordinates x', $(y' + y'')/2$, and c, respectively, together forming the 3-vector of the spatial direction from O' to $\mathcal{X}'$ in the first image. The factor $S = -B/(x'' - x') = Z/c$ is the image scale number at points $\mathcal{X}'$ and $\mathcal{X}''$.

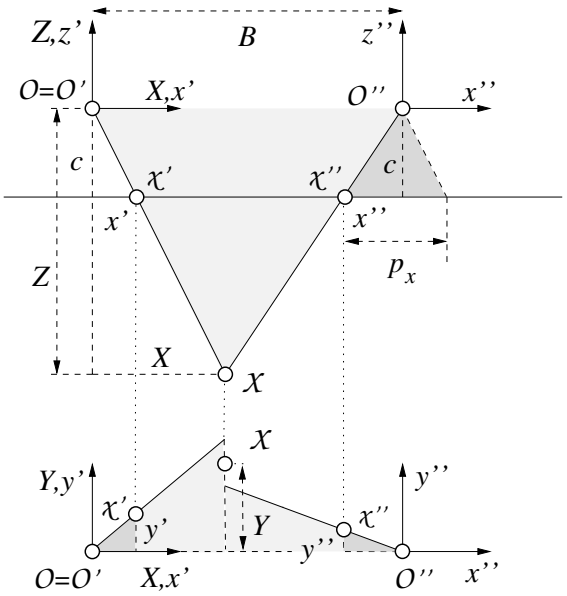


Fig. 13.16 Canonical photogrammetric solution of triangulation for the normal case. **Top**: view in the XZ plane. **Bottom**: view in the XY plane. The coordinate system of the photogrammetric model with its origin O is identical to the left camera coordinate system

Obviously, the parallaxes of the image coordinates play a decisive role:

1. The x-parallax $p_x = x'' - x'$, generally $\neq 0$, is responsible for the depth, the height, or the distance of the point from the base line, depending on the context.
2. The y-parallax $p_y = y'' - y'$ should equal 0 and is responsible for the consistency of the relative orientation, namely the quality of the intersection of the projection rays.

For a set of $\{x', y'\}$ in the first image, the set $\{x', y', p_x\}$ is called the *parallax map*. There is a full analogy between this parallax map and the corresponding set $\{X, Y, Z\}$ of 3D points. If $p_y = 0$, thus $(y'' + y')/2 = y'$, (13.274) yields

$$\begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ c \\ p_x \end{bmatrix} . \tag{13.276}$$

This is a straight line-preserving mapping or a homography of image space $[x', y', p_x]$ to object space $[X, Y, Z]$ (see Chumerin and Van Hulle, 2008). Thus for checking the collinearity or coplanarity of points, we do not need to determine their 3D coordinates but may check these relations using the image-related coordinates $[x', y', p_x]$ of the parallax map. Observe, for $B > 0$ the mapping (13.276) changes the chirality of spatial relations as the determinant of the homography matrix is negative, see (9.40), p. 357.

The precision of 3D points can be easily derived from these relations, see Sect. (13.4.1.4), p. 603.

13.4.1.3 Direct Algebraic Solution for a 3D Point from Multiple Images

The determination of the intersection of several 3D rays has been discussed in Sect. 10.5.3.2, p. 401.

We now give a direct algebraic solution with three or more images which can handle points at infinity. The constraint on image point $\mathbf{x}'_t$ in the tth image which with projection matrix P_t is the projection of an unknown 3D point $\mathbf{X}$ is

$$\mathbf{x}'_t \times \mathsf{P}_t \, \mathbf{X} = \mathsf{S}_{x'_t} \mathsf{P}_t \, \mathbf{X} = \mathbf{w}_t \stackrel{!}{=} \mathbf{0} \qquad t = 1, ..., T \,. \tag{13.277}$$

Collecting all 3×4 matrices $\mathsf{A}_t = \mathsf{S}_{x'_t} \mathsf{P}_t$ in the $3I \times 4$ matrix $\mathsf{A} = [\mathsf{A}_t]$, and all residuals $\mathbf{w}_t$ in the $3T$-vector $\mathbf{w} = [\mathbf{w}_t]$, we can represent all constraints as

$$\mathsf{A}\mathbf{X} = \begin{bmatrix} \mathsf{S}_{x'_1} \mathsf{P}_1 \\ \dots \\ \mathsf{S}_{x'_t} \mathsf{P}_t \\ \dots \\ \mathsf{S}_{x'_T} \mathsf{P}_T \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{w}_1 \\ \dots \\ \mathbf{w}_t \\ \dots \\ \mathbf{w}_T \end{bmatrix} = \mathbf{w} \stackrel{!}{=} \mathbf{0} \,. \tag{13.278}$$

Alternatively, the skew symmetric matrices $\mathsf{S}^{(s)}_{x'_t}$ with selected rows could be used as in (12.122), p. 495. Thus the optimal point, minimizing the algebraic error $\mathbf{w}^\mathsf{T}\mathbf{w}$, is the right eigenvector of A belonging to its smallest eigenvalue, using an SVD. The calculations should be performed after a proper conditioning, see Sect. 6.9, p. 286, especially centring and scaling, such that all elements are inside a centred unit cube. The solution is then a very good approximation to the optimal one (Hartley, 1997a; Wolff and Förstner, 2000; Ressl, 2003). The covariance matrix of the solution can be derived using (4.521), p. 181. Observe, the solution does not require selecting independent constraints or working with reduced coordinates.

13.4.1.4 Quality of 3D Points from Two Images

The canonical photogrammetric solution for the spatial intersection can be used to determine the theoretical precision of 3D points. It depends on the uncertainty of the relative orientation and the uncertainty of the measured corresponding points.

For simplicity, we again assume the uncertainty of the relative orientation to be negligible. The precision of the image coordinates are in a first step assumed to be $\sigma_{x'} = \sigma_{y'} = \sigma_{x''} = \sigma_{y''}$, i.e., we assume both points are positioned independently using a key point detector. We will later discuss the uncertainty if the point in the first image is determined by some key point detector and the parallax is measured with a correlation technique.

By variance propagation, we obtain from (13.275), with $|Z/c| = |B/p_x|$,

$$\sigma_X^2 = \frac{Z^2}{c^2} \frac{x'^2 + x''^2}{p_x^2} \sigma_{x'}^2 \tag{13.279}$$

$$\sigma_Y^2 = \frac{1}{2} \left(\frac{Z^2}{c^2} \frac{4y''^2 + p_x^2}{p_x^2} \right) \sigma_{x'}^2 \tag{13.280}$$

$$\sigma_Z^2 = 2 \frac{Z^2}{c^2} \frac{Z^2}{B^2} \sigma_{x'}^2 \,, \tag{13.281}$$

where the Z-coordinate is the distance of the point from the principal plane $\mathcal{C}$ of the camera, see Fig. 12.10, p. 474 right. For points below the middle of the basis we have $[x', y'] \approx [-p_x/2, 0]$, $y'' \approx 0$, and therefore $x'' \approx p_x/2$. For these points we first obtain the standard deviations (always taking absolute values of Z and c)

$$\sigma_X = \sigma_Y = \frac{1}{\sqrt{2}} \frac{Z}{c} \sigma_{x'} \,. \tag{13.282}$$

The standard deviation of the X- and Y-coordinates is the standard deviation of the measured image coordinate multiplied by the scale factor Z/c, except for the factor $1/\sqrt{2}$ which results from averaging the coordinates. Using the standard deviation $\sigma_{p_y} = \sqrt{2}\sigma_{x'}$ of the parallax p_x, we obtain the standard deviation of the depth in various forms using $Z/c = B/p_x$ (again always taking absolute values of Z, c, and p_x),

$$\sigma_Z = \frac{Z}{p_x}\sigma_{p_x} = \frac{cB}{p_x^2}\sigma_{p_x} = \frac{Z^2}{cB}\sigma_{p_x} = \frac{Z}{c}\frac{1}{B/Z}\sigma_{p_x}\,. \tag{13.283}$$

This result deserves a few comments:

- First we observe that the relative depth accuracy σ_Z/Z is identical to the relative parallax accuracy, which is intuitive.
- The standard deviation σ_Z is inversely proportional to the square of the x-parallax for a given geometry of the images, thus for fixed (c, B).
- It is proportional to the square Z^2 of the distance from the base line for a given geometry.
- It is proportional to the scale factor Z/c *and* inversely proportional to the base-to-height ratio B/Z, in contrast to the planimetric precision, which is proportional only to the scale number. The base-to-height ratio B/Z is closely related to the parallactic angle; in the symmetric case, we have

$$\tan\frac{\alpha}{2} = \frac{1}{2}\frac{B}{Z}\,. \tag{13.284}$$

  This is why very small parallactic angles, i.e., short base lines, lead to very uncertain distances.
- We will give standard deviations for the case of multiple rays in Sect. 15.7, p. 715, see (15.221), p. 717.

Example 13.4.48: ***Precision of 3D coordinates.*** We discuss the theoretical uncertainty of 3D coordinates, first visually inspecting the uncertainty structure in front of a camera pair and then determining the expected precision of a 3D point for three practical cases.

1. **Comparing different setups for measuring correspondences.** Fig. 13.17, left, shows the standard ellipses of a grid of points in front of a camera pair if the image points are independently measured, e.g., using a key point detector with homogeneous accuracy within the image plane. We show them in one epipolar plane, say in the XZ-plane. The visualization takes the standard deviations in X and Z directions, see (13.279), p. 603, but also the correlations between the X and Z coordinates, into account. The dependency of the depth uncertainty on the depth is clearly visible.

We now assume the point in the first image to be located first, e.g., using a key point detector, and the parallaxes to be determined next, e.g., using some image correlation technique. The second step is stochastically independent of the first. Then the image coordinates $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are correlated, see Sect. 13.3.1, p. 569. Since parallaxes can be measured more precisely than points can be located, the situation improves if we take this into account. If we assume the parallax to be determinable with the standard deviation $\sigma_{p_x} = \sigma_{x'}/\sqrt{2}$, we obtain the uncertainty field in Fig. 13.17, right. Not only has the depth accuracy improved, but the uncertainty in the X-direction now is smaller as well.

Finally, we compare the two uncertainty fields of the perspective camera to the one obtained for the spherical camera in Fig. 13.15. The uncertainty of the 3D points on average is less for the perspective camera than for the spherical camera. The reason is simply the different stochastical model, see the discussion of Figs. (10.9), p. 369 and (10.11), p. 372: The directional uncertainty in both models is the same only for rays in viewing direction, perpendicular to the base line. While the directional uncertainty for the rays in the spherical cameras is assumed to be homogeneous, the directional uncertainty for the rays in perspective cameras greatly increases with the angle β' between the viewing direction and the direction to the scene point. Only scene points with small angles β' and β'' have comparable accuracy for the spherical and the perspective models. Both models are highly simplified. The localization accuracy in perspective cameras will generally decrease close to the border of the image, mostly due to imperfection of the lens, causing image blur.

The next two examples give an idea of the accuracy achievable in two representative applications.

rule of thumb for aerial images with approximately 60% overlap: σ_Z is 0.1‰ of flying height over ground

2. **Aerial camera:** We assume two images of a high-resolution aerial camera, say from the frame camera DMC 250 of Zeiss. It has five camera heads, one with a panchromatic sensor and four with infrared sensors. The panchromatic sensor has 14 015 pixels in flight direction and 16 768 pixels across flight direction, and a principal distance of 20 000 pixels. The overlap between two images usually is at least 60%, i.e., the base line, measured in pixels, is 40% of the 14 000 pixels along the flight path. We assume a measuring accuracy of $\sigma_{x'} = \sigma_{y'} = 0.3$ pixel for the independently measured coordinates and $\sigma_{p_x} = 0.5$ pixel for the parallax, which is realistic for smooth surfaces. With $p_x \approx b$ we first obtain

$$\sigma_Z = \frac{Z}{b}\sigma_{p_x} = \frac{Z}{5\,600\ [\text{pixel}]}0.5\ [\text{pixel}] \approx \frac{1}{10\,000}\,Z\,.$$

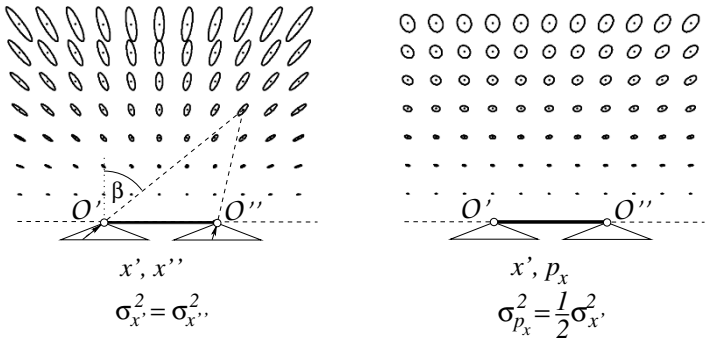


Fig. 13.17 Uncertainty field around the basis for a pair of perspective cameras. **Left:** Corresponding points in both images are measured independently. **Right:** The points in the left image and the parallaxes to the point in the second image are measured independently

So in a first approximation, the accuracy of the heights is $0.1‰$ of the flying height. This is shown by experience for all realizable flying heights between 300 m and 10 km (Schwidefsky and Ackermann, 1976, Sect. 3.6).

As an example: For a flying height of $Z = 1\,500$ m we obtain $\sigma_Z \approx 15$ cm, and for the planimetric coordinates,

$$\sigma_X = \sigma_Y = \frac{1}{\sqrt{2}}\frac{Z}{c}\sigma_{x'} = \frac{1}{\sqrt{2}}\frac{1\,500\ [\mathrm{m}]}{20\,000\ [\mathrm{pixel}]}0.3[\mathrm{pixel}] \approx 1.8\ \mathrm{cm}\,.$$

3. **Stereo video camera.** We assume a stereo video camera system with two video cameras with a 1024×768-pixel sensor and principal distance of 1 500 pixels, say, mounted in a car looking ahead. The base line has a length of $B = 0.3$ m. We again assume that image points are measured with $\sigma_{x'} = \sigma_{y'} = 0.3$ pixel, yielding the standard deviation $\sigma_{p_x} = 0.5$ pixel for the parallax. Then we obtain

$$\sigma_Z = \frac{Z^2}{cB}\sigma_{p_x} = \frac{1}{1\,500\ [\mathrm{pixel}]\ 0.3\ [\mathrm{m}]}\ 0.5\ [\mathrm{pixel}] \approx \frac{Z^2}{1\,000\ [\mathrm{m}]}\ Z^2\,.$$

The expected precision of the distance Z of 3D points from the camera system is given in Table 13.6.

Table 13.6 Expected precision and relative precision of distance Z (depth) derived from a stereo camera system. Base line $B = 0.3$ m, $c = 1500$ pixel, $\sigma_{x'} = 0.3$ pixel, $\sigma_{p_x} = 0.5$ pixel

distance Z [m]	2	5	10	20	50
precision σ_Z [m]	0.004	0.025	0.1	0.4	2.5
relative precision σ_Z/Z [%]	0.2	0.5	1.0	2.0	5.0

These theoretical expectations can be used for planning purposes. But the theoretical standard deviations need to be empirically checked by controlled experiments for each application area using sufficiently precise reference data, see the discussion in Sect. 4.6.2.2, p. 118. ◇

13.4.2 Reconstruction of 3D Lines

Reconstruction of 3D Lines from Two Images. For observed corresponding image lines $\mathbf{l}'$ and $\mathbf{l}''$ we obtain the 3D line $\mathbf{L}$ directly as the intersection of the two projecting planes $\mathbf{A}_{l'}$ and $\mathbf{A}_{l''}$, see (12.83), p. 483,

$$\mathbf{L} = \mathbf{A}_{l'} \cap \mathbf{A}_{l''} = \overline{\Pi}(\mathsf{P}'^{\mathsf{T}}\mathbf{l}')\mathsf{P}''^{\mathsf{T}}\mathbf{l}''\,. \tag{13.285}$$

The uncertainty of the 3D line depends on the uncertainty of the observed lines and of the projection matrices. They allow us to derive the covariance matrix Σ_{LL}. Exercise 13.18

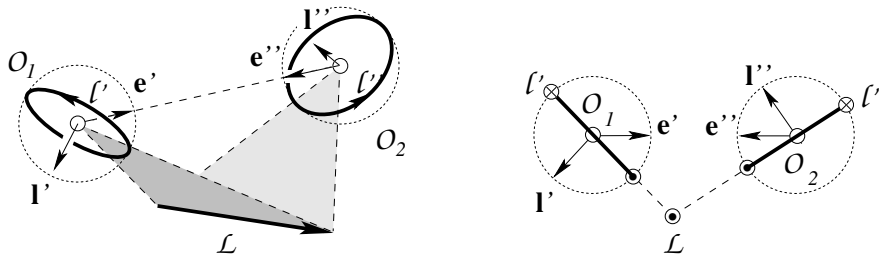


Fig. 13.18 Sign of image lines in two views: In each epipolar plane the projection centres O_1 and O_2 lie on the same side of the 3D line $\mathcal{L}$ and of their projections ℓ' and ℓ'', respectively. The figure on the right is a projection of the left figure parallel to the 3D line: The 3D line $\mathcal{L}$ is perpendicular to the drawing plane, and we draw the tip of the arrow of the directed line, indicated with a dotted circle. The oriented great circles are the images ℓ' and ℓ'' of the oriented 3D line and are shown by the thick diameters of the two circles indicating the two viewing spheres. In this projection, the 3D line lies below the basis O_1O_2. Therefore the two line vectors $\mathbf{l}'$ and $\mathbf{l}''$ point to the left. As the epipolar vectors point in opposite directions, the scalar products with the line vectors have different signs: $\text{sign}(\mathbf{l}' \cdot \mathbf{e}') = -\text{sign}(\mathbf{l}'' \cdot \mathbf{e}'')$

If the two lines ℓ' and ℓ'' are directed, they need to be consistent, i.e., they need to be images of an oriented 3D line $\mathbf{L}$, see Fig. 13.18. Given the two image lines $\mathbf{l}' = \mathsf{Q}'\mathbf{L}$ and $\mathbf{l}'' = \mathsf{Q}''\mathbf{L}$ (see (12.72), p. 480) and the two epipoles $\mathbf{e}'$ and $\mathbf{e}''$ from (13.72), p. 565, it can be shown that

Exercise 13.16

$$\mathbf{l}'^{\mathsf{T}}\mathbf{e}' + \mathbf{l}''^{\mathsf{T}}\mathbf{e}'' = 0\,. \tag{13.286}$$

If all elements are oriented homogeneous vectors, we have the sign constraint for two corresponding directed 2D lines,

$$\text{sign}(\mathbf{l}'^{\mathsf{T}}\mathbf{e}') = -\text{sign}(\mathbf{l}''^{\mathsf{T}}\mathbf{e}'')\,. \tag{13.287}$$

Given two corresponding and consistent directed 2D lines ℓ' and ℓ'', the 3D line $\mathbf{L}$ derived from (13.285) is not guaranteed to have the correct sign, since the sign of $\mathbf{L}$ should change when we change the signs of the two image lines such that they remain consistent. Also, exchanging the two images should not change the sign of $\mathbf{L}$. For obtaining the proper direction of the 3D line $\mathbf{L}$, it can be proven that (13.285) needs to be modified by a factor (see Werner and Pajdla, 2001):

Exercise 13.17

$$\mathbf{L} = \text{sign}(\mathbf{l}'^{\mathsf{T}}\mathbf{e}')\ \mathbf{A}_{l'} \cap \mathbf{A}_{l''} = \text{sign}(\mathbf{l}'^{\mathsf{T}}\mathbf{e}')\ \overline{\mathsf{TT}}(\mathsf{P}'^{\mathsf{T}}\mathbf{l}')\mathsf{P}''^{\mathsf{T}}\mathbf{l}''\,. \tag{13.288}$$

Direct Algebraic Solution for 3D Lines from Multiple Images. Analogously, we have the constraint for an image line $\mathbf{l}'_j$ in the jth image with projection matrix P_j, which is the image of an unknown 3D line $\mathbf{L}$ (7.64), p. 305,

$$\mathsf{TT}^{\mathsf{T}}(\mathsf{P}_j^{\mathsf{T}}\mathbf{l}'_j)\,\mathbf{L} = \mathbf{w}_j \stackrel{!}{=} \mathbf{0} \qquad j = 1, ..., J\,. \tag{13.289}$$

Therefore the optimal line minimizing the algebraic error $\sum \mathbf{w}_j^{\mathsf{T}}\mathbf{w}_j$ is the right eigenvector of the matrix

$$\begin{bmatrix} \mathsf{TT}^{\mathsf{T}}(\mathsf{P}_1^{\mathsf{T}}\mathbf{l}'_1) \\ \dots \\ \mathsf{TT}^{\mathsf{T}}(\mathsf{P}_j^{\mathsf{T}}\mathbf{l}'_j) \\ \dots \\ \mathsf{TT}^{\mathsf{T}}(\mathsf{P}_J^{\mathsf{T}}\mathbf{l}'_J) \end{bmatrix} \tag{13.290}$$

belonging to its smallest eigenvalue, based on an SVD. Again, the calculations should be performed after a proper conditioning. Also, the covariance matrix of the resulting 3D line can be determined using (4.521), p. 181.

13.5 Absolute Orientation and Spatial Similarity Transformation

photogrammetric model

The result of the relative orientation is a *photogrammetric model* of the scene. It includes the following information:

1. The parameters of the relative orientation of the two images, e.g., the pair $\{R, \boldsymbol{B}\}$ in the case of the parametrization with dependent images, in a local coordinate system S_m of the photogrammetric model. The model coordinate system S_m often is identical to the camera coordinate system S_{c_1} of the first camera, but any other choice is admissible.
2. 3D coordinates of N space points in the model coordinate system

$$^m\boldsymbol{X}_n = [^mX_n, ^mY_n, ^mZ_n]^\mathsf{T}, \qquad n = 1, ..., N\,. \tag{13.291}$$

This model is identical to the object, represented by a set of N points, up to a transformation. For calibrated cameras this is a spatial similarity transformation between the 3D coordinates in both systems. For uncalibrated perspective cameras it is a spatial homography, see the discussion in Sect. (13.2.1), p. 550. Absolute orientation is the determination of this transformation. If we have to take the similarity transformation we have for each model

absolute orientation

$$^m\boldsymbol{X}_i = {}^m\lambda \; {}^m\!R\boldsymbol{X}_i + \boldsymbol{T} \tag{13.292}$$

from the scene coordinate system S_0 to the model coordinate system S_m.

We need a specific *model scale*,

$$^m\lambda = \frac{^md}{D}\,, \tag{13.293}$$

i.e., the ratio of distances ^md in model space to corresponding distances D in object space. We cannot recover the scale of the scene from image data alone. The rotation and the translation are not necessarily those of one of the two images involved, as the coordinate system of the photogrammetric model can be chosen arbitrarily. The rotation matrix $^m\!R$ is only identical to the rotation matrix R_1 from the scene into the camera coordinate system of the first camera if the model coordinate system is chosen accordingly.

model scale

For an optimal estimation we need the covariance matrices of the scene and the model points.

Once enough points are given in both systems, we can determine the transformation using the methods discussed in Sect. 10.5, p. 395. However, the direct least squares solution presented in Sect. 10.5.4.3, p. 408 is only an *approximate* one in our situation, even if the scene points are assumed to have zero variance, in which case we would use the inverse transformation of (13.292).

The reason is simple: the uncertainty of the model coordinates depends on the uncertainty of pairs of corresponding image points *and* on the uncertainty of the relative orientation of the images of the two cameras. While the point pairs may realistically be assumed to be mutually uncorrelated, the uncertainty of the relative orientation affects *all* 3D points of the model. They therefore are all mutually correlated. As the confidence ellipsoids generally are not spherical, see Example 13.4.1.4, p. 604, the covariance matrix will be full and not block diagonal with $\Sigma = \mathrm{Diag}(\{\sigma_i^2 I_3\})$. Therefore the determination of the absolute orientation with the direct least squares solution or any other technique which neglects the correlations between the 3D model points is statistically suboptimal. The difference to an optimal estimate, which we will discuss in the next section, will be acceptable if enough points are used for the determination. However, testing the absolute orientation using an approximate stochastical model may easily lead to totally incorrect decisions if the number of points is not large or if the distribution of the 3D points is not homogeneous.

direct LS solution is suboptimal

13.6 Orientation of the Image Pair and Its Quality

13.6.1 Overview 608
13.6.2 Comparison of the Solutions 610
13.6.3 Synopsis 614

13.6.1 Overview

We now compare different procedures for orienting two images and discuss their quality in order to provide decision criteria for adequately choosing from the procedures presented above in a particular situation.

13.6.1.1 Basic Setup for Comparison

We assume the following situation (Fig. 13.19):

- We have two images of N_O object points. They are not necessarily visible in both images nor do we necessarily know their 3D coordinates.

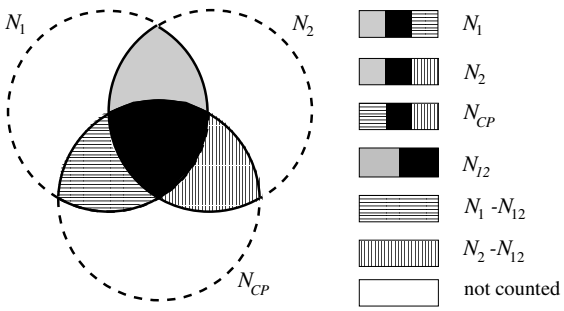


Fig. 13.19 Number of points for orienting two images. The number of points in the two images is N_1 (grey, black, horizontally hashed) and N_2 (grey, black, vertically hashed). Observations which appear only in one image and do not refer to control points are not counted (white). The number of tie points is N_{12} (grey, black). They are either pure tie points (grey) or common control points (black). The number of pure tie points is identical to the number of new points N_{NP} (grey). The number of control points N_{CP} contains points visible in one image (hashed) or both images (black). The number of all observed scene points N_O is identical to the number of all points in the nonwhite area. Observe, the symmetry of the figure suggests the set of control points can be interpreted as a third image, a 'ground image'

  - We observe $N_1 \leq N_O$ points in the first image. We only count points for which we observe a corresponding point in the other image or for which 3D coordinates are available.
  - Likewise, we accept $N_2 \leq N_O$ points observable in the second image.
  - We have $0 \leq N_{12} \leq \min(N_1, N_2)$ points observed in both images. These are *tie points*.

tie points

  Obviously $N_O = N_1 + N_2 - N_{12}$. The most favourable case, where all object points are observed in both images, occurs when $N_1 = N_2 = N_{12} = N_O$.
- Some of the object points, namely N_{CP}, have known coordinates and are control points. We distinguish between (see Fig. 13.19):
  - Full control points. Their number is N_{FCP}, with all three coordinates $[X, Y, Z]$ known.

  - Planimetric control points. Two of their coordinates, namely $[X, Y]$, are known. Their number is N_{PCP}. Geometrically, each defines a 3D line perpendicular to the XY plane.
  - Height control points, whose Z-coordinate is known. Their number is N_{HCP}. Geometrically, each defines a horizontal plane.

  This distinction between full and partial control points is useful in bundle adjustment for one or two images or in absolute orientation.
  Thus we have $N_{CP} = N_{FCP} + N_{PCP} + N_{HCP}$ control points, and the number of new unknown points is $N_{NP} = N_O - N_{CP}$.
  We do not discuss here the possible use of control lines or control planes in arbitrary pose.
- We assume the correspondences of the image and the object points to be correct in general. However, we need to expect at least a few outliers; when applying automatic image analysis techniques, the percentage might be large. Robust techniques, such as RANSAC or M-type estimation (see Sect. 4.7, p. 141), may be able to detect and locate these blunders depending on their size, their number, and their distribution in relation to the good observations.
- The observed image coordinates may be assumed to be uncorrelated and of equal precision. This is a practical assumption, especially when there is no information on the measurement procedure. Otherwise, especially when using automatic image analysis techniques for mensuration, they might provide information about the measurement precision, which then should be used to advantage, as discussed in Sects. 12.2.1, p. 490 and 13.3.1, p. 569.

13.6.1.2 Quality Criteria

The procedures have different qualities. For their evaluation and comparison we use two quality criteria:

- The *precision of estimated object coordinates and orientation parameters.*
  We are interested in whether the procedures are optimal in a statistical sense. If the assumed functional and stochastical models are correct and an ML estimation is applied the results are designated optimal. With any violation of this assumption, we obtain suboptimal results, especially if not all geometric constraints are used, or, equivalently, if more parameters are determined than necessary. This is also the case if the procedure does not use all the necessary statistical information, be it information about the precision or about mutual correlations.
- The *checkability of the observations.*
  Checkability depends on the redundancy $R = N - U + H$, where N is the number of observations, U the number of unknown parameters, and H the number of constraints among the unknown parameters. Here we only discuss the necessary condition $R > 0$ for checkability and the qualitative differences between the procedures with respect to the redundancy. We do not use the detailed analysis based on the redundancy numbers or the analysis of the sensitivity of the result with respect to nondetectable errors (see Sect. 4.6.2, p. 117).

We discuss four procedures for orienting two images:

1. One-step procedure with bundle adjustment.
2. Independent direct estimation of the projection matrix (DLT) for each image.
3. Independent spatial resection (SRS) for each image.
4. Two-step procedure with a relative and an absolute orientation.

The main results are summarized in Tables (13.7), p. 614 and (13.8), p. 614. We start with the bundle solution as it is the most general case.

13.6.2 Comparison of the Solutions

13.6.2.1 Bundle Adjustment for the Image Pair

The bundle solution, whose techniques will be discussed in Chap. 15, p. 643, simultaneously determines the orientation of the two bundles of rays and the unknown coordinates of the object points (Fig. 13.20). It is optimal since it exploits all available information and takes the stochastic properties of the observations into account, and since this simultaneous integration is easily realized, the bundle adjustment is superior to the other procedures.

The bundle adjustment, as well as absolute orientation, can treat control points as unknowns and use observations of them simultaneously, which offers the advantage that the observed coordinates of the control points can be tested. For simplicity, however, we here treat these coordinates as fixed given values. This has no effect on the redundancy, and therefore on the comparison of the procedures as described below.

The mathematical model is the following. We observe the coordinates of the $N_1 + N_2$ image points $\underline{x}_{it}$ of the two images taken, $t = 1, 2$, the index t possibly representing time. We want to simultaneously determine the two projection matrices P_t containing unknown parameters $\boldsymbol{p}_t$ of the exterior orientation in R_t and $\boldsymbol{Z}_t$, and possibly also unknown parameters $\boldsymbol{s}_t$ of the interior orientation in K_t. The projection matrices P_t thus explicitly depend on parameters $\boldsymbol{s}_t$ for the interior and parameters $\boldsymbol{p}_t$ for the exterior orientation. They may be known, partially unknown, or completely unknown. In addition, the intrinsic parameters of the images may be assumed to be the same for all images. This leaves enough degrees of freedom for realistic modelling.

bundle adjustment with points

The relation between the image coordinates and the coordinates of the corresponding scene points $\mathcal{X}_i$ is given by

$$\mathbb{E}(\underline{\mathbf{x}}'_{it}) = \lambda_{it}\mathsf{P}_t\,\mathbf{X}_i = \lambda_{it}\mathsf{K}_t R_t[I| - \boldsymbol{Z}_t]\,\mathbf{X}_i \tag{13.294}$$

$$\mathbb{D}(\underline{\boldsymbol{x}}'_{it}) = \Sigma_{x'_{it}x'_{it}}\,, \qquad i = 1, ..., N_t, t = 1, 2\,. \tag{13.295}$$

Eq. (13.294) refers to two different types of scene points:

- Scene points which are visible in both images. Some of them may be control points. Most of them will be tie points, whose coordinates are unknown. The observation equations enforce the intersection of corresponding projection rays in one object point.
- Scene points which are visible in only one of the two images and need to be control points.

The model is a generalization of the optimal estimation of the pose of a single view: it refers to two images instead of one and some of the 3D points may be fully unknown, a situation which cannot occur when orienting a single view.

We may iteratively estimate the unknown orientation parameters and the unknown 3D elements in an optimal way, possibly with constraints between the unknown parameters. This *requires approximate values* for the parameters $\widehat{\boldsymbol{p}}_t$ and $\widehat{\boldsymbol{s}}_t$, and some prior knowledge about the precision of the observed image points $\mathbf{x}'_{it}$.

The number of unknown parameters depends on the assumed camera model and the number of individual control point types. The number U_{EO} of the unknown parameters of the exterior orientation is 12, as we have six parameters for each camera. Furthermore we may have unknown interior orientation parameters:

- If the interior orientation of both cameras is assumed to be known, the number of parameters U_{IO} of the interior orientation is zero.
- If the interior orientation is unknown but the cameras are straight line-preserving, we have $U_{IO} = 10$ as each camera requires five parameters.
- If the interior orientation of the two straight line preserving cameras is unknown but identical, we have $U_{IO} = 5$.

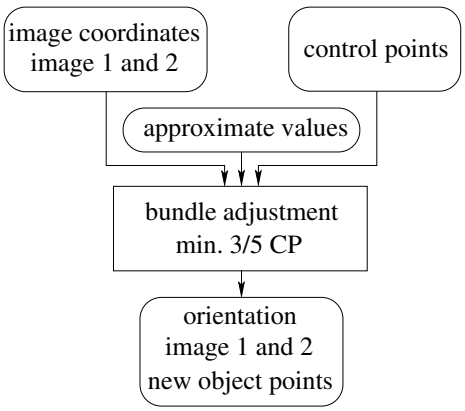


Fig. 13.20 Bundle solution for the orientation of the image pair. For calibrated cameras we need at least three control points, while for uncalibrated straight line-preserving cameras we need at least five control points

The number of unknown coordinates of new points and control points is $U_{coor.} = 3 \times N_{NP} + 2 \times N_{HCP} + N_{PCP}$. We have $N = 2 \times (N_1 + N_2)$ observed image coordinates. Therefore the total redundancy of the bundle adjustment for the orientation of two images is

$$R = 2 \times (N_1 + N_2) - (U_{EO} + U_{IO} + 3 \times N_{NP} + 2 \times N_{HCP} + N_{PCP}) \,, \quad (13.296)$$

which necessarily needs to be ≥ 0.

Moreover, the minimum number of control points is three or five, depending on whether the cameras are calibrated or uncalibrated straight line-preserving, as without control points the image pair can only be determined up to a spatial similarity or projective transformation with seven or 15 degrees of freedom, respectively, which have to be fixed by control points. If only 3D lines are used as control, we only need two and four control lines for calibrated and straight line-preserving cameras, respectively.

13.6.2.2 Independent Direct Linear Transformations

We use the same observation equations as in the bundle solution, namely $\mathbf{x}'_{it} = \mathsf{P}_t \mathbf{X}_i$, but just for full control points observed in each image, see Fig. 13.21.

The solution is not optimal in general, especially if new points are observed or some information about the interior orientation, beyond their property of it being straight line preserving, is known. However, this information cannot be used to improve the orientation, as is possible for bundle adjustment.

The number of observations is $N = 2 \times (N_1 + N_2 - 2N_{NP})$ as the N_{NP} new points measured in both images cannot be used. The number of unknown parameters is $U = 22$. Therefore the redundancy is

$$R = 2 \times (N_1 + N_2 - 2N_{NP}) - 22 \,. \quad (13.297)$$

The DLT cannot be used for orientation if the object points lie in a plane, and if the object points are nearly coplanar the solution is unstable, which is a strong drawback (Tables 13.7 and 13.8). The direct algebraic solution does not exploit the statistical properties of the image coordinates. With the minimal number six of control points in each image, we have a redundant system with $R = 1$.

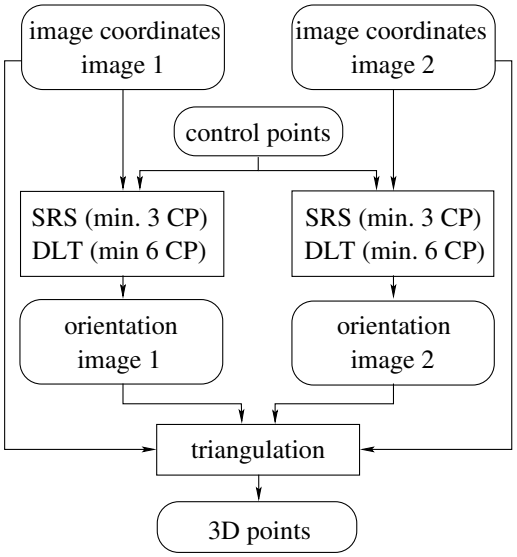


Fig. 13.21 Orientation of the image pair by determining the orientations of the two images separately: spatial resection (SRS) for calibrated and direct linear transformation (DLT) for uncalibrated straight line-preserving cameras

13.6.2.3 Independent Spatial Resections

The same observation equations are used again, namely $\mathbf{x}'_{it} = \mathsf{K}_t \mathsf{R}_t [I| - \boldsymbol{Z}_t]\mathbf{X}_i$, only for full control points observed in each image. Control lines may also be included. Moreover, only six parameters are unknown, namely the rotation parameters in R_t and the projection centre $\boldsymbol{Z}_t$. As in the case of two independent DLTs, the solution generally is suboptimal, especially if we have new points observed in both images. The result is optimal only if no new points are observed in the two images.

With a similar argument as in the previous case, the redundancy is

$$R = 2 \times (N_1 + N_2 - 2N_{NP}) - 12\,. \tag{13.298}$$

The direct algebraic solution of the SRS with three points leads to up to four solutions for each image. If only three control points are available and if they represent the correct choice, the direct solution is optimal. The projection centre must not lie on the circular cylinder through the three control points (Table 13.8).

13.6.2.4 Two-Step Procedure with Relative and Absolute Orientation

The orientation is performed in two steps, see Fig. 13.22:

1. *relative orientation* using image information only, including the determination of 3D points of the photogrammetric model, and
2. *absolute orientation* of the photogrammetric model utilizing the control points.

Relative Orientation. Relative orientation (RO) for images of straight line-preserving cameras applies the coplanarity constraint

$$\mathbf{x}_i{}'^{\mathsf{T}} \mathsf{F} \mathbf{x}''_i = 0 \qquad i = 1, ..., N_{12} \tag{13.299}$$

to determine the fundamental matrix F from the image coordinates $\mathbf{x}'_i$ and $\mathbf{x}''_i$ of all corresponding points, new points, and control points, or, with calibrated cameras,

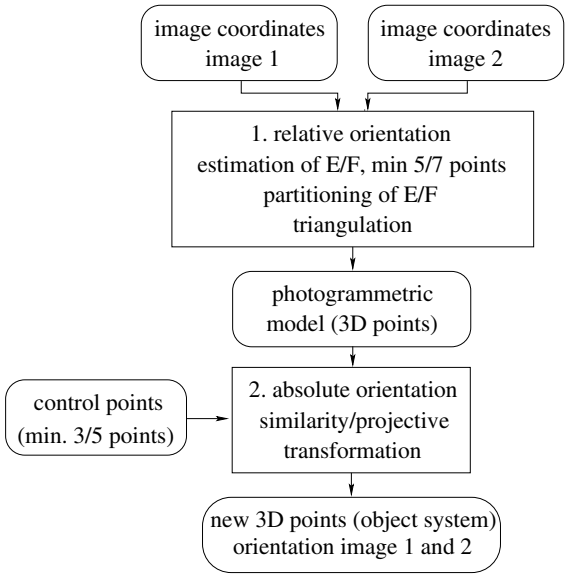


Fig. 13.22 Two-step procedure for the orientation of the image pair of calibrated cameras or of uncalibrated straight line-preserving cameras

$$ {}^c\mathbf{x}_i'^{\mathsf{T}} \mathsf{E} {}^c\mathbf{x}_i'' = 0 \qquad i = 1, ..., N_{12} \,, \tag{13.300} $$

to determine the essential matrix E from the direction vectors ${}^c\mathbf{x}_i' = \mathsf{K}_1^{-1}\mathbf{x}_i'$ and ${}^c\mathbf{x}_i'' = \mathsf{K}_2^{-1}\mathbf{x}_i''$ to the image points in the camera coordinate system. The number of observations is $N = N_{12}$, and the number of unknowns U_{RO}, five or seven, depends on the type of solution. The redundancy is

$$ R = N_{12} - U_{RO} \,. \tag{13.301} $$

The six-, seven- and eight-point solutions for the determination of F or E do not allow the object points to lie on a plane (Tables 13.7 and 13.8). The five-point solution of Nistér (2003) for E can handle this case even if more than five points are used, except when the base vector is perpendicular to the scene plane.

Relative orientation does not use the control information and therefore is not optimal in general, namely if more than the minimum control point configuration is available.

Absolute Orientation. Absolute orientation for uncalibrated straight line-preserving cameras uses the 3D homography

$$ \underset{4\times1}{\mathbf{X}_i} = \underset{4\times4}{\mathsf{H}_m} \; \underset{4\times1}{{}^m\mathbf{X}_i} \qquad i = 1, ..., N_{CP} \tag{13.302} $$

between the model coordinates ${}^m\mathbf{X}_i$ and the object coordinates $\mathbf{X}_i$. It requires $N_{CP} \geq 5$ full control points for the $U_{AO} = 15$ parameters of the 3D homography H_m.

For calibrated cameras, we use the 3D similarity transformation

$$ \boldsymbol{X}_i = \lambda_m \, R_m \, ({}^m\boldsymbol{X}_i - \boldsymbol{T}) \qquad i = 1, ..., N_{CP} \,, \tag{13.303} $$

which requires $N_{CP} \geq 3$ control points for the $U_{AO} = 7$ parameters. The given direct least squares solution with independent points is always suboptimal as the mutual correlations between the points of the photogrammetric model are neglected.

The redundancy is

$$ R = 3 \times N_{CP} - U_{AO} \,. \tag{13.304} $$

If an iterative optimal estimation procedure is chosen, partial control points can be used. For calibrated cameras, at least seven control point coordinates are necessary, e.g., two planimetric and three height control points.

13.6.3 Synopsis

Tables 13.7 and 13.8 collect the main properties of the orientation procedures for image pairs. First we give necessary constraints for the number N_{CP} of control points and N_{12} of tie points. A nonnegative redundancy R is always required for obtaining a solution. For the bundle procedure, no general constraint on the necessary number N_{12} of tie points can be given, indicated by the dash. Here the only constraint is $R \geq 0$. Not all critical configurations are given in the tables.

Table 13.7 Properties of procedures for orienting two uncalibrated straight line-preserving cameras, with number U of unknown orientation parameters, lower bound on number N_{CP} of control points, number N_{12} of corresponding image points. Direct linear transformation (DLT). Critical configurations: (a) coplanar object points, (b) twisted cubic curve also containing the projection centre, (c) ruled quadrics, especially cylinders, also containing the projection centres, (d) $N_{CP} - 1$ control points coplanar. Existence of direct solution procedures including maximum number of solutions. Critical configurations for the bundle solution generally cannot be characterized easily; only one is given

procedure	Eq.	U	N_{CP}	N_{12}	crit. conf.	direct sol.
one-step: bundle solution	(13.294)	22	≥ 5	–	(d)	no
2 × DLT	(12.116)	22	≥ 6	–	(a), (b)	yes (1)
two-step		22				
1. relative orient.						
direct (F)	(13.95)	7		≥ 8	(c) incl. (a)	yes (1)
direct (F)	(13.102)	7		≥ 7	(c) incl. (a)	yes (3)
2. absolute orient.	(13.302)	15	≥ 5		(d)	yes (1)

Table 13.8 Properties of procedures for orienting two calibrated cameras. Number U of unknown orientation parameters, minimum required number N_{CP} of control points, number N_{12} of corresponding image points. Spatial resection (SRS). Critical configurations: (a) coplanar object points, (b) projection centre on cylinder (only for three-point solution), 3D horopter curve also containing the projection centre, (c) ruled quadric containing the projection centres, (d) all control points collinear, (e) orthogonal ruled quadric, especially cylinder, also containing case where the scene is planar and perpendicular to basis, (f) $N_{12} - 1$ object points collinear. Existence of direct solution procedures, including maximum number of solutions, in brackets. Procedures not given explicitly in this book are indicated with *. Again, only one example for a critical configuration for the bundle adjustment is given

procedure	Eq.	U	N_{CP}	N_{12}	crit. conf.	direct sol.
one-step: bundle solution	(13.294)	12	≥ 3	–	(d)	no
2 × SRS	(12.222)	12	≥ 3	–	(b)	yes (4)
two-step		12				
1. rel. orient.						
direct (E)	(13.123)	5		≥ 8	(c) incl. (a)	yes (1)
direct (E)	(13.128)	5		≥ 5	(e)	yes (10)
direct, planar *	–	5		≥ 4	(f)	yes (2)
direct (given R)	(13.146)	2		≥ 2	-	yes (1)
direct (symmetric)	(13.154)	2		≥ 2	-	yes (1)
iterative	(13.186)	5		≥ 5	(e)	
2. absolute orient.	(13.292)	7	≥ 3		(d)	yes

The different procedures can be evaluated as follows:

- The one-step procedure using the bundle adjustment always yields optimal results but requires good approximate values. A robust estimation procedure can handle a small percentage of sufficiently small blunders.
- The two-step procedure with a relative and an absolute orientation allows separate testing of image measurements and control point measurements, especially outliers caused by a wrong identification. It provides good approximate values for a bundle adjustment. Due to the small number of unknowns, the relative and the absolute orientation can handle large blunders using RANSAC. The precision of the two-step procedure may be sufficient for certain purposes. If image coordinates of the control points are only available in one of the two images, no absolute orientation can be performed, as the length of the base vector cannot be determined.
- The solution with two independent spatial resections is applicable only if at least three control points are observable in both images. This generally requires good approximate values. In order to overcome the fourfold ambiguity of the direct solution with three points, at least four control points observed in the two images or good approximate values for the orientation are necessary.

  If many new points could be used for orientation, the precision of the orientation with two separate spatial resections is significantly lower than that for the two-step procedure, as the epipolar constraints for the corresponding new points are not exploited.

  If many control points are available, independent spatial resection is useful for blunder detection using RANSAC, since a direct solution with three points is available.
- The solution with two independent direct linear transformations is applicable only if at least six control points are observable in the two images and if these control points are not coplanar.

  This method is useful for getting approximate values for a bundle solution only for uncalibrated straight line-preserving cameras if enough noncoplanar control points are available. Otherwise, the solution is not precise enough.

  As the solution requires at least six control points, blunder detection is more involved and less secure compared to separate spatial resections.

13.7 Exercises

Basics

1. (1) Your colleague claims that he can estimate distances just by taking a bearing with his thumb. He demonstrates this with a parked car, see Fig. 13.23 Is this possible? Is

Fig. 13.23 Images taken with the left and the right eye. Can you derive the distance from the car from these two images?

   further information necessary? Argue with techniques and concepts of the image pair. If possible give equations for estimating the distance.
2. (1) Two synchronised video angle cameras are mounted on a car, say in the middle of the roof. Camera A is looking ahead. Camera B is looking to the right side. Their viewing angle is large enough such that they have overlap. The car moves forward.

a. Where are the epipoles of the camera pair?
b. Where are the epipoles of consecutive images in the video stream for cameras A and B?
c. Where are the epipoles of the camera pair if camera A is tilted a few degrees downwards and camera B looks more forward?

Hint: Draw a sketch.

3. (2) You can find a Java applet for inspecting the epipolar geometry of the image pair at `HOME/epipolar-geometry.html`[8], cf. Fig. 13.24.

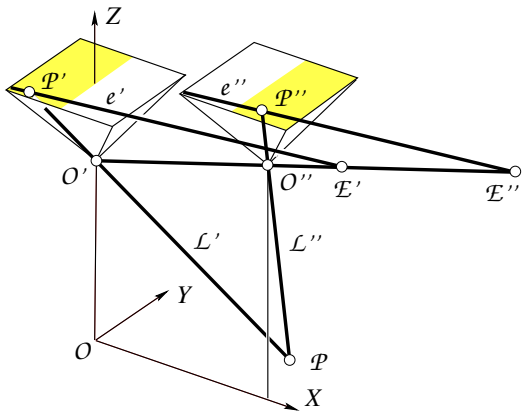


Fig. 13.24 Epipolar geometry of the image pair. The two projection centres have different height

Confirm that all elements of the epipolar geometry lie in the epipolar plane by moving the point $\mathcal{P}$.

a. Search for a configuration where the epipolar lines within one image are nearly parallel. Describe the configuration.
b. Search for a configuration where the epipolar lines pass through the principal point. Describe the configuration.
c. Search for a configuration where the bright area/yellow in the left image is mapped exactly to the bright/yellow area in the right image. What does this configuration tell you about the height of the point $\mathcal{P}$?
d. Inspect the configuration where the principal distance is negative. Which image areas are corresponding?

4. (1) Given an image point $\mathcal{X}'$ in the left image, the image point in the second image $\mathcal{X}''$ lies on the epipolar line. This holds for noiseless data. Investigate the uncertainty area around the epipolar line in more detail using the Java applet from Exerc. 3. Assume that both cameras have the same calibration matrix. Investigate the effect of the following – possibly random – changes, while all other conditions remain unchanged.

a. The measurement of the coordinates $\boldsymbol{x}'$ is uncertain.
b. The principal point $\mathcal{H}'$ in the first image is uncertain.
c. The rotation ω'' around the viewing direction of the second camera is uncertain.

5. (1) Show that the three parameters of the rotation between two calibrated cameras can be determined if the basis is zero. *Hint:* Explore (13.215), p. 589. Explain why in general the eight point algorithm for determining the relative orientation does not break down if the true basis is zero.

[8] See Sect. 1.3.2.4, p. 16.

6. (1) Given an essential matrix E, so is E^T also an essential matrix? If not, explain what conditions for an essential matrix it does not fulfil. If yes, what does this matrix mean and why?
7. (1) Given is the fundamental matrix

$$\mathsf{F} = \begin{bmatrix} -2 & -4 & 12 \\ 6 & -2 & -8 \\ 8 & -4 & -8 \end{bmatrix}.$$

   How can you check whether an image point is an epipole? Determine the two epipoles and check whether the constraint for epipoles is fulfilled.
8. (1) Given are two ideal cameras with $c = 1000$ pixels in normal position with a base line of 30 m. Assume that two corresponding points $\mathcal{x}'$ and $\mathcal{x}''$ have been measured with a known precision. Assume the relative orientation is fixed.
   a. Give the fundamental matrix F.
   b. Give a statistically optimal procedure for checking whether the two image points are corresponding. What assumptions have you made about the measurement precision?
   c. Is the procedure also optimal if the coordinates $\boldsymbol{x}'$ of the first point and the parallax $\boldsymbol{p} = \boldsymbol{x}'' - \boldsymbol{x}'$ are measured? If not, which procedure is optimal?
9. (1) For three images taken with a calibrated camera, you have the two essential matrices E_{12} and E_{13}. Can you derive the essential matrix E_{23}? Why?
10. (2) The task is to determine the elements of the fundamental matrix

$$\mathsf{F} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}.$$

    a. Given are the coordinates of corresponding image points,

$$[\mathbf{x}'_a, \mathbf{x}''_a] = \begin{bmatrix} 1 & 0.815 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad [\mathbf{x}'_b, \mathbf{x}''_b] = \begin{bmatrix} 0 & 0 \\ 1 & 0.815 \\ 0 & 0 \end{bmatrix}.$$

       i. Where do the points $\mathbf{x}'_a$ and $\mathbf{x}'_b$ lie in image 1?
       ii. Which elements of the fundamental matrix result from the coplanarity condition for the two points $\boldsymbol{a}$ and $\boldsymbol{b}$. Determine these elements.
    b. In addition you are provided with the coordinates of the two epipoles $\boldsymbol{e}' = [5, 0]^\mathsf{T}$ and $\boldsymbol{e}'' = [-5, 0]^\mathsf{T}$. Use the constraints for the epipoles to determine the matrix F up to two parameters.
    c. Finally, you obtain a third pair of homologous points,

$$[\boldsymbol{x}'_c, \boldsymbol{x}''_c] = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

       Determine the remaining parameters of the fundamental matrix.

11. (2) Determine the essential matrix using the following information: The two rotation and calibration matrices are

$$R' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \quad R'' = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathsf{K}' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathsf{K}'' = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (13.305)$$

    and the corresponding points are

$$\boldsymbol{x}'_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \boldsymbol{x}''_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}'_2 = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \qquad \boldsymbol{x}''_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} .$$

(You do not need a computer!)

Methods

12. (2) Determine the 3D point closest to two 3D lines given in point–direction form. Given

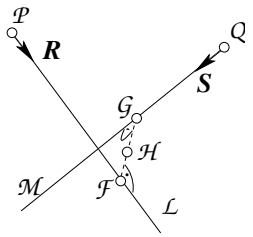


Fig. 13.25 Determining the point $\mathcal{H}$ closest to the two projecting rays $\mathcal{L}$ and $\mathcal{M}$

are two points $\mathcal{P}(\boldsymbol{P})$ and $\mathcal{Q}(\boldsymbol{Q})$ and the corresponding normalized directions $\boldsymbol{r}$ and $\boldsymbol{s}$, see Fig. 13.25. The point $\mathcal{H}$ closest to the two lines $\mathcal{L}(\boldsymbol{P} + \lambda\boldsymbol{R})$ and $\mathcal{M}(\boldsymbol{Q} + \mu\boldsymbol{S})$. Show that it is given as the midpoint of $\mathcal{F} \in \mathcal{L}$ and $\mathcal{G} \in \mathcal{M}$, with

$$\begin{bmatrix} \boldsymbol{R}\cdot\boldsymbol{R} & -\boldsymbol{R}\cdot\boldsymbol{S} \\ -\boldsymbol{R}\cdot\boldsymbol{S} & \boldsymbol{S}\cdot\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} (\boldsymbol{Q}-\boldsymbol{P})\cdot\boldsymbol{R} \\ -(\boldsymbol{Q}-\boldsymbol{P})\cdot\boldsymbol{S} \end{bmatrix} . \tag{13.306}$$

13. (1) You have an image pair taken with a calibrated camera. The calibration matrix is given. For taking the first image, the camera is mounted on a tripod positioned on a horizontal plane and oriented such that the viewing direction is horizontal. For taking the second image, (i) the tripod is moved horizontally in the X-direction of the camera, (ii) the height of the camera is changed, and (iii) the camera is tilted by 40° around the (vertical) Y-axis of the camera.

    a. How many homologous points do you need to determine the epipolar geometry of the two images taking all given information into account? Sketch a procedure for solving the problem in no more than three short sentences.
    b. Does the described procedure always give a solution?

14. (2) We have described several algorithms for the relative orientation of two images. Given a certain number of homologous points, characterize the suitability of the given algorithms in the following table:

	Name of algorithm	unique solution	direct solution	required constraints	critical configurations
9 points					
8 points					
7 points					
6 points					
5 points					
4 points					

15. (2) The task is to compare the applicability of the different methods for orienting the image pair. Discuss both cases, namely whether the bundle adjustment adopts calibrated cameras or assumes both cameras to be perspective with possibly different interior orientations. Compare five different methods.

Given is the following information:

- Image coordinates in image 1 for the points 10–18, 20–25, 27–30.
- Image coordinates in image 2 for the points 10–16, 20–25, 27, 28, 30.
- Scene coordinates $\boldsymbol{X}_i$ for the points 11, 15, 17, 18, 21, 26, 27, 31.

a. Draw a diagram similar to the one in Fig. 13.19, p. 608. Determine the numbers N_1, N_2, N_{12}, N_0, N_{CP}, and N_{NP}.
b. Evaluate the suitability of the different procedures for orienting the image pair. Among other criteria, use the relative redundancy $\bar{r} = R/N$ for the evaluation:

$$\begin{aligned} \bar{r} &< 0.1 \rightarrow \text{bad} \\ 0.1 \le \bar{r} &< 0.5 \rightarrow \text{moderate} \\ 0.5 \le \bar{r} &< 0.8 \rightarrow \text{good} \\ 0.8 \le \bar{r} &\le 1.0 \rightarrow \text{very good} \end{aligned}$$

Proofs

16. (2) Prove (13.286), p. 606. *Hint:* Express the coordinates of the image lines and the epipoles as a function of $\mathbf{L}$ and the projection centres and use the result from Exerc. 10, p. 538.
17. (2) Prove (13.288), p. 606. *Hint:* Determine the projected line $\mathbf{m}' = \mathsf{Q}_1(\mathbf{A}_{l''} \cap \mathbf{A}_{l'})$ and compare it to $\mathbf{l}'$. Use the representation of the three elements by the camera planes, e.g., $\mathsf{Q}_1^\mathsf{T} = [\mathbf{B}_1 \wedge \mathbf{C}_1, \mathbf{C}_1 \wedge \mathbf{A}_1, \mathbf{A}_1 \wedge \mathbf{B}_1]$ and e.g., $\mathbf{A}_{l'} = l'_1\mathbf{A}_1 + l'_2\mathbf{B}' + l'_3\mathbf{C}_1$, and the representation of the epipoles in (13.72), p. 565.
18. (1) Using (13.285), p. 605, show that the Jacobian of the line parameters $\mathbf{L}$ w.r.t. the vector $\boldsymbol{y} = [\mathbf{l}'^\mathsf{T}, \mathbf{l}''^\mathsf{T}, \mathbf{p}_1^\mathsf{T}, \mathbf{p}_2^\mathsf{T}]^\mathsf{T}$ of the observed image lines $\mathbf{l}'$ and $\mathbf{l}''$ and the parameters $\mathbf{p}_i = \mathrm{vec}\mathsf{P}_i, i = 1, 2$, is given by

$$\frac{\partial \mathbf{L}}{\partial \boldsymbol{y}} \left[\overline{\mathsf{T}}(\mathsf{P}_2^\mathsf{T}\mathbf{l}'')\mathsf{P}_1^\mathsf{T} \mid \overline{\mathsf{T}}(\mathsf{P}_1^\mathsf{T}\mathbf{l}')\mathsf{P}_2^\mathsf{T} \mid \mathbf{l}'^\mathsf{T} \otimes \overline{\mathsf{T}}(\mathsf{P}_2^\mathsf{T}\mathbf{l}'') \mid \mathbf{l}''^\mathsf{T} \otimes \overline{\mathsf{T}}(\mathsf{P}_1^\mathsf{T}\mathbf{l}') \right] . \tag{13.307}$$

19. (2) Show that the two solutions for the rotation matrices derived from a mirror image (13.154), p. 580 differ by a rotation of 180° around an axis perpendicular to the basis. *Hint:* Show that the two rotation quaternions $\mathbf{q}_1^{[1]}$ and $\mathbf{q}_1^{[2]}$ have a ratio of $[0, 0, w, -v]/(1-u)$.

Computer Experiments

20. (2) Given are the image coordinates of two homologous points,

$$(x', y')_P = (474.14, -761.97) \text{ pixel} \qquad (x'', y'')_P = (626.52, -1211.41) \text{ pixel}$$

and

$$(x', y')_Q = (-455.20, -225.49) \text{ pixel} \qquad (x'', y'')_Q = (-298.70, -284.16) \text{ pixel},$$

observed in two images taken with an ideal camera with principal distance $c = 1530$ pixels. Furthermore, you have the two rotation matrices

$$R' = \frac{1}{57}\begin{bmatrix} 44 & -28 & 23 \\ 32 & 47 & -4 \\ -17 & 16 & 52 \end{bmatrix} \qquad R'' = \frac{1}{57}\begin{bmatrix} 52 & -16 & -17 \\ 4 & 47 & -32 \\ 23 & 28 & 44 \end{bmatrix} .$$

a. Determine the base vector.

b. Draw a sketch of the situation. Draw the Z-axis of the two cameras. Assume the projection centres are in front of the images. Determine whether the first camera is left of the second camera. Determine the correct sign of the basis.

21. (2) Given are the two projection centres $\boldsymbol{Z}_t$ of an ideal camera with $c = 1530$ pixels and rotation matrices $R_t = I_3, t = 1, 2$, and the image coordinates ${}^i\boldsymbol{x}'_t, t = 1, 2$ of two measured corresponding points:

$$\boldsymbol{Z}_1 = \begin{bmatrix} 790 \\ 365 \\ 110 \end{bmatrix}, \quad \boldsymbol{Z}_2 = \begin{bmatrix} 782 \\ 365 \\ 110 \end{bmatrix}, \quad {}^i\boldsymbol{x}' = \begin{bmatrix} 140 \\ 913 \end{bmatrix}, \quad {}^i\boldsymbol{x}'' = \begin{bmatrix} -72 \\ 895 \end{bmatrix}.$$

Determine the 3D coordinates of the scene point $\mathcal{X}$

a. using the approximate solution for the normal case of the image pair,
b. as the 3D point closest to the two projection rays,
c. using the two constraints $\mathsf{S}(\mathbf{x}')\mathsf{P}_1\mathbf{X} = \mathbf{0}$ and $\mathsf{S}(\mathbf{x}'')\mathsf{P}_2\mathbf{X} = \mathbf{0}$, and
d. as the statistically optimal solution assuming the image coordinates to have isotropic uncertainty $\sigma = 0.5$ pixel.

Compare the solutions of the four methods. Take the statistically optimal solution as a reference. Change the y' and y'' coordinates by 300 pixels. What effect does this have on the consistency of the four solutions?

22. (2) Given are two images of a building (`HOME/ex-12-F-eins.jpg` and `...F-zwei.jpg`), see Fig. 13.26. Determine the fundamental matrix and verify it by drawing an epipolar line to a point.
Use a measuring tool with which you can identify and measure the sensor coordinates of image points, e.g., MATLAB's `[x,y] = ginput(1)`.
Hint: Avoid critical configurations. Apply conditioning.

Fig. 13.26 An image pair

23. (3) Derive an estimation scheme for triangulating a 3D point from two rays using the projection lines which is useful for more than two rays. Assume the directions $\mathbf{u}'_i := {}^c\mathbf{x}'^s_i$ in images $i = 1, ..., I$ that have the covariance matrices $\Sigma_{u'_i u'_i} = \sigma^2 I_3, i = 1, ..., I$.

a. Give the Plücker coordinates of the projection lines $\mathcal{L}_i = \mathcal{Z}_i \wedge \mathcal{T}_i$ with $\boldsymbol{T}_i = \boldsymbol{Z}_i + R_i^\mathsf{T}\boldsymbol{u}'_i$.
b. Give the covariance matrices of the Plücker coordinate vectors $\mathbf{L}_i$.
c. Give the constraint $\boldsymbol{g}_i = \boldsymbol{g}_i(\mathbf{L}_i, \mathbf{X}) = \mathbf{0}$ for the 3D point to lie on the line $\mathcal{L}_i$.
d. Express the selection of two independent constraints $\boldsymbol{g}_{r,i}$.
e. Follow the procedure for estimating the vanishing points in Sect. 10.6.2, p. 417.
f. Realize the estimation procedure and test it for correctness following Sect. 4.6.8, p. 139.

24. (3) Write an algorithm `RO` for the iterative solution of the relative orientation for a general configuration following Sect. 13.3.5, p. 585. Follow the development steps and tests of Exerc. 26, p. 540, possibly adapting them.

Chapter 14
Geometry and Orientation of the Image Triplet

14.1 Geometry of the Image Triplet 622
14.2 Relative Orientation of the Image Triplet 632
14.3 Exercises 641

This chapter discusses the basic geometry and orientation of image triplets. The higher redundancy caused by observing the scene in three instead of only two images, as before, leads to a number of advantages, so it is useful to treat the image triplet in detail.

We will first develop methods for predicting image features in an image, given the image features in the two other images, and establish constraints between corresponding image features in three images. These can then be used for finding and checking corresponding image features and for determining the orientation of the three images.

There are several reasons to analyse image triplets:

- Given three images, say 1, 2, and 3, the relative orientation of two pairs of them, say $(1, 2)$ and $(2, 3)$, does not tell us anything about the mutual scale of the resulting photogrammetric models.
- The relative orientation of three images gives constraints on all image coordinates involved. This is in contrast to the relative orientation of an image pair, which only gives constraints in one direction, in the normal case for the y-coordinates, while the x-coordinates cannot be checked at all.
- The relative orientation of image triplets can be based on both *corresponding points* and *corresponding lines*, in contrast to image pairs, where corresponding lines give no constraint for the relative orientation, see Fig. 14.1. Analogously to the image

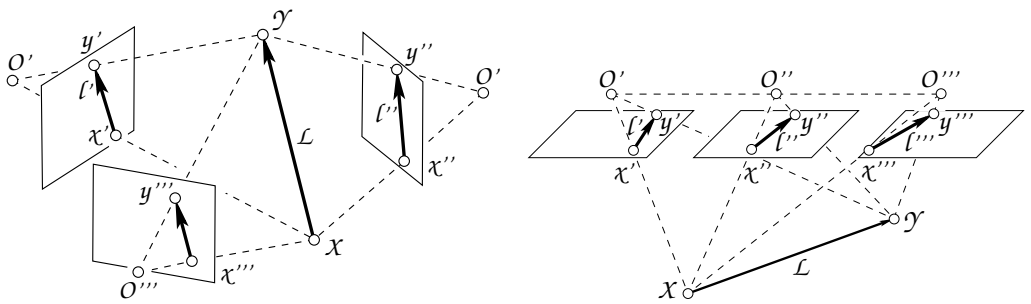


Fig. 14.1 The relative orientation of the image triplet is fully captured in the trifocal tensor. **Left:** general configuration. **Right:** Collinear projection centres. The trifocal tensor allows us to establish constraints between corresponding points (x', x'', x''') and lines (l', l'', l''') in three images. It also allows us to predict points and lines from two given points and lines in all feasible configurations without determining the 3D point X or 3D line L. This is possible also if the epipolar geometry would not be sufficient, namely if the 3D points and the three projection centres (O', O'', O''') are coplanar as in the important case of collinear projection centres as shown in the right figure

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_14

pair, the constraints between corresponding points and lines, respectively are linear in homogeneous coordinates of the entities. In addition, the constraints are linearly dependent on 27 parameters, collected in what is called the *trifocal tensor* T. The trifocal tensor is a $3 \times 3 \times 3$ array representing the complete geometry of the image triplet, in full analogy to the nine parameters of the fundamental matrix F which captures the complete geometry of the image pair.

- The prediction of points and lines from two images in the third can also be based on the trifocal tensor. This prediction is linear in its parameters and in the homogeneous coordinates of the points and lines. Similarly to the determination of the epipolar line in the case of the image pair, the prediction of a point or line in a third image can be performed without first determining the 3D point or 3D line.

  In the important situation of collinear projection centres (Fig. 14.1, right), the prediction of the point $\mathbf{x}'''$ in image 3 based on the points $\mathbf{x}'$ and $\mathbf{x}''$ in the first two images, obtained from the epipolar geometry of image pairs $(1,2)$ and $(1,3)$, is not possible using corresponding epipolar lines, as they are identical in the third image and do not give a unique prediction. Prediction using the trifocal tensor does not have this deficiency, as it implicitly works with the 3D point.

14.1 Geometry of the Image Triplet

14.1.1 Number of Parameters 622
14.1.2 The Coplanarity Constraints of the Image Triplet 623
14.1.3 The Trifocal Tensor 625
14.1.4 Predictions and Constraints for Points and Lines 629

We consider central cameras with distortion-free lenses. If the cameras are uncalibrated, we assume they are perspective, but straight line-preserving. They may have individual interior orientation. If the cameras are calibrated, we use the spherical camera model.

We first determine the degrees of freedom of the geometric entities involved, then derive constraints between corresponding points and lines, and finally establish expressions for predicting image features observed in two images in a third one.

14.1.1 Number of Parameters

Three *uncalibrated* perspective cameras require 33 parameters for their interior and exterior orientation, 11 for each camera. However, the reconstructable 3D scene, also called the photogrammetric model, can only be determined up to a straight line-preserving 3D transformation, a 3D homography, requiring 15 parameters. Therefore, the relative orientation of the image triplet with uncalibrated cameras requires $18 = 33 - 15$ independent parameters.

This is consistent with the number derived from a two step procedure: first, we perform the relative orientation of two images, requiring 7 parameters to determine the fundamental matrix, and obtain a photogrammetric model.

We then perform a direct linear transformation (DLT) of the third image, based on the 3D coordinates of the photogrammetric model and requiring 11 parameters for the exterior and interior orientation. This yields a total of $18 = 7 + 11$ parameters.

Three *calibrated* cameras require 18 parameters to represent their exterior orientation, six for each camera. Here the photogrammetric model can only be determined up to a similarity transformation requiring 7 parameters. Therefore the relative orientation of the image triplet with calibrated cameras can be described by $11 = 18 - 7$ independent parameters.

This number is again consistent with the one derived from a two step procedure: we first perform the relative orientation of two images, requiring five parameters for determining the essential matrix and yielding a photogrammetric model. We then perform a spatial resection of the third image, based on the 3D coordinates of the photogrammetric model, requiring six parameters for the exterior orientation. This yields a total of $11 = 5 + 6$ parameters.

The situation is summarized in Table 14.1.

Table 14.1 Number of parameters of the orientation of an image triplet (O=EO+IO), the relative orientation RO, the absolute orientation AO

camera	# O /image	# O /triplet	# RO param.	# AO param.
calibrated	6	18	11	7
straight line-preserving	11	33	18	15

While the parametrization of the image triplet with calibrated cameras is straightforward, the parametrization of an image triplet with uncalibrated cameras is not simple. In this section, however, we only derive explicit expressions for the trifocal tensor as a function of projection matrices, as well as constraints and predictions for corresponding points and lines.

14.1.2 The Coplanarity Constraints of the Image Triplet

We start with the generic situation where the three projection centres are not collinear. The three projection centres $\mathcal{O}'$, $\mathcal{O}''$, and $\mathcal{O}'''$ then uniquely span the *trifocal plane*, see Fig. 14.2 *trifocal plane*

Let the three projections be given by

$$\mathcal{X}' = \mathcal{P}'(\mathcal{X}): \quad \mathbf{x}' = \mathsf{P}_1\mathbf{X} \tag{14.1}$$
$$\mathcal{X}'' = \mathcal{P}''(\mathcal{X}): \quad \mathbf{x}'' = \mathsf{P}_2\mathbf{X} \tag{14.2}$$
$$\mathcal{X}''' = \mathcal{P}'''(\mathcal{X}): \quad \mathbf{x}''' = \mathsf{P}_3\mathbf{X}\,, \tag{14.3}$$

with projection matrices $\mathsf{P}_t = \mathsf{K}_t R_t [I_3 | -\boldsymbol{Z}_t] = [A_t | \boldsymbol{a}_t], t \in \{1, 2, 3\}$.

Remark: Elements of the three images are denoted either by primes or by numbers. If these are variables where the order does not matter, we use $t = 1, 2, 3$. If we refer to pairs or triplets of indices, where the order is important, we use $i, j, k \in \{1, 2, 3\}$ in order to avoid double indices. ◇

In the general case, the prediction of points can be based on the epipolar geometry of two image pairs. We have three fundamental matrices,

$$\mathsf{F}_{ij} = \mathsf{A}_i^{-\mathsf{T}} \mathsf{S}_{\mathrm{b}_{ij}} \mathsf{A}_j^{-1} \qquad (i, j) \in \{(1, 2), (2, 3), (3, 1)\}\,, \tag{14.4}$$

with $\mathbf{b}_{ij} = \boldsymbol{Z}_j - \boldsymbol{Z}_i$.

Let the two points $\mathcal{X}''$ and $\mathcal{X}'''$ in the second and the third images be given. Then the intersection of the epipolar lines in the first image,

$$\mathbf{l}'(\mathbf{x}'') = \mathsf{F}_{12}\mathbf{x}'' \qquad \mathbf{l}'(\mathbf{x}''') = \mathsf{F}_{13}\mathbf{x}'''\,, \tag{14.5}$$

yields the predicted point $\mathcal{X}' = \mathcal{l}'(\mathcal{X}'') \times \mathcal{l}'(\mathcal{X}''')$, thus

$$\mathbf{x}' = \mathsf{F}_{12}\mathbf{x}'' \times \mathsf{F}_{13}\mathbf{x}'''\,. \tag{14.6}$$

Similar expressions can be found for predicting image points in the other two images.

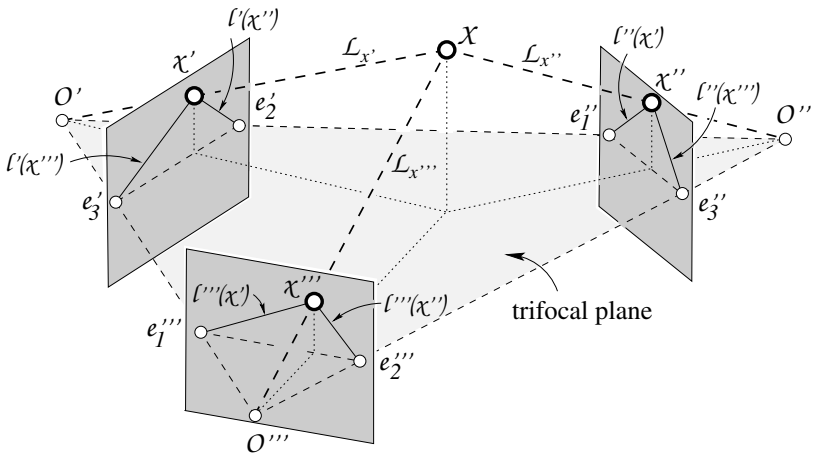


Fig. 14.2 Geometry of the image triplet with points. Three projection centres O', O'', and O'''. Three image points $\mathcal{X}'$, $\mathcal{X}''$, and $\mathcal{X}'''$. Six epipoles, e.g., $e'_3 = \mathcal{P}'(O''')$. Three projection rays $\mathcal{L}_{x'}$, $\mathcal{L}_{x''}$, and $\mathcal{L}_{x'''}$. Six epipolar lines, e.g., $l'(\mathcal{X}''') = \mathcal{P}'(\mathcal{L}_{x'''})$. If the 3D point $\mathcal{X}$ is outside the trifocal plane, the two epipolar lines in each image have a unique intersection point, namely the image point of $\mathcal{X}$. Otherwise, if the 3D point is on the trifocal plane, the two epipolar lines in each image are identical and do not yield a unique intersection point

If all elements are oriented and the projection matrices have the proper sign, we have the following *sign constraints* for three corresponding image points:

$$\text{sign}(|\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \mathbf{X}|) = \text{sign}(|\mathbf{e}'_2, \mathbf{e}'_3, \mathbf{x}'|) = \text{sign}(|\mathbf{e}''_3, \mathbf{e}''_1, \mathbf{x}''|) = \text{sign}(|\mathbf{e}'''_1, \mathbf{e}'''_2, \mathbf{x}'''|)\,. \quad (14.7)$$

sign constraints for three corresponding points

The proof starts from a canonical situation, e.g., one in Fig. 14.2, where all image points are positive and uses the definition of chirality of point triplets in 2D and quadruplets in 3D, cf. (9.14), p. 349, and (9.21), p. 350. Then exchanging two projection centres or moving the point below the trifocal plane does not change the constraints.

This method of prediction only works if the 3D point and the three projection centres are not coplanar, or – equivalently – if the 3D point is not on the trifocal plane. Then the two projection lines $\mathcal{L}_{x''}$ and $\mathcal{L}_{x'''}$ lie in the trifocal plane, the two epipolar lines $l'(\mathcal{X}'')$ and $l'(\mathcal{X}''')$ are identical, and therefore the intersection point $\mathcal{X}' = l'(\mathcal{X}'') \cap l'(\mathcal{X}''')$ is not unique. Practically, even 3D points close to the trifocal plane cause numerical uncertainties in this type of prediction.

Unfortunately, this unfavourable situation occurs often, especially in image sequences, where consecutive projection centres are collinear or nearly collinear. If the three projection centres are collinear, they and the 3D points are always coplanar, so the prediction of an image point using the epipolar geometry of two pairs of images fails for all 3D points. This can easily be visualized: Any three projection rays which are coplanar lead to three 3D intersection points, $\mathcal{X}_{12} = \mathcal{L}_{x'} \cap \mathcal{L}_{x''}$, $\mathcal{X}_{23}$ and $\mathcal{X}_{31}$. These three points need not be identical. Thus the epipolar constraints are fulfilled, in spite of the fact that the three image points are not corresponding. The epipolar constraints thus are only necessary but not sufficient conditions for the correspondence of three image points.

But a prediction of a point and thus a constraint can be achieved. This can be seen by first determining the 3D point, e.g., $\mathcal{X}_{23}$, by triangulation and then projecting it into the other image, which results in the constraint $\mathcal{X}' \equiv \mathcal{P}'(\mathcal{X}_{23})$. While the intersection of two rays due to observational noise is generally cumbersome, the situation is simple for corresponding lines. We discuss this in the next section.

14.1.3 The Trifocal Tensor

14.1.3.1 Predicting an Image Line

Given a corresponding line in two images we can predict it in a third image.

Predicting a Line in the First Image. We assume we are given two corresponding lines $\mathcal{l}''$ and $\mathcal{l}'''$ in the second and the third image, respectively, and we want to predict the corresponding line $\mathcal{l}'$ in the first image. For this we first use the projection planes $\mathcal{A}_{l''}$ and $\mathcal{A}_{l'''}$,

$$\mathbf{A}_{l''} = \mathsf{P}_2^\mathsf{T}\mathbf{l}'' \qquad \mathbf{A}_{l'''} = \mathsf{P}_3^\mathsf{T}\mathbf{l}''' \tag{14.8}$$

(cf. (12.83), p. 483), for determining the 3D line $\mathcal{L}$ as their intersection,

$$\mathbf{L} = \mathbf{A}_{l''} \cap \mathbf{A}_{l'''} = \overline{\mathsf{TT}}(\mathsf{P}_2^\mathsf{T}\mathbf{l}'')\,\mathsf{P}_3^\mathsf{T}\mathbf{l}''' \tag{14.9}$$

(cf. (7.44), p. 301), and project it into the first image,

$$\mathbf{l}' = \mathsf{Q}_1\mathbf{L} = \mathsf{Q}_1\,\overline{\mathsf{TT}}(\mathsf{P}_2^\mathsf{T}\mathbf{l}'')\,\mathsf{P}_3^\mathsf{T}\mathbf{l}''' \tag{14.10}$$

(cf. (12.72), p. 480). Given the three projection matrices $\mathsf{P}_t, t = 1,2,3$, and thus the projection matrix Q_1, the expression for the coordinates $\mathbf{l}'$ of the predicted line $\mathcal{l}'$ are *linear* in the coordinates $\mathbf{l}''$ and $\mathbf{l}'''$ as $\overline{\mathsf{TT}}(\mathsf{P}_2^\mathsf{T}\mathbf{l}'')$ is linear in $\mathbf{l}''$. Therefore we can write the coordinates of the predicted line with bilinear forms as

$$\mathbf{l}' = \begin{bmatrix} \mathbf{l}''^\mathsf{T}\mathsf{T}_1\mathbf{l}''' \\ \mathbf{l}''^\mathsf{T}\mathsf{T}_2\mathbf{l}''' \\ \mathbf{l}''^\mathsf{T}\mathsf{T}_3\mathbf{l}''' \end{bmatrix}, \tag{14.11}$$

with the three matrices T_i called *trifocal matrices*, which depend on the given projection matrices, see below. Obviously the prediction of the line $\mathbf{l}'$ is linear in the homogeneous coordinates of the two other lines and also *linear* in the elements of the trifocal matrices T_i.

trifocal matrices

To simplify notation we will write (Faugeras and Luong, 2001; Ressl, 2003)

prediction of $\mathcal{l}'$

$$\mathcal{l}' = \ell(\mathcal{l}'', \mathcal{l}''') : \quad \mathbf{l}' = \mathsf{T}(\mathbf{l}'', \mathbf{l}''') \tag{14.12}$$

for the prediction of $\mathbf{l}'$ from the two other lines. The stack of the three trifocal matrices $\mathsf{T}_i = [T_{i,jk}]$ yields the *trifocal tensor*

trifocal tensor

$$\mathsf{T} \doteq [\mathsf{T}_i] = [[T_{i,jk}]] \tag{14.13}$$

with $3 \times 3 \times 3 = 27$ elements.

Expressions for the Trifocal Tensor. We now give explicit expressions for the elements $T_{i,jk}$ of the trifocal tensor as a function of the given projection matrices, especially their camera planes $\mathbf{A}_t$, $\mathbf{B}_t$, and $\mathbf{C}_t$. We prove

trifocal matrices for perspective cameras

$$T_{i,jk} = \overline{\mathbf{L}}_i^\mathsf{T}\mathbf{L}_{jk} \qquad i,j,k \in \{1,2,3\} \tag{14.14}$$

and

$$\mathsf{T}_i = \mathsf{P}_2\overline{\mathsf{\Gamma}}(\mathbf{L}_i)\mathsf{P}_3^\mathsf{T} \qquad i \in \{1,2,3\} \tag{14.15}$$

with the camera lines

$$\underset{18\times1}{[\mathbf{L}_i]} = \begin{bmatrix} \mathbf{B}_1 \cap \mathbf{C}_1 \\ \mathbf{C}_1 \cap \mathbf{A}_1 \\ \mathbf{A}_1 \cap \mathbf{B}_1 \end{bmatrix}, \quad \underset{18\times3}{[\mathbf{L}_{jk}]} = \begin{bmatrix} \mathbf{A}_2 \cap \mathbf{A}_3 & \mathbf{A}_2 \cap \mathbf{B}_3 & \mathbf{A}_2 \cap \mathbf{C}_3 \\ \mathbf{B}_2 \cap \mathbf{A}_3 & \mathbf{B}_2 \cap \mathbf{B}_3 & \mathbf{B}_2 \cap \mathbf{C}_3 \\ \mathbf{C}_2 \cap \mathbf{A}_3 & \mathbf{C}_2 \cap \mathbf{B}_3 & \mathbf{C}_2 \cap \mathbf{C}_3 \end{bmatrix}. \tag{14.16}$$

Explicitly, the three trifocal matrices therefore are

$$\mathsf{T}_1 = -\begin{bmatrix} |\mathbf{B}_1,\mathbf{C}_1,\mathbf{A}_2,\mathbf{A}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{A}_2,\mathbf{B}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{A}_2,\mathbf{C}_3| \\ |\mathbf{B}_1,\mathbf{C}_1,\mathbf{B}_2,\mathbf{A}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{B}_2,\mathbf{B}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{B}_2,\mathbf{C}_3| \\ |\mathbf{B}_1,\mathbf{C}_1,\mathbf{C}_2,\mathbf{A}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{C}_2,\mathbf{B}_3| & |\mathbf{B}_1,\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3| \end{bmatrix} \tag{14.17}$$

$$= \mathsf{P}_2\overline{\Gamma}(\mathbf{B}_1 \cap \mathbf{C}_1)\mathsf{P}_3^\mathsf{T} \tag{14.18}$$

$$\mathsf{T}_2 = -\begin{bmatrix} |\mathbf{C}_1,\mathbf{A}_1,\mathbf{A}_2,\mathbf{A}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{A}_2,\mathbf{B}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{A}_2,\mathbf{C}_3| \\ |\mathbf{C}_1,\mathbf{A}_1,\mathbf{B}_2,\mathbf{A}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{B}_2,\mathbf{B}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{B}_2,\mathbf{C}_3| \\ |\mathbf{C}_1,\mathbf{A}_1,\mathbf{C}_2,\mathbf{A}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{C}_2,\mathbf{B}_3| & |\mathbf{C}_1,\mathbf{A}_1,\mathbf{C}_2,\mathbf{C}_3| \end{bmatrix} \tag{14.19}$$

$$= \mathsf{P}_2\overline{\Gamma}(\mathbf{C}_1 \cap \mathbf{A}_1)\mathsf{P}_3^\mathsf{T} \tag{14.20}$$

$$\mathsf{T}_3 = -\begin{bmatrix} |\mathbf{A}_1,\mathbf{B}_1,\mathbf{A}_2,\mathbf{A}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{A}_2,\mathbf{B}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{A}_2,\mathbf{C}_3| \\ |\mathbf{A}_1,\mathbf{B}_1,\mathbf{B}_2,\mathbf{A}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{B}_2,\mathbf{B}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{B}_2,\mathbf{C}_3| \\ |\mathbf{A}_1,\mathbf{B}_1,\mathbf{C}_2,\mathbf{A}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{C}_2,\mathbf{B}_3| & |\mathbf{A}_1,\mathbf{B}_1,\mathbf{C}_2,\mathbf{C}_3| \end{bmatrix} \tag{14.21}$$

$$= \mathsf{P}_2\overline{\Gamma}(\mathbf{A}_1 \cap \mathbf{B}_1)\mathsf{P}_3^\mathsf{T}\,. \tag{14.22}$$

efficient determination of trifocal tensor

These expressions are valid for perspective cameras.

An efficient determination of all 27 elements needs a maximum of 252 multiplications if following (14.15).

Proof: We first determine the 3D line $\mathcal{L}$ as the intersection of the projection planes $\mathbf{A}_{l''} = \mathsf{P}_2^\mathsf{T}\mathbf{l}''$ and $\mathbf{A}_{l'''} = \mathsf{P}_3^\mathsf{T}\mathbf{l}'''$, cf. (12.83), p. 483. Then we project it into the first image using $\mathbf{l}' = \mathsf{Q}_1\mathbf{L}$, cf. (12.72), p. 480.

We start from the representation of the projection matrices, cf. (12.44), p. 474 and (12.77), p. 481, and the homogeneous coordinates of the given lines,

$$\mathsf{P}_t = \begin{bmatrix} \mathbf{A}_t^\mathsf{T} \\ \mathbf{B}_t^\mathsf{T} \\ \mathbf{C}_t^\mathsf{T} \end{bmatrix} \qquad \mathsf{Q}_1 = \begin{bmatrix} \overline{(\mathbf{B}_1 \cap \mathbf{C}_1)}^\mathsf{T} \\ \overline{(\mathbf{C}_1 \cap \mathbf{A}_1)}^\mathsf{T} \\ \overline{(\mathbf{A}_1 \cap \mathbf{B}_1)}^\mathsf{T} \end{bmatrix} \quad \mathbf{l}'' = \begin{bmatrix} l_1'' \\ l_2'' \\ l_3'' \end{bmatrix} \quad \mathbf{l}''' = \begin{bmatrix} l_1''' \\ l_2''' \\ l_3''' \end{bmatrix}. \tag{14.23}$$

The two projection planes are

$$\mathbf{A}_{l''} = \mathsf{P}_2^\mathsf{T}\mathbf{l}'' = l_1''\mathbf{A}_2 + l_2''\mathbf{B}_2 + l_3''\mathbf{C}_2\,, \quad \mathbf{A}_{l'''} = \mathsf{P}_3^\mathsf{T}\mathbf{l}''' = l_1'''\mathbf{A}_3 + l_2'''\mathbf{B}_3 + l_3'''\mathbf{C}_3\,; \tag{14.24}$$

cf. (12.83), p. 483. We have

$$\mathbf{l}' = \mathsf{Q}_1\,(\mathsf{P}_2^\mathsf{T}\mathbf{l}'' \cap \mathsf{P}_3^\mathsf{T}\mathbf{l}''') = \begin{bmatrix} \overline{(\mathbf{B}_1 \cap \mathbf{C}_1)}^\mathsf{T} \\ \overline{(\mathbf{C}_1 \cap \mathbf{A}_1)}^\mathsf{T} \\ \overline{(\mathbf{A}_1 \cap \mathbf{B}_1)}^\mathsf{T} \end{bmatrix} (l_1''\mathbf{A}_2 + l_2''\mathbf{B}_2 + l_3''\mathbf{C}_2) \cap (l_1'''\mathbf{A}_3 + l_2'''\mathbf{B}_3 + l_3'''\mathbf{C}_3))\,. \tag{14.25}$$

Therefore the three trifocal matrices T_i contain 4×4 determinants composed of the rows of the projection matrices. This can be seen from the following example: the first element l_1' linearly depends on l_1'' and l_1''' with the coefficient

$$(\overline{\mathbf{B}_1 \cap \mathbf{C}_1})^\mathsf{T}(\mathbf{A}_2 \cap \mathbf{A}_3) = \overline{\mathbf{L}}_1^\mathsf{T}\mathbf{L}_{23} = -|\mathbf{B}_1,\mathbf{C}_1,\mathbf{A}_2,\mathbf{A}_3| = \mathbf{A}_2^\mathsf{T}\overline{\Gamma}(\mathbf{B}_1 \cap \mathbf{C}_1)\mathbf{A}_3\,; \tag{14.26}$$

cf. (7.60) and (7.61), p. 304. Observe the minus sign before the determinant, cf. the definition of the Plücker coordinates (5.107), p. 226 and (5.117), p. 227.

For an efficient determination of the 27 elements following (14.15) we first determine the three lines $\mathbf{L}_i, i = 1,2,3$ which requires 12 multiplications for each, e.g., $\mathbf{L}_1 = \mathbf{B}_1 \cap \mathbf{C}_1$. The multiplications with P_2 requires the intersection of the three lines $\mathbf{L}_i$ with its three rows j, with 12 multiplications each, e.g., $\mathbf{X}_{12} = \mathbf{L}_1 \cap \mathbf{A}_2$. Finally we determine the 27 dot products of the nine points $\mathbf{X}_{ij}$ with the three columns of P_3^T, with four multiplications for each. This yields $3 \times 12 + 9 \times 12 + 27 \times 4 = 252$ multiplications in total. ◇

critical configuration

Critical Configurations for the Prediction. There exist critical configurations where the prediction fails, assuming all projection centres are distinct:

1. The prediction $\ell(\mathcal{l}'', \mathcal{l}''')$ fails if both lines $\mathcal{l}''$ and $\mathcal{l}'''$ are epipolar lines $\mathcal{l}''(\mathcal{x}')$ and $\mathcal{l}'''(\mathcal{x}')$ w.r.t. some point $\mathcal{x}'$ in the first image I_1. Then the 3D line is a projection ray

$\mathbf{L}_x = \overline{\mathsf{Q}}^\mathsf{T}\mathbf{x}$ of the first image, which leads to an indefinite predicted line $\mathbf{l}' = \mathsf{Q}\mathbf{L}_x = \mathsf{Q}\overline{\mathsf{Q}}^\mathsf{T}\mathbf{x}' = \mathbf{0}$, as the three rows of Q, interpreted as 3D lines, intersect and fulfil the Plücker constraint; hence, we have $\mathsf{Q}\overline{\mathsf{Q}}^\mathsf{T} = \mathit{0}$, cf. (14.23); we therefore need to check the constraints

$$\angle(\mathbf{l}'', \mathbf{l}''(\mathbf{x}')) \notin \{0, \pi\} \quad \text{and} \quad \angle(\mathbf{l}''', \mathbf{l}'''(\mathbf{x}')) \notin \{0, \pi\}, \tag{14.27}$$

taking into account that the lines do not have opposite directions. Using the line coordinates in the camera system allows us to check the angles with a fixed threshold, e.g., 1°.

2. If the two lines l'' and l''' are epipolar lines of the image pair I_2 and I_3, the projection planes are identical and the 3D line is not defined. As the two projection planes pass through O'' and O''', it is sufficient to check the angle between their normals. This yields the constraint

$$\angle(\mathsf{A}_2^\mathsf{T}\mathbf{l}'', \mathsf{A}_3^\mathsf{T}\mathbf{l}'''(\mathbf{x}')) \notin \{0, \pi\}. \tag{14.28}$$

Predicting Directed Image Lines. When working with *directed* lines and proper projection matrices, we first need to assume that the given two lines l'' and l''' are consistently directed. With the epipoles of the second and the third image

$$\mathbf{e}_3'' = \mathsf{P}_2\mathbf{Z}_3 \qquad \mathbf{e}_2''' = \mathsf{P}_3\mathbf{Z}_2, \tag{14.29}$$

the constraint

$$\operatorname{sign}(\mathbf{l}''^\mathsf{T}\mathbf{e}_3'') = -\operatorname{sign}(\mathbf{l}'''^\mathsf{T}\mathbf{e}_2''') \tag{14.30}$$

needs to be fulfilled, cf. (13.287), p. 606. But even then, changing the direction of both lines l'' and l''' does not change the direction of $\mathbf{l}$. Moreover, the intersection of the two projection planes is not unique: they could be exchanged, yielding the line l' in the opposite direction.

Therefore we need to enforce the correct direction of the line l', provided the two others are consistently oriented. Using (13.288), p. 606 this yields the prediction of the directed image line, cf. Werner and Pajdla (2001),

prediction of directed image line

$$\mathit{l}' = \ell(\mathit{l}'', \mathit{l}''') : \quad \mathbf{l}' = \operatorname{sign}(\mathbf{l}''^\mathsf{T}\mathbf{e}_3'') \begin{bmatrix} \mathbf{l}''^\mathsf{T}\mathsf{T}_1\mathbf{l}''' \\ \mathbf{l}''^\mathsf{T}\mathsf{T}_2\mathbf{l}''' \\ \mathbf{l}''^\mathsf{T}\mathsf{T}_3\mathbf{l}''' \end{bmatrix}, \tag{14.31}$$

with the trifocal matrices from (14.17).

14.1.3.2 The Normalized Trifocal Tensor

The trifocal tensor has a simple form if the first camera is normalized,

$$\mathsf{P}_1 = [I_3|\mathbf{0}] = \begin{bmatrix} \boldsymbol{e}_1^{[4]\mathsf{T}} \\ \boldsymbol{e}_2^{[4]\mathsf{T}} \\ \boldsymbol{e}_3^{[4]\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^\mathsf{T} \\ \mathbf{B}_1^\mathsf{T} \\ \mathbf{C}_1^\mathsf{T} \end{bmatrix}, \tag{14.32}$$

with the unit 4-vectors $\boldsymbol{e}_i^{[4]}$. For a general projection matrix $\mathsf{P}_1 = [\mathsf{A}_1|\mathbf{a}_1]$, the normalization could be achieved by the regular transformations of the projection matrices and the 3D points,

$$\mathsf{P}_t := \mathsf{P}_t\mathsf{M} = [\mathsf{A}_1|\mathbf{a}_1]\begin{bmatrix}\mathsf{A}_1^{-1} & -\mathsf{A}_1^{-1}\mathbf{a}_1 \\ \mathbf{0}^\mathsf{T} & 1\end{bmatrix}, \quad t = 1,2,3\,, \tag{14.33}$$

$$\mathbf{X} := \mathsf{M}^{-1}\mathbf{X} = \begin{bmatrix}\mathsf{A}_1 & \mathbf{a}_1 \\ \mathbf{0}^\mathsf{T} & 1\end{bmatrix}\mathbf{X}\,, \tag{14.34}$$

as then the predicted image points $\mathsf{P}_t\mathbf{X}$ in the three images remain the same.

Using the two other projection matrices characterized by their columns,

$$\mathsf{P}_2 = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4] \qquad \mathsf{P}_3 = [\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4]\,, \tag{14.35}$$

normalized trifocal tensor

the three trifocal matrices of the *normalized trifocal tensor* have the form of differences of dyadic products of the columns of P_2 and P_3,

$$\mathsf{T}_i = \mathbf{y}_4\mathbf{z}_i^\mathsf{T} - \mathbf{y}_i\mathbf{z}_4^\mathsf{T}\,, i = 1,2,3\,. \tag{14.36}$$

The determination of these trifocal matrices is very efficient: it only requires $3 \times 18 = 54$ multiplications, at the expense of working in the coordinate system of the first normalized camera.

Proof: First we observe, cf. (12.44), p. 474

$$\overline{\mathbf{L}}_1 = \overline{\mathbf{B}_1 \cap \mathbf{C}_1} = \overline{\boldsymbol{e}_2^{[4]} \cap \boldsymbol{e}_3^{[4]}} = \boldsymbol{e}_4^{[6]} = \begin{bmatrix}\mathbf{0} \\ \boldsymbol{e}_1^{[3]}\end{bmatrix} \tag{14.37}$$

$$\overline{\mathbf{L}}_2 = \overline{\mathbf{C}_1 \cap \mathbf{A}_1} = \overline{\boldsymbol{e}_3^{[4]} \cap \boldsymbol{e}_1^{[4]}} = \boldsymbol{e}_5^{[6]} = \begin{bmatrix}\mathbf{0} \\ \boldsymbol{e}_2^{[3]}\end{bmatrix} \tag{14.38}$$

$$\overline{\mathbf{L}}_3 = \overline{\mathbf{A}_1 \cap \mathbf{B}_1} = \overline{\boldsymbol{e}_1^{[4]} \cap \boldsymbol{e}_2^{[4]}} = \boldsymbol{e}_6^{[6]} = \begin{bmatrix}\mathbf{0} \\ \boldsymbol{e}_3^{[3]}\end{bmatrix}. \tag{14.39}$$

The elements $\mathbf{L}_{jk}$ in (14.16) can be expressed as elements of the two projection matrices,

$$\mathsf{P}_2 = [P_{2;j,l}]\,, \qquad \mathsf{P}_3 = [P_{3;k,l}]\,. \tag{14.40}$$

For example, the line $\mathbf{L}_{12}$ is given by the 6-vector, cf. (7.41), p. 301,

$$\mathbf{L}_{12} = \mathbf{A}_2 \cap \mathbf{B}_3 = \begin{bmatrix}\boldsymbol{A}_{2,h} \times \boldsymbol{B}_{3,h} \\ A_{2,0}\boldsymbol{B}_{3,h} - \boldsymbol{A}_{2,h}B_{2,0}\end{bmatrix} = P_{2;1\bullet} \cap P_{3;2\bullet}\,, \tag{14.41}$$

where $P_{2;1\bullet}$ indicates the first row of the projection matrix P_2. Generally we have

$$\mathbf{L}_{jk} = P_{2;j\bullet} \cap P_{3;k\bullet} = \begin{bmatrix}P_{2;j1} \\ P_{2;j2} \\ P_{2;j3} \\ P_{2;j4}\end{bmatrix} \cap \begin{bmatrix}P_{3;k1} \\ P_{3;k2} \\ P_{3;k3} \\ P_{3;k4}\end{bmatrix}. \tag{14.42}$$

Therefore we obtain

$$T_1(1,2) = \overline{\mathbf{B}_1 \cap \mathbf{C}_1}^\mathsf{T}\mathbf{A}_2 \cap \mathbf{B}_3 = \boldsymbol{e}_4^{[6]\mathsf{T}}\,\mathbf{L}_{12} = P_{2;1,4}P_{3;2,1} - P_{2;1,1}P_{3;2,4}\,, \tag{14.43}$$

or, generally,

$$T_i(j,k) = P_{2;j,4}P_{3;k,i} - P_{2;j,i}P_{3;k,4}\,. \tag{14.44}$$

With the columns of the projection matrices from (14.35), the three submatrices of the normalized trifocal tensor can be expressed as (14.36). ◇

14.1.3.3 Trifocal Tensor for Normalized Cameras in the Trifocal Plane

Without loss of generality we can choose the coordinate system such that all projection centres lie in the trifocal plane $O_1 \wedge O_2 \wedge O_3$ passing through the three projection centres and use normalized cameras with $\mathsf{K}_t R_t = I_3$. Actually, the first use of trifocal constraints by Mikhail (1962, 1963) for checking the consistency of observed image coordinates was based on normalized cameras. Then the three normal cameras have special projection matrices,

$$\mathsf{P}_t = [I_3 | -\boldsymbol{Z}_t], \qquad \boldsymbol{Z}_t = \begin{bmatrix} X_{Ot} \\ Y_{Ot} \\ 0 \end{bmatrix} \qquad t = 1, 2, 3, \tag{14.45}$$

and the matrices T_k can explicitly be given:

$$\mathsf{T}_1 = \begin{bmatrix} X_{O3} - X_{O2} & Y_{O3} - Y_{O1} & 0 \\ -(Y_{O2} - Y_{O1}) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{14.46}$$

$$\mathsf{T}_2 = \begin{bmatrix} 0 & -X_{O2} + X_{O1} & 0 \\ X_{O3} - X_{O1} & Y_{O3} - Y_{O2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{14.47}$$

$$\mathsf{T}_3 = \begin{bmatrix} 0 & 0 & -X_{O2} + X_{O1} \\ 0 & 0 & -Y_{O2} + Y_{O1} \\ X_{O3} - X_{O1} & Y_{O3} - Y_{O1} & 0 \end{bmatrix}. \tag{14.48}$$

14.1.3.4 The Geometry of One, Two, and Three Images

It is interesting to compare the different representations for the geometry of one, two and three images. Table 14.2 gives the expressions for the projection matrix P, the fundamental matrix F and the trifocal tensor T. The different expressions are useful for different tasks. They all are expressed in terms of the projection matrices, thus allowing the expression of all entities as functions of the given interior and exterior orientation parameters:

- The geometry of one and two images can be expressed as a function of the parameters of the interior and the exterior orientation, see lines 1, 4, and 7.
- The geometry of one and two images also can be represented by the infinite homograph $\mathsf{H}_\infty = \mathsf{A}$ and the projection centre, see lines 2, 5, and 8.
- The camera lines $\mathcal{L}_i, i = 1, 2, 3$, see line 6, link the projection matrix for lines and the fundamental matrix, see lines 6 and 9.
- The camera planes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ link all representations: the projection matrices for points and lines, see lines 3 and 6, and (via the camera lines) also the fundamental matrix and the trifocal tensor, see lines 10 and 11.

The representations are valid for calibrated and uncalibrated cameras.

14.1.4 Predictions and Constraints for Points and Lines

So far, we have only discussed the prediction of a line into the first image when two observed lines in the second and the third image are given. Now we want to derive predictions

- for lines into the second and the third image,
- for points from any two images into a third, and
- for a given mixture of points and lines, again from any two images into a third.

They are the basis for constraints between lines and points, and mixtures of points and lines (cf. Ressl, 2003).

14.1.4.1 Predicting Points and Lines

The derivation starts with a constraint involving the first image point. For a point $\mathcal{X}'$ on the line $\mathcal{l}'$ predicted from $\mathcal{l}''$ and $\mathcal{l}'''$ in the other two images, we have $\mathbf{x}'^{\mathsf{T}}\mathbf{l}' = 0$. Thus, with (14.11) for an observed point $\mathcal{X}'$ and two lines $\mathcal{l}''$ and $\mathcal{l}'''$,

Table 14.2 Explicit expressions for projection matrices P and Q for points and lines, respectively; fundamental matrix F and trifocal tensor T. The lines $\overline{\mathbf{L}}_i$ used for determining the elements for the trifocal tensor refer to the first camera

	entity	representation	Eq.
1	P	$\mathsf{K}\boldsymbol{R}[\boldsymbol{I}_3 \mid -\boldsymbol{Z}]$	(12.34)
2		$[\mathsf{A}\mid\mathbf{a}] = [\mathsf{A} \mid -\mathsf{A}\boldsymbol{Z}]$	(12.44)
3		$\begin{bmatrix}\mathbf{A}^\mathsf{T}\\ \mathbf{B}^\mathsf{T}\\ \mathbf{C}^\mathsf{T}\end{bmatrix}$	(12.44)
4	Q	$(\mathsf{K}\boldsymbol{R})^\circ[-\boldsymbol{S}(\boldsymbol{Z})\mid\boldsymbol{I}_3]$	(12.79)
5		$[\mathsf{Y}\mid\mathsf{N}] = [-\mathsf{A}^\circ\boldsymbol{S}(\boldsymbol{Z}) \mid \mathsf{A}^\circ]$	(12.77)
6		$[\overline{\mathbf{L}}_i] = \begin{bmatrix}\overline{(\mathbf{B}\cap\mathbf{C})}^\mathsf{T}\\ \overline{(\mathbf{C}\cap\mathbf{A})}^\mathsf{T}\\ \overline{(\mathbf{A}\cap\mathbf{B})}^\mathsf{T}\end{bmatrix} = \begin{bmatrix}(\mathbf{B}\wedge\mathbf{C})^\mathsf{T}\\ (\mathbf{C}\wedge\mathbf{A})^\mathsf{T}\\ (\mathbf{A}\wedge\mathbf{B})^\mathsf{T}\end{bmatrix}$	(12.71)
7	F	$\mathsf{K}_1^{-\mathsf{T}}\boldsymbol{R}_1\mathsf{S}(\mathbf{b})\boldsymbol{R}_2^\mathsf{T}\mathsf{K}_2^{-1}$	(13.8)
		$\mathbf{b} = \boldsymbol{Z}_2 - \boldsymbol{Z}_1$	
8		$\mathsf{F} = \mathsf{A}_1^\circ\mathsf{S}(\mathbf{b})\mathsf{A}_2^{\circ\mathsf{T}}$	(13.12)
		$\mathbf{b} = \mathsf{A}_2^{-1}\mathbf{a}_2 - \mathsf{A}_1^{-1}\mathbf{a}_1$	
9		$\mathsf{Q}'\overline{\mathsf{Q}''}^\mathsf{T}$	(13.70)
10		$[F_{ij}] = [\overline{\mathbf{L}}_{1i}^\mathsf{T}\mathbf{L}_{2j}]$	(13.19)
11	T	$[T_{i,jk}] = [\overline{\mathbf{L}}_i^\mathsf{T}\mathbf{L}_{jk}]$	(14.14)
		$[\mathbf{L}_{jk}] = \begin{bmatrix}\mathbf{A}_2\cap\mathbf{A}_3 & \mathbf{A}_2\cap\mathbf{B}_3 & \mathbf{A}_2\cap\mathbf{C}_3\\ \mathbf{B}_2\cap\mathbf{A}_3 & \mathbf{B}_2\cap\mathbf{B}_3 & \mathbf{B}_2\cap\mathbf{C}_3\\ \mathbf{C}_2\cap\mathbf{A}_3 & \mathbf{C}_2\cap\mathbf{B}_3 & \mathbf{C}_2\cap\mathbf{C}_3\end{bmatrix}$	

$$\mathbf{x}'^\mathsf{T}\begin{bmatrix}\mathbf{l}''^\mathsf{T}\mathsf{T}_1\mathbf{l}'''\\ \mathbf{l}''^\mathsf{T}\mathsf{T}_2\mathbf{l}'''\\ \mathbf{l}''^\mathsf{T}\mathsf{T}_3\mathbf{l}'''\end{bmatrix} = 0\,. \tag{14.49}$$

With the matrix

$$\mathsf{T}_\diamond(\mathbf{x}') = x_1'\mathsf{T}_1 + x_2'\mathsf{T}_2 + x_3'\mathsf{T}_3\,, \tag{14.50}$$

this constraint c for the correspondence of the triplet $(\mathcal{x}', \mathcal{l}'', \mathcal{l}''')$ reads

$$c(\mathcal{x}', \mathcal{l}'', \mathcal{l}''') : \quad \mathbf{l}''^\mathsf{T}\mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}''' = 0\,. \tag{14.51}$$

When partitioning this bilinear form as

$$\underbrace{\mathbf{l}''^\mathsf{T}\mathsf{T}_\diamond(\mathbf{x}')}_{\mathbf{x}'''^\mathsf{T}}\mathbf{l}''' = \mathbf{l}''^\mathsf{T}\underbrace{\mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}'''}_{\mathbf{x}''} = 0\,, \tag{14.52}$$

we obtain prediction equations for points in the second and the third image

$$(\mathcal{x}', \mathcal{l}''') \to \mathcal{x}'' : \quad \mathbf{x}'' = \mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}''' \tag{14.53}$$

$$(\mathcal{x}', \mathcal{l}'') \to \mathcal{x}''' : \quad \mathbf{x}''' = \mathsf{T}_\diamond^\mathsf{T}(\mathbf{x}')\mathbf{l}''\,. \tag{14.54}$$

The predictions discussed so far involve the trifocal tensor only once. Predicting the first point involving image features in the second or third image is achieved by representing these points as the intersection of two lines and using the line prediction. For example, the prediction $(\mathcal{x}'', \mathcal{l}''') \to \mathcal{x}'$ is achieved by choosing two lines $\mathcal{l}_i'', i = 1, 2$ such that $\mathcal{x}'' = \mathcal{l}_1'' \cap \mathcal{l}_2''$, predicting these lines into the first image via $\mathcal{l}_i' = \ell(\mathcal{l}_i'', \mathcal{l}''')$, and determining the predicted point $\mathcal{x}'$ as their intersection:

$$\mathcal{x}' = \ell(\mathcal{l}_1'', \mathcal{l}''') \times \ell(\mathcal{l}_2'', \mathcal{l}''')\,. \tag{14.55}$$

The selected lines need to pass through the given image point and should not cause a singularity. Therefore we draw a random direction α'' and determine two lines in the second image with two orthogonal directions $\alpha_1'' = \alpha''$ and $\alpha_2'' = \alpha'' + 90°$. Then we obtain

$$\mathcal{l}_1'' = \mathcal{v}_1'' \wedge \mathcal{x}'' , \qquad \mathcal{l}_2'' = \mathcal{v}_2'' \wedge \mathcal{x}'' , \qquad \mathcal{v}_i'' = \mathcal{v}''(\alpha_i'') , \; i = 1, 2 \tag{14.56}$$

and the directions $\mathcal{v}(\alpha)$

$$\mathbf{v}(\alpha) = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix} . \tag{14.57}$$

In order to guarantee that no singularity occurs, the constraints (14.27), p. 627 and (14.28), p. 627 need to be fulfilled.

Other projections involving points and lines can be realized similarly. With the prediction operators $\mathcal{l}' = \ell(\mathcal{l}'', \mathcal{l}''')$ from (14.12) and

prediction operators $\wp_2(\mathcal{x}', \mathcal{l}''')$ and $\wp_3(\mathcal{x}', \mathcal{l}'')$ for points

$$\wp_2(\mathcal{x}', \mathcal{l}''') \to \mathcal{x}'' : \; \mathbf{x}'' = \mathsf{T}_\diamond^\mathsf{T}(\mathbf{x}')\mathbf{l}''' , \qquad \wp_3(\mathcal{x}', \mathcal{l}'') \to \mathcal{x}''' : \; \mathbf{x}''' = \mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}'' , \tag{14.58}$$

we obtain all cases collected in Table 14.3 (cf. Ressl, 2003, Table 7.6).

Table 14.3 Prediction relations in an image triplet, adapted from Ressl (2003, table 7.6) using the prediction operators ℓ, $\wp_2$ and $\wp_3$ in (14.58)

	entities	prediction
	in image 1	
1	$\{\mathcal{l}'', \mathcal{l}'''\} \to \mathcal{l}'$	$\mathcal{l}' = \ell(\mathcal{l}'', \mathcal{l}''')$
2	$\{\mathcal{x}'', \mathcal{l}'''\} \to \mathcal{x}'$	$\mathcal{x}' = \ell(\mathcal{l}_1'', \mathcal{l}''') \times \ell(\mathcal{l}_2'', \mathcal{l}''')$ with two lines $\mathcal{l}_i''$ such that $\mathcal{x}'' = \mathcal{l}_1'' \cap \mathcal{l}_2''$
3	$\{\mathcal{l}'', \mathcal{x}'''\} \to \mathcal{x}'$	$\mathcal{x}' = \ell(\mathcal{l}'', \mathcal{l}_1''') \times \ell(\mathcal{l}'', \mathcal{l}_2''')$ with two lines $\mathcal{l}_i'''$ such that $\mathcal{x}''' = \mathcal{l}_1''' \cap \mathcal{l}_2'''$
4	$\{\mathcal{x}'', \mathcal{x}'''\} \to \mathcal{x}'$	$\mathcal{x}' = \ell(\mathcal{l}_1'', \mathcal{l}''') \times \ell(\mathcal{l}_2'', \mathcal{l}''')$ with two lines $\mathcal{l}_i''$ and a line $\mathcal{l}'''$ such that $\mathcal{x}'' = \mathcal{l}_1'' \cap \mathcal{l}_2''$, $\mathcal{l}''' \ni \mathcal{x}'''$
	in image 2	
5	$\{\mathcal{x}', \mathcal{l}'''\} \to \mathcal{x}''$	$\mathcal{x}'' = \wp_2(\mathcal{x}', \mathcal{l}''')$
6	$\{\mathcal{x}', \mathcal{x}'''\} \to \mathcal{x}''$	$\mathcal{x}'' = \wp_2(\mathcal{x}', \mathcal{l}''')$ with some $\mathcal{l}'''$ such that $\mathcal{l}''' \ni \mathcal{x}'''$
7	$\{\mathcal{l}', \mathcal{l}'''\} \to \mathcal{l}''$	$\mathcal{l}'' = \wp_2(\mathcal{x}_1', \mathcal{l}''') \times \wp_2(\mathcal{x}_2', \mathcal{l}''')$ with two points $\mathcal{x}_i'$ such that $\mathcal{l}' = \mathcal{x}_1' \wedge \mathcal{x}_2'$
8	$\{\mathcal{l}', \mathcal{x}'''\} \to \mathcal{x}''$	$\mathcal{x}'' = \left(\wp_2(\mathcal{x}_1', \mathcal{l}_1''') \times \wp_2(\mathcal{x}_1', \mathcal{l}_2''')\right) \times \left(\wp_2(\mathcal{x}_2', \mathcal{l}_1''') \times \wp_2(\mathcal{x}_2', \mathcal{l}_2''')\right)$ with points $\mathcal{x}_i'$ and lines $\mathcal{l}_i'''$ such that $\mathcal{l}' = \mathcal{x}_1' \wedge \mathcal{x}_2'$, $\mathcal{x}''' = \mathcal{l}_1''' \cap \mathcal{l}_2'''$
	in image 3	
9	$\{\mathcal{x}', \mathcal{l}''\} \to \mathcal{x}'''$	$\mathcal{x}''' = \wp_3(\mathcal{x}', \mathcal{l}'')$
10	$\{\mathcal{x}', \mathcal{x}''\} \to \mathcal{x}'''$	$\mathcal{x}''' = \wp_3(\mathcal{x}', \mathcal{l}'')$ with some $\mathcal{l}''$ such that $\mathcal{l}'' \ni \mathcal{x}''$
11	$\{\mathcal{l}', \mathcal{l}''\} \to \mathcal{l}'''$	$\mathcal{l}''' = \wp_3(\mathcal{x}_1', \mathcal{l}'') \times \wp_3(\mathcal{x}_2', \mathcal{l}'')$ with two points $\mathcal{x}_i'$ such that $\mathcal{l}' = \mathcal{x}_1' \wedge \mathcal{x}_2'$
12	$\{\mathcal{l}', \mathcal{x}''\} \to \mathcal{x}'''$	$\mathcal{x}''' = \left(\wp_3(\mathcal{x}_1', \mathcal{l}_1'') \times \wp_3(\mathcal{x}_1', \mathcal{l}_2'')\right) \times \left(\wp_3(\mathcal{x}_2', \mathcal{l}_1'') \times \wp_3(\mathcal{x}_2', \mathcal{l}_2'')\right)$ with points $\mathcal{x}_i'$ and lines $\mathcal{l}_i''$ such that $\mathcal{l}' = \mathcal{x}_1' \wedge \mathcal{x}_2'$, $\mathcal{x}'' = \mathcal{l}_1'' \cap \mathcal{l}_2''$

14.1.4.2 Constraints for Points and Lines

In Sect. 14.1.4.1, p. 629 we already collected some constraints for three corresponding image features involving image points. Further constraints can be derived easily.

From the basic line prediction $\mathcal{l}' = \ell(\mathcal{l}'', \mathcal{l}''')$ we can derive the constraint for three corresponding lines, i.e., the observed line and the predicted line in the first image should be identical,[1]

[1] Remember, the matrix $\mathsf{S}^{(s)}(\mathbf{l}')$ is a 2×3 matrix, cf. Sect. 7.4.1, p. 317 and the footnote there.

Table 14.4 Constraints using the trifocal tensor. For critical configurations, cf. Ressl (2003, p. 82ff.)

elements	relation	dof
$\mathcal{l}', \mathcal{l}'', \mathcal{l}'''$	$\mathsf{S}^{(s)}(\mathbf{l}')\ \mathsf{T}(\mathbf{l}'', \mathbf{l}''') = \mathbf{0}$	2
$\mathcal{x}', \mathcal{l}'', \mathcal{l}'''$	$\mathbf{l}''^{\mathsf{T}}\ \mathsf{T}_\diamond(\mathbf{x}')\ \mathbf{l}''' = 0$	1
$\mathcal{x}', \mathcal{x}'', \mathcal{l}'''$	$\mathsf{S}^{(s)}(\mathbf{x}'')\ \mathsf{T}_\diamond(\mathbf{x}')\ \mathbf{l}''' = \mathbf{0}$	2
$\mathcal{x}', \mathcal{l}'', \mathcal{x}'''$	$\mathbf{l}''^{\mathsf{T}}\ \mathsf{T}_\diamond(\mathbf{x}')\ \mathsf{S}^{(s)\mathsf{T}}(\mathbf{x}''') = \mathbf{0}^{\mathsf{T}}$	2
$\mathcal{x}', \mathcal{x}'', \mathcal{x}'''$	$\mathsf{S}^{r\mathsf{T}}(\mathbf{x}''')\ \mathsf{T}_\diamond(\mathbf{x}')\ \mathsf{S}^{(s)\mathsf{T}}(\mathbf{x}'') = \mathsf{0}$	4

$$c(\mathcal{l}', \mathcal{l}'', \mathcal{l}''') : \quad \mathsf{S}^{(s)}(\mathbf{l}') \begin{bmatrix} \mathbf{l}''^{\mathsf{T}}\mathsf{T}_1\mathbf{l}''' \\ \mathbf{l}''^{\mathsf{T}}\mathsf{T}_2\mathbf{l}''' \\ \mathbf{l}''^{\mathsf{T}}\mathsf{T}_3\mathbf{l}''' \end{bmatrix} = \mathbf{0}\,. \tag{14.59}$$

Generally, only two constraints are linearly independent. Here we choose the skew symmetric matrix with two independent columns selected in order to arrive at two independent constraints. However, this choice does not necessarily avoid a critical configuration, cf. the discussion in Ressl (2003, p. 82ff.).

From the predictions $\mathbf{x}'' = \mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}'''$ and $\mathbf{x}''' = \mathsf{T}_\diamond^{\mathsf{T}}(\mathbf{x}')\mathbf{l}''$ we can similarly derive the constraints for two further configurations,

$$c(\mathcal{x}', \mathcal{x}'', \mathcal{l}''') : \ \mathsf{S}^{(s)}(\mathbf{x}'')\mathsf{T}_\diamond(\mathbf{x}')\mathbf{l}''' = \mathbf{0} \tag{14.60}$$

$$c(\mathcal{x}', \mathcal{l}'', \mathcal{x}''') : \ \mathbf{l}''^{\mathsf{T}}\mathsf{T}_\diamond(\mathbf{x}')\mathsf{S}^{(s)}(\mathbf{x}''') = \mathbf{0}^{\mathsf{T}}\,. \tag{14.61}$$

Again, each of these equations only represents two linearly independent constraints. In the same way, we finally obtain a constraint for three corresponding points,

$$c(\mathcal{x}', \mathcal{x}'', \mathcal{x}''') : \ \mathsf{S}^{s}(\mathbf{x}'')\mathsf{T}_\diamond(\mathbf{x}')\mathsf{S}^{s\mathsf{T}}(\mathbf{x}''')^{\mathsf{T}} = \mathsf{0}\,. \tag{14.62}$$

These are four linearly independent constraints in general.

All other constraints can be derived from Table 14.3 by forcing the predicted entity to be identical to an observed one.

Table 14.4 collects all constraints where the trifocal tensor is involved once. The table also contains the degrees of freedom for the constraints.

14.2 Relative Orientation of the Image Triplet

14.2.1 Sequential Relative Orientation of the Image Triplet 633
14.2.2 Direct Solutions for the Trifocal Tensor 636
14.2.3 Iterative Solution for the Triplet 637

The orientation of three images relies on the procedures of the orientation of two images but additionally has some unique features. On the one hand we can also perform a simultaneous orientation using a bundle adjustment or an orientation in steps, where the first step only uses image information determining the trifocal tensor representing the relative orientation, whereas the second step also uses control information in object space. On the other hand, we may use straight lines for relative orientation of three images, which is not possible for only two images.

Due to the algebraic complexity of the trifocal tensor, which is needed to describe the relations between corresponding features in three images, the orientation procedures are complicated. Although we discuss methods for both uncalibrated and calibrated cameras, the focus is on evaluating the image triplet with calibrated cameras.

We assume that the three images have unknown projection matrices $\mathsf{P}_t, t = 1, 2, 3$, sometimes also denoted by P', P'', and P'''. We assume that we have observed image features, namely points $\mathbf{x}'_{it}$, sometimes denoted by $\mathbf{x}'_i$, $\mathbf{x}''_i$, $\mathbf{x}''_i$, and lines $\mathbf{l}'_{jt}$, sometimes denoted by $\mathbf{l}'_j$, $\mathbf{l}''_j$, $\mathbf{l}''_j$. Indices i and j refer to the same point $\mathcal{X}_i$ or line $\mathcal{L}_j$ in object space, thus establish corresponding image features. Finally, we assume some control points or control lines in object space to be known. The task is to derive the unknown projection matrices. As soon as the orientation parameters are known, one may determine the 3D points and lines by triangulation, cf. Sect. 13.4, p. 596.

We have various stepwise procedures using the methods from single-view and two-view orientation, see Table 14.5. In addition we have a two-step procedure based on a relative orientation of the image triplet and a subsequent absolute orientation. For some of the mentioned tasks, minimal or direct solutions are available. All procedures can be used to obtain approximate values for a final bundle adjustment, for which there exists no closed form solution.

Therefore we mainly discuss methods for the relative orientation of the image triplet and their individual merits.

Table 14.5 Orientation of the image triplet. Procedures, participating image IDs (in brackets). RO = relative orientation, EO = exterior orientation, AO = absolute orientation, BA = bundle adjustment. Number U of parameters or number H of constraints, minimum number N of observations for direct solution, p = number of points, l = number of lines. Corresponding points and lines are assumed to be visible in the images mentioned in the first column. No direct solution, only iterative solutions are known for the problems indicated by ^{i}. Solutions indicated by * are not described in this section

	camera:	calibrated		uncalibrated perspective	
	procedure	U,(H)	min N	U	min N
1	three-step procedure I				
	1. RO (1,2)	5	$p \geq 5$	7	$p \geq 7$
	2. EO (3)	6	$2(p + l^*) \geq 6$	11	$2(p + l) \geq 11$
	3. AO	7	$3p + 4l^* \geq 7$	15	$3p + 4l \geq 15$
2	three-step procedure II				
	1. ROs (1,2), (2,3)	10	$p \geq 5$	14	$p \geq 7$
	2. RO (1,2,3)	1	$p + l^* \geq 1$	4	$2(p + l) \geq 4^*$
	3. AO	7	$3p + 4l^* \geq 7$	15	$3p + 4l \geq 15$
3	three-step procedure III				
	1. ROs (1,2), (2,3), (3,1)	15	$p \geq 5$	21	$p \geq 7$
	2. RO (1,2,3)	(4)	–	(3)	–
	3. AO	7	$3p + 4l^* \geq 7$	15	$3p + 4l \geq 15$
4	two-step procedure				
	1. RO (1,2,3)	11	$3p + 2l \geq 11^i$	18	$3p + 2l \geq 18$
	2. AO	7	$3p + 4l^* \geq 7$	15	$3p + 4l \geq 15$
5	one-step: BA	18	$3p + 4l \geq 18^i$	33	$3p + 4l \geq 33^i$

14.2.1 Sequential Relative Orientation of the Image Triplet

14.2.1.1 Procedure with Relative and Exterior Orientation

The relative orientation of three images can start with the relative orientation of two images leading to a photogrammetric model with 3D points or lines in a local model coordinate system, which can be used to perform a subsequent exterior orientation.

The last step could be repeated in the case of more than three images. The final photogrammetric model then is represented in the coordinate system used in the relative orientation of the first two images.

Formally, we have the following steps:

1. Relative orientation of the image pair, say $(1, 2)$, yielding the fundamental matrix F_{12} or the essential matrix E_{12} from the constraints
$$\mathbf{x}_i'^{\mathsf{T}}\,\mathsf{F}_{12}\,\mathbf{x}_i'' = 0 \qquad \text{or} \qquad {}^c\mathbf{x}_i'^{\mathsf{T}}\,\mathsf{E}_{12}\,{}^c\mathbf{x}_i'' = 0\,, \tag{14.63}$$
the projection matrices ${}^m\mathsf{P}_1$ and ${}^m\mathsf{P}_2$ and points ${}^m\mathbf{X}_i$ or lines ${}^m\mathbf{L}_j$ in the coordinate system S_m of the photogrammetric model $(1, 2)$.
For simplicity let us assume the coordinate system is defined by the first image. For calibrated images, the length $|\mathbf{b}_{12}|$ of the base vector defines the scale of the photogrammetric model. For uncalibrated cameras we need to fix four parameters, say $\{\alpha_{12}, \boldsymbol{D}_{12}\}$, cf. Sect. 13.3.7.1, p. 594.
2. The exterior orientation of the third image with respect to the image pair yields the third projection matrix ${}^m\mathsf{P}_3$ in the coordinate system S_m of the first image pair. The most intuitive solution is to use 3D points in the photogrammetric model visible in the third image: At least six are required for a direct linear transformation, or at least three for a spatial resection.
Additional points ${}^m\mathbf{X}_i$ or lines ${}^m\mathbf{L}_j$ may be determined in the local coordinate system S_m by triangulation.

This method is the simplest one, as it can be based on well-established procedures and does not require any post-processing. Furthermore, outliers in the third image can be identified. However, the second step cannot be performed independently of the first one; thus, the procedure does not allow for parallelization.

14.2.1.2 Procedure with Two Relative Orientations

The second step of the previous procedure can be replaced by the relative orientation of a second image pair, say (1,3). Then we obtain the fundamental matrix or essential matrix for this image pair from the constraints
$$\mathbf{x}_i'^{\mathsf{T}}\mathsf{F}_{13}\mathbf{x}_i''' = 0 \qquad \text{or} \qquad {}^c\mathbf{x}_i'^{\mathsf{T}}\mathsf{E}_{13}{}^c\mathbf{x}_i''' = 0\,. \tag{14.64}$$

We could determine the projection matrix P''' for the third image using the coordinate system of the first camera. However, the two projection matrices P'' and P''' will not be consistent due to the freedom in choosing the free parameters, namely the length of the basis $|\mathbf{b}_{13}|$ for calibrated cameras and the four parameters $\{\alpha_{13}, \boldsymbol{D}_{13}\}$ for uncalibrated cameras. They need to be made consistent w.r.t. the parameters of the first image pair, which we will only discuss for calibrated cameras. For uncalibrated cameras, cf. Avidan and Shashua (1998).

scale transfer

The *scale transfer* for two photogrammetric models consists in determining the ratio of the lengths of the two bases or the scale ratio of the two models, cf. (13.293), p. 607, using the distances of 3D points $\mathcal{X}_i$ in both models from the common projection centre O':
$$\frac{{}^m\lambda_3}{{}^m\lambda_2} = \frac{|\mathbf{b}_{13}|}{|\mathbf{b}_{12}|} = \frac{\sum_i w_i \frac{|{}^c\boldsymbol{X}_{i3}|}{|{}^c\boldsymbol{X}_{i2}|}}{\sum_i w_i}\,, \tag{14.65}$$
where the weights should reflect the accuracy of the distances (e.g., $\sigma_{Z_i} \propto Z_i^2$ leads to $w_i \propto 1/Z_i^4$, cf. (13.283), p. 604).

This two step procedure can be parallelized and used to check the correspondence. However, this checking is weak for all correspondences, as only outliers across the epipolar lines can be identified.

14.2.1.3 Procedure with Three Relative Orientations

The difficulty of checking correspondences with only two relative orientations and the asymmetry w.r.t. the three images motivate fusing the results of the relative orientations of all image pairs. Using triples of image pairs is also a classical method to check correspondences between more than three images.

For uncalibrated cameras we obtain the three fundamental matrices F_{12}, F_{23}, and F_{31}. They are represented by 21 parameters. As the relative orientation of the image triplet has only 18 degrees of freedom, the three fundamental matrices generally are not consistent when independently estimated from image correspondences. They need to fulfil three constraints. If the three projection centres are not collinear, these three constraints can be used to check the consistency of the fundamental matrices which result from using each of the three projection centres as an observed point in the other two images,

$$\mathbf{e}_1''^{\mathsf{T}}\mathsf{F}_{23}\mathbf{e}_1''' \stackrel{!}{=} 0\,, \quad \mathbf{e}_2'''^{\mathsf{T}}\mathsf{F}_{31}\mathbf{e}_2' \stackrel{!}{=} 0\,, \quad \mathbf{e}_3'^{\mathsf{T}}\mathsf{F}_{12}\mathbf{e}_3'' \stackrel{!}{=} 0\,. \tag{14.66}$$

A method to arrive at consistent fundamental matrices is given by Sinha and Pollefeys (2010).

For calibrated cameras we obtain the three essential matrices E_{12}, E_{23}, and E_{31}, represented by 15 parameters. As the relative orientation of the image triplet has 11 degrees of freedom, the three essential matrices need to fulfil four constraints. Three of them capture the closure of the relative rotations:

$$R_{12}R_{23}R_{31} \stackrel{!}{=} I_3\,. \tag{14.67}$$

In the case of random errors, the product $R = R_{12}R_{23}R_{31}$ will be close to a unit rotation, thus close to a skew symmetric matrix $S(\boldsymbol{r})$ with the entries $\boldsymbol{r}$ on the off-diagonal terms representing small angles. If no covariance matrix for the rotations is available, an approximate test of the total angle $|\boldsymbol{r}|$ can be based on the average uncertainty of the angles.

The fourth constraint refers to the three basis vectors, which should be coplanar. Thus the determinant

$$b = |\mathbf{b}_{12}, \mathbf{b}_{23}, \mathbf{b}_{31}| \stackrel{!}{=} 0 \tag{14.68}$$

of the three vectors should be zero. If the three base vectors are unit vectors, the determinant measures the angle between the normals of two of the vectors with the third vector. The angle can be compared with the expected uncertainty of the base directions.

A joint statistical test on the vector $\boldsymbol{d} = [\boldsymbol{r}^{\mathsf{T}}, b]^{\mathsf{T}}$ is based on the Mahalanobis distance $\underline{\boldsymbol{d}}^{\mathsf{T}}\Sigma_{dd}\underline{\boldsymbol{d}} \sim \chi^2_4$, where the covariance matrix Σ_{dd} depends on the covariance matrix of the rotations and base directions. Exercise 14.3

The last constraint can only be used if the three projection centres are not collinear. Otherwise we could use a scalar constraint involving the trifocal tensor, e.g., the one for checking the consistency of a point in the first image and two lines in the second and the third image, cf. (14.51), p. 630,

$$\mathbf{l}''^{\mathsf{T}}\mathsf{T}_{\diamond}(\mathbf{x}')\mathbf{l}''' \stackrel{!}{=} 0\,, \tag{14.69}$$

where the line l'' passes through x'' and the line l''' passes through x'''. This constraint can be replaced by the following, which avoids the determination of the trifocal tensor:

$$|\mathsf{P}_1^{\mathsf{T}}\mathbf{l}_1'\,, \mathsf{P}_1^{\mathsf{T}}\mathbf{l}_2'\,, \mathsf{P}_2^{\mathsf{T}}\mathbf{l}''\,, \mathsf{P}_3^{\mathsf{T}}\mathbf{l}'''| \stackrel{!}{=} 0\,, \tag{14.70}$$

where the two lines l_1' and l_2' are chosen to define the point x'. In both cases, care has to be taken when selecting the lines through the given image points in order to avoid a critical configuration, see the discussion in Sect. 14.2.3.2, p. 638.

14.2.2 Direct Solutions for the Trifocal Tensor

The relative orientation of the image triplet uses corresponding points $(\mathbf{x}'_i, \mathbf{x}''_i, \mathbf{x}'''_i)$ or corresponding lines $(\mathbf{l}'_j, \mathbf{l}''_j, \mathbf{l}'''_j)$ and exploits the constraints to yield the trifocal tensor, from which consistent projection matrices can be derived. We will only sketch the available direct solutions.

The relative orientation of three uncalibrated views aims at directly estimating the trifocal tensor (Hartley, 1997b). We mention two of them. They differ in the parametrization of the trifocal tensor and in their ability to handle more than the minimum number of correspondences:

1. The 27 entries of the $3 \times 3 \times 3$ tensor can be determined in closed form. This solution is based on the constraints between corresponding image points and lines in Table (14.4), p. 632. The constraints are all linear in the entries of T. This allows us to write all constraints in the form $\mathbf{A}\mathbf{t} = \mathbf{0}$, so the estimation can proceed according to Sect. 4.9.2, p. 177. The entries of the matrix $\mathbf{A}$ depend on the observed entities, and the 27-vector $\mathbf{t}$ contains the elements of the trifocal tensor. We need at least seven corresponding points or 13 corresponding lines to obtain a direct solution of this type.
   The advantage of this procedure is that any number of corresponding points or lines can be used to obtain an over-constrained solution for the entries of T. The disadvantage is that the resulting tensor is not consistent. As the tensor contains 27 entries, but has only 18 degrees of freedom, there are nine constraints between the entries of T (cf. Hartley and Zisserman, 2000; Ressl, 2003). These constraints are not taken into consideration during the estimation procedure. However, empirical studies by Ressl (2003) suggest that the effect of not taking the constraints into account is negligible if the 3D points are not close to a plane.
   The solution for straight line-preserving cameras fails or is unstable in the case of coplanar or nearly coplanar object points. Investigations by Ressl (2003) suggest that small deviations (even below 5%) from coplanarity yield stable results, except in the case of forward motion of the camera towards the 3D points.
2. A minimum parametrization of the trifocal tensor with 18 parameters is used. The parameters can be directly determined from six corresponding points in the three images (Torr and Zisserman, 1997).
   The advantages of this procedure are the consistency of the resulting tensor and the ability to use it for a RANSAC procedure in the case of erroneous correspondences. The disadvantage is that redundant observations cannot be directly integrated into the procedure and only corresponding points – not lines – are involved. The description of the algorithm is lengthy and can be found in Hartley and Zisserman (2000).

In both cases consistent fundamental matrices and projection matrices need to be derived (cf. Ressl, 2003, Sect. 7.4). Both procedures fail in the presence of a critical configuration, especially if all 3D points are coplanar.

The relative orientation of three calibrated images requires at least four points to determine the 11 parameters, cf. Table (14.1), p. 623 and Fig. 14.3. It has been shown that this solution is unique in general (cf. Holt and Netravali, 1995). Each point observed in all three images gives rise to three constraints, as we have six observed image coordinates for three scene coordinates. Therefore at least four points are generally necessary to yield a solution. Then we already have one redundancy; thus, the image coordinates cannot have arbitrary values.

There exist two solutions to this highly complex problem, for which we refer to the original papers.

1. The first solution, given by Nistér and Schaffalitzky (2006), starts from the observation that the relative orientation of two images requires five correspondences. Therefore, having only four correspondences in two views yields a one-parameter family of relative orientations. The epipole of one image can be explicitly parametrized by one

parameter, which yields a tenth-order curve for that epipole. Searching for a solution consistent with the observations in the third image for this particular parameter leads to the final solution. A fifth point is used to rule out certain branches of the solution space. Finally, using triangulation (cf. Sect. 13.4, p. 596), the 3D coordinates of the four scene points, and – using spatial resection (cf. Sect. 12.2.4, p. 513) – the pose of the third image can be determined.

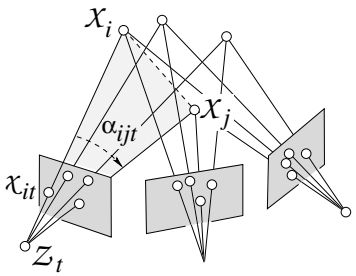


Fig. 14.3 Minimum solution for the relative orientation of three calibrated images requires four points, $\mathcal{X}_i$, seen in all images. The direct solution by Li (2010) determines all 12 distances from the three cameras to the four scene points and the six distances between the scene points in a local scale from the 18 angles derivable from the observed image points using convex programming

2. The second solution, given by Li (2010), determines the 3D position of the scene points without calculating the orientation, inspired by the direct solution for the spatial resection, where the distances from the scene points are also determined first, see Fig. 14.3. It uses the cosine law of a triangle to relate the distances $s_{it} = \overline{\mathcal{X}_i\mathcal{Z}_t}$ and $s_{jt} = \overline{\mathcal{X}_j\mathcal{Z}_t}$ from one camera to two scene points $\mathcal{X}_i$ and $\mathcal{X}_j$, their distance $l_{ij} = \overline{\mathcal{X}_i\mathcal{X}_j}$ and the observed angle α_{ijt}. This can be done for all 18 triangles where one point is one of the three camera centres and the other two points are the ends of the six distances between the four scene points. These 18 constraints are quadratic in the distances. By relaxing the problem it is transformed into a convex problem for the 12 distances from the scene points and the six distances between the scene points. Using the six distances between the scene points, the relative position of the four 3D points can easily be determined. The method is designed to also handle more than four 3D points.

Further details are given in the publications.

14.2.3 Iterative Estimation of the Relative Orientation of the Image Triplet

We now discuss statistically optimal solutions for the orientation of the image triplet with outstanding properties: They yield best estimates for all parameters, as they take into account all available information and constraints; and they yield the best tests for small and medium outliers. We therefore assume that the determination of approximate values was combined with the identification of large outliers.

The solutions can be used for calibrated, partially calibrated, or totally uncalibrated cameras.

The bundle solution can be based on the projection equation for points and lines and leads to a relative orientation or possibly a complete orientation including control points or lines. Alternatively, the relative orientation of the three images can be based on constraints between the images and need not include the unknown 3D points in the estimation.

14.2.3.1 Bundle Solution with the Projection Equations

Let the three projections for the three images $t = 1, 2, 3$, i.e., the nonlinear observation equations, be given by

$$\mathbb{E}(\mathbf{x}'^s_{it}) = \mathsf{N}\left(\mathsf{P}_t(\widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{s}}_t)\, \widehat{\mathbf{X}}_i\right) \quad i = 1, ..., I; t = 1, 2, 3 \tag{14.71}$$

$$\mathbb{E}(\mathbf{l}'^s_{jt}) = \mathsf{N}\left(\mathsf{Q}_t(\widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{s}}_t)\, \widehat{\mathbf{L}}_j\right) \quad j = 1, ..., J; t = 1, 2, 3\,, \tag{14.72}$$

referring to I unknown 3D points $\mathcal{X}_i$ or J unknown 3D lines $\mathcal{L}_j$ and yielding the coordinates of the image points $\mathcal{x}'_{it}$ and the image lines $\mathcal{l}'_{jt}$. Observe that this setup also allows for observed points or lines which are only observed in one or two images. Using the spherically normalized image coordinate vectors eliminates the unknown scale of the homogeneous coordinates. The projection matrices P_t and Q_t explicitly depend on parameters $\widehat{\boldsymbol{s}}_t$ for the interior and parameters $\widehat{\boldsymbol{p}}_t$ for the exterior orientation and may be known, partially unknown, or completely unknown. In addition, the intrinsic parameters of the images may be assumed to be the same for all images, which leaves enough degrees of freedom for realistic modelling.

The unknown parameters of the exterior orientation may be collected in the vector $\widehat{\boldsymbol{p}} = [\widehat{\boldsymbol{p}}_t]$, and the parameters of the interior orientation in the vector $\widehat{\boldsymbol{s}} = [\widehat{\boldsymbol{s}}_t]$.

Together with the I_{CP} control points $\mathcal{X}_{CP,i}$ and J_{CL} control lines $\mathcal{L}_{CL,j}$, with similar observation equations,

$$\mathbb{E}(\mathbf{x}'^s_{it}) = \mathsf{N}\left(\mathsf{P}_t(\widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{s}}_t)\, \mathbf{X}_{CP,i}\right) \quad i = 1, ..., I_{CP}; t = 1, 2, 3 \tag{14.73}$$

$$\mathbb{E}(\mathbf{l}'^s_{jt}) = \mathsf{N}\left(\mathsf{Q}_t(\widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{s}}_t)\, \mathbf{L}_{CL,j}\right) \quad j = 1, ..., J_{CL}; t = 1, 2, 3\,, \tag{14.74}$$

we may iteratively determine the optimal estimate of the unknown orientation parameters and the unknown 3D elements by a bundle adjustment based on the Gauss–Markov model, possibly with constraints between the unknown parameters. This requires minimal representations for the homogeneous coordinate vectors, cf. Sect. (10.2.2.1), p. 369. We need approximate values for all parameters in (14.71) and (14.72), namely $\widehat{\boldsymbol{p}}_t$, $\widehat{\boldsymbol{s}}_t$, $\widehat{\mathbf{X}}_i$, and $\widehat{\mathbf{L}}_j$, and prior knowledge about the precision of the observed image points $\mathbf{x}'_{it}$ and lines $\mathbf{l}'_{jt}$. These approximate values can be determined with one of the procedures from the previous section, 14.2.1, p. 633.

14.2.3.2 Bundle Solution with Constraints

If the number of points or lines is large, it may be advantageous to only solve for the unknown camera parameters and use the constraints between the observed image points or lines, namely the coplanarity or epipolar constraints and the trifocal constraints. These constraints depend on the image features and the fundamental matrices or the trifocal tensor. As the fundamental matrices and the trifocal tensor may be explicitly expressed in terms of the rows of the projection matrices, which themselves depend on the parameters of the exterior and the interior orientation, we may write the constraints directly in terms of these parameters. As a result, we can express the estimation problem in the form of the Gauss–Helmert model with constraints, cf. Sect. 4.8.2, p. 163.

Three corresponding image points yield six observed coordinates, while the corresponding 3D point is described by three coordinates. Thus we need *three independent constraints* to exploit the correspondence.

For estimation we represent the geometry using the three projection matrices, where the first one is normalized. The three view geometry is represented with two projection matrices $\mathsf{P}_t, t = 2, 3$, which are represented with more than the required parameters. Therefore we need additional constraints between the parameters.

For *uncalibrated cameras* we use the representation

$$\mathsf{P}_1 = [I_3|\mathbf{0}]\,, \qquad \mathsf{P}_2 = [\mathsf{A}_2|\mathbf{a}_2]\,, \qquad \mathsf{P}_3 = [\mathsf{A}_3|\mathbf{a}_3]\,, \tag{14.75}$$

with the unknown 24-vector $\boldsymbol{x} = [\text{vec}\mathsf{P}_1; \text{vec}\mathsf{P}_2]$. As the trifocal tensor only has 18 degrees of freedom, cf. Table (14.1), p. 623, we use the following six additional constraints between the 24 unknown parameters representing the projection matrices (cf. Ressl, 2003),

$$|\mathbf{a}_2| = 1 \qquad |\mathbf{a}_3| = 1\,, \qquad \mathsf{A}_2^\mathsf{T}\mathbf{a}_2 = \mathbf{0}\,, \qquad ||\mathsf{T}||^2 = 1\,, \tag{14.76}$$

with $T_{i,jk}$ from (14.14), p. 625. For *calibrated cameras* we use

$${}^c\mathsf{P}_1 = [I_3|\mathbf{0}]\,, \qquad {}^c\mathsf{P}_2 = R_2[I_3| - \mathbf{b}_2]\,, \qquad {}^c\mathsf{P}_3 = R_3[I_3| - \mathbf{b}_3]\,. \tag{14.77}$$

Here we only have one constraint between the 12 parameters: the basis to the second camera must have length 1,

$$|\mathbf{b}_2| = 1\,. \tag{14.78}$$

For each 3D point which is visible in all three images we always use two of the epipolar constraints and one trifocal constraint, as the epipolar constraints are simpler than the trifocal constraints and three epipolar constraints cannot be used in the case of collinear projection centres.

The *first two constraints* are the two epipolar constraints w.r.t. the first image

$$g_1 := \mathbf{x}'^\mathsf{T}\mathsf{F}_{12}\mathbf{x}'' = 0 \quad \text{and} \quad g_2 := \mathbf{x}'^\mathsf{T}\mathsf{F}_{13}\mathbf{x}''' = 0\,, \tag{14.79}$$

enforcing the position of the points in the second and third image across their epipolar lines. With a specific choice of the coordinate system, the two fundamental matrices are $\mathsf{F}_{1t} = A_k\mathsf{S}(\mathbf{a}_k)$, cf. (13.13), p. 554, leading to the two coplanarity constraints

$$g_1(\mathbf{x}', \mathbf{x}''; \mathsf{A}_2, \mathbf{a}_2) := \mathbf{x}'^\mathsf{T}\mathsf{A}_2\mathsf{S}(\mathbf{a}_2)\mathbf{x}'' = 0 \tag{14.80}$$

$$g_2(\mathbf{x}', \mathbf{x}'''; \mathsf{A}_3, \mathbf{a}_3) := \mathbf{x}'^\mathsf{T}\mathsf{A}_3\mathsf{S}(\mathbf{a}_3)\mathbf{x}''' = 0\,. \tag{14.81}$$

For calibrated cameras with projection matrices these constraints reduce to

$$g_1({}^c\mathbf{x}', {}^c\mathbf{x}''; \mathsf{R}_2, \mathbf{b}_2) := {}^c\mathbf{x}'^\mathsf{T}\mathsf{S}(\mathbf{b}_2)R_2^\mathsf{T}\,{}^c\mathbf{x}'' = 0 \tag{14.82}$$

$$g_2({}^c\mathbf{x}', {}^c\mathbf{x}'''; \mathsf{R}_3, \mathbf{b}_3) := {}^c\mathbf{x}'^\mathsf{T}\mathsf{S}(\mathbf{b}_3)R_3^\mathsf{T}\,{}^c\mathbf{x}''' = 0\,. \tag{14.83}$$

The *third constraint*, for checking the direction along the epipolar lines, can be determined in the following way. The image points can be represented by the intersection of two image lines. Each of these image lines gives rise to a projection plane, four of which can be checked for intersection at a single point. Thus we arrive at a constraint of the form given in (14.70), p. 635, namely

$$|\mathbf{A}_{l'_1}\,, \mathbf{A}_{l'_2}\,, \mathbf{A}_{l''}\,, \mathbf{A}_{l'''}| = |\mathsf{P}_1^\mathsf{T}\mathbf{l}'_1\,, \mathsf{P}_1^\mathsf{T}\mathbf{l}'_2\,, \mathsf{P}_2^\mathsf{T}\mathbf{l}''\,, \mathsf{P}_3^\mathsf{T}\mathbf{l}'''| \stackrel{!}{=} 0\,; \tag{14.84}$$

see Fig. 14.4.

Since the two epipolar constraints (14.79) involving the image point $\mathcal{X}'$ in the first image guarantee that the two points $\mathcal{X}''$ and $\mathcal{X}'''$ pass through the epipolar lines in the second and the third image, we in addition need to guarantee that the two projection lines $\mathcal{L}_{X''}$ and $\mathcal{L}_{X'''}$ intersect the projection line $\mathcal{L}_{X'}$ in the same point. This is equivalent to require that the two projection planes $\mathcal{A}_{l''}$ and $\mathcal{A}_{l'''}$ intersect $\mathcal{L}_{X'}$ in the same point, or that the two base vectors $\mathbf{b}_2$ and $\mathbf{b}_3$ have the proper scale ratio, see the scale transfer in (14.65).

In order to achieve a numerically stable constraint we choose the following four lines through the three points and specify them by their directions $\mathcal{V}$:

$$\mathcal{L}'_1 = \mathcal{V}'_x \wedge \mathcal{X}'\,, \quad \mathcal{L}'_2 = \mathcal{V}'_y \wedge \mathcal{X}'\,, \quad \mathcal{L}'' = \mathcal{V}'' \wedge \mathcal{X}''\,, \quad \mathcal{L}''' = \mathcal{V}''' \wedge \mathcal{X}'''\,, \tag{14.85}$$

where

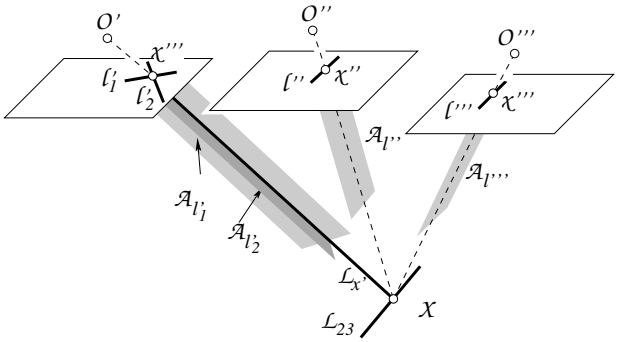


Fig. 14.4 Trifocal constraint. We choose four lines through the three given image points. The corresponding four projection planes must intersect in one point. The constraint also requires that the projection line $\mathcal{L}_{x'}$ and the intersection line $\mathcal{L}_{23}$ of the two other projection planes are coplanar

- the lines

$$\ell_1' = \begin{bmatrix} 0 \\ w' \\ -v' \end{bmatrix}, \quad \text{since} \quad \mathbf{v}_x' = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \tag{14.86}$$

  and

$$\ell_2' = \begin{bmatrix} w' \\ 0 \\ -u' \end{bmatrix}, \quad \text{since} \quad \mathbf{v}_y' = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{14.87}$$

  pass through $\mathbf{x}'$ and are parallel to the two coordinate axes and
- the lines ℓ'' and ℓ''' are perpendicular to the epipolar lines in the second and the third image; e.g., for epipolar line $[a'', b'', c'']^\mathsf{T}$ in the second image with (7.16) we have

$$\mathbf{l}'' = \begin{bmatrix} b''w'' \\ -a''w'' \\ a''v'' - b''u'' \end{bmatrix}, \quad \text{since} \quad \mathbf{v}'' = \begin{bmatrix} a'' \\ b'' \\ 0 \end{bmatrix}; \tag{14.88}$$

  and equivalently for the line ℓ''' through $\mathbf{x}'''$ in third image.

The resulting third constraint for a point triplet finally reads as

$$g_3(\mathsf{T}, \mathbf{x}', \mathbf{x}'', \mathbf{x}''') = |\mathsf{P}_1^\mathsf{T}\mathsf{S}(\mathbf{v}_1')\mathbf{x}', \mathsf{P}_1^\mathsf{T}\mathsf{S}(\mathbf{v}_2')\mathbf{x}', \mathsf{P}_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')\mathbf{x}'', \mathsf{P}_3^\mathsf{T}\mathsf{S}(\mathbf{v}''')\mathbf{x}'''| = 0\,. \tag{14.89}$$

When using this constraint in an estimation procedure, the vectors $\mathbf{v}$ can be treated as fixed entities.

Using the specific choice of the projection matrices with $\mathsf{P}_1 = [I_3|\mathbf{0}]$ for uncalibrated cameras, the constraint then reads as

$$g_3 := \begin{vmatrix} \mathsf{S}(\mathbf{v}_1')\mathbf{x}' & \mathsf{S}(\mathbf{v}_2')\mathbf{x}' & \mathsf{A}_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')\mathbf{x}'' & \mathsf{A}_3^\mathsf{T}\mathsf{S}(\mathbf{v}''')\mathbf{x}''' \\ 0 & 0 & \mathbf{a}_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')\mathbf{x}'' & \mathbf{a}_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')\mathbf{x}'' \end{vmatrix} = 0\,. \tag{14.90}$$

For calibrated cameras the constraint is

$$g_3 := \begin{vmatrix} \mathsf{S}(\mathbf{v}_1')^c\mathbf{x}' & \mathsf{S}(\mathbf{v}_2')^c\mathbf{x}' & R_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')^c\mathbf{x}'' & R_3^\mathsf{T}\mathsf{S}(\mathbf{v}''')^c\mathbf{x}''' \\ 0 & 0 & -\mathbf{b}_2^\mathsf{T}R_2^\mathsf{T}\mathsf{S}(\mathbf{v}'')^c\mathbf{x}'' & -\mathbf{b}_3^\mathsf{T}R_3^\mathsf{T}\mathsf{S}(\mathbf{v}''')^c\mathbf{x}''' \end{vmatrix} = 0\,. \tag{14.91}$$

These constraints work for all points if they are not close to an epipole. Then at least two projection planes are nearly parallel and the intersecting line is numerically unstable or, in the case of observational noise, inaccurate. This especially holds for forward motion, for which the image points close to the focus of expansion, i.e., the epipole, cannot be handled.

Corresponding lines (l'_j, l''_j, l'''_j) can directly use the trifocal constraint (14.59), p. 632 and the representation of the normalized trifocal tensor (14.36), p. 628. We again have to avoid critical configurations, as discussed in Sect. 14.1.3.1, p. 626.

14.3 Exercises

1. (2) Your supervisor has heard that evaluating image pairs sometimes leads to difficulties and wants to know whether and how these difficulties can be solved using a third image of the object.
   a. Is your boss right? Why? Give a simple example which will convince your boss to use three images instead of two.
   b. Assume the first two images are in normal position. You have two choices when taking the third image:
      (A) The projection centre O''' is in the prolongation of O' and O''.
      (B) The projection centre O''' lies outside the basis $(O'O'')$ in a plane through O' and parallel to the common image plane of the two first images.

      Given are two putative corresponding points x' and x'' in the first two images, you want to determine/predict the coordinates of the corresponding point x''' in the third image. For both configurations (A) and (B) discuss the two proposals for a procedure:
      (a) Determine the 3D point X by triangulation and project it into the third image using the projection matrix P''', which is assumed to be known.
      (b) Intersect the two epipolar lines $l'''(x')$ and $l'''(x'')$ in the third image.
      (c) Use the trifocal tensor and (14.58), p. 631, see line 10 in Table 14.3.

      Answer the following questions concerning the six cases (Aa) to (Bc):
      i. Does the procedure work in all cases? Take into account that points may be at infinity.
      ii. Give an approximation for the number of operations (multiplications) per image pair without taking into account possible zeros in the matrices (i.e., worst case scenario). Assume the prediction of x''' needs to be performed very often (e.g., more than 1 000 times), such that providing the relevant matrices does not play a role. *Hint:* How do you determine the epipolar lines?
   c. Which camera arrangement, (A) or (B), and which method, (a), (b), or (c), would you recommend?

2. (2) The orientation of three images $t = 1, 2, 19$, has been determined by spatial resection. The fundamental matrices between image 19 and the other two images 1 and 2 are
$$\mathsf{F}_{19,1} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & +1 \\ -1 & +1 & 0 \end{bmatrix} \quad \text{and} \quad \mathsf{F}_{19,2} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} . \tag{14.92}$$
   a. The vector between the two projection centres O_1 and O_{19} is $\boldsymbol{d}_{1,19} = \boldsymbol{Z}_{19} - \boldsymbol{Z}_1 = [736, 736, 0]^\mathsf{T}$ m. Due to the flight path the two images 1 and 19 are mutually rotated by 180°around the Z-axis. Confirm $\mathsf{F}_{19,1}$ using this information.
   b. The image coordinates of the point X_{22} are measured in the two images, leading to ${}^1\boldsymbol{x}_{22} = [-690, -460]^\mathsf{T}$ pixel and ${}^2\boldsymbol{x}_{22} = [230, -460]^\mathsf{T}$ pixel. Determine the corresponding point coordinates ${}^{19}\boldsymbol{x}_{22}$ in image 19.

3. (2) Refer to Sect. 14.2.1.3, p. 635. Derive a statistical test for the consistency of three essential matrices. Specifically derive the covariance matrix of the 5-vector $\boldsymbol{d} = [\boldsymbol{r}^\mathsf{T}, \boldsymbol{b}^\mathsf{T}]^\mathsf{T}$ if the covariance matrices of the parameters of the three relative orientations are given.

Chapter 15
Bundle Adjustment

15.1 Motivation for Bundle Adjustment and Its Tasks 644
15.2 Block Adjustment 645
15.3 Sparsity of Matrices, Free Adjustment and Theoretical Precision 651
15.4 Self-calibrating Bundle Adjustment 674
15.5 Camera Calibration 696
15.6 Outlier Detection and Approximate Values 707
15.7 View Planning 715
15.8 Exercises 722

Bundle adjustment is a unified method to simultaneously estimate the internal and external camera parameters and the 3D coordinates of the scene points in a statistically optimal manner. The number of cameras and scene points can be arbitrarily large. Therefore it can be used to solve the previously discussed problems with only one, two, or three images, including triangulation and absolute orientation in a unified manner. Conceptually, it solves the inverse problem to computer graphics: given the images of an unknown scene the task is to recover the scene structure, i.e., the visible surface together with the parameters describing the cameras used for taking the images, thus exploiting all available information. In our context we start from observed image bundles, from observed or given 3D scene points, and from priors of the internal or external parameters. As for the orientation tasks for one, two and three images, we assume that the image points or lines are available and their correspondence established. Integrated approaches for recovering the scene structure and the camera poses from the original digital images are discussed in the second volume.

Bundle adjustment is a special case of what are called block adjustments, where many units available in local coordinate systems are simultaneously fused and transformed into a global coordinate system. Such units may be point clouds derived from image pairs, but also from laser scans. The aggregated set of all units is called a *block*, a term borrowed from the French *en bloc*. The transformations may be 2D if tilts are nonexistent or negligible, or 3D in the general case. In all cases, the number of units may be very large, up to several thousands, requiring solutions to handle very large equation systems. Exploiting the sparsity of the Jacobians and the normal equations makes it possible to efficiently solve such adjustments on desktop computers.

The geometric setup of such blocks may vary from very regular to very irregular. Therefore, not only are statistically optimal techniques indispensable in order to guarantee good results, they are also needed for the evaluation of the 3D geometry of the setup, which is generally complex.

This chapter first gives an overview of the general setup of a block adjustment and – for pedagogical reasons – discusses its properties in detail using a planar setting where 2D point clouds are fused via similarity transformations. These properties refer to (1) the sparsity structures of the matrices, (2) the reduction of the normal equations, (3) the realization of free block adjustments where no scene control is used, and the corresponding

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_15

gauge problem. The chapter closes with an in-depth analysis of the theoretical quality which can be expected for different configurations, including the specific case of what is called the loop closing problem. We then generalize the results to the self-calibrating bundle adjustment in 3D, first formalized in modern terms by H. H. Schmid (1958) and successfully applied since then (cf. McGlone, 2013, Chap. 14). We discuss the choice of additional parameters for modelling the internal geometry of cameras and how to evaluate bundle adjustment results using the sensitivity analysis techniques discussed in Part I. Camera calibration requires special attention, as proper calibration of cameras is a prerequisite for efficient use of imagery to reliably recover 3D scene geometry. We provide closed form and incremental methods for the determination of approximate values required to solve the nonlinear optimization problem of bundle adjustment. Based on the previous findings we provide guidelines for view planning in order to achieve certain user requirements in terms of accuracy or completeness.

15.1 Motivation for Bundle Adjustment and Its Tasks

In the previous chapters we have discussed the orientation of only a few images and the reconstruction of scene points visible in these images. However, there are quite a few tasks where many more images are necessary:

extended scenes
- The geometric complexity of the scene is high, such that many images are required in order to observe all parts of the scene with at least three images. Examples are extended scenes like landscapes, complete cities, indoor or outdoor scenes of single buildings, and panorama images generated via stitching.

high accuracy
- The envisaged accuracy is too high to be achieveed by using only a few images. Increasing the accuracy and the resolution by reducing the average distance to the scene then requires an increased number of images in order to cover the complete scene. A prominent example is the mapping of large areas using aerial images, the generation of a geo-referenced image mosaic, familiar from traditional photogrammetric products such as orthophotos or from Google maps. Stitching of many images to obtain a high-resolution image composition also falls into this category.

visual odometry
- The images are taken with a video camera from a moving platform in order to support visual odometry, i.e., real-time ego-motion determination or scene exploration, e.g., when performing simultaneous localization and mapping (SLAM).

photo collections
- The images are taken from photo collections on the internet in order to obtain a 3D reconstruction of the scene.

In all these cases, we can join the images by determining their spatial pose during exposure based on scene details visible in several images. We encounter a number of problems when addressing this task:

- **View planning and control point distribution.** In many cases, the user has control over where the views are taken, such that there is a possibility of arriving at a desired quality of the bundle adjustment result, e.g., by using a camera with an adequate viewing angle and by enforcing a certain minimum overlap to guarantee a certain minimum precision and reliability of the result. Also, if the scene is extended so that many images are necessary, it is not necessary to have control points whose scene coordinates are known and which are related to the cameras or images for *each* camera or image. Therefore we discuss the problem of planning the geometric distribution of views and control points in Sect. 15.7, p. 715.
- **Approximate values.** The models for block adjustment are inherently nonlinear and so require approximate values for all parameters. They need to be determined efficiently in the presence of possible outliers. Direct solutions only exist for special configurations. Therefore we will also discuss efficient robust sequential methods for determining approximate values of large blocks in Sect. 15.6, p. 707

- **Camera calibration.** In order to exploit the accuracy potential of digital images for scene reconstruction and pose estimation, calibration of cameras is indispensable. This requires cameras to be metric, i.e., to be stable over time, as the observed images can be undistorted before further processing, as presumed in the previous chapters. Here we will discuss how to perform camera calibration, thus to reliably and efficiently determine the envisaged corrections leading to a perspective or a spherical camera in Sect. 15.5, p. 696.
- **Self-calibrating bundle adjustment.** As real cameras often are not stable enough over time, bundle adjustment can be augmented by additional parameters which compensate for possible systematic effects. This leads to the concept of self-calibrating bundle adjustment. Depending on the application, self-calibrating bundle adjustment tasks are *scene reconstruction*, *ego-motion determination* and *camera calibration*, or a combination of them, such as simultaneous localization and mapping, where scene reconstruction and ego-motion determination are integrated. We will especially discuss the evaluation of bundle adjustment results in Sect. 15.4, p. 674.
- **Handling large equation systems.** The resulting normal equation systems become extremely large, possibly with numbers of unknowns in the order of 10^6, which would make it impossible to handle these tasks on normal computers. However, the equation systems turn out to be quite sparse, i.e., only a very small percentage of the elements of the normal equation matrix is nonzero. This results from the fact that most scene features are visible only in a small number of images. Take the example of image stitching for generating a panorama: here scene features are only seen in two or three neighbouring images. We will address the *sparsity* of the normal equation system and how it can be exploited for efficient numerical solutions in Sect. 15.3.3, p. 655.
- **Gauge:** Without any scene information, the coordinate system of the orientation parameters and the scene features cannot be determined uniquely. Therefore the coordinate system needs to be fixed without imposing constraints on the given image features, e.g., by the orientation of the first camera and possibly some additional parameters. But this choice is arbitrary and influences the resulting parameters and their precision. The chosen coordinate system fixes the *gauge* of the resulting parameters of the scene coordinates and the poses as discussed in Sect. 4.5, p. 108 and will be described in Sect. 15.3.4, p. 663.

We start with describing the setup of block adjustment i.e., the general scheme for fusing many units by statistically optimal estimation techniques. For details on open software see the tutorial by Leotta et al. (2015).

15.2 Block Adjustment

15.2.1 General Setup of Block Adjustment 646
15.2.2 Examples for Block Adjustments 648

Block adjustment, understood as simultaneous registration of geometric units such as bundles of rays captured by cameras or geometric models captured by image analysis or laser range finders, shows a generic structure and has prominent instances useful in scene recovery and pose estimation. We assume the scene is static. For the spatiotemporal reconstruction of non-static scenes from images cf. Vo et al. (2016).

15.2.1 General Setup of Block Adjustment

The general setup of all registration problems is the following.

scene features

- We assume the scene to consist of a set of I *scene features* $\mathcal{F}_i$, which are to be determined ($\mathcal{I} = \{1, ..., i, ..., I\}$). The scene may be flat or 3D, therefore the scene features may be 2D or 3D points, lines, or regions. The unknown parameters of the scene features are collected in the vector $\boldsymbol{k}_i$, where the letter k stands for coordinate parameters, say of 2D or 3D points. Thus we have scene features with their unknown parameters,

$$\mathcal{F}_i(\boldsymbol{k}_i)\,, \quad i \in \mathcal{I}\,. \tag{15.1}$$

transformation parameters

- We assume that we have a set $\mathcal{T} = \{1, ..., t, ...T\}$ of T images of the scene. Each image $t \in \mathcal{T}$ is described by its projection $\mathcal{P}_t$ of the scene modelled by some unknown parameters $\boldsymbol{p}_t$. The *transformation parameters* of all images are collected in the vector $\boldsymbol{p}$. The transformations may be motions, projections, or a combination of both. Hence we have the projections with their unknown parameters,

$$\mathcal{P}_t(\boldsymbol{p}_t)\,, \quad t \in \mathcal{T}\,. \tag{15.2}$$

image features

- In each image $t \in \mathcal{T}$ we have observed *image features* f_{it} of the scene features $\mathcal{F}_i$. Again, these may be points, lines, or regions. If they are points, they geometrically establish bundles of rays. The indices indicate the image t and the scene feature i. Thus we also assume the association between image and scene features to be given, a problem to be solved a priori. Not each scene feature needs to be observed in each image; thus, the index pairs (it) are elements of some subset $\mathcal{E} \subset \mathcal{I} \times \mathcal{T}$. The observations are parametrized by $\boldsymbol{l}_{it}$. Hence we have given

$$f_{it}(\boldsymbol{l}_{it})\,, \quad (it) \in \mathcal{E} \subset \mathcal{I} \times \mathcal{T}\,. \tag{15.3}$$

  We can interpret the setup using a bipartite graph, see Fig. 15.1, with nodes $i \in \mathcal{I}$ of scene features and nodes $t \in \mathcal{T}$ of projecting images which are connected by edges $(it) \in \mathcal{E}$ of observed image features f_{it}, which is also the reason we use the letter $\mathcal{E}$.

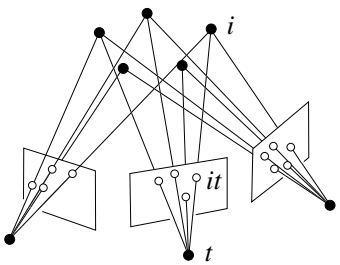


Fig. 15.1 A bundle block as bipartite graph: nodes $i \in \mathcal{I}$ of scene features $\mathcal{F}_i$ and nodes $t \in \mathcal{T}$ of images joined by edges $(it) \in \mathcal{E}$ of observations $\boldsymbol{l}_{it}$

If the scene is 3D, the captured data also may be 3D, as when the scene is observed using a laser range finder or some other 3D acquisition device, e.g., using a pair of mutually fixed cameras. Instead of image features we would then use features derived from the laser range data or the stereo images, such as corners, planes or quadrics. These features are generally represented by three or more parameters. The projections are then replaced by a spatial transformation, e.g., a motion or a similarity. In the following we will talk about images and image features, assuming that for laser range data or data from stereo cameras these features are of different nature.

In addition, we assume that the observation process provides some internal measure of the uncertainty of the observed image features. In most cases, we can assume the

observations in image t to be statistically independent of those in image t'. But this is not necessarily the case, e.g., if a point is tracked through an image sequence, the coordinates of points at least in neighbouring images will be statistically dependent. Mostly we will assume the observations $\boldsymbol{l}_{it}$ in different images and within one image to be mutually independent and represented by a covariance matrix $\Sigma_{l_{it}l_{it}}$.
Observe, for simplicity we exclude the situation in which one image feature is related to two or more scene features, e.g., we exclude the case where an image line is related to two scene points. This case could of course be addressed in the general setup, but would require a more advanced indexing scheme.

- In addition, we may have observed I_0 of the scene features $\mathcal{F}_{i0}(\boldsymbol{k}_{i0}), i \in \mathcal{I}_0$, in the following called *control features* as they define the reference coordinate system. These may be 3D points or lines from a map. Thus we assume observed parameters $\boldsymbol{k}_{i0}$ together with their covariance matrix $\Sigma_{k_{i0}k_{i0}}$. We can interpret this set of observed scene features as a set with index $t = 0$:

control features

$$\mathcal{F}_{i0}(\boldsymbol{k}_{i0}), \quad i \in \mathcal{I}_0 . \tag{15.4}$$

Similarly, we may have some T_0 direct observations $\boldsymbol{p}_t, t \in \mathcal{T}_0$, of the transformation parameters of the camera, measured in the scene coordinate system, e.g., by GPS or an inertial system:

$$\mathcal{P}_t(\boldsymbol{p}_t), \quad t \in \mathcal{T}_0 . \tag{15.5}$$

Again, we assume the uncertainty of these parameters to be available in the form of a covariance matrix $\Sigma_{p_t p_t}$.

- The mathematical model of the imaging process then can be described by its functional model, and the stochastical model, representing the assumed stochastic properties of the observed entities involved.
The *functional model* reads as

functional model

$$\widetilde{\boldsymbol{l}}_{it} = \boldsymbol{f}_{it}(\tilde{\boldsymbol{k}}_i, \tilde{\boldsymbol{p}}_t), \quad (it) \in \mathcal{E} . \tag{15.6}$$

It refers to the true observations $\tilde{\boldsymbol{l}}_{it}$, the true scene parameters $\tilde{\boldsymbol{k}} = [\tilde{\boldsymbol{k}}_i]$, and the true transformation parameters $\tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{p}}_t]$. It simply states that the coordinates $\boldsymbol{f}_{it}(\tilde{\boldsymbol{k}}_i, \tilde{\boldsymbol{p}}_t)$ of the scene feature with true coordinates $\tilde{\boldsymbol{k}}_i$ transformed into the image t with true parameters $\tilde{\boldsymbol{p}}_t$ are identical to the true coordinates $\tilde{\boldsymbol{l}}_{it}$ of the observed feature.
The functional model may depend on additional parameters $\boldsymbol{s}$ which take into account the unknown internal structure of the camera or the 3D acquisition device, e.g., if the observed image features are not corrected for image distortions and these distortions are too large to be acceptable.
We need to write the functional model as a set of implicit constraints if the observed features cannot be expressed as an explicit function of the scene coordinates and the transformation parameters,

$$\boldsymbol{g}_{it}(\widetilde{\boldsymbol{l}}_{it}, \tilde{\boldsymbol{k}}_i, \tilde{\boldsymbol{p}}_t) = \boldsymbol{0}, \quad (it) \in \mathcal{E} , \tag{15.7}$$

between the entities involved.
The *stochastical model* formalizes the assumptions about the statistical properties of the observations. For the observed image features, we assume

stochastical model

$$\mathbb{E}(\underline{\boldsymbol{l}}_{it}) = \tilde{\boldsymbol{l}}_{it}, \quad \mathbb{D}(\underline{\boldsymbol{l}}_{it}) = \Sigma_{l_{it}l_{it}} \quad (it) \in \mathcal{E} . \tag{15.8}$$

For the possibly observed scene features and the possibly observed transformation parameters we have, analogously,

$$\mathbb{E}(\underline{\boldsymbol{k}}_{i0}) = \tilde{\boldsymbol{k}}_i, \quad \mathbb{D}(\underline{\boldsymbol{k}}_{i0}) = \Sigma_{k_{i0}k_{i0}}, \quad i \in \mathcal{I}_0 , \tag{15.9}$$

$$\mathbb{E}(\underline{\boldsymbol{p}}_{0t}) = \tilde{\boldsymbol{p}}_j, \quad \mathbb{D}(\underline{\boldsymbol{p}}_{0t}) = \Sigma_{p_{0t}p_{0t}}, \quad t \in \mathcal{T}_0 . \tag{15.10}$$

Since we want to evaluate the result of the estimation, we assume in general that the observations are normally distributed,

$$\underline{\boldsymbol{l}}_{it} \sim \mathcal{N}(\tilde{\boldsymbol{l}}_{it}, \Sigma_{l_{it}l_{it}}), \quad \underline{\boldsymbol{k}}_{i0} \sim \mathcal{N}(\tilde{\boldsymbol{k}}_{i0}, \Sigma_{k_{i0}k_{i0}}), \quad \underline{\boldsymbol{p}}_{0t} \sim \mathcal{N}(\tilde{\boldsymbol{p}}_{0t}, \Sigma_{p_{0t}p_{0t}}). \tag{15.11}$$

- The goal is to obtain optimal estimates $\{\widehat{\boldsymbol{k}}, \widehat{\boldsymbol{p}}\}$ simultaneously for all parameters $\{\boldsymbol{k}, \boldsymbol{p}\}$ and the fitted observations $\widehat{\boldsymbol{l}}_{it}$ which satisfy the given functional model (15.6) or (15.7).

MAP estimate for block adjustment

We obtain the maximum a posteriori estimate (MAP estimate)

$$\{\widehat{\boldsymbol{k}}_i, \widehat{\boldsymbol{p}}_t\} = \text{argmax}_{k_i, p_t}\, p(\boldsymbol{k}_i, \boldsymbol{p}_t | \boldsymbol{l}_{it}) \tag{15.12}$$

with the a posteriori density of the unknown parameters,

$$p(\boldsymbol{k}_i, \boldsymbol{p}_t |\, l_{it}) \propto p(\boldsymbol{l}_{it} | \boldsymbol{k}_i, \boldsymbol{p}_t)\, p(\boldsymbol{k}_i)\, p(\boldsymbol{p}_t). \tag{15.13}$$

As we assume the observations to be normally distributed, cf. (15.11), taking the negative logarithm of (15.13), this is achieved by minimizing the weighted residuals or reprojection errors:

fusion of all observations to a complete block

$$\begin{aligned}\{\widehat{\boldsymbol{l}}_{it}, \widehat{\boldsymbol{k}}_i, \widehat{\boldsymbol{p}}_t\} = \text{argmin}_{l_{it}^*, k_i^*, p_t^*} \Bigg\{ & \sum_{(it)\in\mathcal{E}} (\boldsymbol{l}_{it} - \boldsymbol{l}_{it}^*)^\mathsf{T} \Sigma_{l_{it}l_{it}}^{-1} (\boldsymbol{l}_{it} - \boldsymbol{l}_{it}^*) \\ & + \sum_{i\in\mathcal{I}_0} (\boldsymbol{k}_{i0} - \boldsymbol{k}_i^*)^\mathsf{T} \Sigma_{k_{i0}k_{i0}}^{-1} (\boldsymbol{k}_{i0} - \boldsymbol{k}_i^*) \\ & + \sum_{t\in\mathcal{T}_0} (\boldsymbol{p}_{0t} - \boldsymbol{p}_t^*)^\mathsf{T} \Sigma_{p_{0t}p_{0t}}^{-1} (\boldsymbol{p}_{0t} - \boldsymbol{p}_t^*) \Bigg\}\end{aligned} \tag{15.14}$$

subject to the constraints (15.6) or (15.7). The weighting is done using the inverse covariance matrices. The estimation process allows us to derive the covariance matrix of estimated parameters, which is a lower bound on the true covariance matrix, namely the Cramer–Rao bound. Eq. (15.14) realizes the fusion of the different observations into a complete block.

This estimation can be interpreted as a Bayesian estimate for the parameters $(\boldsymbol{k}, \boldsymbol{t})$, with the priors resulting from the observed parameters and their covariance matrices. In addition we can use the residuals to perform a robust estimation procedure if necessary.

15.2.2 Examples for Block Adjustments

The described setup for geometrically fusing many images is very general and has a number of important instances; some of them are collected in Table 15.1. In addition, we give an example for the dimensions of the individual entities, namely the dimension D_I of the scene features, the dimension D_T of the transformation, and the dimension D_E of the observed features.

15.2.2.1 Bundle Adjustment

If we have images in general position and the scene has a general 3D structure, we arrive at the classical bundle adjustment. Here we have a perspective mapping of points from the 3D space to the image space ($D_I = 3$, $D_E = 2$). The unknown transformation parameters primarily refer to the 3D pose of each camera when taking the image. For perspective cameras, which are uncalibrated and straight line-preserving, we use a bundle adjustment

Table 15.1 Types of block adjustments. BA = bundle adjustment: observations are bundles of rays; MA = model block adjustment: observations are model points; PH = photogrammetry; CV = computer vision

	name	scene feature	transformation	observations	main application
1	BA	3D points	projective transformation/ 3D motion	image points	PH/CV
		$D_I = 3$	$D_T = 11/6$	$D_E = 2$	
2	3D MA	3D points	3D homography/ 3D similarity/ 3D motion	local 3D points	PH/CV
		$D_I = 3$	$D_T = 15/7/6$	$D_E = 3$	
3	2D MA	2D points	2D homography/ 2D similarity/ 2D motion	2D points	photography/ PH/CV/ robotics
		$D_I = 2$	$D_T = 8/4/3$	$D_E = 2$	

with $D_T = 11$, which we call *projective bundle adjustment*. It includes individual calibration parameters for each image. For spherical or calibrated perspective cameras we use a bundle adjustment with $D_T = 6$, which we call *Euclidean bundle adjustment*, or simply *bundle adjustment*. In its most general form it has been proposed by Schmid (1958); an excellent review is given by Triggs et al. (2000); open source software SBA (sparse bundle adjustment) is provided by Lourakis and Argyros (2009).

projective bundle adjustment

Euclidean bundle adjustment

The mathematical model has already been given for orienting one, two, or three images. It is the projective bundle adjustment, and is cited here for completeness. For each image point x_{it} observed in image t, we have, cf. (12.128), p. 497

$$\mathbb{E}(\underline{\boldsymbol{x}}'_{it}) = \boldsymbol{c}(\tilde{\mathsf{P}}_t \tilde{\mathbf{X}}_i)\,, \quad \mathbb{D}(\underline{\boldsymbol{x}}'_{it}) = \Sigma_{x'_{it}x'_{it}}\,, \quad (it) \in \mathcal{E}\,, \quad \text{with} \quad \boldsymbol{c}(\mathbf{x}) = \frac{\mathbf{x}_0}{x_h}\,. \tag{15.15}$$

Thus we assume the perspective camera model. It can be written in the classical form, cf. (12.35), p. 472,

$$\mathbb{E}(\underline{x}'_{it}) = \frac{\tilde{p}_{t,11}\tilde{X}_i + \tilde{p}_{t,12}Y_i + \tilde{p}_{t,13}Z_i + \tilde{p}_{t,14}}{\tilde{p}_{t,31}X_i + \tilde{p}_{t,32}Y_i + \tilde{p}_{t,33}Z_i + \tilde{p}_{t,34}} \tag{15.16}$$

$$\mathbb{E}(\underline{y}'_{it}) = \frac{\tilde{p}_{t,21}X_i + \tilde{p}_{t,22}Y_i + \tilde{p}_{t,23}Z_i + \tilde{p}_{t,24}}{\tilde{p}_{t,31}X_i + \tilde{p}_{t,32}Y_i + \tilde{p}_{t,33}Z_i + \tilde{p}_{t,34}}\,. \tag{15.17}$$

For calibrated cameras, we have the model for a Euclidean bundle adjustment with $D_T = 6$ parameters for each image. For each camera ray ${}^c\mathbf{x}'_{it}$ in image t, we have

$$\mathbb{E}({}^c\underline{\mathbf{x}}'_{it}) = \mathsf{N}(\tilde{R}_t[I_3 \mid -\tilde{\boldsymbol{Z}}_t]\,\tilde{\mathbf{X}}_i)\,, \quad \mathbb{D}({}^c\underline{\mathbf{x}}'_{it}) = \Sigma_{{}^c\mathbf{x}'_{it}\,{}^c\mathbf{x}'_{it}}\,, \quad (it) \in \mathcal{E}\,; \tag{15.18}$$

cf. (12.222), p. 520. Hence, due to the known interior parameters of the camera we use the spherical camera model. We will discuss this model in detail in Sect. 15.4, p. 674.

15.2.2.2 3D Model Block Adjustment

If we have observed 3D point clouds, e.g., using a laser ranger or using some stereo evaluation software, we arrive at the classical *spatial model block adjustment*, or just spatial model adjustment. Here we have a 3D motion ($D_T = 6$), a similarity ($D_T = 7$) or a homography ($D_T = 15$) as transformation of point features from the scene ($D_I = 3$) into the local point cloud ($D_E = 3$). In its general form, it has been proposed by Ackermann et al. (1970) for fusing photogrammetric 3D models derived from image pairs. It has been independently developed by Ikeuchi (cf. Oishi et al., 2005) for matching point clouds derived by laser scanning.

The mathematical model for fusing projective photogrammetric models leads to the *projective model block adjustment*, derived e.g., from image pairs of uncalibrated cameras.

spatial projective model block adjustment

With the $D_T = 15$ parameters per projective transformation, H_t it is given by

$$\mathbb{E}(\underline{\mathbf{X}}'_{it}) = \tilde{\lambda}_{it}\tilde{\mathsf{H}}_t\,\tilde{\mathbf{X}}_i\,, \quad \mathbb{D}(\underline{\mathbf{X}}'_{it}) = \Sigma_{\mathrm{X}'_{it}\mathrm{X}'_{it}}\,, \quad (it) \in \mathcal{E}\,. \tag{15.19}$$

It can be specialized for fusing photogrammetric models derived from calibrated cameras determines the $D_T = 7$ parameters of each similarity transformation $\mathcal{M}_t(R_t, \boldsymbol{T}_t, \lambda_t)$ with $D_T = 7$ parameters per model. Its mathematical model is given by

$$\mathbb{E}(\underline{\boldsymbol{X}}'_{it}) = \tilde{\lambda}_t\tilde{\mathsf{R}}_t\tilde{\boldsymbol{X}}_i + \tilde{\boldsymbol{T}}_t\,, \quad \mathbb{D}(\underline{\boldsymbol{X}}'_{it}) = \Sigma_{X'_{it}X'_{it}}\,, \quad (it) \in \mathcal{E}\,. \tag{15.20}$$

spatial Euclidean model block adjustment

We arrive at a *Euclidean model block adjustment*, which may include additional control information. If the scale between the models is known, as when fusing 3D point clouds, the similarity transformation is specialized to a 3D motion with $D_T = 6$ parameters, omitting the scale parameter λ_t in the mathematical model.

15.2.2.3 2D Model Block Adjustment

We arrive at the *planar model adjustment* if we have locally observed 2D point clouds, e.g., when fusing 2D images by an *image stitching* process or when neglecting the third dimension when using a levelled laser range finder, or when the rotation of the cameras used for a stereo evaluation are known. Here we have a 2D motion ($D_T = 3$), a similarity transformation ($D_T = 4$), or a homography ($D_T = 8$) of 2D points ($D_I = 2$) into local coordinate systems ($D_E = 2$).

The most general functional model assumes a planar homography per model.[1] At the same time it is the simplest model for stitching arbitrary images taken at a common viewpoint. Here the model is given, fully equivalent to (15.19), except that all entities refer to 2D.

The most specific functional model assumes a planar motion and is given by

$$\mathbb{E}(\underline{\boldsymbol{x}}'_{it}) = \tilde{\mathsf{R}}_t\tilde{\boldsymbol{x}}_i + \tilde{\boldsymbol{t}}_t\,, \quad \mathbb{D}(\underline{\boldsymbol{x}}'_{it}) = \Sigma_{x'_{it}x'_{it}}\,, \quad (it) \in \mathcal{E}\,. \tag{15.21}$$

An important situation arises when fusing 2D images with planar similarities, since then the parametrization can be made linear, cf. (6.14), p. 252,

$$\mathbb{E}\left(\begin{bmatrix}\underline{x}'_{it}\\ \underline{y}'_{it}\end{bmatrix}\right) = \begin{bmatrix}\tilde{a}_t & -\tilde{b}_t\\ \tilde{b}_t & \tilde{a}_t\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{y}_i\end{bmatrix} + \begin{bmatrix}\tilde{c}_i\\ \tilde{d}_i\end{bmatrix}\,, \quad \mathbb{D}(\underline{\boldsymbol{x}}'_{it}) = \Sigma_{x'_{it}x'_{it}}\,, \quad (it) \in \mathcal{E}\,. \tag{15.22}$$

This functional model can be used (1) for stitching images, which have been rectified for perspective distortion, using e.g., vanishing points, or (2) for fusing photogrammetric models from calibrated cameras, if they have been rotated such that the ${}^mZ$-axes of the models are parallel to the Z-axis of the scene coordinate system and only the horizontal coordinates are of concern. The rotation may be derived from the nadir or zenith point derived from vertical lines in a Manhattan scene.

Due to its relevance and simplicity, we use this functional model in the next section for analysing sparsity patterns of the normal equations, the gauge transformations of the coordinates, and the distribution of control points in block adjustment. The results qualitatively transfer to the other functional models for fusing images.

[1] Observe, the notion 'model' here is used in two different ways: (1) in its sense as the (simplified) *description* of an object, and (2) in its sense as a functional model, where it consists of a particular (e.g., algebraic) *structure* and a certain set of parameters.

15.3 Sparsity of Matrices, Free Adjustment and Theoretical Precision

15.3.1 The Mathematical Model of 2D Block Adjustment 651
15.3.2 The Optimization Function 654
15.3.3 Sparse Structure of Matrices 655
15.3.4 Free Block Adjustment 663
15.3.5 Theoretical Quality of Regular Strips and Blocks 670

The 2D model block adjustment is representative for all block adjustments and, in particular, also the bundle adjustment. This section addresses all steps of the process:

- the setup of the nonlinear model and its linearization,
- the setup of the normal equations, the analysis of their sparsity structure,
- the iteration solution sequence,
- the fixation of the gauge in case no control points are available,
- the evaluation of the estimated parameters, and
- the analysis of the quality of certain image configurations w.r.t. the distribution of control points.

We assume that the block adjustment is used for stitching of images rectified such that their mutual transformation is a planar similarity. Where appropriate, we provide notes on the situation of bundle adjustment.

15.3.1 The Mathematical Model of 2D Block Adjustment

We will first discuss the mathematical model of the 2D model block adjustment, also called model adjustment, in more detail. This establishes a direct connection to Chap. 4 on parameter estimation, illustrates the role of all entities w.r.t. the stochastical model, and is the basis for the analysis of the sparseness patterns in the normal equations.

Aiming at a maximum likelihood estimation, we represent the *functional model* in the form of a Gauss–Markov model. We start by identifying the different entities as observed or unknown parameters.

From now on we *identify the model coordinates with image coordinates* in the context of stitching many images into an image mosaic, assuming similarity transformations are sufficient.

The scene coordinates $\boldsymbol{x}_i = [x_i, y_i]^\mathsf{T}, i \in \mathcal{I}$, are observed in image t, leading to the observed image coordinates $\boldsymbol{x}_{it} = [x_{it}, y_{it}]^\mathsf{T}, (it) \in \mathcal{E}$. Then the model (15.22), p. 650 can be written in compact form,

$$\mathbb{E}(\underline{\boldsymbol{x}}_{it}) = \boldsymbol{r}_t + \boldsymbol{Z}(\boldsymbol{s}_t)\boldsymbol{x}_i\,, \quad (it) \in \mathcal{E}\,, \tag{15.23}$$

using the parameters

$$\boldsymbol{r}_t = \begin{bmatrix} a_t \\ b_t \end{bmatrix}\,, \qquad \boldsymbol{s}_t = \begin{bmatrix} c_t \\ d_t \end{bmatrix} \tag{15.24}$$

and the 2×2 matrix function[2]

$$\boldsymbol{Z}(\boldsymbol{s}) = \begin{bmatrix} s_1 & -s_2 \\ s_2 & s_1 \end{bmatrix}\,. \tag{15.25}$$

[2] The motivation for this modelling is the isomorphy between complex numbers $c = a + b\sqrt{-1}$ and 2×2 matrices $\boldsymbol{Z}(\boldsymbol{c})$ (the vector $\boldsymbol{c} = [a, b]^\mathsf{T}$ contains the real and the imaginary part of c), with the commutative matrix product and the transposition including the complex conjugation. Interpreting the vectors as complex numbers, the basic similarity model can be written as a linear function $\mathbb{E}(x_{it}) = r_t + s_t x_i$, where all variables are complex numbers.

The functional model for the observed control points is simple, namely

$$\mathbb{E}(\boldsymbol{x}_{i0}) = \underset{2\times 1}{\boldsymbol{x}_i}\,, \quad i \in \mathcal{I}_0\,, \tag{15.26}$$

not depending on transformation parameters.

observation vector including observed control points

We now collect the observed image and scene coordinates $\boldsymbol{x}_{it}$ and $\boldsymbol{x}_{i0}$ in the observation vector,

$$\boldsymbol{l} = \begin{bmatrix} [\boldsymbol{l}_{it}] \\ [\boldsymbol{l}_{i0}] \end{bmatrix} := \begin{bmatrix} [\boldsymbol{x}_{it}] \\ [\boldsymbol{x}_{i0}] \end{bmatrix}\,. \tag{15.27}$$

On the left-hand side we use the notation from adjustment theory and on the right-hand side the notation of the current application, namely planar model block adjustment. Similarly, the unknown parameters

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{k} \\ \boldsymbol{p} \end{bmatrix} \tag{15.28}$$

are partitioned into the vectors $\boldsymbol{k}$ and $\boldsymbol{p}$ for the unknown scene coordinates and the unknown transformation parameters,

$$\boldsymbol{k} = [\boldsymbol{k}_i] := [\boldsymbol{x}_i] \quad \text{and} \quad \boldsymbol{p} = [\boldsymbol{p}_t]\,, \tag{15.29}$$

transformation parameters containing shift $\boldsymbol{r}_t$ and scaled rotation $\boldsymbol{s}_t$

with the transformation parameters

$$\boldsymbol{p}_t := \begin{bmatrix} \boldsymbol{r}_t \\ \boldsymbol{s}_t \end{bmatrix} = \begin{bmatrix} a_t \\ b_t \\ c_t \\ d_t \end{bmatrix} \tag{15.30}$$

related to each image t.

The *stochastical model* describes the stochastical properties of the observations. In our applications, we usually can assume that the observed image points $\boldsymbol{l}_{it}$ and the observed scene points $\boldsymbol{l}_{i0}$ deviate by additive observational errors from the true values,

$$\boldsymbol{l}_{it} = \tilde{\boldsymbol{l}}_{it} + \boldsymbol{e}_{it}\,, \qquad (it) \in \mathcal{E} \cup \mathcal{E}_0\,, \tag{15.31}$$

where we use the index set of observations of the control points,

$$\mathcal{E}_0 = \mathcal{I}_0 \times \{0\}\,. \tag{15.32}$$

The observational errors in most cases can be assumed to be stochastically independent. In the ideal case we can assume the deviations $\underline{\boldsymbol{e}}_{it}$ from the true values to be normally distributed with mean zero and some individual covariance matrix. The observational errors thus are modelled by

$$\underline{\boldsymbol{e}}_{it} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{e_{it}e_{it}})\,, \qquad (it) \in \mathcal{E}_0 \cup \mathcal{E}\,, \tag{15.33}$$

and all covariances $\Sigma_{e_{it}e_{i't'}} = \boldsymbol{0}$ for $(it) \neq (i't')$ vanish.

stochastical model for planar model block adjustment

As the true values have zero variance, we have

$$\Sigma_{l_{it}l_{it}} = \Sigma_{e_{it}e_{it}}\,, \qquad (it) \in \mathcal{E}_0 \cup \mathcal{E}\,. \tag{15.34}$$

The covariance matrix Σ_{ll} includes the uncertainty of the observed image and the observed scene coordinates.

The complete mathematical model of block adjustment therefore can be written as:

$$\underline{\boldsymbol{l}}_{it} \sim \mathcal{N}(\mathbb{E}(\boldsymbol{l}_{it}), \Sigma_{l_{it}l_{it}})\,, \qquad (it) \in \mathcal{E}_0 \cup \mathcal{E}\,, \qquad (i \in \mathcal{I}\,, \quad t \in \{0, \mathcal{T}\})\,, \tag{15.35}$$

where the indices (it) indicate which scene point $\boldsymbol{p}_i$ is observed in which image t, $t = 0$ indicating that the observed point refers to a control point. The mean values of the

observations $\mathbb{E}(\boldsymbol{l}_{it})$ depend on the unknown values of the transformation parameters,

$$\mathbb{E}(\underline{\boldsymbol{l}}_{it}) = \boldsymbol{r}_t + Z(\boldsymbol{s}_t)\boldsymbol{k}_i\,, \quad t \in \mathcal{T} \tag{15.36}$$

$$\mathbb{E}(\underline{\boldsymbol{l}}_{i0}) = \boldsymbol{k}_i\,. \tag{15.37}$$

Generally, the numbers N and U of observations and unknown parameters depend on

- the dimension D_E of the observation vectors $\boldsymbol{l}_{it}$ in the image, which we for simplicity assume to be the same for each observed image feature; in our case we have two-dimensional coordinates of the image points: $D_E = 2$;
- the dimension D_I of the parameter vectors $\boldsymbol{k}_i$ the scene, again generally assuming this to be the same for all scene features; in our case we have two-dimensional coordinates of the scene points: $D_I = 2$;
- the total number $E = \sum_{t=1}^{T} I_t$ of observed image features;
- the number I_0 of observed scene features;
- the number T_0 of observed parameter vectors $\boldsymbol{p}_t$ for transformations; this does not apply in our special case; therefore we have $T_0 = 0$; in bundle adjustment, such observations could be possible using the global positioning system (GPS) for measuring the position of the projection centres and inertial measurement units (IMUs) for measuring the rotation angles of the camera;
- the number I of unknown scene features, here all scene points; this *includes* the unknown features, which are observed;
- the number D_T of the transformation parameters $\boldsymbol{p}_t$ per image; in our case $D_T = 4$.

The total number N of observations and the number U of unknown parameters then are

$$N = D_E \sum_{t=1}^{T} I_t + D_I\, I_0 + D_T\, T_0\,, \qquad U = D_I\, I + D_T\, T\,. \tag{15.38}$$

The redundancy of the problem is $R = N - U$. A necessary, though not sufficient, condition for the problem to be solvable is $R \geq 0$. With the assumed specifications, we have the number of observations and of unknown parameters

$$N = 2\left(I_0 + \sum_{t=1}^{t} I_t\right)\,, \qquad U = 2\,I + 4\,T\,. \tag{15.39}$$

Example 15.3.49: ***An example image set for stitching.*** We will illustrate the sparse structures using an example, see Fig. 15.2. Given are six images, arranged in two rows and three columns, which

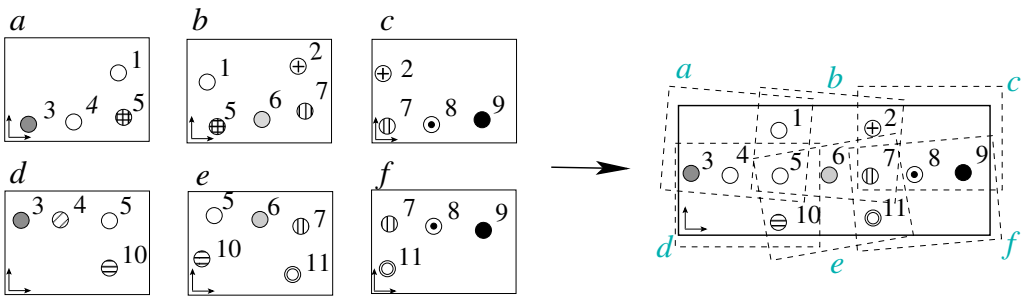


Fig. 15.2 Example of stitching six images. **Top:** key points with their scene point number and coordinated in six individual local coordinate systems. **Below:** key points in a common coordinate system (solid rectangle). Points 3 and 9 are assumed to be also observed in the scene coordinate system

overlap along their borders. A key point detector is assumed to have identified some points in each image and coordinated them in the local coordinate system; only a small number of key points is assumed for clarity purposes. The correspondence between these key points has been established. We are interested

in generating an image composite with the six images overlapping at the borders, such that image points referring to the same scene point have the same scene coordinates. Some of the scene points are assumed to be observed in the scene coordinate system.

The total number of observed image points is $B = 4 + 5 + 4 + 4 + 5 + 4 = 26$; thus, the number of observed coordinates is $N = D_E\, E + D_I\, I_0 = 2 \times 26 + 2 \times 2 = 56$. The number of unknown scene points is $I = 11$; therefore, the number of unknown scene coordinates is $U_K = D_I\, I = 2 \times 11 = 22$. Together with the $U_T = D_T\, T = 4 \times 6 = 24$ unknown transformation parameters we in total have $U = U_T + U_K = 24 + 22 = 46$ unknown parameters. The redundancy of the problem thus is $R = N - U = 56 - 46 = 10 > 0$, indicating the problem to be generally solvable. ◇

15.3.2 The Optimization Function

The optimization function for the maximum likelihood estimation now reads as

$$p(\{\boldsymbol{k}_i^*\}, \{\boldsymbol{p}_t^*\} | \{\boldsymbol{l}_{it}\}) \propto \prod_{(it)\in\mathcal{E}} p(\boldsymbol{k}_i^*, \boldsymbol{p}_t^* | \boldsymbol{l}_{it})\,. \tag{15.40}$$

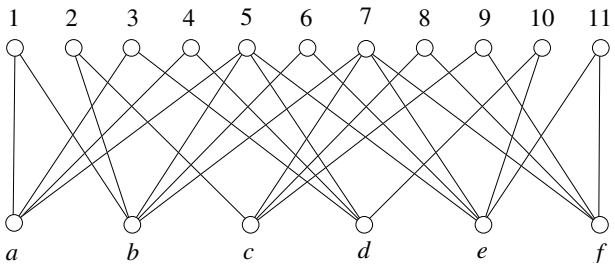


Fig. 15.3 A factor graph for the block adjustment example. The scene points have indices $\mathcal{I} = \{1, ..., 11\}$, the images have indices $\mathcal{T} = \{a, ..., f\}$. Each edge $(it) \in \mathcal{E}$ in the graph relates to an observation $\boldsymbol{l}_{it}$ and represents an observation equation, causing the introduction of a factor ϕ_{it} in the optimization function for the maximum likelihood estimation. Each image $t \in \mathcal{T}$ contains scene points with index $i \in \mathcal{I}_t$, the set of all neighbours of node t. Each scene point $i \in \mathcal{I}$ is observed in images $t \in \mathcal{T}_i$, the set of all neighbours of node i. The size of the index sets $\mathcal{I}_t$ and $\mathcal{T}_i$ are usually small. Not shown is the node for the observed control points.

It is a product related to the observations, where each factor is the exponential of the Mahalanobis distance of the observed value from its predicted value weighted with the inverse covariance matrix, e.g.,

$$\phi_{it} := p(\boldsymbol{k}_i^*, \boldsymbol{p}_t^* | \boldsymbol{l}_{it}) \propto \exp\left(-\frac{1}{2}\,(\boldsymbol{l}_{it} - \boldsymbol{l}_{it}^*(\boldsymbol{k}_i^*, \boldsymbol{p}_k^*))^\mathsf{T}\, \boldsymbol{\Sigma}_{l_{it}l_{it}}^{-1}\, (\boldsymbol{l}_{it} - \boldsymbol{l}_{it}^*(\boldsymbol{k}_i^*, \boldsymbol{p}_k^*))\right)\,. \tag{15.41}$$

Each factor depends on an image feature $\boldsymbol{f}_{it}$, thus one observational group $\widehat{\boldsymbol{l}}_{it}$, and establishes a relation between the parameters $\boldsymbol{k}_i$ of the unknown scene points and the parameters of the unknown transformations $\boldsymbol{p}_t$ only. No other relations exist. As mentioned above, this can be easily visualized using the bipartite adjacency graph of the unknown parameters. Each edge corresponds to a factor, which is why this graph, interpreted as a stochastical model, also is called *factor graph*. We will use it to illustrate the sparsity structure of the Jacobians and the normal equation matrices occurring within the estimation procedure.

factor graph

The factor graph carries all information about the structure of the estimation problem and shows the estimation problem to be a specific instance of a Markov random field (cf. Kschischang et al., 2001; Dellaert and Kaess, 2006; Förstner, 2013). The factor graph for the example is shown in Fig. 15.3. The observations of the control points are subsumed in one of the factors referring to that point, e.g., the factor for observation $\widehat{\boldsymbol{l}}_{3d}$ including the

observation $\boldsymbol{l}_{3,0}$ of the control point 3,

$$\phi_{3,d} = p(\boldsymbol{k}_3^*, \boldsymbol{p}_d^* | \boldsymbol{l}_{3,d})\; p(\boldsymbol{k}_3^* | \boldsymbol{l}_{3,0})\,, \tag{15.42}$$

thus augmenting the expression in (15.41).

15.3.3 Sparse Structure of Matrices

In this section we discuss the sparsity of the observation and normal equation matrices. We first do this for the general setup and then specialize for the planar model block adjustment.

15.3.3.1 Observation and Normal Equations of the Linearized Model

We start from the linearized model. For this we adopt the notation of the estimation theory in Sect. 4.2.6, p. 94. The observations are collected in the vector $\boldsymbol{l}$. The unknown parameters are collected in the vector $\boldsymbol{x} = [\boldsymbol{k}^\mathsf{T}, \boldsymbol{p}^\mathsf{T}]^\mathsf{T}$. The general model,

$$\boldsymbol{l} + \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{l}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) = \boldsymbol{f}(\widehat{\boldsymbol{k}}, \widehat{\boldsymbol{p}})\,, \tag{15.43}$$

is linearized using approximate values $\widehat{\boldsymbol{x}}^a = [\widehat{\boldsymbol{k}}^{a\mathsf{T}}, \widehat{\boldsymbol{p}}^{a\mathsf{T}}]^\mathsf{T}$ and $\widehat{\boldsymbol{l}}^a = \boldsymbol{f}(\widehat{\boldsymbol{x}}^a)$ for the fitted parameters $\widehat{\boldsymbol{x}}$, namely the fitted scene coordinates $\widehat{\boldsymbol{k}}$ and the fitted transformations parameters $\widehat{\boldsymbol{p}}$, and for the linearized observations $\Delta\boldsymbol{l}$. With the corrections

$$\Delta\boldsymbol{l} = \boldsymbol{l} - \widehat{\boldsymbol{l}}^a\,, \quad \widehat{\Delta\boldsymbol{x}} = \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}^a\,, \quad \widehat{\Delta\boldsymbol{k}} = \widehat{\boldsymbol{k}} - \widehat{\boldsymbol{k}}^a\,, \quad \widehat{\Delta\boldsymbol{p}} = \widehat{\boldsymbol{p}} - \widehat{\boldsymbol{p}}^a\,, \tag{15.44}$$

this leads to the linearized functional model

$$\Delta\boldsymbol{l} + \widehat{\boldsymbol{v}} = A\widehat{\Delta\boldsymbol{x}} = C\widehat{\Delta\boldsymbol{k}} + D\widehat{\Delta\boldsymbol{p}}\,, \tag{15.45}$$

with the partitioning of the Jacobian,

$$A = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{x=\widehat{x}^a} = [C\ ,\ D] = \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{k}}\ ,\ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\right]_{x=\widehat{x}^a}\,. \tag{15.46}$$

The Jacobians have to be evaluated at the current approximate values within the iteration process.

With the weight matrix $W_{ll} = \Sigma_{ll}^{-1}$, the normal equations then have the following form:

$$N\widehat{\Delta\boldsymbol{x}} - \boldsymbol{h} = \boldsymbol{0} \quad \text{or} \quad A^\mathsf{T} W_{ll} A\widehat{\Delta\boldsymbol{x}} - A^\mathsf{T} W_{ll}\Delta\boldsymbol{l} = \boldsymbol{0}\,. \tag{15.47}$$

Explicitly, this reads as

$$\begin{bmatrix} N_{kk} & N_{kp} \\ N_{pk} & N_{pp} \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{k}} \\ \widehat{\Delta\boldsymbol{p}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{h}_k \\ \boldsymbol{h}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{15.48}$$

or

$$\begin{bmatrix} C^\mathsf{T} W_{ll} C & C^\mathsf{T} W_{ll} D \\ D^\mathsf{T} W_{ll} C & D^\mathsf{T} W_{ll} D \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{k}} \\ \widehat{\Delta\boldsymbol{p}} \end{bmatrix} - \begin{bmatrix} C^\mathsf{T} W_{ll}\Delta\boldsymbol{l} \\ D^\mathsf{T} W_{ll}\Delta\boldsymbol{l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}\,. \tag{15.49}$$

Solving (15.47) for the correction $\widehat{\Delta\boldsymbol{x}}$ allows us to obtain improved approximate values in an iterative manner.

We now examine the structure of the Jacobian A and the normal equation matrix N.

The Jacobians. The functional model explicitly shows that each observed image point $\mathcal{f}_{it}$ only depends on the transformation parameters of the image t it is observed in and on the coordinates of the scene point $\mathcal{F}_i$. Moreover, in most practical cases the observed image points are stochastically independent, which preserves the sparseness pattern. Therefore, the Jacobian $\mathsf{A} = [\mathsf{C}, \mathsf{D}]$ and the weight or precision matrix W_{ll} are sparse.

We now determine the design matrix, specifically the Jacobians w.r.t. the parameter sets. Using a multiplicative correction of the scale-rotation part $\boldsymbol{s}_t$ of the transformation, we obtain the observation equations

$$\boldsymbol{l}_{it} + \widehat{\boldsymbol{v}}_{it} = (\widehat{\boldsymbol{r}}_t^a + \widehat{\Delta\boldsymbol{r}}_t) + \mathsf{Z}(\widehat{\Delta\boldsymbol{s}}_t)\mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)(\widehat{\boldsymbol{k}}_i^a + \widehat{\Delta\boldsymbol{k}}_i) \tag{15.50}$$

(cf. (15.23), p. 651), with the unknown corrections $\widehat{\Delta\boldsymbol{r}}_t$ and $\widehat{\Delta\boldsymbol{s}}_t$ for the translations and scaled rotation and omitting the hats $\widehat{\ }$ on the approximate values to simplify notation.

Using the approximate transformation

$$\widehat{\boldsymbol{l}}_{it}^a = \widehat{\boldsymbol{r}}_t^a + \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)\widehat{\boldsymbol{k}}_i^a \tag{15.51}$$

and image coordinates scaled and rotated into the tth image system

$${}^t\widehat{\boldsymbol{k}}_i^a = \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)\widehat{\boldsymbol{k}}_i^a\,, \tag{15.52}$$

we finally have the linearized model

$$\begin{aligned}
\widehat{\Delta\boldsymbol{l}}_{it} + \widehat{\boldsymbol{v}}_{it} &= (\boldsymbol{l}_{it} - \widehat{\boldsymbol{l}}_{it}^a) + \widehat{\boldsymbol{v}}_{it} && (15.53)\\
&= \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)\widehat{\Delta\boldsymbol{k}}_i + \widehat{\Delta\boldsymbol{r}}_i + \mathsf{Z}({}^t\widehat{\boldsymbol{k}}_i^a)\widehat{\Delta\boldsymbol{s}}_t && (15.54)\\
&= \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)\widehat{\Delta\boldsymbol{k}}_i + [\mathsf{I}_2 \mid \mathsf{Z}({}^t\widehat{\boldsymbol{k}}_i^a)]\widehat{\Delta\boldsymbol{p}}_t\,, && (15.55)
\end{aligned}$$

where for arbitrary 2-vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ we have $\mathsf{Z}(\boldsymbol{a})\boldsymbol{b} = \mathsf{Z}(\boldsymbol{b})\boldsymbol{a}$. The Jacobian C consists of 2×2 submatrices C_{it},

$$\mathsf{C}_{it} = \left.\frac{\partial \boldsymbol{f}_{it}}{\partial \boldsymbol{k}_i}\right|_{x=\widehat{x}^a} = \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a) = \begin{bmatrix} \widehat{c}_t & -\widehat{d}_t \\ \widehat{d}_t & \widehat{c}_t \end{bmatrix}^a\,, \tag{15.56}$$

for index pairs $(it) \in \mathcal{E}$ to be evaluated at the approximate values $(\widehat{c}_t, \widehat{d}_t)^a$. The Jacobians for the observed control points are unit matrices I_2. The Jacobian D has dimension $N \times U_T = 56 \times 24$ and consists of 2×4 submatrices D_{it},

$$\mathsf{D}_{it} = \left.\frac{\partial \boldsymbol{f}_{it}}{\partial \boldsymbol{p}_t}\right|_{x=\widehat{x}^a} = [\mathsf{I}_2 \mid \mathsf{Z}({}^t\widehat{\boldsymbol{k}}_i^a)] = \begin{bmatrix} 1 & 0 & {}^t\widehat{x}_i & -{}^t\widehat{y}_i \\ 0 & 1 & {}^t\widehat{y}_i & {}^t\widehat{x}_i \end{bmatrix}^a \quad \text{with} \quad \begin{bmatrix} {}^t\widehat{x}_i^a \\ {}^t\widehat{y}_i^a \end{bmatrix} = {}^t\widehat{\boldsymbol{k}}_i^a\,. \tag{15.57}$$

Again only the submatrices of D for index pairs $(it) \in \mathcal{E}$ are nonzero. The coordinates $[{}^t x_i^{a\mathsf{T}}, {}^t y_i a\mathsf{T}]^\mathsf{T}$ are taken from (15.52). The precision matrix W_{ll} is a block diagonal matrix with $E + I_0$ entries, one for each observational group $\boldsymbol{l}_{it}$ or $\boldsymbol{l}_{i0}$,

$$\mathsf{W}_{ll} = \mathrm{Diag}(\{\mathsf{W}_{l_{it}l_{it}}\})\,, \quad (it) \in \mathcal{E} \cup \mathcal{E}_0\,. \tag{15.58}$$

Fig. 15.4 shows the sparse structure of the design matrix A for the example data set.

In our case the Jacobian A has dimension $N \times U_K = 56 \times 22$. The upper left 10×10 submatrix of C explicitly is

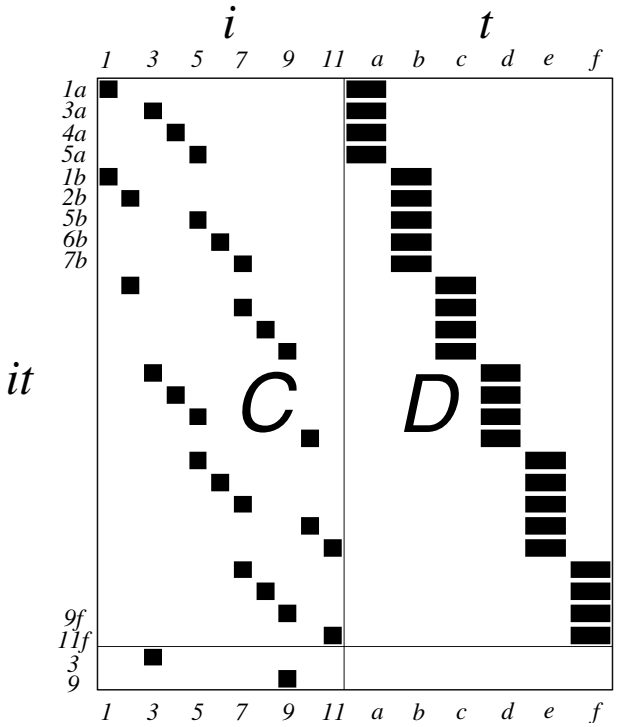


Fig. 15.4 Jacobian $A = [C \mid D]$ for the parameters of the example data set in Fig. 15.2. In C, only those 2×2 matrices are nonzero where the image point x_{it} of scene point X_i is measured and – in the two last rows – where a scene point X_i is measured, see the factor graph in Fig. 15.3, p. 654. In D, only those 2×4 submatrices are nonzero where a scene point X_i is measured in image t. The horizontal white stripes between the blocks in D are only to increase visibility; they do not indicate zeros

$$C = \begin{bmatrix} \begin{matrix} c_a & -d_a \\ d_a & c_a \end{matrix} & & & & & \dots \\ & & \begin{matrix} c_a & -d_a \\ d_a & c_a \end{matrix} & & & \dots \\ & & & \begin{matrix} c_a & -d_a \\ d_a & c_a \end{matrix} & & \dots \\ & & & & \begin{matrix} c_a & -d_a \\ d_a & c_a \end{matrix} & \dots \\ \begin{matrix} c_b & -d_b \\ d_b & c_b \end{matrix} & & & & & \dots \\ \dots \quad \dots & \begin{matrix} c_b & -d_b \end{matrix} & \dots \quad \dots & \dots \quad \dots & \dots \quad \dots & \dots \end{bmatrix}^a , \tag{15.59}$$

to be evaluated at the approximate values. The upper left 10×10 submatrix of D is

$$D = \begin{bmatrix} \begin{matrix} 1 & 0 & {}^a x_1 & -{}^a y_1 \\ 0 & 1 & {}^a y_1 & {}^a x_1 \end{matrix} & & \dots \\ \begin{matrix} 1 & 0 & {}^a x_3 & -{}^a y_3 \\ 0 & 1 & {}^a y_3 & {}^a x_3 \end{matrix} & & \dots \\ \begin{matrix} 1 & 0 & {}^a x_4 & -{}^a y_4 \\ 0 & 1 & {}^a y_4 & {}^a x_4 \end{matrix} & & \dots \\ \begin{matrix} 1 & 0 & {}^a x_5 & -{}^a y_5 \\ 0 & 1 & {}^a y_5 & {}^a x_5 \end{matrix} & & \dots \\ & \begin{matrix} 1 & 0 & {}^b x_1 & -{}^b y_1 \\ 0 & 1 & {}^b y_1 & {}^b x_1 \end{matrix} & \dots \\ \dots \quad \dots \quad \dots \quad \dots & \begin{matrix} 1 & 0 & {}^b x_2 & -{}^b y_2 \end{matrix} & \dots \quad \dots \quad \dots \end{bmatrix}^a , \tag{15.60}$$

to be evaluated at the approximate values. Obviously the design matrix A is sparse: Only five elements in each row are not zero, independently of the number of parameters.

sparsity of design matrix

The Normal Equations and the Effect of Conditioning. Due to the sparsity of the design matrix and the simple rule for the sparsity pattern, the normal equation matrix can be expressed explicitly. We first analyse the general sparsity pattern and then give explicit expressions for the components of the normal equation matrix, especially for the case of isotropic uncertainty of the observed points, and demonstrate the effect of conditioning. We analyse the three components N_{kk}, N_{pp}, and N_{kp} separately.

Both submatrices N_{kk} and N_{pp} are block diagonal, as each row of C and D only contains a single submatrix C_{it} and D_{it}, respectively. The submatrix N_{kk} is given by

$$\mathsf{N}_{kk} = \mathrm{Diag}\left(\{\mathsf{N}_{k_i k_i}\}\right), \qquad \text{with} \qquad \mathsf{N}_{k_i k_i} = \sum_{t \in \mathcal{T}_i} \mathsf{C}_{it}^{\mathsf{T}} \mathsf{W}_{l_{it} l_{it}} \mathsf{C}_{it}, \tag{15.61}$$

where the sum is taken over the set $\mathcal{T}_i$ of all images where point i is observed, including the set $t = 0$, which is the reference set.

Similarly, the submatrix N_{pp} is given by

$$\mathsf{N}_{pp} = \mathrm{Diag}\left(\{\mathsf{N}_{p_t p_t}\}\right), \qquad \text{with} \qquad \mathsf{N}_{p_t p_t} = \sum_{i \in \mathcal{I}_t} \mathsf{D}_{it}^{\mathsf{T}} \mathsf{W}_{l_{it} l_{it}} \mathsf{D}_{it}, \tag{15.62}$$

where the sum is taken over the set $\mathcal{I}_t$ of all points in image t.

The off-diagonal matrix N_{kp} has nonzero submatrices $\mathsf{N}_{k_i p_t}$ where point i is observed in image t,

$$\mathsf{N}_{kp} = [\mathsf{N}_{k_i p_t}], \tag{15.63}$$

with

$$\mathsf{N}_{k_i p_t} = \mathsf{C}_{it}^{\mathsf{T}} \mathsf{W}_{l_{it} l_{it}} \mathsf{D}_{it} \quad \text{for } \{i,t\} \in \mathcal{E}, \quad \text{and } \mathsf{N}_{k_i p_t} = \mathsf{0} \text{ else.} \tag{15.64}$$

The right-hand sides are given explicitly as

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_k \\ \boldsymbol{h}_p \end{bmatrix} = \begin{bmatrix} \sum_{t \in \mathcal{T}_i} \mathsf{C}_{it}^{\mathsf{T}} \mathsf{W}_{l_{it} l_{it}} \Delta \boldsymbol{l}_{it} \\ \sum_{i \in \mathcal{I}_t} \mathsf{D}_{it}^{\mathsf{T}} \mathsf{W}_{l_{it} l_{it}} \Delta \boldsymbol{l}_{it} \end{bmatrix}. \tag{15.65}$$

sparsity of normal equation matrix

Fig. 15.5 shows the sparse structure of the normal equation matrix. The two diagonal

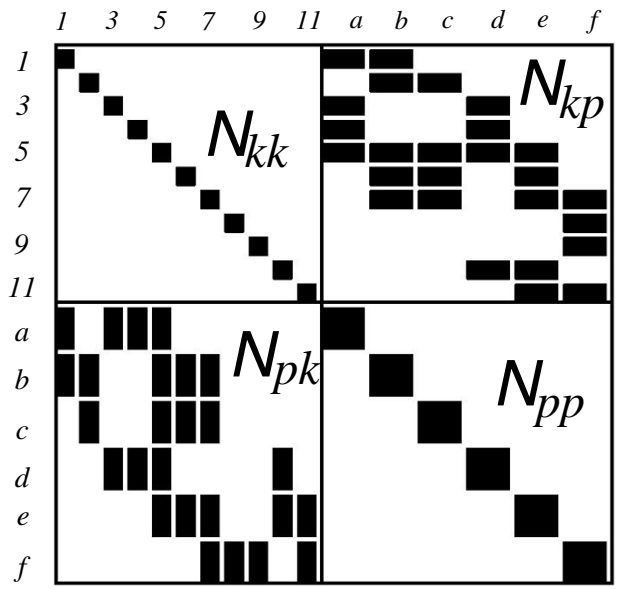


Fig. 15.5 Normal equation matrix for the example data set. The submatrices N_{kk} and N_{pp} are block diagonal, the submatrix N_{kp} reflects the incidence of points i and images t. The white lines between the blocks are introduced for separating the blocks; they are not zeros

submatrices N_{kk} and N_{pp} are block matrices with 2×2 and 4×4 matrices on the diagonals. The off-diagonal submatrix N_{kp} has nonzero 2×4-blocks $\mathsf{N}_{k_i p_t}$ where a point i is observed in image t. In many practical cases, not all points are observed in all images; therefore,

in these cases N_{kp} is sparse. The indices of the nonzero blocks in the off-diagonal matrix N_{kp} correspond to the edges in the factor graph in Fig. 15.6, p. 661. We give now explicit expressions for the three matrices $\mathsf{N}_{k_ik_i}$, $\mathsf{N}_{p_tp_t}$ and $\mathsf{N}_{k_ip_t}$.

The diagonals of the block matrix for the points are given by

$$\mathsf{N}_{k_ik_i} = \sum_{t \in \mathcal{T}_i} \mathsf{Z}^\mathsf{T}(\widehat{\boldsymbol{s}}_t^a)^\mathsf{T} \mathsf{W}_{l_{it}l_{it}} \mathsf{Z}(\widehat{\boldsymbol{s}}_t^a)\,, \tag{15.66}$$

which for isotropic uncertainty of the image points, i.e., $\mathsf{W}_{l_{it}l_{it}} = w_{it}\mathsf{I}_2$, specializes to the diagonal matrix

$$\mathsf{N}_{k_ik_i} = \left(\sum_{t \in \mathcal{T}_i} \lambda_t^2 w_{it}\right)^a \mathsf{I}_2\,, \tag{15.67}$$

with the squared scales $\lambda_t^2 = |\boldsymbol{s}_t|^2 = c_t^2 + d_t^2$, to be evaluated at the approximate values, and the weights $w_{it} = 1/\sigma_{it}^2$.

Analogously, for the diagonal matrices $\mathsf{N}_{p_tp_t}$ for the transformation parameters, we have

$$\mathsf{N}_{p_tp_t} = \sum_{i \in \mathcal{I}_t} \begin{bmatrix} \mathsf{W}_{l_{it}l_{it}} & \mathsf{W}_{l_{it}l_{it}} \mathsf{Z}({}^t\widehat{\boldsymbol{x}}_i) \\ \mathsf{Z}^\mathsf{T}({}^t\widehat{\boldsymbol{x}}_i) \mathsf{W}_{l_{it}l_{it}} & \mathsf{Z}^\mathsf{T}({}^t\widehat{\boldsymbol{x}}_i) \mathsf{W}_{l_{it}l_{it}} \mathsf{Z}({}^t\widehat{\boldsymbol{x}}_i) \end{bmatrix}^a . \tag{15.68}$$

If we assume isotropic uncertainty, this yields

$$\mathsf{N}_{p_tp_t} = \sum_{i \in \mathcal{I}_t} \begin{bmatrix} w_{it} & 0 & w_{it}\,{}^t x_i & -w_{it}\,{}^t y_i \\ 0 & w_{it} & w_{it}\,{}^t y_i & w_{it}\,{}^t x_i \\ w_{it}\,{}^t x_i & w_{it}\,{}^t y_i & w_{it}|{}^t\boldsymbol{x}_i|^2 & 0 \\ -w_{it}\,{}^t y_i & w_{it}\,{}^t x_i & 0 & w_{it}|{}^t\boldsymbol{x}_i|^2 \end{bmatrix}^a , \tag{15.69}$$

to be evaluated at the approximate values. Again, the coordinates $[{}^t\widehat{x}_i^a; {}^t\widehat{y}_i^a]$ are taken from (15.52). The matrices in the off-diagonal block matrix N_{kp} explicitly are

$$\mathsf{N}_{k_ip_t} = \mathsf{Z}^\mathsf{T}(\widehat{\boldsymbol{s}}_t^a) \mathsf{W}_{l_{it}l_{it}} [\mathsf{I}_2 \mid \mathsf{Z}({}^t\widehat{\boldsymbol{k}}_i^a)]\,, \tag{15.70}$$

which for $\mathsf{W}_{l_{it}l_{it}} = w_{it}\mathsf{I}_2$ reads

$$\mathsf{N}_{k_ip_t} = w_{it} \begin{bmatrix} c_t & d_t & {}^t x_i & -{}^t y_i \\ -d_t & c_t & {}^t y_i & {}^t x_i \end{bmatrix}^a , \tag{15.71}$$

to be evaluated at the approximate values.

We now investigate the effect of conditioning, namely centring and scaling, of the image coordinates on the condition number of the submatrices $\mathsf{N}_{p_tp_t}$. We use the weighted centroid μ_t of the image points within an image t as the image coordinate system and scale the coordinates by the quadratic mean σ_t of the centred coordinates: *effect of conditioning*

$$\begin{bmatrix} \overline{\boldsymbol{x}}_{it} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathsf{I}_2 & -\boldsymbol{\mu}_t \\ 0 & \sigma_t \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{it} \\ 1 \end{bmatrix}, \tag{15.72}$$

with

$$\boldsymbol{\mu}_t = \frac{\sum_{i \in \mathcal{I}_t} w_{it} \boldsymbol{x}_{it}}{\sum_{i \in \mathcal{I}_t} w_{it}} \qquad \sigma_t^2 = \frac{\sum_{i \in \mathcal{I}_t} w_{it} |\boldsymbol{x}_{it} - \boldsymbol{\mu}_t|^2}{\sum_{i \in \mathcal{I}_t} w_{it}} \tag{15.73}$$

to be evaluated at the approximate values. Then the matrix

$$\mathsf{N}_{p_tp_t} = \sum_{i \in \mathcal{I}_t} \begin{bmatrix} w_{it}\,\mathsf{I}_2 & \mathsf{0} \\ \mathsf{0} & w_{it}\,|{}^t\widehat{\overline{\boldsymbol{x}}}_i^a|^2\,\mathsf{I}_2 \end{bmatrix} = \left(\sum_{i \in \mathcal{I}_t} w_{it}\right) \mathsf{I}_4 \tag{15.74}$$

becomes a multiple of the unit matrix, having condition number $\kappa_t = \lambda_1/\lambda_2 = 1$. This can be compared to the condition number of the matrix in (15.69), which for centred data is

$\kappa_t = \sigma_t^4$ and for scaled data is $\kappa_t = (1+\mu_t)^2/(1-\mu_t)^2$. Observe that after conditioning, the two matrices $\mathbf{N}_{k_i k_i}$ and $\mathbf{N}_{p_t p_t}$ are multiples of the unit matrix, but with different factors.

Summarizing, the normal equation matrix easily can be built up directly from the given approximate values for the unknown parameters. Especially for isotropic uncertainty and proper conditioning, the two matrices $\mathbf{N}_{kk}$ and $\mathbf{N}_{pp}$ turn out to be diagonal matrices, which simplifies the reduction of the normal equations to the coordinates or the parameters, discussed next.

15.3.3.2 The Reduced Normal Equations

For many practical problems where the number of the images is in the thousands and the number of the points is in the hundreds of thousands, the normal equation matrix may be too large to fit into the computer's memory, especially if scene points are observed in many images, e.g., when analysing video streams. Therefore it may be useful to directly build the reduced normal equations. The reduction can be performed either to the scene parameters $\boldsymbol{k}$ or to the transformation parameters $\boldsymbol{p}$. In most cases, the reduction to the transformation parameters is preferable. For a general discussion of the reduction, cf. Sect. 4.2.6, p. 94. Here we make the reduction process more explicit, investigate the possibility of setting up the reduced equations directly and analyse the sparsity patterns of the resulting matrices.

We determine $\widehat{\Delta \boldsymbol{k}}$ from the 2×2-block equation system

$$\mathbf{N}_{kk}\widehat{\Delta \boldsymbol{k}} + \mathbf{N}_{kp}\widehat{\Delta \boldsymbol{p}} - \boldsymbol{h}_k = \mathbf{0} \tag{15.75}$$

$$\mathbf{N}_{pk}\widehat{\Delta \boldsymbol{k}} + \mathbf{N}_{pp}\widehat{\Delta \boldsymbol{p}} - \boldsymbol{h}_p = \mathbf{0} \tag{15.76}$$

by solving (15.75), getting

$$\widehat{\Delta \boldsymbol{k}} = \mathbf{N}_{kk}^{-1}(\boldsymbol{h}_k - \mathbf{N}_{kp}\widehat{\Delta \boldsymbol{p}}) , \tag{15.77}$$

and substitute it in (15.76). We obtain the normal equations reduced to the transformation parameters with the reduced normal equation matrix $\overline{\mathbf{N}}_{pp}$, which is also called the *Schur complement* of $\mathbf{N}$, and the reduced right-hand sides $\overline{\boldsymbol{h}}_p$,

Schur complement

$$\overline{\mathbf{N}}_{pp}\widehat{\Delta \boldsymbol{p}} = \overline{\boldsymbol{h}}_p \quad \text{with} \quad \overline{\mathbf{N}}_{pp} = \mathbf{N}_{pp} - \mathbf{N}_{pk}\mathbf{N}_{kk}^{-1}\mathbf{N}_{kp} , \; \overline{\boldsymbol{h}}_p = \boldsymbol{h}_p - \mathbf{N}_{pk}\mathbf{N}_{kk}^{-1}\boldsymbol{h}_k . \tag{15.78}$$

To simplify the other derivations, we express the reduced normal equation system using the reduced coefficients $\overline{\mathbf{D}}$ and get

$$\overline{\mathbf{N}}_{pp}\widehat{\Delta \boldsymbol{p}} - \overline{\boldsymbol{h}}_p = \overline{\mathbf{D}}^{\mathsf{T}}\mathbf{W}_{ll}\overline{\mathbf{D}} - \overline{\mathbf{D}}^{\mathsf{T}}\mathbf{W}_{ll}\Delta \boldsymbol{l} = \mathbf{0} \quad \text{with} \quad \overline{\mathbf{D}} = \mathbf{D} - \mathbf{C}\mathbf{N}_{kk}^{-1}\mathbf{N}_{kp} , \tag{15.79}$$

in full analogy to (4.123).

sparsity of reduced normal equation matrix

The reduced normal equation matrix $\overline{\mathbf{N}}_{pp}$ is also sparse. It has nonzero submatrices $\overline{\mathbf{N}}_{p_t p_{t'}}$ where two images share a common image point. For the diagonal submatrices we explicitly obtain

$$\overline{\mathbf{N}}_{p_t p_t} = \mathbf{N}_{p_t p_t} - \sum_{i \in \mathcal{I}_t} \mathbf{N}_{p_t k_i}\mathbf{N}_{k_i k_i}^{-1}\mathbf{N}_{k_i p_t} . \tag{15.80}$$

The off-diagonal submatrices are

$$\overline{\mathbf{N}}_{p_t p_{t'}} = - \sum_{i \in \mathcal{I}_t \cap \mathcal{I}_{t'}} \mathbf{N}_{p_t k_i}\mathbf{N}_{k_i k_i}^{-1}\mathbf{N}_{k_i p_{t'}} . \tag{15.81}$$

Figure 15.6 shows the adjacency graph for the images and points which reflects the sparse structure of the reduced normal equation matrices $\overline{\mathbf{N}}_{pp}$ and $\overline{\mathbf{N}}_{kk} = \mathbf{N}_{kk} - \mathbf{N}_{kp}\mathbf{N}_{pp}^{-1}\mathbf{N}_{pk}$, cf. (4.121), p. 95. For example, images a and f have no common point (left), and points 1 and 11 are not in the same image (right). The two graphs can be obtained

from the original factor graph (Fig. 15.3, p. 654) by edge contraction, which fuses the two end nodes of an edge and transfers the connections. The adjacency graph for the images (left) is obtained by sequentially contracting edges which belong to points, whereas the adjacency graph for the points is obtained by contacting all edges belonging to images. The graphs in Fig. 15.6 refer to the images and the scene features, not to the individual

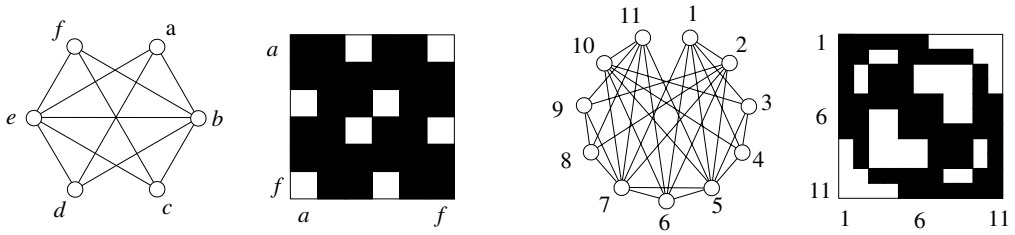


Fig. 15.6 Adjacency graph of images (left) and points (right) and a visualization of the corresponding reduced normal equation matrices $\overline{\mathsf{N}}_{pp}$ and $\overline{\mathsf{N}}_{kk}$ for the example data set, which also can be interpreted as the adjacency matrices of the two corresponding graphs

parameters. Therefore the storage and manipulation of these graphs is much more efficient than those of the graphs corresponding to the nonzero elements in the matrices.

15.3.3.3 Solution for the Parameters

Determining the Unknown Parameters. In the following we assume the normal equations are reduced w.r.t. the transformation parameters, $\widehat{\Delta\boldsymbol{p}}$. These are determined by solving the equation system $\overline{\mathsf{N}}_{pp}\widehat{\Delta\boldsymbol{p}} = \overline{\boldsymbol{h}}_p$. This determination can exploit the sparseness, as discussed below, and therefore generally is significantly more efficient than using $\widehat{\Delta\boldsymbol{p}} = \overline{\mathsf{N}}_{pp}^{-1}\overline{\boldsymbol{h}}_p$.

The coordinate parameters $\widehat{\Delta\boldsymbol{k}}$ can be easily determined individually from (15.77)

$$\widehat{\Delta\boldsymbol{k}}_i = \mathsf{N}_{k_ik_i}^{-1}\left(\boldsymbol{h}_{k_i} - \sum_{t\in\mathcal{T}} \mathsf{N}_{k_ip_t}\widehat{\Delta\boldsymbol{p}}_t\right), \tag{15.82}$$

where the sum is taken over all images t in the set $\mathcal{T}_i$ containing the point i.

Solving the Sparse Normal Equations. The solution of the sparse normal equations $\mathsf{N}\widehat{\Delta\boldsymbol{x}} = \boldsymbol{h}$, or their reduced variants, usually is performed by some triangular reduction, either LU or Cholesky decomposition, and back substitution.

This has a number of advantages, all resulting from the following observation. The inverse of a sparse normal equation matrix usually is a full matrix. In contrast, the reduction of the equation system by a triangular reduction preserves the zeros to a large extent. This situation can easily be recognized from the basic reduction operation in the kth reduction step of an element in the upper triangular part of the $U \times U$-normal equation matrix $\mathsf{N} = [N_{ij}]$,

$$N_{ij} := N_{ij} - \frac{N_{ik}N_{kj}}{N_{kk}}, \quad k = 1, ..., U-1, i = k+1, ..., U, j = k+1, ..., U; \tag{15.83}$$

cf. Fig. 15.7. From this we see two main effects:

1. If an off-diagonal element N_{ij} of the upper triangular part of N is zero and all elements N_{kj} lie above this element, i.e., for all $k > j$, are zero too, nothing changes. Thus in the columns of the upper triangular part of the normal equation matrix, leading zeros are preserved. This is the basis for an efficient storage scheme using the *profile* of the

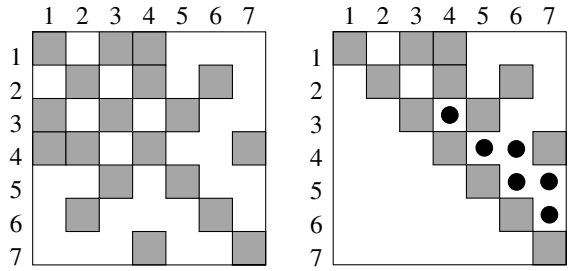


Fig. 15.7 Principle of the reduction of a sparse symmetric matrix (**left**) to upper triangular form (**right**); zeros are white regions. Sparsity is preserved, i.e., leading zeros in columns of the upper triangular matrix remain zero, e.g., element (3,7). *Fill-in* occurs if zeros below the upper nonzero element in the columns of the upper triangular matrix become nonzero, e.g., element (3,4)

profile of normal equation matrix

normal equation matrix, which is the set of nonzeros with lowest index in each column: No elements above the profile need to be stored.

2. If an off-diagonal element N_{ij} of the upper triangular part of N is zero and only one factor $N_{ik}N_{kj}$ is nonzero, the element will become nonzero. Thus the number of nonzeros during the reduction process will increase. This effect is called *fill-in*.

fill-in

The algorithmic complexity of the reduction mainly depends on the number of nonzeros after the reduction process. As a consequence, the algorithmic complexity can be influenced by proper sorting of the unknown parameters, see Fig. 15.8. Whereas the number

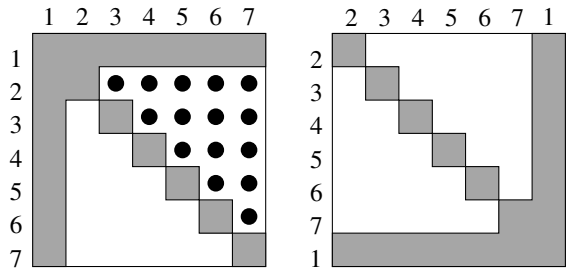


Fig. 15.8 Optimal sorting for diminishing algorithmic complexity when reducing the normal equation matrix; white regions are zeros. **Left:** Full fill-in, as all columns have nonzeros in the first row. **Right:** no fill-in, as all elements below the first nonzero in each column are nonzero until the diagonal

of nonzeros of the normal equation matrix is invariant to the sorting of the unknown parameters, the fill-in heavily depends on the ordering.

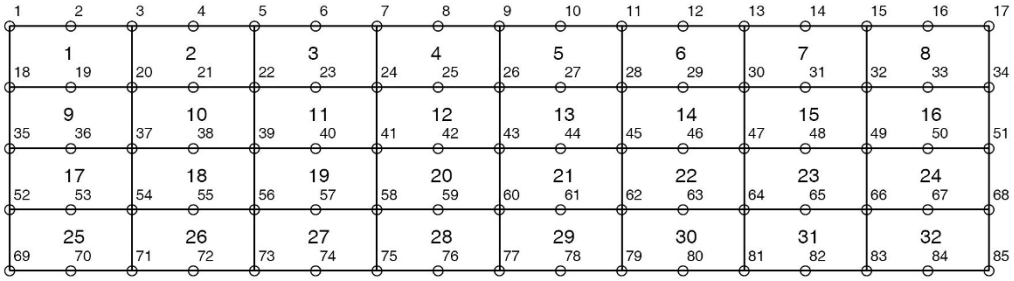
There are several principle ways to minimize fill-in. An intuitive way is to sort the unknowns with increasing number of nonzeros in rows or columns. Alternative options are to minimize the profile (Snay, 1976) or the *bandwidth* of the normal equation matrix. The bandwidth is the maximum distance of an off-diagonal element from the main diagonal. This principle is the basis for the reverse Cuthill–McKee algorithm (1969).

bandwidth

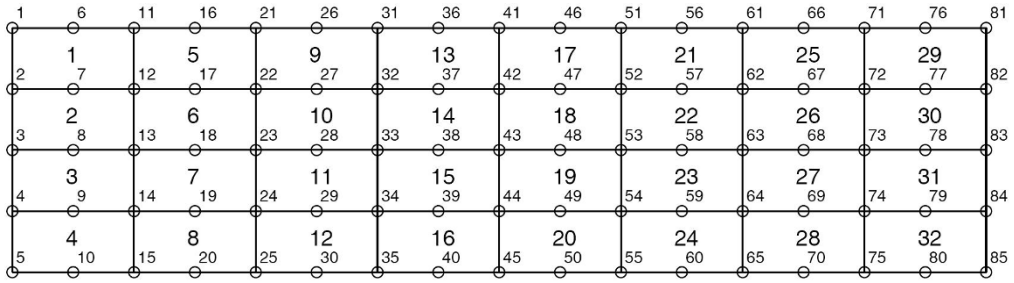
An example demonstrates the effect of ordering. Figure 15.9 shows the same block with two different numberings of the unknown scene points and image parameters. The effects of fill-in can be compared in Fig. 15.10.

15.3.3.4 Elements of the Covariance Matrix of Parameters

Evaluating the result of an estimation requires the variances and covariances of the estimated parameters. For large blocks of images the calculation of the full inverse is prohibitive, even of the reduced normal equations N_{pp} .



Numbering along the strip



Numbering across the strip

Fig. 15.9 Two numberings of scene points and images of a planar block: along the strip and across the strip, scene points from 1 to 85, images from 1 to 32

In many cases, knowing the variances and covariances $\Sigma_{p_t p_t}$ or $\Sigma_{k_i k_i}$, referring to the transformation parameters of individual images or to the coordinates of individual scene points, may be sufficient. Also, for statistically evaluating the residuals of individual image points only, their 2×2 covariance matrix, cf. (4.59), p. 87,

$$\Sigma_{v_{it}v_{it}} = \Sigma_{l_{it}l_{it}} - E_{it}^{\mathsf{T}} A \Sigma_{\hat{x}\hat{x}} A^{\mathsf{T}} E_{it} = \Sigma_{l_{it}l_{it}} - E_{it}^{\mathsf{T}} C N_{kk}^{-1} C^{\mathsf{T}} E_{it} - E_{it}^{\mathsf{T}} \overline{D} \Sigma_{\hat{p}\hat{p}} \overline{D}^{\mathsf{T}} E_{it}\,, \quad (15.84)$$

with the reduced coefficients $\overline{D}$ (cf. (15.79)), would be sufficient. The $2 \times 2(B + I_0)$ matrix E_{it} has an entry I_2 at the position of observational values of $\boldsymbol{l}_{it}$ and thus selects the corresponding rows and columns of the large matrix $A\Sigma_{\hat{x}\hat{x}}A^{\mathsf{T}}$. As $\overline{D}^{\mathsf{T}} E_{it}$ is sparse, we only need a few elements of $\Sigma_{\hat{p}\hat{p}} = \overline{N}_{pp}^{-1}$, namely where $\overline{N}_{pp}$ is nonzero.

sparse covariance matrix

These individual covariance matrices can be efficiently derived, provided those elements of the covariance matrix $\Sigma_{pp} = \overline{N}_{pp}^{-1}$ where there are nonzeros in the normal equation matrix, are known. These elements can be determined *without* needing to determine the other elements of the covariance matrix Σ_{pp}, cf. Triggs et al. (2000, Eq. (53)), Takahashi et al. (1973, cf. MATLAB-code `sparseinv.m`), and Vanhatalo and Vehtari (2008).

15.3.4 Free Block Adjustment

A free block adjustment is an adjustment without control information or constraints, except for those given by the correspondences between the images. This is the standard situation when stitching a set of images: no coordinates of control points in the mosaicked image are required. As a consequence, the coordinate system of the mosaic usually can be chosen freely, e.g., parallel to one of the given images or in another meaningful manner. For the similarity model, the position of the origin, the direction of the axis, and the scaling are not fixed uniquely. A general discussion on how to handle this situation can be found in Sect. 4.5, p. 108ff.

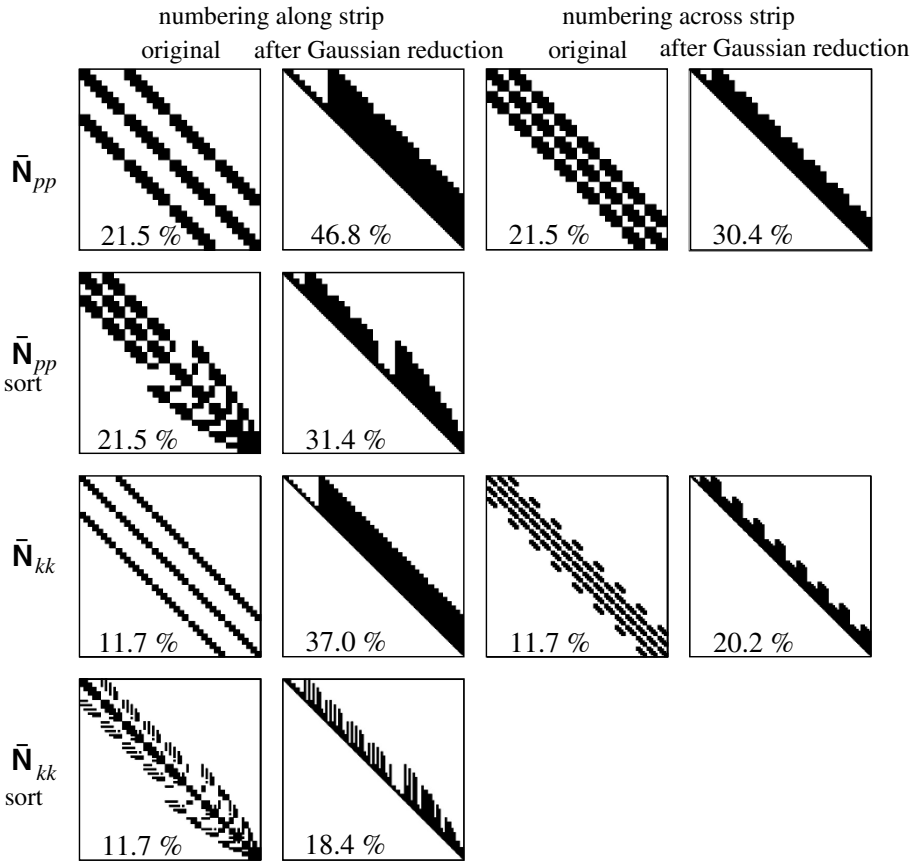


Fig. 15.10 Effect of sorting on the fill-in. Shown are the nonzero elements in the reduced normal equation matrices before and after Gaussian elimination for several sortings together with the percentage of nonzeros. **First two rows:** normal equation matrix reduced to transformation parameters $\bar{N}_{pp}$ (128×128), without and with sorting with reverse Cuthill–McKee. **Lower two rows:** normal equation matrix reduced to scene coordinates $\bar{N}_{kk}$ (170×170), without and with sorting. **Left two columns:** numbering along the strip before and after Gaussian reduction. **Right two columns:** numbering across the strip before and after Gaussian reduction. The numbering across the strip is favourable. In the case of an unfavourable original sorting – along the strip – sorting helps. Reverse Cuthill–McKee does not outperform numbering across strip in this case. The results after sorting, of course, are independent of the original numbering

15.3.4.1 Minimal Control for Fixing the Gauge

Fixing the gauge by a minimal number of four parameters in our context of stitching can be achieved in several ways. The most important ones are fixing

1. arbitrary scene points, here two, e.g., $\boldsymbol{x}_i := [0,0]^\mathsf{T}$ and $\boldsymbol{x}_{i'} := [1,0]^\mathsf{T}$,
2. arbitrary translation parameters, here two, e.g., $\boldsymbol{r}_t := [0,0]^\mathsf{T}$ and $\boldsymbol{r}_{t'} := [1,0]^\mathsf{T}$, cf. (15.23), p. 651, or
3. the parameters of an arbitrary image t, here four, e.g., $\boldsymbol{p}_t := [0,0,1,0]^\mathsf{T}$.

This most easily is realized by omitting the parameters as unknowns in the estimation process.

Though the choice of these parameters is open, for numerical reasons it is recommended that we choose in case 1. the pairs (i, i') of points, in case 2. the pairs (t, t') of images to be far apart, and in case 3. the image t to be in the centre of the configuration.

An *approximate* solution is to introduce the selected parameters as observations with small standard deviations or large weights. The standard deviations of these observations should be at least four orders of magnitude smaller than the standard deviations of the other parameters of the same type.

15.3.4.2 Gauge Constraints

In order to avoid the elimination of a subset of parameters, we can introduce constraints fixing the selected parameters, see the general discussion in Sect. 4.5.3, p. 111. Fixing the first of the T cameras then would be realized by introducing the constraints

fixing the gauge with constraints

$$H_p^\mathsf{T}(\widehat{\boldsymbol{p}} - \boldsymbol{p}^{(0)}) = \mathbf{0} \tag{15.85}$$

with the $(4 \times 4T)$ matrix and the $4T$-vector $\boldsymbol{p}^{(0)}$,

$$\underset{4\times 4T}{H_p^\mathsf{T}} = [I_4 \mid 0 \mid \dots] \qquad \boldsymbol{p}^{(0)\mathsf{T}} = [0\ 0\ 1\ 0\ |\dots]\,. \tag{15.86}$$

In order to keep the fill-in small, the constraint matrix should be at the lower and right borders of the normal equation matrix, leading to the extended normal equation system

$$\begin{bmatrix} N_{kk} & N_{kp} & 0 \\ N_{pk} & N_{pp} & H_p \\ 0 & H_p^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \Delta\widehat{\boldsymbol{k}} \\ \Delta\widehat{\boldsymbol{p}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_k \\ \boldsymbol{n}_p \\ H_p^\mathsf{T}(\widehat{\boldsymbol{p}}^{(\nu)} - \boldsymbol{p}^{(0)}) \end{bmatrix}, \tag{15.87}$$

where the $\boldsymbol{p}^{(0)}$ are the prespecified approximate values and the $\widehat{\boldsymbol{p}}^{(\nu)}$ are the approximate values for the estimation in the νth iteration, and $\boldsymbol{\mu}$ is a 4-vector of Lagrangian multipliers. As discussed in Sect. 4.3.1, p. 100, the normal equation matrix is not positive definite, requiring care when solving the equation system.

Similarly, we can define the coordinate system by fixing two scene points using $H_k^\mathsf{T}(\widehat{\boldsymbol{k}} - \boldsymbol{k}^{(0)}) = \mathbf{0}$. If the first two points are chosen to fix the gauge, the constraint matrix H_k has the same form, $H_k^\mathsf{T} = [I_4 \mid 0 \mid \dots]$.

fixing the gauge using scene points

Alternatively, we can define the gauge by prespecified approximate values $\boldsymbol{k}^{(0)}$ of the scene points $\boldsymbol{k}$ in a fully symmetric manner. We require

symmetric gauge fixation w.r.t. scene points

1. the centroid of the estimated coordinates and the centroid of the approximate values to be identical;
2. the rotation of an estimated scene point to its approximate scene point on average of all points to be zero, again possibly using a weighting; and
3. the average squared distance of all scene points from their centroid and the average squared distance of all approximate points from their centroid to be the same.

The constraints can be expressed in the form (cf. (4.227), p. 113)

$$H_k^\mathsf{T} W^\mathsf{T} \left(\widehat{\boldsymbol{k}} - \boldsymbol{k}^{(0)}\right) = \begin{bmatrix} \sum_i w_i \left(\widehat{\boldsymbol{k}} - \boldsymbol{k}^{(0)}\right) \\ \sum_i w_i Z(\boldsymbol{k}_i^{(0)}) \left(\widehat{\boldsymbol{k}} - \boldsymbol{k}^{(0)}\right) \end{bmatrix} = \mathbf{0}\,, \tag{15.88}$$

with

$$H_k = [H_{k_i}^\mathsf{T}]\,, \qquad H_{k_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_i^{(0)} & -y_i^{(0)} \\ y_i^{(0)} & x_i^{(0)} \end{bmatrix} \tag{15.89}$$

and the matrix

$$W = \mathrm{Diag}(\{W_i\})\,, \qquad W_i = w_i I_2\,, \quad w_i \in \{0, 1\}\,, \tag{15.90}$$

indicating which of the scene points are to be used. Observe, the matrix H_k is used to fixed the four gauge parameters of a 2D similarity transformation and is a special case of the matrix H in (4.221), p. 112, which can be used to fix the seven parameters of a spatial similarity transformation.

15.3.4.3 The Inner Precision of a Free Block

inner precision
estimable quantities
invariants

The precision of the scene point coordinates and the transformation parameters depends on (1) the choice of the gauge, (2) the geometric configuration, and (3) the precision of the observed image coordinates. As discussed in Sect. 4.5, p. 108, coordinates or transformation parameters are not estimable quantities, as they depend on the chosen gauge. However, distance ratios or angles between three scene points or positions of cameras, and angles and scale ratios between two images as invariants of the images, cf. Sect. 6.4, p. 266, in our similarity model are estimable quantities: their values and their variances and covariances do not depend on the chosen gauge. Such estimable quantities characterize the *inner precision* of a free block of images.

There are two ways to visualize the inner precision of the result of a free bundle adjustment:

1. The standard ellipses of the scene points or the positions of the images with the gauge referring to the centroid of the scene coordinates are shown. Figure 15.11 presents the

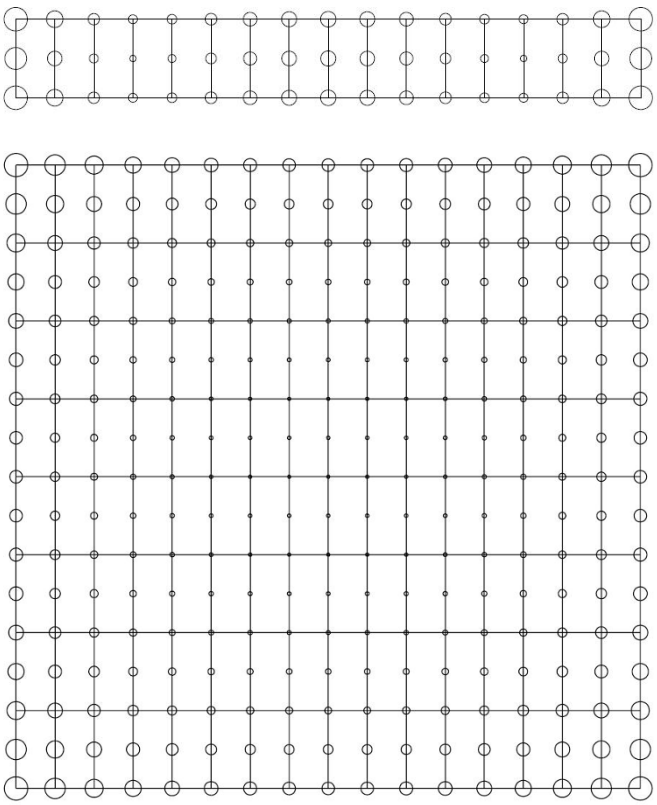


Fig. 15.11 Inner precision of a free strip and a free block of rectangular images in 2D visualized by standard ellipses, which here are circles. The lowest standard deviations in the strip occur at 1/5 and 4/5 of its length, indicating it behaves similarly to a long elastic stick

   standard ellipses, in this case circles, of the scene points of a strip with 16 images and a block with 8×16 images. There are six image points per image which establish the connection to the neighbouring images.
2. Alternatively we can show the largest effect of the random perturbations of the image measurements on the scene coordinates. A long strip of images behaves quite similarly to a free swinging stick. We therefore would expect a bend across the strip axis with fix points close to 1/5 and 4/5 of the length of the strip (cf. Berliner, 1928, p. 254). To determine the maximum effect we perform an eigenvalue decomposition, or, equivalently, a principal component analysis (PCA), of the $N_k \times N_k$ covariance matrix ${}^c\Sigma_{\widehat{x}\widehat{x}}$

for the N_k coordinates

PCA of covariance matrix

$$ {}^{c}\boldsymbol{\Sigma}_{\widehat{x}\widehat{x}} = \sum_{j=1}^{N_k} \boldsymbol{u}_j \boldsymbol{u}_j^{\mathsf{T}} \sigma_j^2 \,. \tag{15.91} $$

We assume the gauge to be defined by the centroid of the scene points, hence the superscript c. The eigenvalues of the covariance matrix can be interpreted as variances, which allows us to write the uncertain estimated coordinates as

$$ \underline{\widehat{\boldsymbol{x}}} = \boldsymbol{\mu}_{\widehat{x}} + \sum_j \boldsymbol{u}_j \underline{z}_j \,, \quad \text{with} \quad \underline{z}_j \sim \mathcal{M}(0, \sigma_j^2) \,. \tag{15.92} $$

This equation explains the uncertainty of $\underline{\widehat{\boldsymbol{x}}}$ as the sum of N_k independent random effects $\boldsymbol{u}_j \underline{z}_j$ caused by independent error sources $\underline{z}_j$. Due to the symmetric nature of the model w.r.t. scale and rotation, cf. (15.22), p. 650, all eigenvalues appear in pairs. The eigenvalues of the covariance matrix rapidly decay. The second eigenvalue pair usually is less than $1/4$ of the first one. In order to obtain a first impression about the impact of random effects on the scene coordinates, we can show the standard deviation vector $\boldsymbol{u}_j \sigma_j$ for the first few j.

Figure 15.12 shows two examples. The expected weakness of the strip with respect

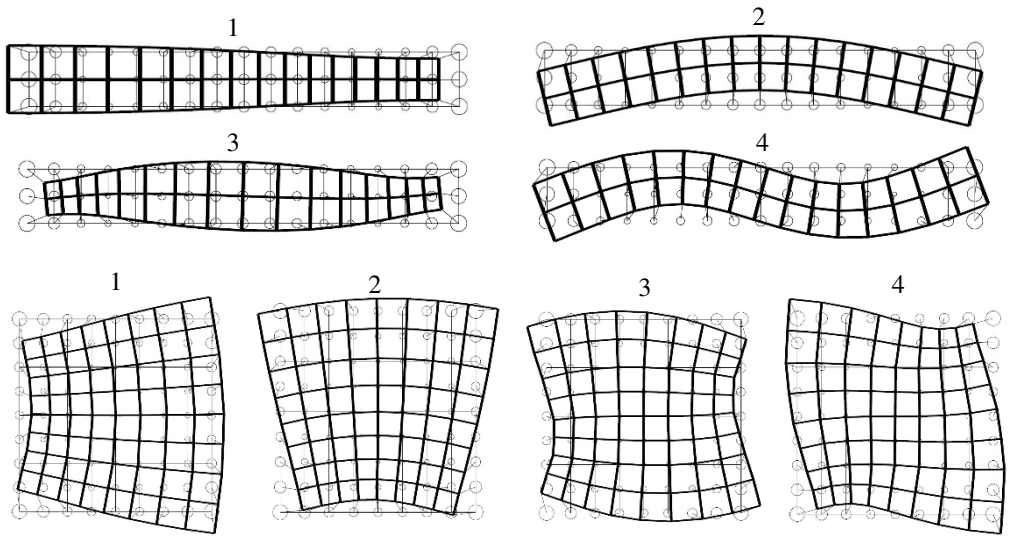


Fig. 15.12 Maximum effect of random perturbations on free 2D blocks: the largest four eigenvectors $\boldsymbol{u}_j, j = 1, ..., 4$, of the covariance matrix. **Top rows:** The strip is likely to be bent, caused by random errors in the directional transfer between the images. But the scale transfer may be also uncertain, leading to a quasi-systematic scale error. **Bottom row:** The block is likely to be bent, and at the same time – in perpendicular direction – the scale transfer may be systematically erroneous. The third and the fourth eigenvectors lead to more complex random deformations. The scaling of the random perturbations is not the same as the scaling of the standard ellipses, and different for the two cases

to a global bend, but also with respect to a global scale change, is clearly visible. The deformation consists of a bend of the block in one direction and a scale change in the other. Such deformations are to be expected. They obviously are caused by random effects. Sometimes they are called *quasi-systematic*, as they appear to be of systematic nature, though they are not caused by model errors, such as neglected lens distortion.

quasi-systematic errors

15.3.4.4 Gauge Transformations

Gauge or S-transformations, cf. Sect. 4.5.3, p. 111, are necessary when comparing results of free bundle adjustments following the procedure discussed in Sect. 4.6.2.2, p. 118ff. if the results of the two free bundle adjustments refer to different gauges or if their gauges are not known, e.g., when the documentation of a software package does not provide this information. Then both results first need to be brought into the same coordinate system and the same gauge. Afterwards all scene coordinates or all transformation parameters not defining the gauge can be compared using the test in (4.257), p. 119.

A gauge transformation can be seen as a weighted differential coordinate transformation of the approximate values $\widehat{\boldsymbol{x}}^a$ to the final estimates $\widehat{\boldsymbol{x}}$. The given covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$ refers to small changes $\widehat{\Delta\boldsymbol{x}}$ of the parameters.

The used weight matrix ${}^{\ell}W$ defines the gauge G_ℓ using the name ℓ for the coordinate system of the covariance matrix. The residuals ${}^{\ell}\underline{\widehat{\Delta\boldsymbol{x}}}$ of the parameters after this estimation contain the stochastic component of the parameters without the effect of the gauge parameters, here the four parameters of a similarity transformation. The covariance matrix ${}^{\ell}\Sigma_{\widehat{x}\widehat{x}}$ of the residuals is the covariance matrix sought with the specified gauge.

Technically, we reach this goal by determining the parameters of the similarity transformation – the small shift $\Delta\boldsymbol{a}$ and the small scale/rotation $\Delta\boldsymbol{b}$ – in a least squares sense from

$$\underbrace{\begin{bmatrix}\widehat{\Delta\boldsymbol{k}}_i\\ \widehat{\Delta\boldsymbol{r}}_t\\ \widehat{\Delta\boldsymbol{s}}_t\end{bmatrix}}_{\boldsymbol{l}:\ \widehat{\Delta\boldsymbol{x}}_{i,t}} - \underbrace{\begin{bmatrix}{}^{\ell}\widehat{\Delta\boldsymbol{k}}_i\\ {}^{\ell}\widehat{\Delta\boldsymbol{r}}_t\\ {}^{\ell}\widehat{\Delta\boldsymbol{s}}_t\end{bmatrix}}_{\widehat{\boldsymbol{v}}:\ {}^{\ell}\widehat{\Delta\boldsymbol{x}}_{i,t}} = \begin{bmatrix}\widehat{\Delta\boldsymbol{a}} + Z(\widehat{\boldsymbol{k}}_i^a)\widehat{\Delta\boldsymbol{b}}\\ \widehat{\Delta\boldsymbol{a}} + Z(\widehat{\boldsymbol{r}}_t^a)\widehat{\Delta\boldsymbol{b}}\\ Z(\widehat{\boldsymbol{s}}_t^a)\widehat{\Delta\boldsymbol{b}}\end{bmatrix} = \underbrace{\begin{bmatrix}I_2 & Z(\widehat{\boldsymbol{k}}_i^a)\\ I_2 & Z(\widehat{\boldsymbol{r}}_t^a)\\ 0 & Z(\widehat{\boldsymbol{s}}_t^a)\end{bmatrix}}_{A:\ H_{i,t}} \underbrace{\begin{bmatrix}\widehat{\Delta\boldsymbol{a}}\\ \widehat{\Delta\boldsymbol{b}}\end{bmatrix}}_{\widehat{\boldsymbol{x}}} \tag{15.93}$$

for $i \in \mathcal{I}, t \in \mathcal{T}$, and using the weights

$${}^{\ell}W_i = {}^{\ell}w_i I_2 \qquad {}^{\ell}W_t = {}^{\ell}w_i I_4\,, \tag{15.94}$$

which define how the gauge depends on the scene points and transformation parameters, see the basic model (15.23), p. 651. Since the gauge transformation only changes the covariance matrix, the similarity transformation (15.93) only serves to determine the Jacobian H_{it}.

Observe, we transform both the coordinates and the transformation parameters, in contrast to the setup in Sect. 4.5.3, p. 111, where we only transformed the coordinates. The comparison of block adjustment results based on the same images may be based on points only if the scene points in both block adjustments are the same. This is not guaranteed if the scene points result from some automatic key point detector, which might be different in the two block adjustments. Therefore it is advisable to compare the results of the camera parameters, i.e., the transformation parameters, which therefore need to be transformed into the same coordinate system so that their gauge is identical (cf. Dickscheid et al., 2008).

Linearizing with respect to the transformation parameters $\boldsymbol{a}$ and $\boldsymbol{b}$, we obtain the Jacobian

$$H_{it} = \begin{bmatrix}H_{k_i}\\ H_{p_t}\end{bmatrix} = \left[\begin{array}{cccc}1 & 0 & x_i^a & -y_i^a\\ 0 & 1 & y_i^a & x_i^a\\ \hline 1 & 0 & a_t^a & -b_t^a\\ 0 & 1 & b_t^a & a_t^a\\ 0 & 0 & c_t^a & -d_t^a\\ 0 & 0 & d_t^a & c_t^a\end{array}\right]\,, \tag{15.95}$$

to be evaluated at the approximate values. We obtain the best estimates for the residuals

$${}^{\ell}\widehat{\Delta\boldsymbol{x}} = [\{{}^{\ell}\widehat{\Delta\boldsymbol{x}}_{it}\}] = [\{{}^{\ell}\widehat{\Delta\boldsymbol{k}}_i\}, \{{}^{\ell}\widehat{\Delta\boldsymbol{p}}_t\}, \{{}^{\ell}\widehat{\Delta\boldsymbol{s}}_t\}] \tag{15.96}$$

under this assumption by using the redundancy matrix R as in (4.62), p. 87, here called the S-matrix,

$$ {}^{\ell}S = I - H(H^{\mathsf{T}}\,{}^{\ell}WH)^{-1}H^{\mathsf{T}}\,{}^{\ell}W\,, \tag{15.97} $$

leading to the similarity transformation

$$ {}^{\ell}\widehat{\Delta \boldsymbol{x}} = {}^{\ell}S\,\widehat{\Delta \boldsymbol{x}}\,. \tag{15.98} $$

S- or gauge transformation

The resulting estimates, now in the gauge G_c, therefore have the covariance matrix

$$ {}^{\ell}\Sigma_{\widehat{x}\widehat{x}} = {}^{\ell}S\,\Sigma_{\widehat{x}\widehat{x}}\,{}^{\ell}S^{\mathsf{T}}\,. \tag{15.99} $$

The transformation in (15.99) is the envisaged S-transformation or gauge transformation into the gauge system G_ℓ. This gauge or S-transformation results in the constraint

$$ H^{\mathsf{T}}\,{}^{\ell}W\,{}^{\ell}\widehat{\Delta \boldsymbol{x}} = \mathbf{0}\,, \tag{15.100} $$

as minimizing $\widehat{\boldsymbol{v}}^{\mathsf{T}}W\widehat{\boldsymbol{v}}$ in the Gauss–Markov model $\mathbb{E}(\underline{\boldsymbol{l}}) = A\boldsymbol{x}$ leads to the constraint $A^{\mathsf{T}}W\widehat{\boldsymbol{v}} = \mathbf{0}$, cf. (4.74), p. 88. Thus directly imposing these constraints during the bundle adjustment, cf. (15.88), leads to the same result as imposing the constraints via an S-transformation in a second step.

Example 15.3.50: ***Gauge transformation.*** For example, take the first transformation parameters $\widehat{\boldsymbol{p}}_1$ of two $\widehat{\boldsymbol{p}} = [\widehat{\boldsymbol{p}}_1; \widehat{\boldsymbol{p}}_2]$ as the gauge for the covariance matrix of the transformation parameters. We expect it to have a zero covariance matrix after the gauge transformation, since the four parameters $\widehat{\boldsymbol{p}}_1$ uniquely define the differential transformation, thus have residuals zero. With ${}^{\ell}W = \mathrm{Diag}(\{I_4^{\mathsf{T}}, 0_4\})$ and $H = [H_1; H_2]$, and therefore $H^{\mathsf{T}\ell}WH = H_1^{\mathsf{T}}H_1$, the structure of the matrix ${}^{\ell}S$ now is

$$ {}^{\ell}S = \begin{bmatrix} 0_{4\times 4} & 0_{4\times 4} \\ -H_2H_1^{-1} & I_4 \end{bmatrix}\,, \tag{15.101} $$

since $I_4 - H_1(H_1^{\mathsf{T}}H_1)^{-1}H_1^{\mathsf{T}} = 0$, because H_1 is regular. Obviously, the first parameter $\widehat{\Delta \boldsymbol{p}}_1$ transformed into gauge G_ℓ is zero, thus also has covariance matrix zero, independent of the covariance matrix $\Sigma_{\widehat{p}_1\widehat{p}_1}$. The second parameter is transformed to (cf. (4.225), p. 112)

$$ {}^{\ell}\widehat{\Delta \boldsymbol{p}}_2 = \widehat{\Delta \boldsymbol{p}}_2 - H_2H_1^{-1}\widehat{\Delta \boldsymbol{p}}_1\,. \tag{15.102} $$

◇

15.3.4.5 Evaluating the Covariance Matrix of the Parameters

If the bundle adjustment aims at reliably determining scene points or transformation parameters, e.g., the ego-motion of the camera, this can be done by specifying a *reference covariance matrix* $\Sigma_{xx}^{(\mathrm{ref})}$ which serves either as an upper bound for the desired precision or as a precision to be achieved on average.

For scene points, such a reference matrix $\Sigma_{kk}^{\mathrm{ref}}$ in the most simple case may be the multiple of a unit matrix, say $\sigma^2 I$, with an unspecified gauge. For transformation parameters, this specification, $\Sigma_{pp}^{\mathrm{ref}}$, needs to be done separately for the translation and the rotation/scale component.

Now let the covariance $\Sigma_{\widehat{x}\widehat{x}}$ of the estimated parameters be derived from the bundle adjustment. Then we can apply the evaluation schemes discussed in Sect. 4.6.2.3, p. 120. For consistency, we need to first transform the two covariance matrices which are to be compared into the same gauge using the above-mentioned gauge transformation. The choice of the gauge is arbitrary, as the comparison only refers to the uncertainty of the form, which is invariant to gauge transformations. It is easiest to choose the minimal number of parameters, which is necessary to fix the gauge, thus to apply a regular S-transformation (4.225), p. 112. These parameters obtain zero covariance matrices, can be left out from the comparison. Both criteria can be applied to scene points or transformations parameters separately, or also to subgroups of interest.

15.3.5 Theoretical Quality of Regular Strips and Blocks

The following two sections provide insight into the theoretical quality of regular image strips and blocks.

Quality here is understood as the precision of the estimated parameters, the detectability of outliers, and the effect of nondetectable outliers on the result. Especially, we discuss the theoretical precision of the estimated coordinates, thus the effect of random errors on the result. Whereas outliers in the image points usually can be detected quite reliably, the detectability of outliers in the control points usually is limited. Outliers in the control points mainly result from a wrong identification and, if not detected, lead to strong deterioration of the result.

Image strips occur regularly when building panoramas. If the panorama is made of many images taken, say, with a telecentric lens, we expect the image strip to bend due to the accumulation of random effects at the boundaries between the images. Closing the loop to a 360° panorama will significantly reduce this effect. This behaviour is typical for long (straight) image strips occurring in visual navigation, where – without external control, say from a GPS – long paths will bend and closing the loop will stabilize the configuration. Theoretically, the variance σ_x^2 of the position of the end point of a strip increases with the third power of the strip with T images, thus

$$\sigma_x^2(T) = a\, T^3\,, \tag{15.103}$$

Exercise 15.8

with some factor a, as it can be modelled as a doubly integrated white noise process. Introducing control or closing the loop reduces the factor but does not change the general rule (cf. Ackermann, 1966).

The situation is different in blocks where images are not arranged in a linear pattern but cover a complete region. During stitching, image blocks occur, covering, for example, a large facade or the painting on a ceiling. Here the inner geometry does not show effects of instability except at the border. In robotics, a similar situation occurs after having explored a complete area with images. Here also the inner stability of the recovered scene will be quite high, again except for the borders, where the images are only connected to others by one side. Generally, the variance σ_x^2 of the position increases very slowly, namely with the logarithm of the diameter d of the block, thus

$$\sigma_x^2(d) = b\, \log d \tag{15.104}$$

for large enough d if control points only lie at the border (cf. Meissl, 1972). These rules can be used to predict the precision performance of a certain configuration or to choose a configuration to achieve a prespecified precision.

The situation is quite different for the detectability of outliers in control points as it highly depends on the geometric configuration. Though we give examples of lower bounds for detectable gross errors in control points, they are meant to indicate the necessity of using enough control points and of applying rigorous statistical testing.

In the following we will give some representative examples to offer the reader some insight into the general performance of image strips and blocks w.r.t. their expected quality. The examples are based on given configurations, specifying the Jacobian A, and on assumptions about the precision of the image and the control point coordinates, specifying the covariance matrix Σ_{ll}. We use the derived theoretical covariance matrix $\Sigma_{\widehat{x}\widehat{x}} = (\mathsf{A}^\mathsf{T}\Sigma_{ll}\mathsf{A})^{-1}$ and the redundancy numbers $r_n = (\Sigma_{\widehat{v}\widehat{v}}\Sigma_{ll}^{-1})_{nn}$ of the control point coordinates, which are assumed to be uncertain with a standard deviation corresponding to the uncertainty of the image points, cf. (4.64), p. 87. We always assume each image to contain six image points. Neighbouring images within a strip have three points in common, neighbouring images across strips have two points in common. Whereas the general structures can be transferred to other situations, the parameters a and b of the equations (15.103) and (15.104) need to be determined by investigating the configuration of interest.

15.3.5.1 Theoretical Quality of Regular Strips

We start with the analysis of regular straight strips. Figure 15.13 shows for strips with 4, 8, and 16 images (1) the standard ellipses for the scene points and (2) the lower bounds for detectable outliers in the control points, once with the left border fixed and once with both borders fixed.

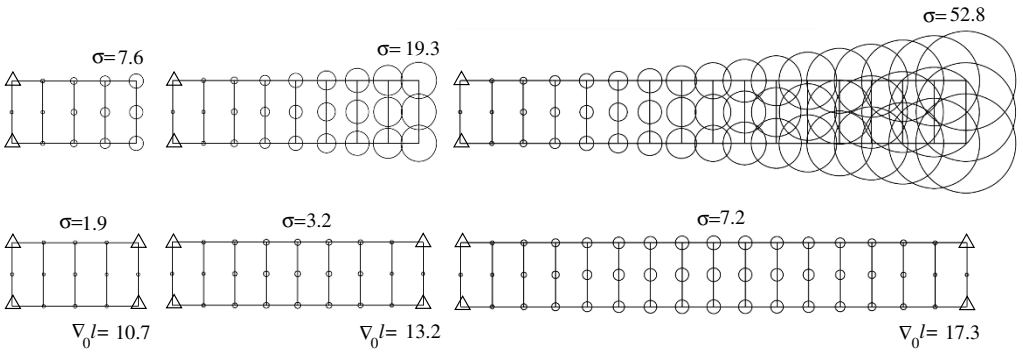


Fig. 15.13 Quality of scene points of a strip with four, eight and 16 images. Triangles indicate control points. **First row:** Fixation at one end with two points, which is the minimum for fixing the gauge of the strip. **Second row:** Fixation at both ends. Maximum standard deviations $\sigma := \max(\sigma_{\hat{x}})$ in units of the precision of image points $\sigma_{x'}$ occur at the end or in the middle of the strips, respectively. The positive effect of control points at both ends is clearly visible. Minimum lower bounds $\nabla_0 l$ for detectable outliers in control point coordinates when using a statistical test, cf. Table 15.2, column 11. When using a simple test, the lower bounds are much higher, cf. Table 15.2, column 12

The precision deteriorates with the distance from the control points. Fixing both ends significantly improves the theoretical precision. The lower bounds for detectable outliers in the control points indicate that for large distances between the control points outliers must be quite large to be detectable when applying a statistical test.

Table 15.2 provides (1) the means and the maximum standard deviations in units of the image measuring standard deviation $\sigma_{x'}$, and (2) the redundancy numbers r_n and the lower bounds for detectable outliers in the control points for the same strips and for two strips with 32 and 64 images.

Table 15.2 Examples of the quality of strips of images as a function of the number T of images. The images scale is 1. Standard deviations of the control points and of the image points are 1. Standard deviations $\sigma_{\hat{x}}$ and average standard deviations $\overline{\sigma_{\hat{x}}}$. **Columns 2–5:** left end of strip fixed. **Columns 6–9:** both ends of strip fixed. **Column 10:** redundancy numbers for control point coordinates. **Columns 11-12:** lower bounds for detectable outliers in control point coordinates when using the statistically rigorous test statistic v_n/σ_{v_n} and when using the simple test statistic v_n/σ_{l_n}. **Column 13:** relation of $\nabla_0 l_n/\sqrt{T}$. **Column 14:** sensitivity factors μ_n. See text for further explanation.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
T	$\overline{\sigma_{\hat{x}}}$	$\frac{\overline{\sigma_{\hat{x}}}}{T^{3/2}}$	$\max(\sigma_{\hat{x}})$	$\frac{\max(\sigma_{\hat{x}})}{T^{3/2}}$	$\overline{\sigma_{\hat{x}}}$	$\frac{\overline{\sigma_{\hat{x}}}}{T^{3/2}}$	$\max(\sigma_{\hat{x}})$	$\frac{\max(\sigma_{\hat{x}})}{T^{3/2}}$	r_n	$\nabla_0 l_n$	$\nabla_0^* l_n$	$\frac{\nabla_0 l_n}{\sqrt{T}}$	μ_n
4	3.76	0.4704	7.61	0.9519	1.27	0.1593	1.94	0.2429	0.1393	10.7	28.7	5.36	2.48
8	8.49	0.3754	19.27	0.8516	2.12	0.0940	3.22	0.1426	0.0915	13.2	43.7	4.67	3.15
16	21.94	0.3428	52.82	0.8253	4.44	0.0694	7.20	0.1125	0.0534	17.3	74.9	4.32	4.21
32	60.13	0.3322	148.20	0.8187	11.22	0.0620	18.95	0.1046	0.0289	23.5	138.4	4.16	5.80
64	168.28	0.3286	418.32	0.8170	30.82	0.0602	52.59	0.1027	0.0150	32.6	266.2	4.08	8.10

The third-power law for the variance as a function of the strip length is confirmed (see columns 3, 7, and 9), independently of whether the fixation is only at one end or on both ends. However, fixing the strip at both ends improves the standard deviation of the points by a factor of 8 approximately. The redundancy numbers for the control point coordinates decrease linearly with the length of the strip (see column 10). They

demonstrate that only a small percentage of errors in the control point coordinates show up in their residuals. For instance, only 9.15% of an error in a control point coordinate shows up in the corresponding residual if the strip has eight images. Therefore the lower bound $\nabla_0 l_n$ for detectable outliers in the control points (column 11) increases slowly with the square root of the length of the strip (see column 13). This assumes that the optimal test statistic $z_n = \widehat{v}_n/\sigma_{\widehat{v}_n} = \widehat{v}_n/\sigma_{l_n}/\sqrt{r_n}$ is applied, which requires the determination of the redundancy numbers r_n. If, instead, the simple test statistic $z_n^* = \widehat{v}_n/\sigma_{l_n}$ is used, outliers must be much larger than $\nabla_0^* l_n$ to be detectable (see column 12 and (4.289), p. 126). The sensitivity factors $\mu_n = \sqrt{(1-r_n)/r_n}$ indicate that a large percentage of nondetected outliers in the control points directly deteriorate the result (see column 14 and (4.292), p. 126). For example, a nondetectable outlier in a control point of a strip with eight images may deteriorate the coordinates up to $\delta_0 \mu_n \approx 4 \times 3.15$ i.e., 12 times the standard deviation of the coordinates, mainly in the vicinity of the erroneous ground control, cf. (4.294), p. 127.

effect of loop closing

The last example in Fig. 15.14 shows the effect on the positional precision when closing a loop in a strip of images. Assume a strip of images is taken in a circular path with 32

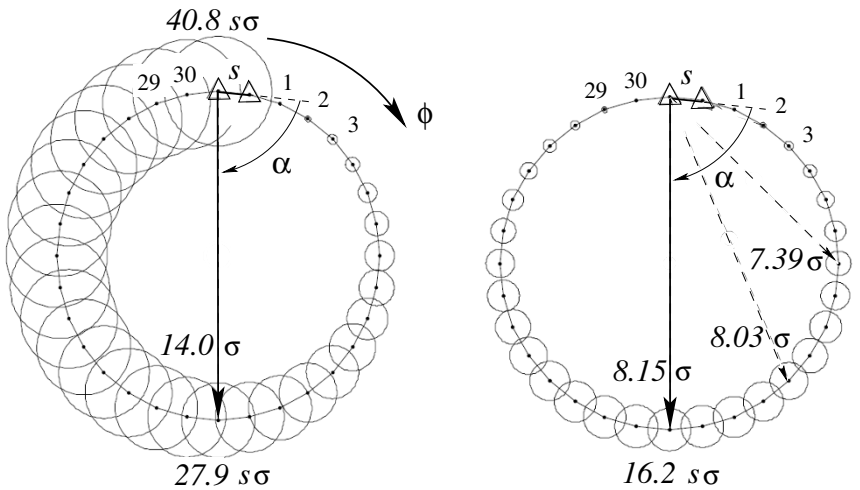


Fig. 15.14 Effect of loop closing on the precision of a strip of images based on a free block adjustment, where the gauge is fixed by two neighbouring cameras (indicated by triangles) having distance s. The uncertainty of the angles and distance ratios between two neighbouring images is σ; this corresponds to a standard deviation $s\sigma$ of the parameters of the scaled rotation of the transformation, namely c_t and d_t in (15.30), p. 652. Shown are standard ellipses, here circles, and standard deviations of relative positions w.r.t. the first control point (in units $s\sigma$) and angles α w.r.t. the direction between the two control points (in units σ). **Left:** Precision of positions and angles without loop closing. **Right:** Precision of positions and angles with loop closing in the same gauge. See text for explanation

regularly spaced positions. The gauge is fixed by the position of two neighbouring images indicated by triangles in the figure. Assume that the angular and the scale uncertainty between an image and its two neighbours is σ. This causes the third image position to be uncertain by $s\sigma$ in all directions, where s is the distance between neighbouring images. The statistical behaviour can be modelled as a polygonal chain, where the angles and distance ratios between neighbouring sides are observed, the standard deviations of the angles and distance ratios are identical, and the gauge is defined by fixing the coordinates of the first two points. We investigate the precision of the positions of the cameras, we do not address their directions.

If the strip is not closed and the gauge is fixed by the first two images, 1 and 2, the precision deteriorates with the sequential number of the image. The variance increases quite well with

$$\sigma_x^2(\phi) = a(\phi - \sin\phi)\,, \quad \phi \geq 0\,, \tag{15.105}$$

where ϕ is the direction to the pose seen from the centre of the circle. For images close to the two gauge point the third power law (15.103) is fulfilled approximately. The theoretical precision of the mid position (opposite the basis) is approximately 28 $s\sigma$, the final position (identical to the first base point) is 41 $s\sigma$. The distance across the circular path from image position 1 to 15 has a standard deviation of approximately 28 $s\sigma$.

If the loop is closed, i.e., the last image, 32, is joined to image 1, the distance across the circle is nearly two times better, namely 16.2 $s\sigma$, than without closure. The angle between the direction of the basis 1-2 and the direction across the circular path now has a standard deviation of approximately 8.2 σ. Also the angles between the direction of the basis and the directions to points closer to the beginning of the strip show comparable precision. The values also hold for the relative scale uncertainty between neighbouring positions when taking σ as relative standard deviation of the scale ratio of two neighbouring distances. Observe, fixing the ends of a circular strip decreases the maximum standard deviation by a factor of 2.5 only, in contrast to fixing the ends of the straight strips, where this decrease happens by a factor of 7.3, see Fig. 15.13.

15.3.5.2 Theoretical Quality of Regular Blocks

The theoretical quality of blocks of stitched images is given in Table 15.3 for quadratic blocks with four control points at the corners and for blocks with full control at the border with a control point at every two image points.

Table 15.3 Examples of the quality of quadratic blocks. The image scale is one. Columns 1 to 7: theoretical precision; Columns 8 to 12: detectability of outliers at control points (CPs) in the corners. **Top table:** four corners are fixed. **Lower table:** the border is fixed. Redundancy numbers for control points along the boundary are larger by a factor of approximately 1.7. The maximum uncertainty (max) and the uncertainty in the middle (mid) of the block is given as absolute value and relative to the linear block size M in base lengths

1	2	3	4	5	6	7	8	9	10	11	12
four CP $T = N \times M$	$\overline{\sigma_x}$	$\frac{\overline{\sigma_x}}{M}$	$\max(\sigma_x)$	$\frac{\max(\sigma_x)}{M}$	$\mathrm{mid}(\sigma_x)$	$\frac{\mathrm{mid}(\sigma_x)}{M}$	r_n	$\nabla_0 l_n$	$\nabla_0^* l_n$	$\frac{\nabla_0 l_n}{\sqrt{M}}$	μ_n
2×4	1.48	0.3712	1.96	0.4915	1.56	0.3907	0.0866	13.6	46.6	4.80	3.3
4×8	2.33	0.2919	3.13	0.3915	2.21	0.2763	0.0281	23.9	142.6	4.22	5.9
8×16	4.02	0.2516	5.69	0.3561	3.72	0.2324	0.0076	45.9	526.6	4.06	11.4
16×32	7.57	0.2366	11.04	0.3452	6.97	0.2177	0.0019	90.8	2062.6	4.01	22.9
border CP $T = N \times M$	$\overline{\sigma_x}$	$\frac{\overline{\sigma_x}}{\ln M}$	$\max(\sigma_x)$	$\frac{\max(\sigma_x)}{\ln M}$	$\mathrm{mid}(\sigma_x)$	$\frac{\mathrm{mid}(\sigma_x)}{\ln M}$	r_n	$\nabla_0 l_n$	$\nabla_0^* l_n$		μ_n
2×4	0.93	0.6727	1.66	1.1983	1.15	0.8330	0.1966	9.0	20.3		2.02
4×8	1.08	0.5209	1.66	0.7993	1.32	0.6325	0.1958	9.0	20.4		2.03
8×16	1.25	0.4500	1.66	0.5995	1.46	0.5250	0.1957	9.0	20.4		2.03
16×32	1.40	0.4054	1.70	0.4898	1.59	0.4576	0.1957	9.0	20.4		2.03

An example of the variations of the standard ellipses is given in Fig. 15.15. The precision is quite homogeneous in the interior of the block. The inhomogeneity at the border can be eliminated by a dense control point chain along the border. These results w.r.t. to the precision were already found by Ackermann (1966) in the context of model block adjustment. Obviously, not fixing the border of a configuration leads to average standard deviations increasing with the block size, whereas fixing the border keeps the precision more or less constant.

The detectability of outliers in control point coordinates is very low if we have only four control points at the corners of the block, linearly decreasing with the side length of the block. Having a dense control point chain at the border leads to quite good detectability of control point outliers. However, control point errors still need to be larger than nine standard deviations, and, if not detected, lead to deteriorations of the result up to eight times its standard deviation, as $\mu = \sqrt{(1 - 0.19)/0.19} \approx 2$, cf. (4.294), p. 127.

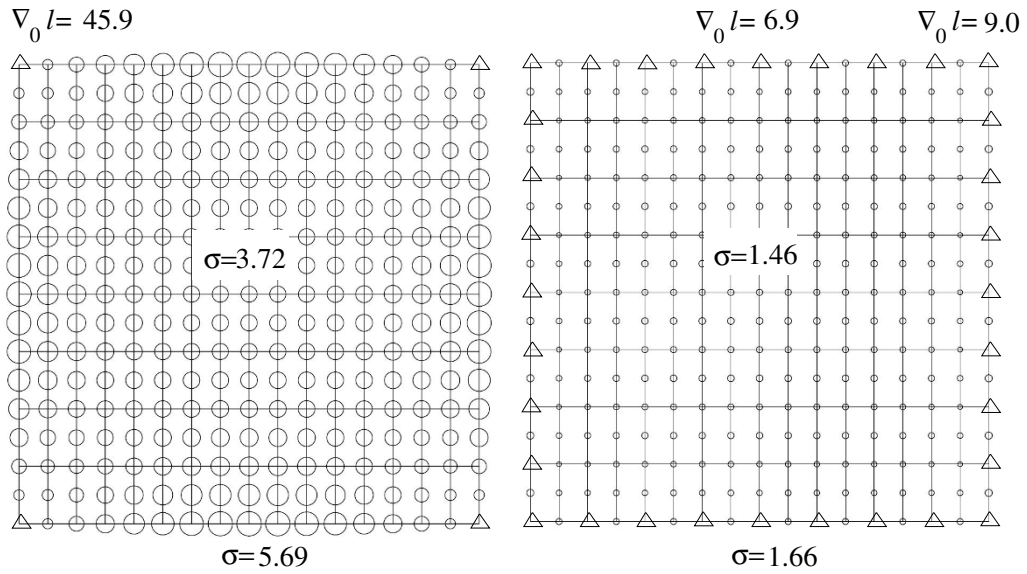


Fig. 15.15 Theoretical quality of image blocks with 8×16 images with six points each. The images scale is one. **Left:** Sparse control at the corners. **Right:** Dense control at the border to achieve a homogeneous precision, required in mobile mapping. Maximum standard deviations $\sigma := \max(\sigma_{\hat{k}})$ of the scene point coordinates in units of the precision of image points $\sigma_{x'}$ occur along the border of the blocks. Lower bounds $\nabla_0 l$ for detectable outliers in control point coordinates

The sensitivity of the resultant coordinates w.r.t. outliers in the control points decreases if we have more observations per image, as then the stability of the block increases and outliers in control points can be detected more easily. For example, if we have 150 observations per image, and only four control points at the corner of a block with 16×32 images, the effect of nondetectable outliers on the result decreases from 90.8 to 10.2 standard deviations.

Since the detectability of outliers in control points highly depends on the configuration and generally is low, it is strongly recommended we perform rigorous testing.

15.4 Self-calibrating Bundle Adjustment

15.4.1 Bundle Adjustment for Perspective Cameras 675
15.4.2 Bundle Adjustment for Spherical Cameras 686
15.4.3 Evaluating Bundle Adjustment Results 687

We now address bundle block adjustment in its general form. Its task is to simultaneously estimate all parameters for (1) the scene features, (2) the poses of the cameras, and (3) the additional parameters for modelling the cameras used. Since it includes the calibration of the cameras, it is called *self-calibrating bundle adjustment*. This setup therefore allows us to capture the intrinsic geometry of the camera at the time of its use.

In principle, the setup allows each image to be taken with an individual camera. Generally, this only leads to reliable results if the overlap between all images is large. In most practical applications this will not be feasible for economic reasons. Therefore, it is advisable to use cameras which are at least stable during the image capture and determine a common set of parameters for each of the cameras used, ideally taking possible changes of the camera parameters into account if the images are taken at different times.

In order to guarantee reliable results we need to (1) perform view planning, (2) carefully select an adequate model for the cameras, and (3) evaluate the final result w.r.t. the prespecified project requirements. Since the structure of the scene has a large impact on

view planning, we discuss this later in Sect. 15.7, p. 715. As the basic model is nonlinear, we need approximate values, the topic of the next section, where also the interleaved problems of outlier detection and sequential estimation are addressed.

We will discuss the basic models for perspective cameras, including linearization, which requires care, since we also want to allow for scene points which are far away or at infinity. We demonstrate the power of applying an optimal statistical method to this geometrically demanding problem by showing how to apply variance component estimation to refine the stochastical model of the observed image coordinates. We generalize the bundle adjustment model to spherical cameras and discuss evaluation of the results in more detail. We refer to the general estimation and evaluation procedures of Chap. 4, p. 75 and the specific aspects discussed in the previous Sect. 15.3, p. 651 on block adjustment. This chapter can be seen as setting the stage for the observation–analysis–modelling loop discussed in the introduction, see Fig. 1.9, p. 10. A generalization of the setup for non-static scenes can be found in Vo et al. (2016).

15.4.1 Bundle Adjustment for Perspective Cameras

The basic model of bundle adjustment is the collinearity equation $\mathbf{x}' = \mathsf{P}\mathbf{X}$, which we already used for optimally estimating the pose of single images, image pairs, or triplets of images. The modelling of deviations from this basic model, in order to capture the imperfectness of real cameras, see Sect. 12.2.3, p. 505, has turned out to be different for perspective and for spherical cameras. The setup of a distortion model within a bundle adjustment requires care, as not all parameters of the interior orientation may be determinable, leading to singular or very unstable normal equation systems.

We first discuss all aspects for perspective cameras, due to their widespread use, and transfer the results to spherical cameras in Sect. 15.4.2, p. 686.

15.4.1.1 The Non-linear Model

For *perspective cameras* we have derived the general projection model in (12.64) and (12.65), p. 479 and provided a specialized version for estimating the projection matrix of a straight line-preserving perspective camera in (12.128), p. 497. We therefore have the following nonlinear Gauss–Markov model for the inhomogeneous coordinates of the image points:

bundle adjustment model for points

$$\mathbb{E}(\underline{\boldsymbol{x}}'_{it}) = \boldsymbol{c}\left(\mathsf{K}_t({}^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{s}_t)\, {}^c\mathsf{P}_t \mathbf{X}_i\right), \qquad (it) \in \mathcal{E} \quad \text{with} \quad \boldsymbol{c}(\mathbf{x}) = \boldsymbol{x}_0/x_h\,. \tag{15.106}$$

This model maps the 3D points $\mathcal{X}_i(\mathbf{X}_i)$ to the points $\mathcal{X}'_{it}(\boldsymbol{x}'_{it})$ observable in the sensor. The perspective projection ${}^c\mathcal{P}_t({}^c\mathsf{P}_t)$, with ${}^c\mathsf{P}_t = R_t[I_3| - \boldsymbol{Z}_t]$ (see (12.17), p. 468), depends on the six parameters in $(R_t, \boldsymbol{Z}_t)$ of the exterior orientation. The general calibration matrices $\mathsf{K}_t({}^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{s}_t)$ are a function of all additional parameters, namely the five parameters $(c_t, x'_{tH}, y'_{tH}, m_t, s_t)$ for a straight line-preserving mapping and the parameters $\boldsymbol{q}_t$ for modelling nonlinear distortions $\Delta\boldsymbol{x}'_{it}({}^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{q}_t)$, see (12.61), p. 478. Observe, the ideal image coordinates of the point $\mathcal{X}'_{it}$ are ${}^i\overline{\boldsymbol{x}}'_{it}$, where the upper left superscript i indicates the image coordinate system, the lower right subscript i indicates the point number. The terms $\Delta\boldsymbol{x}'_{it}$, which model deviations from the ideal perspective projection, depend on these ideal image coordinates. The function $\boldsymbol{c}(\mathbf{x})$ maps the homogeneous coordinates $\mathbf{x} = [\boldsymbol{x}_0^\mathsf{T}, x_h]^\mathsf{T}$ to the inhomogeneous coordinates $\boldsymbol{x}$, see (5.31), p. 206.

If we restrict ourselves to sufficiently close scene points and if the scale difference m between the image coordinates and the skew s are zero, the model can be written as

$$\mathbb{E}({}^i\underline{x}'_{it}) = c_t \frac{r_{t11}(X_i - X_{tO}) + r_{t12}(Y_i - Y_{tO}) + r_{t13}(Z_i - Z_{tO})}{r_{t31}(X_i - X_{tO}) + r_{t32}(Y_i - Y_{tO}) + r_{t33}(Z_i - Z_{tO})} + x'_{tH} + \Delta x'({}^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{q}_t)$$
$$\mathbb{E}({}^i\underline{y}'_{it}) = c_t \frac{r_{t21}(X_i - X_{tO}) + r_{t22}(Y_i - Y_{tO}) + r_{t23}(Z_i - Z_{tO})}{r_{t31}(X_i - X_{tO}) + r_{t32}(Y_i - Y_{tO}) + r_{t33}(Z_i - Z_{tO})} + y'_{tH} + \Delta y'({}^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{q}_t)\,.$$

This model for the bundle adjustment is equivalent to the one used in photogrammetry from the beginning, see (Schmid, 1958) and (12.180), p. 507, but with slight differences: (1) We use error terms $\Delta\boldsymbol{x}'$ instead of correction terms, which is a sign difference; (2) Our terms $\Delta\boldsymbol{x}'$ depend on the ideal image coordinates ${}^i\overline{\boldsymbol{x}}'_{it}$, not on the observed image coordinates; (3) Our model includes the parameters m and s for scale difference and shear, respectively, which in the model (12.180), p. 507 could be included in the correction terms. Due to the possibility of including scene points at infinity and its more compact form, we continue with model (15.106).

bundle adjustment model with lines

Including scene lines into the bundle adjustment is possible with (14.72), p. 638 and (12.79), p. 482 in the form

$$\mathbb{E}(\mathbf{l}'^s_{jt}) = \mathsf{N}\left(\mathsf{Q}_t(\widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{s}}_t)\,\widehat{\mathbf{L}}_j\right) \quad \text{with} \quad \mathsf{Q}_t = (\mathsf{K}_t R_t)^\circ\,[-S(\boldsymbol{Z}_t) \mid I_3] \tag{15.107}$$

for all observed image lines l'_{jt} and with unknown or partially unknown 3D lines $\mathcal{L}_j$.
The model (15.106) covers the following cases:

- A bundle adjustment with additional parameters which are different for each image. This comprises all models of perspective cameras.
  It may be a realistic assumption if nothing is known about the cameras. If possible, it requires careful view planning due to the large set of unknown parameters, namely at least 11 per camera. The model contains the *projective bundle adjustment* as a special case if all cameras are perspective.
- A self-calibrating bundle adjustment with the same additional parameters $\boldsymbol{s}_t = \boldsymbol{s}$ for all cameras, thus with $\mathsf{K}_t = \mathsf{K}$ in the case of perspective cameras.
  This is a realistic model if the same camera has been used for taking all images and the camera can be assumed to be stable during image capture. The calibration parameters may vary if cameras with different interior orientations are used.
- A Euclidean bundle adjustment without additional parameters.
  This is a realistic model if the possibly different cameras used have been properly calibrated beforehand and the calibration parameters are stable and can be applied in the actual situation.

We now analyse the linearization of the nonlinear model.

15.4.1.2 The Linearized Model

Formally the nonlinear model reads as

$$\underline{\boldsymbol{x}}'_{it} + \widehat{\boldsymbol{v}}_{it} = \boldsymbol{f}_{it}(\mathbf{X}_i, \boldsymbol{p}_t, \boldsymbol{s}_t)\,, \quad (it) \in \mathcal{E}\,. \tag{15.108}$$

Here we have

- the observations $\underline{\boldsymbol{x}}'_{it}$, which are the inhomogeneous sensor coordinates of the image points,
- the unknown coordinates $\mathbf{X}_i$ for each scene point, which requires a decision, see below,
- the six transformation parameters $\boldsymbol{p}_t$ per pose t, and
- the additional parameters $\boldsymbol{s}_t$ for modelling the interior orientation of each camera. Omitting the index t indicates the interior orientation of all cameras is the same.

The representation of the scene points in the estimation requires a decision: If all scene points are seen under large parallactic angles, the inhomogeneous coordinates $\boldsymbol{X}_i$ can be

used as unknown parameters. Then there is no danger that scene points are far away or at infinity.

Generally, it is useful to take *the spherically normalized homogeneous coordinates* $\mathbf{X}_i := \mathbf{X}_i^s$ as unknown coordinate parameters, in the following omitting the superscript ^{s}. Then it is of advantage to estimate corrections $\Delta\boldsymbol{X}_{ri}$ for the reduced coordinates of the scene points (see Sect. 10.2.2.1, p. 369) in order to keep the number of unknown parameters small: we only need three parameters for each scene point, corresponding to its degrees of freedom. If we were to estimate corrections to the homogeneous coordinates, we would additionally have the normalization constraint $|\mathbf{X}_i| = 1$ for each scene point. This setup would lead to five unknown parameters per scene point, four for the corrections of the homogeneous coordinates and one for the Lagrangian multiplier which is necessary for each constraint.

We will give the derivations for the general case using reduced coordinates.

linearized model for bundle adjustment

Assuming approximate values $\widehat{\mathbf{X}}^a$ with $|\widehat{\mathbf{X}}^a| = 1$, $\widehat{\boldsymbol{p}}^a$, $\widehat{\boldsymbol{s}}^a$ for the estimates of all parameters, we obtain the linearized model

$$\Delta\boldsymbol{x}'_{it} + \widehat{\boldsymbol{v}}_{it} = \mathsf{C}_{it}\widehat{\Delta\boldsymbol{X}}_{ri} + \mathsf{D}_{it}\widehat{\Delta\boldsymbol{p}}_t + \mathsf{H}_{it}\widehat{\Delta\boldsymbol{s}}_t\,, \tag{15.109}$$

where we use the following expressions:

- The observations $\Delta\boldsymbol{x}'_{it}$ of the linearized model are

$$\Delta\boldsymbol{x}'_{it} := \boldsymbol{x}'_{it} - \widehat{\boldsymbol{x}}'^a_{it}\,, \quad \text{with} \quad \widehat{\boldsymbol{x}}'^a_{it} = \boldsymbol{f}_{it}(\widehat{\mathbf{X}}^a_i, \widehat{\boldsymbol{p}}^a_t, \widehat{\boldsymbol{s}}^a_t)\,. \tag{15.110}$$

- The corrections $\widehat{\Delta\mathbf{X}}_{ri}$ to the reduced scene coordinates are related to the corrections $\widehat{\Delta\mathbf{X}}_i$ of the homogeneous coordinates by

$$\widehat{\Delta\mathbf{X}}_i = J_r(\widehat{\mathbf{X}}^a_i)\,\widehat{\Delta\mathbf{X}}_{ri} \quad \text{with} \quad J_r(\widehat{\mathbf{X}}^a_i) = \mathsf{null}(\widehat{\mathbf{X}}^{a\mathsf{T}}_i) \tag{15.111}$$

  and (10.26), p. 370.
- Therefore we obtain the Jacobian w.r.t. the unknown reduced scene coordinates,

$$\mathsf{C}_{it} = \left.\frac{\partial \boldsymbol{f}_{it}(\mathbf{X}_i, \boldsymbol{p}_t, \boldsymbol{s}_t)}{\partial \boldsymbol{X}_{ri}}\right|_{\widehat{X}=\widehat{X}^a, \widehat{p}=\widehat{p}^a, \widehat{s}=\widehat{s}^a} = \left.\frac{\partial \boldsymbol{f}_{it}(\mathbf{X}_i, \boldsymbol{p}_t, \boldsymbol{s}_t)}{\partial \mathbf{X}_i}\, J_r(\mathbf{X}_i)\right|_{\widehat{X}=\widehat{X}^a, \widehat{p}=\widehat{p}^a, \widehat{s}=\widehat{s}^a}\,, \tag{15.112}$$

  evaluated at the approximate values.

  Formally, the scene coordinates $\mathbf{X}_i$ appear twice in (15.106), p. 675: once directly, as to be projected using $^c\mathsf{P}$, but also hidden, namely in $^i\overline{\boldsymbol{x}}'_{it}$ in the terms $\Delta\boldsymbol{x}'(^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{s}_t)$. If the distortions $\Delta\boldsymbol{x}'_{it}$ do not change much with χ'_{it}, this dependency on the scene points can be neglected for all lenses, except for extreme wide angle lenses or fish-eye lenses, which may be better modelled as spherical cameras.

  Therefore, in the following we work with the Jacobian

$$\mathsf{C}_{it} \approx \frac{\partial \boldsymbol{c}(\widehat{\mathbf{x}}'^a_{it})}{\partial \widehat{\mathbf{x}}'^a_{it}} \frac{\partial \widehat{\mathbf{x}}'^a_{it}}{\partial \widehat{\mathbf{X}}_{ri}} = \frac{1}{\widehat{x}'^a_{it,h}} \left[I_2 \mid -\widehat{\boldsymbol{x}}'^a_{it,0}\right] \widehat{\mathsf{P}}^a_t\, J_r(\widehat{\mathbf{X}}^a_i)\,, \tag{15.113}$$

  with $\widehat{\mathbf{x}}'^a_{it} = \widehat{\mathsf{P}}^a_t \widehat{\mathbf{X}}^a_i$, the partitioning $\mathbf{x} = [\boldsymbol{x}_0^\mathsf{T}, x_h]^\mathsf{T}$, and using $J_c(\mathbf{x}) = \partial\boldsymbol{x}/\partial\mathbf{x}$ from (12.129), p. 497.

  This expression is independent of the parametrization of the projection matrix $\mathsf{P}_t = \mathsf{K}_t(^i\overline{\boldsymbol{x}}'_{it}, \boldsymbol{s}_t)\,{}^c\mathsf{P}_t$.
- The Jacobian w.r.t. the transformation parameters

$$\mathsf{D}_{it} = \left.\frac{\partial \boldsymbol{f}_{it}(\mathbf{X}_i, \boldsymbol{p}_t, \boldsymbol{s}_t)}{\partial \boldsymbol{p}_t}\right|_{\widehat{X}=\widehat{X}^a, \widehat{p}=\widehat{p}^a, \widehat{s}=\widehat{s}^a} \tag{15.114}$$

is to be evaluated at the approximate values. Here the same argument allows us to use the approximation

$$D_{it} \approx \frac{\partial \boldsymbol{c}(\widehat{\mathbf{x}}'^a_{it})}{\partial \widehat{\mathbf{x}}'^a_{it}} \frac{\partial \widehat{\mathbf{x}}'^a_{it}}{\partial \widehat{\boldsymbol{p}}_i} \,. \tag{15.115}$$

The Jacobian $\partial \widehat{\mathbf{x}}'^a_{it} / \partial \widehat{\boldsymbol{p}}_i$ depends on the parametrization of the projection, as already seen in Sect. 12.2.2.2, p. 496.

- Finally, we have the Jacobian w.r.t. the additional parameters,

$$H_{it} = \left. \frac{\partial \boldsymbol{f}_{it}(\mathbf{X}_i, \boldsymbol{p}_t, \boldsymbol{s}_t)}{\partial \boldsymbol{s}_t} \right|_{\widehat{X}=\widehat{X}^a, \widehat{p}=\widehat{p}^a, \widehat{s}=\widehat{s}^a} \,, \tag{15.116}$$

evaluated at the approximate values. The Jacobian explicitly is

$$H_{it} = \frac{\partial \boldsymbol{c}(\widehat{\mathbf{x}}'^a_{it})}{\partial \widehat{\mathbf{x}}'^a_{it}} \frac{\partial \widehat{\mathbf{x}}'^a_{it}}{\partial \boldsymbol{s}_t} \,. \tag{15.117}$$

The first factor is $J_c(\mathbf{x}) = \partial \boldsymbol{x} / \partial \mathbf{x}$, from (12.129), p. 497. With the additional parameters $\boldsymbol{s}_t = [c, x'_H, y'_H, m, s, \boldsymbol{q}]^\mathsf{T}_t$, we obtain the second factor,

$$\frac{\partial \widehat{\mathbf{x}}'^a_{it}}{\partial \boldsymbol{s}_t} = \left[\begin{array}{c|c|c|c|c} {}^c x'_{it} + s_t\, {}^c y'_{it} & 1 & 0 & 0 & c_t\, {}^c y'_{it} \\ (1+m_t) {}^c y'_{it} & 0 & 1 & c_t\, {}^c y'_{it} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right| \left. \begin{array}{c} \partial \Delta \widehat{x}'_{it} / \partial \boldsymbol{q}_t \\ \partial \Delta \widehat{y}'_{it} / \partial \boldsymbol{q}_t \\ 0 \end{array} \right]^a \,, \tag{15.118}$$

to be evaluated at the approximate values again with $\widehat{\mathbf{x}}'^a_{it} = \widehat{\mathsf{P}}'^a_t \widehat{\mathbf{X}}^a_i$, see (12.158), p. 501. If the additive corrections $\Delta \boldsymbol{x}'_{it}$ are polynomials or other basis functions $\boldsymbol{b}_k({}^i\widehat{\overline{\boldsymbol{x}}}'^a_{it})$ linear in the parameters, e.g.,

$$\begin{bmatrix} \Delta x'_{it}({}^i\overline{\boldsymbol{x}}'^a_{it}, \boldsymbol{s}_t) \\ \Delta y'_{it}({}^i\overline{\boldsymbol{x}}'^a_{it}, \boldsymbol{s}_t) \end{bmatrix} = \sum_{k=1}^{K} \begin{bmatrix} q_{t,x,k}\; b_{x,k}({}^i\overline{\boldsymbol{x}}'^a_{it}) \\ q_{t,y,k}\; b_{y,k}({}^i\overline{\boldsymbol{x}}'^a_{it}) \end{bmatrix} \,, \tag{15.119}$$

then the Jacobian $\partial \Delta \widehat{\boldsymbol{x}}'^a_{it} / \partial \boldsymbol{q}_t$ is a $2 \times Q$ matrix, where Q is the number of parameters in $\boldsymbol{q}_t$ and the entries are the basis functions $b_{x,k}({}^i\overline{\boldsymbol{x}}'^a_{it})$ and $b_{y,k}({}^i\overline{\boldsymbol{x}}'^a_{it})$, again to be evaluated at the approximate values ${}^i\widehat{\overline{\boldsymbol{x}}}'^a_{it}$ for the image coordinates.

We now collect the observations of the linearized model and the unknown corrections to the reduced coordinates in the vectors,

$$\Delta \boldsymbol{l} := [\Delta \boldsymbol{x}'_{it}] \quad \text{and} \quad \Delta \boldsymbol{k} := [\Delta \boldsymbol{X}_{ri}] \,, \tag{15.120}$$

where the transition to the parameters $\Delta \boldsymbol{k}$ simplifies notation within the estimation. We hence achieve the Gauss–Markov model,

$$\Delta \boldsymbol{l} + \widehat{\boldsymbol{v}} = A \widehat{\Delta \boldsymbol{x}} = [C\ D\ H] \begin{bmatrix} \widehat{\Delta \boldsymbol{k}} \\ \widehat{\Delta \boldsymbol{p}} \\ \widehat{\Delta \boldsymbol{s}} \end{bmatrix} \,, \quad \mathbb{D}(\underline{\Delta \boldsymbol{l}}) = \mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll} \,. \tag{15.121}$$

The normal equations can be written in the compact form $N \Delta \widehat{\boldsymbol{x}} = \boldsymbol{n}$, with

$$N = \begin{bmatrix} N_{kk} & N_{kp} & N_{ks} \\ N_{pk} & N_{pp} & N_{ps} \\ N_{sk} & N_{sp} & N_{ss} \end{bmatrix} = \begin{bmatrix} C^\mathsf{T} W_{ll} C & C^\mathsf{T} W_{ll} D & C^\mathsf{T} W_{ll} H \\ D^\mathsf{T} W_{ll} C & D^\mathsf{T} W_{ll} D & D^\mathsf{T} W_{ll} H \\ H^\mathsf{T} W_{ll} C & H^\mathsf{T} W_{ll} D & H^\mathsf{T} W_{ll} H \end{bmatrix} \tag{15.122}$$

and

$$\boldsymbol{n} = \begin{bmatrix} \boldsymbol{n}_k \\ \boldsymbol{n}_p \\ \boldsymbol{n}_s \end{bmatrix} = \begin{bmatrix} C^\mathsf{T} W_{ll} \Delta \boldsymbol{l} \\ D^\mathsf{T} W_{ll} \Delta \boldsymbol{l} \\ H^\mathsf{T} W_{ll} \Delta \boldsymbol{l} \end{bmatrix} \,. \tag{15.123}$$

As discussed in the previous section, 15.3.3.1, p. 657, the two matrices $\mathbf{N}_{kk}$ and $\mathbf{N}_{pp}$ are block diagonal or, after proper conditioning, diagonal matrices if the observations have a block diagonal structured covariance matrix.

The solution of the normal equation system leads to corrections to the parameters, which may then be used as new approximate values until convergence is reached,

$$\begin{bmatrix} \widehat{\mathbf{X}}_i \\ \widehat{\boldsymbol{p}}_t \\ \widehat{\boldsymbol{s}}_t \end{bmatrix}^{(\nu+1)} = \begin{bmatrix} \mathsf{N}\left(\widehat{\mathbf{X}}_i^{(\nu)} + J_r(\widehat{\mathbf{X}}_i^{(\nu)})\,\widehat{\Delta\boldsymbol{X}}_{ri}\right) \\ \widehat{\boldsymbol{p}}_t^{(\nu)} + \widehat{\Delta\boldsymbol{p}}_t^{(\nu)} \\ \widehat{\boldsymbol{s}}_t^{(\nu)} + \widehat{\Delta\boldsymbol{s}}_t^{(\nu)} \end{bmatrix}, \tag{15.124}$$

using the reduced coordinates of the scene points, $\widehat{\Delta\boldsymbol{X}}_{ri} := \widehat{\Delta\boldsymbol{k}}_i$, from the estimation (15.121).

We will regularly use the normal equation system reduced by one of the parameter sets, which due to the block diagonal structure of $\mathbf{N}_{kk}$ and $\mathbf{N}_{pp}$ can be performed efficiently, see 4.2.6, p. 94. This reduction will be by $\widehat{\Delta\boldsymbol{k}}$ or $\widehat{\Delta\boldsymbol{p}}$, depending on the scope of the analysis. The reduced submatrices will now have a superscript to indicate the eliminated set of parameters. As an example, when reducing $\mathbf{N}_{ps}$ by the coordinates, we have

$$\overline{N}_{ps}^{(k)} = N_{ps} - N_{pk} N_{kk}^{-1} N_{ks}\,. \tag{15.125}$$

Sometimes we use reduced design matrices

$$\overline{D}^{(k)} = D - C N_{kk}^{-1} N_{kp} = (I - C(C^{\mathsf{T}} W_{ll} C)^{-1} C^{\mathsf{T}} W_{ll}) D \tag{15.126}$$

or

$$\overline{H}^{(k)} = H - C N_{kk}^{-1} N_{ks} = (I - C(C^{\mathsf{T}} W_{ll} C)^{-1} C^{\mathsf{T}} W_{ll}) H\,. \tag{15.127}$$

Then we also have

$$\overline{N}_{ps}^{(k)} = \overline{D}^{(k),\mathsf{T}} W_{ll} \overline{H}^{(k)}\,. \tag{15.128}$$

All these matrices can easily be expressed as functions of the individual partial derivatives $C_{k_i p_t}$ and $D_{k_i p_t}$, as discussed in the previous section.

15.4.1.3 Variance Analysis in a Free Bundle Adjustment

The following example presents the analysis of the residuals of a free bundle adjustment in order to arrive at a more refined model for the uncertainty of the given observations. In particular, we show how the standard deviation of the image coordinates of automatically detected Lowe key points (Lowe, 2004) depends on the scale of their detection, see Sect. 12.2.1, p. 490.

The characteristics of the block with 70 images are given in Table 15.4. A few representative images are shown in Fig. 15.16

Table 15.4 Characteristics of the bundle block

Camera	
Image size [pixel]	2 048 × 2 448
Principal distance c [pixel]	1 591
Number of images	70
Number of observed image points	226 537
Number of relative orientations	376
Number of estimated scene points	63 104
Redundancy	263 349
Estimated $\widehat{\sigma}_0$ [1]	0.3631

Fig. 15.16 Images 11, 21, 34, 61, and 66 of the image sequence. **Top row:** Original images. **Bottom row:** Rectified images using the calibration parameters of the camera

The camera `uEyeSE` with 2/3 inch chip and wide angle lens `Lensagon CY0614S23` was calibrated beforehand using the method described in the next section. The set of parameters were K_1' and K_2' from Brown's model, see (12.175), p. 506. The images were rectified (see Fig. 15.16, lower row) such that the calibration matrix $\mathsf{K} = \mathrm{Diag}([c, c, 0])$ could be applied, with the principal distance $c = 1\,591$ pixel resulting from the calibration. Image points were automatically detected using the SIFT operator of Lowe (2004). It detects image points together with their characteristic scale s and a descriptor. The scale can be interpreted as a blurring factor. Thus points with a large scale can be thought of as being detected in an image with a pixel size larger by the factor s. Thus for each detected and measured image point we obtain $(x', y', s)_{it}$. The descriptor characterizes the image around the point and is used to automatically establish correspondence between points of different images. This way each point in each image is assigned the number i of the scene point it belongs to. This correspondence may be erroneous, which is why we have to expect outliers. Using pairs of relative orientations and the technique described in Sect. 15.6.2.2, p. 710, we determine approximate values and perform a robust bundle adjustment without additional parameters. For the $N = 453\,074$ coordinates of the image points we assume an a priori standard deviation of $\sigma_{x'} = \sigma_{y'} = 1$ pixel.

As a result of the bundle adjustment, besides the orientation parameters of all 70 images, we obtain the 3D coordinates of 63 104 scene points. We will use them to derive a surface model of the facade in Sect. 16.4, p. 757. The gauge was chosen by fixing the coordinate system of the first image as the reference system and by fixing the distance between the first two projection centres. The residuals, which are decisive for our analysis, are invariant to the choice of the gauge.

As an indicator for the fit of the assumed model and the given observations, we use the estimated variance factor,

$$\widehat{\sigma}_0^2 = \frac{\widehat{\boldsymbol{v}}^\mathsf{T} \boldsymbol{W}_{ll} \widehat{\boldsymbol{v}}}{R} = 0.363^2\,, \tag{15.129}$$

with the redundancy $R = 263\,349$. Since the a priori standard deviations are assumed to be one pixel, this indicates the automatically detected key points on average have a standard deviation of approximately 0.36 pixel.

The assumed stochastical model, which stipulates that all observations have the same accuracy, is very simple. We can expect deviations from this simple model. The most intuitive one is the following: The standard deviation of the coordinates $\underline{\boldsymbol{x}}'_{it}$ increases with the scale s_{it}. This is reasonable, as the detection of points in blurred images, or in images with larger pixel size, is likely to be worse. Therefore a reasonable model for the noise of the observed image coordinates is

$$\underline{n}_{it} = \underline{a}_{it} + s_{it}\underline{b}_{it}\,, \tag{15.130}$$

where $\underline{a}_{it} \sim \mathcal{M}(0, \sigma_a^2)$ and $\underline{b}_{it} \sim \mathcal{M}(0, \sigma_b^2)$ are two mutually independent noise sources, where the first is independent of the scale s_{it} and the second reflects the expected loss in

accuracy with an increasing scale. Therefore we can assume the variance model

keypoint scale dependent variance model

$$\sigma_{x'_{it}}^2 = \sigma_a^2 + \sigma_b^2 \, s_{it}^2 \,. \tag{15.131}$$

We will determine the two variances σ_a^2 and σ_b^2 from the residuals of the bundle adjustment.

The model suggests performing a variance component estimation, see Sect. 4.2.4, p. 91, especially using (4.92), p. 92. Variance component estimation is based on the residuals and can be seen as fitting a model such that the normalized residuals $\widehat{\boldsymbol{v}}_n / \sigma_{\widehat{v}_n} = \widehat{\boldsymbol{v}}_n / (\sigma_{l_n} \sqrt{r_n})$ follow a Gaussian distribution. For this we would need at least parts of the inverse of the normal equation matrix to determine the redundancy numbers r_n in Eqs. (4.99), p. 92, see (4.69), p. 88. For our analysis, we approximate the redundancy numbers r_{it} for each observation by the average redundancy number for each triangulated 3D point, see (4.70), p. 88,

$$r_{it} \approx \frac{2I_i - 3}{2I_i} \,, \quad t \in \mathcal{T}_i \,, \tag{15.132}$$

for all images t used to determine the 3D point i. Here I_i is the number of rays used to determine the point $\mathcal{X}_i$. This value is the average redundancy number of all image coordinates $\mathcal{x}_{it}$ referring to the scene point $\mathcal{X}_i$ given the assumption that the orientation parameters are fixed values. This is justified, as the number of points per image is quite large, namely > 2500. We further assume the geometric configuration for the triangulation of the point $\mathcal{X}_i$ to be homogeneous w.r.t. the image rays, which is a reasonable assumption for points visible in three or more images.

To investigate the validity of the model (15.131), we partition the data into B equally filled scale bins $[s_b, s_{b+1}], B = 1, ..., 30$, robustly estimate the variance of the residuals using the MAD (see (4.370), p. 146) in these bins, and fit the model through these estimates.

We obtain the result shown in Fig. 15.17. We used 30 bins. The variances obviously

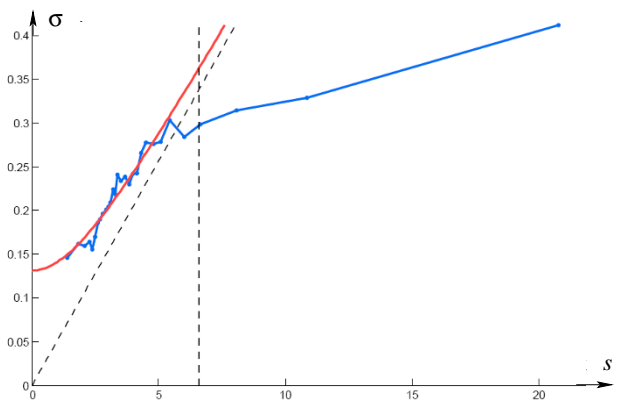


Fig. 15.17 Standard deviations $\sigma_{x'}(s)$ of image coordinates as a function of the scale s [pixel] of the Lowe key points. Polygon (in blue, if seen in colour): Standard deviations from 30 equally filled bins. Curve (in red, if seen in colour): Fitted model $\sigma_{x'}(s) = \sqrt{(0.13)^2 + (0.05\,s)^2}$ for scales between $s = 1.4$ and $s = 6.5$ (vertical dashed line). The standard deviation for points with small scale is below 0.15 [pixel]. The standard deviation increases with the scale by $0.05\,s$ (sloped dashed line)

increase with increasing scale. For small scales, the increase is almost linear with the scale. However, for scales larger than $s = 6.5$, the increase is significantly smaller. This is caused by the fact that the applied outlier detection procedure uses a fixed critical value, independent of the scale. As a consequence, the distribution of the residuals of points with a large scale is heavily truncated. This leads to the observed bias towards smaller estimated standard deviations. Fitting the assumed model to the estimated standard deviations $\widehat{\sigma}_{x'}(s)$ results in a smooth curve, a hyperbola: Within the scale range used, $s \in [1.4, 6.5]$, it appears to be a good model, taking into account that each bin contains approximately 7,000 image points.

The practical result is the following: The key point detector by Lowe is able to locate points with a standard deviation of approximately $\sigma_{x'} = 0.15$ pixel at high resolution. For scales s larger than 5, the standard deviation can be approximated by $\sigma_{x'} = 0.05\,s$ pixel. This result of course refers to the images of the experiment, and may vary to some extent for other images.

Using the scale-dependent standard deviations $\sigma_{x'}(s)$, we can predict that the square-root of the estimated variance factor becomes approximately 0.21 pixel compared to 0.36 pixel, which is a reduction by a factor of $f = 1.7$, see the proof below. This accuracy increase can be expected for all parameters when using more realistic standard deviations for the image coordinates.

The gain in precision can be predicted if only the precision structure of the observations is given, the geometric configuration is homogeneous, and the relative redundancy R/N is large enough, say beyond 2, as in our example.

This prediction uses a simplified substitute model, here for the triangulation, namely the weighted mean of the observations, which gives an indication of how the uncertainty of the XY-coordinates of the scene points change when we change the variances of the observations from the approximate model to a more refined model.

Proof: The Gauss–Markov model for the mean of values z_n is given by $\boldsymbol{z} \sim \mathcal{M}(\mathbf{1}\mu_z, \boldsymbol{\Sigma})$, with $\mathbf{1} = [1, 1, ..., 1]^\mathsf{T}$. Then the weighted arithmetic mean $\widehat{\mu}_z$ of the image coordinates $\{z_n\} := \{x'_{it}\}$, say only of the x'-coordinates, with their variances $\sigma_n^2 := \sigma_{x'_{it}}$, is given by

$$\widehat{\mu}_z = \left(\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{1}\right)^{-1}\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{z} = \frac{\sum_t w_n z_n}{\sum_t w_n} \quad \text{with} \quad \mathsf{W} = \mathrm{Diag}([w_n]) = \mathrm{Diag}([1/\sigma_n^2]) = \boldsymbol{\Sigma}^{-1}\,. \tag{15.133}$$

Its variance for independent observations is

$$\sigma^2_{\widehat{\mu}_z} = \left(\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\,\mathbf{1}\right)^{-1} = \frac{1}{\sum_n \frac{1}{\sigma_n^2}} = \frac{\overline{\sigma^2_{z_n}}}{N}\,, \tag{15.134}$$

where $\overline{\sigma^2_{z_n}}$ is the harmonic mean of the variances $\sigma^2_{z_n}$. We now determine the variance of $\widehat{\mu}_z$ for two choices of $\boldsymbol{\Sigma}$.

In our example, we first we assume all observations z_n to have the same standard deviation $\sigma_n^{(1)} = 0.363$ pixel. Second, we assume the observations z_n to have a scale-dependent standard deviation, thus $\sigma_n^{(2)} := \sigma_{x'_{it}}(s)$. We then have

$$f = \frac{\sigma^{(1)}_{\widehat{\mu}_z}}{\sigma^{(2)}_{\widehat{\mu}_z}} = \sqrt{\frac{(\mathbf{1}^\mathsf{T}(\boldsymbol{\Sigma}^{(1)})^{-1}\,\mathbf{1})^{-1}}{(\mathbf{1}^\mathsf{T}(\boldsymbol{\Sigma}^{(2)})^{-1}\,\mathbf{1})^{-1}}} = \sqrt{\frac{(0.363\ [\text{pixel}])^2}{\overline{\sigma^2_{z_n}}}}\,. \tag{15.135}$$

Assuming we have $N = B$ observations, with the estimated standard deviations taken from the B bins of the variance estimation, we obtain, with $B = 30$,

$$\overline{\sigma_z} = \sqrt{\frac{B}{\sum_{s\in\{s_1,\ldots,s_B\}} \frac{1}{\sigma_z^2(s)}}} = 0.21\ [\text{pixel}]\,. \tag{15.136}$$

This approximates the average situation for estimating the XY-coordinates of a scene point. Assuming an average parallactic angle between the rays to a scene point, the Z-coordinates are less precise than the XY-coordinates a by constant factor. Thus the argumentation is valid for all three coordinates. ◇

15.4.1.4 Empirical Accuracy of Bundle Adjustment for Topographic Mapping

The following external evaluation of bundle adjustment is taken from Cramer (2010) and Jacobsen et al. (2010). The test site Vaihingen/Enz, close to Stuttgart, Germany, covers an area of 7.4 × 4.7 km^2 area. Data are captured with several cameras and two ground sampling distances of 20 cm and 8 cm. We report results for three cameras. Their characteristics of the different flights are given in Table 15.5. The image points were automatically measured and brought into correspondence using the software system MATCH-AT. The

number of unknown scene points for the six different cases is between 4 700 and 3 000, the redundancy of the bundle adjustment is between 44 000 and 630 000. The maximum number of rays per scene point is between 12 and 33, indicating a high overlap which in all cases was at least 60% in both directions. The control and the check points were manually measured with an accuracy of approximately 0.25 pixel. In all cases four control points were used in the corners of the block. The self-calibrating bundle adjustment used the 12 parameters of Ebner (1976) and the 44 parameters of Grün (1978), cf. Sect. 12.2.3.5, p. 509. The table provides the theoretical precision of the 3D points derived from the inverse normal equation matrix of the bundle adjustment. It is the Cramer–Rao bound for the achievable precision, cf. Sect. 4.2.2.2, p. 86. If it is reached by a comparison to

	DMC			Ultracam X			DigiCAMx4		
sensor size [pixel]	7 680×13 824			9 420×14 430			7 216×5 412		
pixel size [μm]	12			7.2			6.8		
principal distance [mm]/[pixel]	120.00/10 000			100.50/13 958			82.00/12 058		
# add. parameters	44			44			12		
GSD 20 [cm]									
flight height [m]	2 160			2 900			2 500		
number of images	42			36			132		
max # of rays/scene point	12			12			18		
end/sidelap [%]	60/60			75/70			62/70		
# check points	180			180			161		
	X	Y	Z	X	Y	Z	X	Y	Z
theoretical precision [m]	0.027	0.039	0.093	0.021	0.029	0.085	0.032	0.044	0.101
empirical accuracy [m]	0.040	0.066	0.108	0.059	0.060	0.154	0.052	0.058	0.131
ratio emp./theor. [1]	1.48	1.69	1.16	2.81	2.07	1.81	1.62	1.31	1.30
GSD 8 [cm]									
flight height [m]	870			1 200			1 060		
number of images	110			175			640		
max # of rays/scene point	13			28			33		
end/sidelap	60/63			75/70			80/70		
# check points	113			111			114		
	X	Y	Z	X	Y	Z	X	Y	Z
theoretical precision [m]	0.012	0.015	0.028	0.008	0.011	0.026	0.009	0.011	0.025
empirical accuracy [m]	0.028	0.044	0.054	0.060	0.025	0.044	0.033	0.039	0.057
ratio emp./theor. [1]	2.33	2.93	1.93	7.50	2.27	1.69	3.67	3.54	2.28

Table 15.5 Empirical accuracy of self-calibrating bundle adjustment achieved with three cameras and two ground sampling distances (GSD). Cameras: DMC from Intergraph, Ultracam X from Vexcel Imaging, DigiCAMx4 from IGI (a camera system with four mutually fixed cameras). Theoretical precision: standard deviation of estimated 3D points from bundle adjustment (internal precision). Empirical accuracy: root mean square errors from differences to ground truth (external accuracy), from Cramer (2010) and Jacobsen et al. (2010)

ground truth, the internal accuracy potential of the data is exploited. The comparison of the 3D coordinates of check points, measured with differential GPS, yields the empirical accuracy in terms of root mean square errors.

For the larger ground sampling distance of GSD= 20 cm the achieved accuracy is in the range of 1/4 of the ground sampling distance in planimetry and approximately 1/2 ground sampling distance in the height. It is in good agreement with the theoretical expectation. The differences can be explained by the accuracy of the reference data, which were determined using differential GPS: they have an accuracy of 0.01 m in planimetry and 0.02 m in height, which is already close to the accuracy obtained when having ground sampling distances of 8 cm. According to the authors also the systematic errors may not be completely modelled. Therefore the empirical accuracies for the smaller ground sampling distances do not quite fit to the theoretical standard deviations, but still allow for an improvement of up to a factor of 2 approximately. This result can be generalized if the guidelines for flight planning are fulfilled, which we discuss in Sect. 15.7, p. 715.

15.4.1.5 On the Choice of Additional Parameters

This section addresses the general choice of additional correction terms for modelling the interior orientation within a self-calibrating bundle adjustment.

At a minimum, between the following situations need to be distinguished:

- If the exterior orientation of the images is *observed externally* with sufficient accuracy, e.g., using an integrated measuring unit with a GPS and an inertial system, the correction terms can be modelled with the complete set of basis functions, of course up to that order, which captures the effects of the lens and reaches the envisaged accuracy, see the discussion in Blazquez and Colomina (2010) and Tang et al. (2012).
- If the exterior orientation of the images is not observed, the correction terms should *not model the same effect on the image coordinates as the exterior orientation*. In the extreme case of a flat scene observed in a single image as discussed in Sect. 12.2.2.3, p. 502, not even the principal distance and the principal point can be estimated. We will discuss this in more detail.
- If the parameters of the *calibration matrix* are part of the model, the additional correction terms should of course not model those effects already modelled by the calibration matrix. This is the reason why in Table (12.7), p. 464 we distinguish between the set $\boldsymbol{s}$ of all additional parameters and the set $\boldsymbol{q}$ of those additional parameters not covered in the calibration matrix.
- If the parameters of the exterior orientation are not determined externally with sufficient accuracy and the scene is flat, the three parameters c, x'_H, and y'_H for the principal distance and the principal point cannot be included.

Though we have presented quite some information on recovering the scene structure or the poses of the cameras, the questions remain about (1) whether to choose the physically motivated model by Brown (12.175), p. 506 or the phenomenologically motivated orthogonal multivariate polynomials, e.g., of (12.197), p. 511, and (2) what order should to choose for the polynomials.

These questions have two aspects:

1. The model should be able to compensate for the real distortions in order to have the smallest possible deviation between the observed data and the model, which may be measured using the a posteriori variance factor $\widehat{\sigma}_0^2$. However, this should not happen at the cost of too many additional parameters. This can be seen as a question of model selection following Sect. 4.6.7, p. 138. Therefore it is useful to compare the usefulness of competing models using a model selection criterion such as the Bayesian information criterion (BIC), cf. (4.351), p. 139.
2. The model should also allow for a reliable estimation of the scene or pose parameters. Thus uncertainty in the estimated calibration parameters should not lead to an uncertainty of the scene and pose parameters. This may be analysed using the sensitivity factors introduced in Sect. 4.6.5.3, p. 134. We will discuss this aspect within our context of bundle adjustment in more detail in Sect. 15.4.3, p. 687, especially in Sect. 15.4.3.4, p. 693.

Using self-calibrating bundle adjustment for camera calibration is the topic of Sect. 15.5, p. 696. We first give an example for the selection of a distortion model and then discuss methods for posthoc analysis of the bundle adjustment, which can be used to check whether the result of a bundle adjustment can be trusted.

15.4.1.6 Example of Self-calibration

The following example is meant to demonstrate the choice of a calibration model. The model selection is based on the Bayesian information criterion, cf. Sect. 4.6.7, p. 138.

We calibrate two cameras using a test field with well-defined circular targets, see Fig. 15.18. The targets can easily be identified in the images. The borders of the elliptical images

of the circular targets are automatically detected and used to determine the best estimate for the image of the centre of the 3D targets, cf. Sect. 12.3.5, p. 534. Its coordinates are

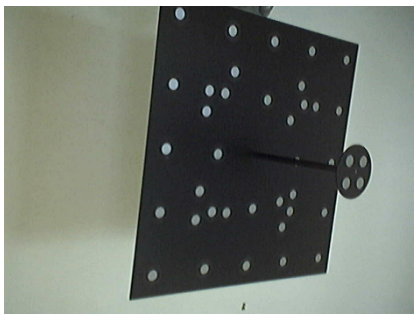


Fig. 15.18 Image of a test field. It consists of 40 circular targets, 36 in a plane and four on a stamp. The arrangement of the targets allows an automatic identification in the images

used in a self-calibrating bundle adjustment for deriving the parameters of a calibration model. The 3D coordinates of the targets are approximately known, which is useful for determining approximate values for the orientation parameters using a DLT.

The Casio camera is calibrated with 36 images using the highest resolution, i.e., images with 3 648 × 2 736 pixel, cf Table 15.6. One camera of the Ladybug 3 camera system is calibrated with 14 images using half the original resolution, i.e., the images have 800 × 600 pixel. We will refer to this single camera as Ladybug 3 camera in the following.

Table 15.6 Characteristics of the calibrated cameras and result of self-calibration. The images were calibrated using Brown's model with two radial and – for the Casio camera – two tangential parameters and the third-order Tschebyscheff polynomials. The Bayesian information criterion BIC clearly suggests the Tschebyscheff model is superior for the Casio camera EX-S10 compared to the radial model of Brown, however only slightly superior to the combined radial and tangential model. It slightly favours Brown's model for the Ladybug 3 camera

		Ladybug 3		Casio EX-S10		
1	principal distance [pixel]	290		3 683		
2	image size [pixel]	800 × 600		3 648 × 2 736		
3	number of images used	14		36		
		Tschebyscheff	Brown: r	Tschebyscheff	Brown: r	Brown: r/t
4	number of additional parameters	17	6	17	6	8
5	number of unknowns	221	210	375	364	366
6	number of observations	736	822	2 114	2 298	2 298
7	estimated $\widehat{\sigma}_0$ ($\widehat{\sigma}_0^2$) [1]	1.94 (3.76)	0.79 (0.624)	4.86 (23.6)	12.5 (156.7)	5.27 (27.8)
7	BIC	2 982	2 393	27 215	161 130	33 877

The comparison of the two camera models follows Sect. 12.1.4, p. 476. The first model adopts polynomials up to third-order, specifically Tschebyscheff polynomials; for details cf. Sect. 12.2.3.5, p. 509. The second model is Brown's model with only radial distortion parameters K_1 and K_2, cf. (12.175), p. 506. The parameters were determined using a self-calibrating bundle adjustment.

Figure 15.19 shows the residuals after the self-calibrating bundle adjustment overlayed in one image. They are magnified by a factor of 100 for the Ladybug 3 camera and by a factor of 50 for the Casio camera. The estimated standard deviation $\widehat{\sigma}_0$ and the variance factors $\widehat{\sigma}_0^2$ are shown in Table 15.6 in line 5.

Visual inspection shows the residual patterns to be partly random and partly systematic. Random patterns indicate that the adopted calibration model is effective, whereas systematic residual patterns indicate that the model does not cover all causes.

We use the variance factor $\widehat{\sigma}_0^2$ for comparing the different results. When using the *Bayesian information criterion* $\text{BIC} = R\widehat{\sigma}_0^2 + {}^1\!/_2 U \log N$, we arrive at the same conclusions.

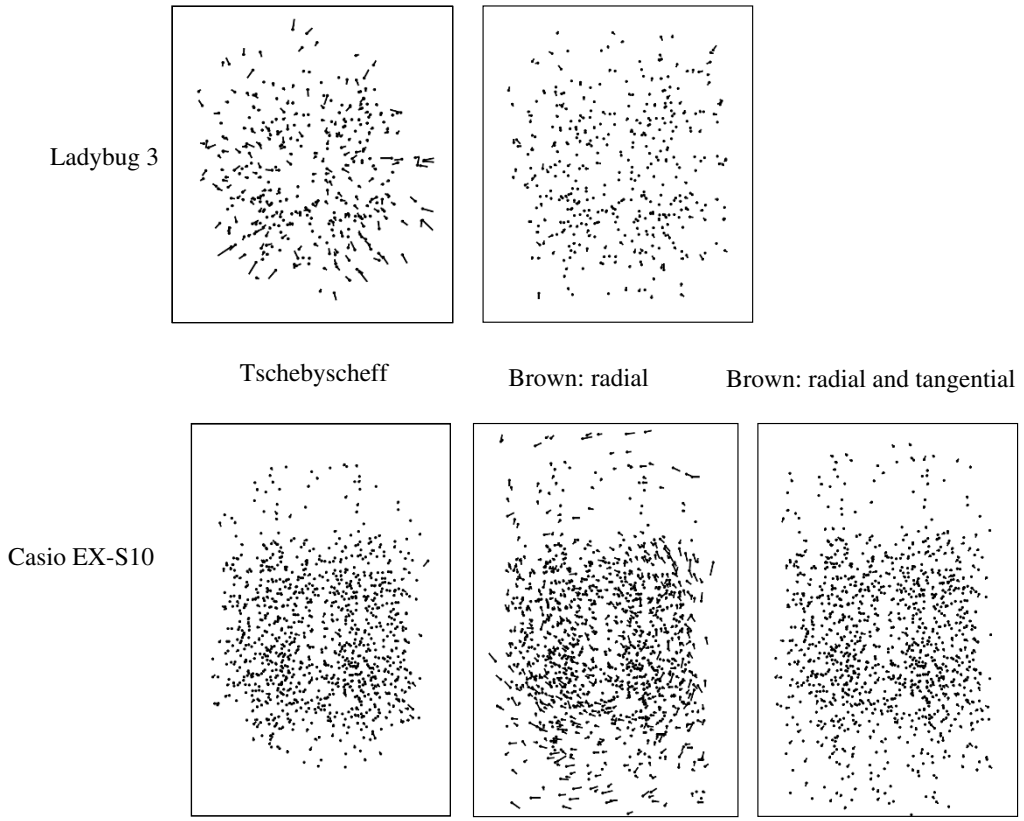


Fig. 15.19 Residuals after calibration. The residuals are magnified by a factor of 100 and 50 for the cameras Ladybug 3 and Casio, respectively. The residual patterns confirm the numerical analysis using the Bayesian information criterion BIC. Observe, the number of observations for the different models slightly differ due to the automatic outlier detection, cf. also Table 15.6

This is clear, as the change of the variance factor $\widehat{\sigma}_0^2$ derived from an estimation with a high redundancy R is decisive compared to the small change of parameters U.

The estimated variance factors suggest that the physical model by Brown with only two parameters is the better model for the Ladybug 3 camera, whereas the phenomenological model, namely the polynomials, is superior for the Casio camera. The residual pattern for Brown's radial model applied to the Casio camera indicates significant residuals, which suggest there may be decentring errors in the lens causing tangential deformations. Therefore we apply Brown's model with the two radial and two tangential parameters P_1 and P_2: The residual pattern becomes much more random and the residuals become smaller, as indicated by a much smaller variance factor $\widehat{\sigma}_0^2 = 27.8$ compared to $\widehat{\sigma}_0^2 = 156.7$. This indicates the extension of the physical model by Brown is effective. However, when applying the polynomial model, the residuals are still smaller, namely with a variance factor of $\widehat{\sigma}_0^2 = 23.6$.

The example clearly demonstrates that it is necessary to evaluate the calibration model used to compensate for systematic errors. Obviously both types of models, physically motivated and phenomenological models, appear to be useful.

15.4.2 Bundle Adjustment for Spherical Cameras

For *spherical cameras* we only give the model with radial distortions. Following the model of Scaramuzza (2008, Eqs. (2.12), (2.15)) the general projection model for bundle adjustment is given by

$$\mathbb{E}\left(\mathsf{N}\begin{bmatrix}\boldsymbol{x}'_{it}\\ g({}^{i}\bar{\boldsymbol{x}}'_{it},\boldsymbol{q}_t)\end{bmatrix}\right) = \mathsf{N}\left({}^{c}\mathsf{P}_t\mathbf{X}_i\right) \tag{15.137}$$

(cf. (12.183), p. 507), with some polynomial with even monomials,

$$g({}^{i}\bar{\boldsymbol{x}}', \boldsymbol{q}_t) = K_0 + K'_1|{}^{i}\bar{\boldsymbol{x}}'|^2 + K'_2|{}^{i}\bar{\boldsymbol{x}}'|^4 + \dots . \tag{15.138}$$

This model maps the 3D points $\mathcal{X}_i(\mathbf{X}_i)$ to the observable camera ray direction ${}^{c}\mathbf{x}'^{s}_{it}$. We assume that this direction is related to the ideal image point ${}^{i}\bar{\boldsymbol{x}}'_{it}$ using the multiplicative correction terms $1/g({}^{i}\bar{\boldsymbol{x}}'_{it}, \boldsymbol{q}_t)$. Normalization is necessary as only the directions are of interest. The projection matrices ${}^{c}\mathsf{P}_t = \mathsf{R}_t[\mathsf{I}_3|-\boldsymbol{Z}_t]$ only contain parameters of the exterior orientation. Besides the parameters $\boldsymbol{q}_t$ for modelling the radial distortion, we also need five parameters, possibly per image t, not made explicit in (15.137), to derive the image coordinates ${}^{i}\boldsymbol{x}'_{it}$ from the observed sensor coordinates $\boldsymbol{x}'^{s}_{it}$, cf. (12.32), p. 472:

$${}^{c}\mathbf{x}'_{it} = -\text{sign}(c)\mathsf{K}_t^{-1}\mathbf{x}'^{s}_{it} . \tag{15.139}$$

Additionally, the original model of Scaramuzza (2008) does not include tangential terms. It assumes odd polynomial terms in the function $g(x)$, which appears unnecessary. We also face the necessity to orthogonalize the different additional parameters in order to achieve stable estimates. We do not further discuss this model and refer the reader to Scaramuzza (2008) for further details.

This model has the structure of a Gauss–Helmert model, as there is no way to solve for the observed image point, which also appears as an argument in the distortion model.

If the camera is calibrated, we arrive at the model for the bundle adjustment with spherical cameras,

$$\mathbb{E}({}^{c}\mathbf{x}'_{it}) = \mathsf{N}\left({}^{c}\mathsf{P}_t\mathbf{X}_i\right) . \tag{15.140}$$

This has the form of a Gauss–Markov model, for which we will give the linearized version (cf. Schneider and Förstner, 2013). In contrast to the perspective camera model, we also need to use reduced coordinates for the camera rays, as they only have two degrees of freedom but are represented by normalized 3-vectors. We now use the relation between the corrections $\Delta\mathbf{x}_{it}$ to the ray directions and the corrections $\Delta\boldsymbol{x}'_{it}$ to their reduced coordinates,

$$\Delta^{c}\boldsymbol{x}'_{r,it} = J_r^{\mathsf{T}}({}^{c}\widehat{\mathbf{x}}'^{a}_{it})\Delta^{c}\mathbf{x}_{it} = J_r^{\mathsf{T}}({}^{c}\widehat{\mathbf{x}}'^{a}_{it})({}^{c}\mathbf{x}'_{it} - {}^{c}\widehat{\mathbf{x}}'^{a}_{it}) , \tag{15.141}$$

together with the regular 2×2 covariance matrix

$$\Sigma_{l_{it}l_{it}} := \mathbb{D}(\underline{\Delta^{c}\boldsymbol{x}}'_{r,it}) = J_r^{\mathsf{T}}({}^{c}\widehat{\mathbf{x}}'^{a}_{it})\ \mathbb{D}({}^{c}\underline{\mathbf{x}}'^{s}_{it})\ J_r({}^{c}\widehat{\mathbf{x}}'^{a}_{it}) . \tag{15.142}$$

Following the model (12.225), p. 520 for the spatial resection, we arrive at the linearized model for the bundle adjustment with spherical cameras in the form of a Gauss–Markov model,

$$\Delta^{c}\boldsymbol{x}'_{r,it} + \widehat{\boldsymbol{v}}_{r,it} = J_r^{\mathsf{T}}({}^{c}\widehat{\mathbf{x}}'^{a}_{it})\ J_s({}^{c}\widehat{\mathbf{x}}'^{a}_{it})\left(\mathsf{C}_{it}\widehat{\Delta\boldsymbol{k}}_i + \mathsf{D}_{it}\widehat{\Delta\boldsymbol{p}}_t\right) , \tag{15.143}$$

using the Jacobians from (15.113) and (15.115) and $J_s(\mathbf{x}) = (\mathsf{I}_3 - \mathbf{x}\mathbf{x}^{\mathsf{T}}/|\mathbf{x}|^2))/|\mathbf{x}|$, cf. (10.18), p. 368. The corrections $\widehat{\Delta\boldsymbol{k}}_i$ again refer to the corrections of the reduced homogeneous coordinates of the scene points.

15.4.3 Evaluating Bundle Adjustment Results

The result of a bundle adjustment always needs to be evaluated. Due to both the geometric complexity of the design and the possibly large number of observations and unknowns,

the evaluation needs to be automated, at least partially. Such an automated evaluation can use the methods discussed in Sect. 4.6, p. 115 to advantage.

We transfer these methods to the following scenario when using a self-calibrating bundle adjustment:

- We assume the bundle adjustment is meant to determine scene or pose information or both. Calibration is discussed in the next section.
- We assume the user has specified criteria for accepting the result of the bundle adjustment. These acceptance criteria should be in the spirit of a traffic light program: *green:* the result is acceptable, *yellow:* the result is acceptable but has a lower quality, and *red:* the result is unacceptable.

The goal is to let the bundle adjustment program automatically check the acceptability of the result, or at least provide a summarizing protocol about the achieved result which can then be evaluated manually.

If the result has been identified as acceptable, it can be used by the next step in the processing pipeline. Otherwise, the software should provide enough information

- about the causes of the nonacceptability, so that a user of the software can take measures to improve the result, and
- about the effect of the possible causes on the quality of the result, so that the user can evaluate the relevance of these effects for the envisaged task

in the spirit of the observation-estimation-evaluation-design loop presented in the introduction, see Fig. (1.9), p. 10.

15.4.3.1 Acceptance Criteria

One central requirement for a bundle adjustment result to be acceptable is the following: Deviations of coordinates/poses from their true values are guaranteed to be below some prespecified tolerance. The tolerance may refer to all parameters, or it may differ for groups of parameters, e.g., for positions and rotations. Violations of this criterion may be caused not only by random perturbations of the observed values but by all kinds of deficiencies of the mathematical model, i.e., outliers or systematic errors.

For example, the user specifies tolerances $\boldsymbol{d} = [d_u]$ for the deviation of the U estimated parameters from the true values $\tilde{\boldsymbol{x}} = [\tilde{x}_u]$, which should not be exceeded. Even if the true values are not likely to be known, we could imagine obtaining them from an ideal measurement procedure with superior accuracy. Often the user then requires

$$|\widehat{x}_u - \tilde{x}_u| < d_u \,. \tag{15.144}$$

But this clearly causes problems for larger numbers of parameters, since the probability that the thresholds d_u are exceeded increases with the number of coordinates, and therefore this requirement is not recommended.

In order to be independent of the number of parameters the criterion (15.144) should be fulfilled with a high probability $P_{\max}$. Thus the requirement formally would read

$$\max_u P(|\widehat{x}_u - \tilde{x}_u| < d_u) < P_{\max} \,, \qquad u = 1, ..., U \,. \tag{15.145}$$

This allows for a certain small percentage, $1 - P_{\max}$, of parameters to exceed the thresholds d_u. For economic reasons, the user will try to reach the limit $\max P(|\widehat{x}_u - \tilde{x}_u| < d_u) = P_{\max}$, as then the measurement accuracy of the image coordinates is lowest.

Often, the user refers to one parameter set only, e.g., the coordinates, and treats all coordinates equally; thus, there is no distinction between far-away and close-by points, which usually show different precisions; hence, he chooses $d_u = d$ for all u.

To be able to accept the result of a bundle adjustment with some confidence, several checks need to be performed, which we now discuss in the sequence that usually is fol-

lowed by a human operator, but which can all be translated into an automatic checking procedure.

15.4.3.2 Fit of Model and Data

Differences between the mathematical model and the data generally can be seen in the residuals. Exceptions are errors in the model or the data which are not visible in the residuals, e.g., outliers in observations with redundancy number $r_n = 0$, (4.69), p. 88. One goal of a good design is to avoid situations where errors in the model or the data are not visible in the residuals.

goal of design: make errors in the model visible in the residuals

A Global Check. A global check can be based on the estimated variance factor $\widehat{\sigma}_0^2$, cf. (4.87), p. 90. It documents the overall deviation between data and model.

global check of residuals

As any deviation between model and data might show in the variance factor, it is not a good indicator for any specific error source. Only if $\widehat{\sigma}_0$ significantly deviates from 1, e.g., by a factor of 1.5 or 2, is there a clear indication of discrepancies between model and data, see discussion 4.6.8.1, p. 140. Therefore a more detailed analysis of the residuals is necessary in order to be able to accept the result with confidence.

RMS$_v$

Remark: The variance factor is closely related to the often used *root mean square residual* RMS_v if the N observations are mutually independent and of equal precision, thus if $\boldsymbol{\Sigma}_{ll} = \sigma_l^2 \boldsymbol{I}_N$. With the redundancy R we then have the relations

$$\mathrm{RMS}_v := \sqrt{\frac{\sum_{n=1}^{N} \widehat{v}_n^2}{N}} = \sqrt{\frac{R}{N}} \widehat{\sigma}_0 \, \sigma_l \quad \text{or} \quad \widehat{\sigma}_0^2 = \frac{R}{N} \left(\frac{\mathrm{RMS}_v}{\sigma_l} \right)^2 , \tag{15.146}$$

since $\widehat{\sigma}_0^2 = 1/R \, \sum_{n=1}^{N} \widehat{v}_n^2 / \sigma_l^2$. ◇

Check on Outliers. A check on outliers can be based on the estimated residuals $\widehat{v}_n$ of individual observations or on the estimated residual vectors $\widehat{\boldsymbol{v}}_i$ of observational groups, cf. Sects. 4.6.4.1, p. 125 and 4.6.4.2, p. 129 on outlier detection. As the bundle adjustment is able to handle neither large percentages of outliers nor large individual outliers, we assume that the following processing pipeline is successfully performed:

1. all large and the majority of the medium-sized outliers are found by some *preprocessing step* (cf. Sect. 15.6, p. 707),
2. the small percentage of medium and small outliers which are not found by some preprocessing steps are eliminated by some *maximum-likelihood type estimator*, and
3. the final iteration steps are performed with a maximum likelihood estimator *using only inliers*.

Then the methods for evaluating the result w.r.t. outliers and systematic errors, discussed in Sect. 4.6, p. 115, can be used to advantage.

optimal test statistic for outliers

The optimal test statistic for outliers in single observations and observational groups is

$$z_n = \frac{\widehat{v}_n}{\sigma_{\widehat{v}_n}} = \frac{\widehat{v}_n}{\sigma_{l_n} \sqrt{r_n}} , \qquad X_i = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i \, . \tag{15.147}$$

Remark: Here we assume that the estimated variance factor is accepted, thus close to 1. In practice, we do not use the theoretical variances or covariances but their estimates, thus the test statistic $\widehat{v}_n / \widehat{\sigma}_{\widehat{v}_n} = \widehat{v}_n / (\widehat{\sigma}_0 \sigma_{\widehat{v}_n})$, and $\widehat{\boldsymbol{v}}_i^{\mathsf{T}} \widehat{\boldsymbol{\Sigma}}_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{v}_i \widehat{v}_i}^{-1} \widehat{\boldsymbol{v}}_i / \widehat{\sigma}_0^2$. If the redundancy is large, say beyond 100, the distributions of $\underline{z}_n$ and $\underline{X}_i$ can safely be approximated by a normal and a chi-square distribution. Otherwise, if the redundancy is low, the estimated variance factor should be corrected for the participating observation or observational group, leading to Fisher tests. ◇

As the bundle adjustment using all available data yields the highest redundancy, it is statistically optimal for detecting (small and few) outliers. Therefore, if all test statistics are acceptable, there is no reason to believe that any outliers remain.

The two test statistics in (15.147) require parts of the covariance matrix of the residuals to be known, namely the diagonal terms or diagonal blocks. Though this can be determined efficiently without needing the complete inverse of the covariance matrix of the parameters, this information may not be available.

If the design is homogeneous, specifically if all scene points are measured in three or more images and the images contain enough observations, then the rigorous test can be replaced by an approximated one: Since the standard deviation of the residuals is smaller than the standard deviation of the observations by a factor of $\sqrt{r_n}$, and the mean redundancy number is $\overline{r_n} = R/N$, we can use the approximate test statistics

approximate test statistics for outliers

$$z_n^{\#} = \sqrt{\frac{N}{R}}\,\frac{\widehat{v}_n}{\sigma_{l_n}}\,, \qquad X_i^{\#} = \frac{N}{R}\,\widehat{\boldsymbol{v}}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{l_i l_i}^{-1}\widehat{\boldsymbol{v}}_i\,, \tag{15.148}$$

which on an average are unbiased. Both test statistics are less efficient than the optimal ones, i.e., they are not able to detect small outliers as well as the optimal test statistics. Often the weighted residuals

$$z_n^{*} = \widehat{v}_n/\sigma_{l_n}\,, \qquad X_i^{*} = \widehat{\boldsymbol{v}}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{l_i l_i}^{-1}\widehat{\boldsymbol{v}}_i \tag{15.149}$$

are used, which however are not corrected for bias.

While the homogeneity of the design can often be realized for the observations, the pose parameters and the new scene points, the detectability of outliers in the *control points* usually is worse as their redundancy numbers r_i often are far below the average $\overline{r_i} = R/N$. Therefore the approximate test statistics in (15.148) are not very useful for control points in general.

testing the control points with optimal test statistics

A remedy against this situation is the following: include the control point coordinates in the set of unknown parameters, cf. (15.9), p. 647. Then the covariance matrix of their residuals can easily be determined from, say, $\boldsymbol{\Sigma}_{\widehat{v}_X\widehat{v}_X} = \boldsymbol{\Sigma}_{XX} - \boldsymbol{\Sigma}_{\widehat{X}\widehat{X}}$, with the inhomogeneous coordinates $\boldsymbol{X}$ of the control point. As the number of control points usually is small, this puts no serious burden on the computational complexity of the bundle adjustment, but it simplifies the identification of outliers in control point coordinates.

A further simplification of the analysis can be achieved if the standard deviations of the observations, again except for the control point coordinates, are identical and the design is homogeneous. Then the analysis can be based on the plain residuals $\widehat{v}_n$ or $\widehat{\boldsymbol{v}}_i$, cf. (15.148).

Homogeneity of Residuals. The residuals should be of homogeneous size in order to make sure there are no groups of outliers or local systematic errors. The homogeneity of the residuals can be checked easily if we evaluate them after grouping. Groups $\mathcal{G}$ could be all observations referring to a scene point or to an image. The average residuals for such groups then should not vary too much.

variance factor for groups of observations

The optimal test statistic for such a group is similar to X_i in (15.147), for better comparison normalized with the group size $G = |\mathcal{G}|$,

$$\widehat{\underline{\sigma}}_{0,G}^2 = \frac{\widehat{\underline{\boldsymbol{v}}}_G^{\mathsf{T}}\,\boldsymbol{\Sigma}_{\widehat{v}_G\widehat{v}_G}^{-1}\,\widehat{\underline{\boldsymbol{v}}}_G}{G} \sim \mathcal{F}(G,\infty)\,. \tag{15.150}$$

It can be interpreted as a local estimate for the variance factor. For sufficiently small groups of observations, we generally can assume the inverse of $\boldsymbol{\Sigma}_{\widehat{v}_G\widehat{v}_G}$ to exist. Again, if the covariance matrix of the residuals is not available, the design is homogeneous, and the observations are mutually independent, we may use the approximate local variance factor,

$$\widehat{\sigma}_{0,G}^{2,*} = \frac{N}{R}\,\frac{1}{G}\,\sum_{g\in\mathcal{G}}\frac{\widehat{v}_g^2}{\sigma_{l_g}^2}\,, \tag{15.151}$$

related to the local RMS_{z^*},

$$\mathrm{RMS}_{z^*} = \sqrt{\frac{1}{G}\sum_{g\in\mathcal{G}}\frac{\widehat{v}_g^2}{\sigma_{l_g}^2}}\,. \tag{15.152}$$

The estimated variance factor $\widehat{\sigma}_{0,G}^*$ is related to the RMS_{z^*} by a factor of $\sqrt{N/R}$.

The Absence of Unmodelled Systematic Errors. The absence of unmodelled systematic errors is difficult to express explicitly, especially if there exists no hypothesis about the structure of such errors. Remaining systematic errors are unlikely

1. if the histogram of the normalized residuals z_n does not significantly deviate from that of a Gaussian distribution; deviations may result from unmodelled systematic errors but also from wrong assumptions about the precision of the given observations; for the same reason, the histogram of the weighted residuals z_n^* (15.149) is not necessarily suited for an evaluation, especially if the geometric design is weak; it is likely to deviate from that of a Gaussian in this situation, as the standard deviations of the residuals are not homogeneous;
2. if the residuals show no patterns in the image plane; this especially holds for the two-dimensional pattern of the residuals $\widehat{\boldsymbol{v}}_i(\boldsymbol{x}_i')$ over all images; such a pattern may indicate a unmodelled lens distortion; the found pattern may be an indication of how to improve the model of the interior orientation;
3. if the estimated variance factors $\widehat{\sigma}_0^2$ of sub-blocks, especially of image pairs or triplets, do not differ from the global estimate.

15.4.3.3 Acceptability of Design

The design refers to the geometric distribution of the camera poses and, if the block is not a free block, of the control points. It is acceptable if the achievable theoretical precision of the estimated parameters fulfils prespecified conditions and the sensitivity of the estimated parameters w.r.t. nondetectable model errors, i.e., outliers and systematic errors, lies within prespecified bounds. The design can be checked in the planning phase of a project, but its acceptability needs to be verified after a bundle adjustment.

If the user has specified some precision structure in the form of a criterion matrix requiring the theoretical precision not to be worse than the prespecified one, the achieved covariance matrix needs to be compared to the prespecified criterion matrix, following the method of Sect. 4.6.2.3, p. 120.

Otherwise a few simple criteria can be checked:

- Each image overlaps with at least three or four other images by more than 50% of the points or contain at least 20 points. The number of neighbouring images does not vary too much, say by factor of 3.
- Each scene point is measured in at least three or four images. Moreover, the cone of rays ${}^c\mathbf{x}_{it}'$ needs to be large enough: The average angle between the rays should be larger than a threshold $\mathrm{median}_{tt'}(\alpha_{i,tt'}) > T$, say $T = 20°$.
- The precision of the orientation parameters agrees with the precision of the image measurements, i.e., the covariance matrix of the pose parameters of an image does not deviate too much from the one obtained by a spatial resection with evenly distributed image points (e.g., using the standard deviations and correlations from (12.229), p. 523ff. and multiplying them with a factor $\sqrt{I_t/8}$, due to the larger number I_t of points used per image, compared to the eight image points assumed for (12.229)).
- The precision of scene points is homogeneous.

15.4.3.4 Acceptability of Accuracy

A bundle adjustment result has acceptable accuracy if the combined effect of random errors in the observations and possible effects of outliers and systematic errors onto the estimated parameters is small enough.

Acceptability of Precision. We first need to check the *acceptability of the achieved precision*, i.e., the expected effect of random perturbations on scene points or on pose parameters. This is reflected in the estimated covariance matrix $\widehat{\Sigma}_{\widehat{x}\widehat{x}} = \widehat{\sigma}_0^2 \Sigma_{\widehat{x}\widehat{x}}$ of the estimated parameters, or equivalently for each estimated parameter, cf. (4.251), p. 118,

$$\widehat{\sigma}_{\widehat{x}_u} = \widehat{\sigma}_0 \, \sigma_{\widehat{x}_u} \, . \tag{15.153}$$

The estimated standard deviations of the estimated parameters take into account the model fit, represented by the estimated variance factor $\widehat{\sigma}_0^2$; otherwise, they just depend on the geometric design and the assumed observational precision.

In the ideal case, if no systematic errors are left, and we are given a probability of $P_{\max}$ of deviations from true values lying in the tolerances $d_u, u = 1, ..., U$, the confidence intervals for the individual parameters is $k(P_{\max})$ multiplied by their standard deviations. These intervals can be compared to the prespecified tolerances d_u,

$$k(P_{\max}) \, \widehat{\sigma}_0 \, \sigma_{\widehat{x}_u} \le d_u \, , \quad u = 1, ..., U \, , \tag{15.154}$$

possibly only referring to scene coordinates or to pose parameters. If this criterion is violated, either the geometric design needs to be improved in order to decrease the standard deviations $\sigma_{\widehat{x}_u}$ of the estimated parameters, or observations with higher accuracy need to be made, diminishing $\widehat{\sigma}_0$.

local empirical covariance matrix

The homogeneity of the precision can be analysed by determining the *local empirical covariance matrices*,

$$\widehat{\Sigma}_{\widehat{x}_G \widehat{x}_G} := \widehat{\sigma}_{0,G}^2 \, \Sigma_{\widehat{x}_G \widehat{x}_G} \, . \tag{15.155}$$

Here G refers to groups of parameters $\widehat{x}_G$, say 3D points or pose parameters, and the local variance factor of the observations referring to that group, i.e., to the scene point or the image. As the estimated variance factor requires the covariance matrix of the residuals to be known, which may be not available, it can be replaced by the root mean square residual compensated by the factor $\sqrt{N/R}$, (15.150), p. 690.

Sensitivity w.r.t. Outliers. Up to now only the effect of random errors onto the final accuracy has been addressed. We now analyse the effect of nondetectable outliers on the estimates. For example, the effect of a nondetectable error in an group i on an estimated parameter $\widehat{x}_u$ is bounded by

$$|\nabla_{0,i} \widehat{x}_u| \le \delta(d_i) \, \mu_i \, \sigma_{\widehat{x}_u} \tag{15.156}$$

(cf. (4.314), p. 130), with the noncentrality parameter $\delta(d_i)$ not depending on the design, the sensitivity factor μ_i, and the standard deviation $\sigma_{\widehat{x}_u}$ of the estimated parameter. In order to guarantee a high accuracy of the parameters the upper bound on the right-hand side of (15.156) needs to be small.

Assuming the acceptability of the precision has been checked, the sensitivity of the result essentially depends on the sensitivity factors μ_i. They measure the magnification or diminution of outliers solely caused by the design of the configuration. The expressions in Tables 4.1, p. 128 and 4.2, p. 131 refer to the effect of outliers on the estimated coordinates. For example, the sensitivity factor for an outlier in group i w.r.t. the coordinates is given by

$$\mu_{ik}^2 = \lambda_{\max} \left(\overline{C}_i^{(p)\mathsf{T}} \Sigma_{\widehat{k}\widehat{k}} \overline{C}_i^{(p)} \Sigma_{\widehat{v}_i \widehat{v}_i}^{-1} \right) \, , \tag{15.157}$$

with the part $\overline{C}_i^{(p)}$ of the reduced design matrix for the coordinates $\widehat{\boldsymbol{k}}$ referring to the ith observational group after eliminating the orientation parameters $\widehat{\boldsymbol{p}}$, cf. (4.122), p. 95. Both

factors, $\overline{C}_i^{(p),\mathsf{T}} \Sigma_{\widehat{k}\widehat{k}} \overline{C}_i^{(p)}$ and $\Sigma_{\widehat{v}_i\widehat{v}_i}$, can be determined efficiently if only parts of the inverse covariance matrix are available, cf. Sect. 15.3.3.4, p. 662. If we are interested in the effects of undetected outliers on the orientation parameters, we use the sensitivity factors w.r.t. the transformation parameters,

$$\mu_{ip}^2 = \lambda_{\max}\left(\overline{D}_i^{(k)\mathsf{T}} \Sigma_{\widehat{p}\widehat{p}} \overline{D}_i^{(k)} \Sigma_{\widehat{v}_i\widehat{v}_i}^{-1}\right) , \tag{15.158}$$

with the design matrices $\overline{D}_i^{(k)}$ reduced for the pose parameters and referring to observational group i.

Larger groups could be chosen, e.g., related to the scene points or images. Then the coordinates of the scene point or the pose parameters of the image in question need to be taken out of the evaluated estimated parameters, i.e., the corresponding columns need to be eliminated in the reduced design matrices and the corresponding rows and columns taken out of the covariance matrix.

As the effect of observations on the estimated parameters is $\delta_0\mu_i$ times their standard deviation and the noncentrality parameter is $\delta_0 \approx 4$ (cf. Table 3.2, p. 67), in order to guarantee that nondetectable outliers have an influence on the result, say, less than three times its standard deviation, the sensitivity factors should be smaller than 3/4. In the most simple case, due to $\mu_n = \sqrt{(1-r_n)/r_n}$ for single observations, the redundancy should be larger than $\frac{16}{9}U$ or approximately larger than two times the number of unknowns,

$$R > 2U\,. \tag{15.159}$$

In bundle adjustment, this requirement can be realized by observing each scene point in five or more images.

effect of leaving out observational groups on the estimated parameters

The effect of leaving out observational group i on the estimated parameters is bounded by

$$|\nabla_i \widehat{x}_u| \le X_i\ \mu_i\ \widehat{\sigma}_0 \sigma_{\widehat{x}_u}\,, \tag{15.160}$$

where $\widehat{x}_u$ refers either to the scene coordinates or to the pose parameters. We also need the test statistic X_i from (4.302), p. 129. Thus the effect of leaving out observational group i is $X_i\mu_i$ times larger than the standard deviation of some selected function of the estimated parameters. This evaluation also assumes that the precision already is checked for acceptability or at least for homogeneity.

Even if we have enough observations per scene point, the sensitivity of the resultant parameters $\widehat{\boldsymbol{x}}$ w.r.t. outliers in the control points may be high, mostly as the number of control points usually is small for economic reasons. Therefore control point coordinates should be introduced as observations with a realistic assumption about their covariance matrix $\Sigma_{k_0k_0}$. Let the covariance matrix of the fitted control points $\widehat{\boldsymbol{k}}_0$ be $\Sigma_{\widehat{k}_0\widehat{k}_0}$, then the sensitivity factors for the ith control point are

$$\mu_{ix}^2 = \lambda_{\max}(I_d - \Sigma_{\widehat{k}_{0i}\widehat{k}_{0i}} \Sigma_{k_{0i}k_{0i}}^{-1})\,, \tag{15.161}$$

where d is the dimension of the control point vector.

It may be of advantage to reduce the normal equation system by the parameters of the new points $\boldsymbol{k}_{\text{new}}$ only and thus only keep the transformation parameters $\boldsymbol{p}$, the coordinates of the control points $\boldsymbol{k}_0$, and the additional parameters $\boldsymbol{s}$ of the interior orientation, in that order, for keeping the fill-in low (see Fig. 15.7, p. 662). We have used this setup for determining the quality values of control points in Sect. 15.3.5, p. 670.

Sensitivity w.r.t. Systematic Errors. In a similar manner, we can analyse the sensitivity of the design w.r.t. unmodelled systematic errors. The effect of omitting the P additional parameters $\boldsymbol{s} = [s_p]$ on an estimated parameter $\widehat{x}_u$ is

$$\nabla_{0s}\widehat{x}_u \le \delta_0(P)\ \mu_s\ \sigma_{\widehat{x}_u} \tag{15.162}$$

(cf. (4.331), p. 135), where again the noncentrality parameter $\delta_0(P)$ does not depend on the geometric design.

The user needs to decide whether it is the sensitivity w.r.t. the coordinates or w.r.t. the orientation parameters that he wants to evaluate. The sensitivity factor μ_{sk} for the effect of additional parameters on the *scene coordinates* $\widehat{\boldsymbol{k}}$ is

$$\mu_{sk}^2 = \lambda_{\max}\left(W_{\widehat{s}\widehat{s}}^{(-k)}\Sigma_{\widehat{s}\widehat{s}} - I\right), \quad \text{with} \quad W_{ss}^{(-k)} = N_{ss} - N_{sp}N_{pp}^{-1}N_{ps}\,; \tag{15.163}$$

see the proof below and (4.328), p. 135. The corresponding factor μ_{sp} w.r.t. the effect on the *orientation parameters* $\widehat{\boldsymbol{p}}$ is

$$\mu_{sp}^2 = \lambda_{\max}\left(W_{\widehat{s}\widehat{s}}^{(-p)}\Sigma_{\widehat{s}\widehat{s}} - I\right), \quad \text{with} \quad W_{ss}^{(-p)} = N_{ss} - N_{sk}N_{kk}^{-1}N_{ks}\,. \tag{15.164}$$

The effect $\nabla_s\widehat{x}_u$ of leaving out some additional parameters from $\widehat{\boldsymbol{s}}$ is

$$|\nabla_s\widehat{x}_u| \le X_s\ \mu_s\ \widehat{\sigma}_0\ \sigma_{\widehat{x}_u} \quad \text{with} \quad X = \widehat{\boldsymbol{s}}^{\mathsf{T}}\Sigma_{\widehat{s}\widehat{s}}^{-1}\widehat{\boldsymbol{s}}\,, \tag{15.165}$$

the estimated parameter $\widehat{x}_u$ again referring to scene coordinates or pose parameters. Correspondingly, the sensitivity factor μ_s is either μ_{sk}, referring to scene coordinates, or μ_{sk}, referring to the pose parameters. Leaving out the additional parameters results in a test statistic, $X_s^2 = \widehat{\boldsymbol{s}}^{\mathsf{T}}\Sigma_{\widehat{s}\widehat{s}}^{-1}\widehat{\boldsymbol{s}}$, cf. (4.317), p. 133.

acceptance criterion for the accuracy of the estimated parameters

Taking all effects into account, we obtain the following *acceptance criterion for the accuracy of the estimated parameters*,

$$\sqrt{k^2(P_{\max}) + \max_i((X_i\mu_i)^2) + (X_s\mu_s)^2}\ \widehat{\sigma}_0\ \sigma_{\widehat{x}_u} \le d_u\,, \tag{15.166}$$

referring to the unknown parameters of interest. It captures the influence on the estimated parameters, especially (1) of all random errors via the tolerance $k(P_{\max})$, (2) of nondetected outliers of observational group $\boldsymbol{l}_i$ via $X_i\mu_i$, and (3) of systematic errors via $X_s\mu_s$, where all are factors for the estimated standard deviation $\widehat{\sigma}_{\widehat{x}_u} = \widehat{\sigma}_0\ \sigma_{\widehat{x}_u}$. The combined influence should be smaller than the prespecified accuracy tolerances. As the effects can be assumed to be independent, we adopt their quadratic sum. Depending on whether the coordinates or the orientation parameters are of primary interest, the sensitivity factors μ_{ik} and μ_{sk} or the factors μ_{ip} and μ_{sp} are to be taken.

Proof: We prove (15.163), which gives the effect of systematic errors, i.e., of the additional parameters $\boldsymbol{s}$ on the estimated coordinates $\widehat{\boldsymbol{k}}$. For this, we reduce the normal equation system to the coordinates $\Delta\boldsymbol{k}$ and the additional parameters $\Delta\boldsymbol{s}$, collected in the vector

$$\Delta\boldsymbol{y} = \begin{bmatrix}\Delta\boldsymbol{k}\\ \Delta\boldsymbol{s}\end{bmatrix}. \tag{15.167}$$

We obtain the reduced normal equation matrix

$$\overline{N}_{yy} = N_{yy} - N_{yp}N_{pp}^{-1}N_{py}\,, \tag{15.168}$$

or, explicitly,

$$\begin{bmatrix}\overline{N}_{kk} & \overline{N}_{ks}\\ \overline{N}_{sk} & \overline{N}_{ss}\end{bmatrix} = \begin{bmatrix}N_{kk} - N_{kp}N_{pp}^{-1}N_{pk} & N_{ks} - N_{kp}N_{pp}^{-1}N_{ps}\\ N_{sk} - N_{sp}N_{pp}^{-1}N_{pk} & N_{ss} - N_{sp}N_{pp}^{-1}N_{ps}\end{bmatrix}. \tag{15.169}$$

In full analogy to the sensitivity w.r.t. all parameters $\widehat{\boldsymbol{x}}$, the sensitivity factor w.r.t. the coordinates therefore is

$$\mu_{sk}^2 = \lambda_{\max}\left(W_{\widehat{s}\widehat{s}}^{(-k)}\Sigma_{\widehat{s}\widehat{s}} - I\right), \tag{15.170}$$

with

$$W_{\widehat{s}\widehat{s}}^{(-k)} := \overline{N}_{ss} = N_{ss} - N_{sp}N_{pp}^{-1}N_{ps} = H^{\mathsf{T}}W_{ll}H - H^{\mathsf{T}}W_{ll}D(D^{\mathsf{T}}W_{ll}D)^{-1}D^{\mathsf{T}}W_{ll}H\,. \tag{15.171}$$

◇

The practical calculation requires some discussion. Usually the normal equations are reduced to the orientation parameters and the additional parameters so that $\Sigma_{\widehat{s}\widehat{s}}$ is avail-

able. But $W_{\widehat{s}\widehat{s}}^{(-k)}$ needs to be determined solely for the sensitivity analysis. It is useful to exploit the sparsity of the matrix D,

$$D = \{ \underset{2\times 6}{D_{it}^{\mathsf{T}}} \}, \tag{15.172}$$

in a bundle adjustment consisting of 2×6 matrices D_{it}^{T} for each point i observed in each image t. The matrix N_{pp} is block diagonal,

$$N_{pp} = \mathrm{Diag}(\{\underbrace{N_{p_t p_t}}_{6\times 6}\}), \tag{15.173}$$

with 6×6 matrices,

$$N_{p_t p_t} = \sum_{i\in\mathcal{I}_t} \underset{6\times 2}{D_{it}} \underset{2\times 2}{W_{l_i l_i}} \underset{2\times 6}{D_{it}^{\mathsf{T}}}, \quad t = 1, ..., T, \tag{15.174}$$

on the diagonal, which exploits the sparsity of D as the sums are taken over all points i in image t.

If we only have one group of additional parameters for modelling systematic errors, the matrix H is full (cf. (15.121), p. 678) with $H = [\ \underbrace{H_i^{\mathsf{T}}}_{2\times N_s}\ ]$, where the H_i^{T} are $2 \times N_s$ matrices for each observational group $\boldsymbol{l}_i$. The matrix

$$\underset{N_p\times N_s}{N_{ps}} = D^{\mathsf{T}} W_{ll} H = [\ \underbrace{N_{p_t s}}_{6\times N_s}\ ] \tag{15.175}$$

usually is full, where the $6 \times N_s$ matrices can be determined from

$$\underset{6\times N_s}{N_{p_t s}} = \sum_{i\in\mathcal{I}_t} \underset{6\times 2}{D_{it}} \underset{2\times 2}{W_{l_i l_i}} \underset{2\times N_s}{H_i^{\mathsf{T}}}, \qquad t = 1, ..., T, \tag{15.176}$$

again exploiting the sparsity of D as the sum is taken over all points i in image τ. Finally, the matrix $W_{\widehat{s}\widehat{s}}^{(-k)}$ finally is

$$W_{\widehat{s}\widehat{s}}^{(-k)} = H^{\mathsf{T}} W_{ll} H - \sum_{t=1}^{N_t} N_{p_t s}^{\mathsf{T}} N_{p_t p_t}^{-1} N_{p_t s}. \tag{15.177}$$

Eliminating Individual Additional Parameters to Achieve Stable Results. Often only individual parameters, i.e., basis functions, which generate the columns $\boldsymbol{h}$ of the matrix H (cf. (15.121), p. 678 and (15.116), p. 678), are responsible for the instability of the result. Then it is advisable to eliminate these parameters from the estimation in the next iteration of the estimation procedure. An intuitive criterion to keep a parameter s_j would be to require that the uncertainty of this parameter has only a limited effect onto the precision of the result, e.g., the estimated coordinates $\widehat{\boldsymbol{k}}$ of the bundle adjustment. Thus the sensitivity factor $\mu_{s_j k}$ should be small, i.e.,

$$\mu_{s_j k} \le T_\mu \quad \text{with} \quad \mu_{s_j k}^2 = w_{\widehat{s}_j}^{(-k)}\, \sigma_{\widehat{s}_j}^2 - 1 \tag{15.178}$$

with a threshold T_μ in the range of $[3, 10]$ and using (15.163), p. 694.

The estimated parameters $\widehat{s}_j$ in general are correlated; therefore, the decisions on eliminating individual parameters s_j will be mutually dependent. This is why an a posteriori orthogonalization is useful, which leads to strictly orthogonal basis functions. The parameters of these basis functions result from a whitening (cf. (2.131))

$$\widehat{\Delta \boldsymbol{s}'} = M \widehat{\Delta \boldsymbol{s}} \quad \text{with} \quad M = \Sigma_{\widehat{s}\widehat{s}}^{-1/2}. \tag{15.179}$$

Thus the systematic errors now are modelled by

$$H\Delta\boldsymbol{s} = (HM^{-1})\,(M\Delta\boldsymbol{s}) = H'\Delta\boldsymbol{s}' \quad \text{with} \quad H' = HM^{-1}\,. \tag{15.180}$$

The estimated parameters $\widehat{\boldsymbol{s}}'$ then have covariance matrix $\Sigma_{\widehat{s}'\widehat{s}'} = M\Sigma_{\widehat{s}\widehat{s}}M^{\mathsf{T}} = I$. After this whitening an individual evaluation and possibly elimination of the parameters $\widehat{s}'_j$ for the strictly orthogonal basis functions can be realized. The decisions can be based on the corresponding influence factors

$$\mu_{s'_j k} = w_{\widehat{s}'_j}^{(-k)} - 1\,, \tag{15.181}$$

which can be derived using the diagonal elements $w_{\widehat{s}'_j}^{(-k)}$ of (15.171) with H' instead of H and $\sigma_{\widehat{s}'_j} = 1$. The following iteration works with H', possibly with some columns eliminated.

15.5 Camera Calibration

15.5.1 Self-calibration with an Unknown 3D Test Field 698
15.5.2 Evaluating Additional Parameters 699
15.5.3 Unmodelled Systematic Errors 700
15.5.4 Effect of Unmodelled Systematic Errors 701
15.5.5 Instability of Systematic Errors and Their Effect 702
15.5.6 An Example for Evaluating Calibration Parameters 702

We now address the problem of camera calibration, i.e., determining the parameters of the interior orientation of the cameras used. Though we already discussed the choice of additional parameters for modelling the interior orientation and the technique of self-calibrating bundle adjustment for pose determination and scene reconstruction, here we focus on camera calibration as a task of its own.

We distinguish between three scenarios when using a camera within a project:

calibrated metric camera

- The interior orientation of the camera is stable from the beginning of the calibration until the end of its use. It can be realized if the calibration is performed under similar conditions as those of the use of the camera. Then the camera can be treated as a *calibrated metric camera*, cf. Sect. 12.1.2, p. 459.
  In fact, no camera is perfectly stable in reality. However, its instability may be small enough for the envisaged application. Then the camera often is called stable.
  If we want to work with calibrated cameras, we need to determine the calibration before its use within a project and check its calibration afterwards, in order to be sure no changes occurred during the image acquisition.

noncalibrated metric camera

- The interior orientation is stable, both during the calibration as well as during the use of the camera, but the two interior orientations may be slightly different.
  This is the classical case for aerial cameras, where the calibration is performed in a laboratory, say at a temperature of 20°, and the camera is used under different weather conditions, say at temperatures of −50°. The calibration in the lab can be repeated in order to confirm that the camera has not changed, e.g., after damage to the camera case.
  Then the camera used can be said to be metric, but not calibrated.
  In this case, we need to determine the parameters of the interior orientation by self-calibration within the bundle adjustment and check the stability of the camera afterwards, if it appears to be necessary.
- The interior orientation varies due to deliberate changes, say of the focus or of the aperture, and the effects of these changes on the interior orientation are repeatable. This assumes the camera to be stable, i.e., have repeatable properties, up to the intended changes.

The deliberate changes may be known, e.g., by calibrating the control module of the camera influencing the focus, with its effect on the principal distance and possibly the lens distortion. In this case the camera can be said to be metric, as the relation between the observable image points and the resulting camera rays is known.

uncalibrated nonmetric camera

Otherwise these changes are not known precisely. They need to be determined by self-calibration and the camera is nonmetric.

If the camera is not stable at all, the only way to determine its interior orientation is by self-calibrating bundle adjustment, or its suboptimal surrogates using only partial information.

The calibration process itself can be performed in various ways:

- *Laboratory calibration.* Here the physical relation between the sensor array and the viewing ray is determined optically, e.g., using a collimator. The resulting protocol contains all required information, including, but not restricted to, information about the origin of the sensor system, the principal point, the principal distance, lens distortion and possibly sensor flatness. The effort for such a calibration is high and requires an adequate laboratory. Therefore this method is rarely used in practice.

  laboratory calibration
- *Camera calibration with a known 3D test field.* Here the relation between the 3D points or lines in the test field and its images can be used to determine the parameters of the interior orientation. Classical test fields are sets of points, possibly arranged in a grid or sets of 3D lines, e.g., physically realized as plumb lines. As the test field is known, the number of images necessary for a reliable calibration is usually low. For example, observing a large enough set of 3D points or 3D lines in an image allows us to determine lens distortion from a single image. However, the five parameters contained in the calibration matrix K, namely the principal distance, the principal point (two), and the two affine parameters shear and scale difference, require either a 3D test field or multiple images taken with a stable camera.

  test field calibration
- *Camera self-calibration.* Here the parameters of the interior orientation are determined from several images within a self-calibrating bundle adjustment. For ensuring precise enough image measurements, a test field with artificial targets is often used, in the most simplest a checkerboard one. But, in contrast to the test field calibration, the 3D coordinates of the test field need not be known, since they are determined within the bundle adjustment. Therefore camera self-calibration can also be applied if no test field is available and the scene contains a sufficiently large number of well-identifiable features. As the scene coordinates are unknown, more images are necessary for obtaining a reliable camera calibration.

  camera self-calibration

  using an unknown test field

The difference between camera self-calibration and test field calibration is small if a test field is used, since test field calibration can also determine the parameters of the interior orientation with a self-calibrating bundle adjustment, where the scene features are known.

As manufacturing a test field with precise enough target coordinates may be costly, we can apply an intermediate procedure: Either the nominal coordinates of the 3D targets may be treated as prior information in a self-calibrating bundle adjustment, the uncertainty of the scene features reflecting the imprecision of the manufacturing process; or, when measuring the targets of the test field with some measuring device, their coordinates may be used as a prior, now with the uncertainty of the mensuration process.

In the following, we start with guidelines for a calibration procedure using camera self-calibration, which guarantees a reliable calibration result. This is the most general setup and practically feasible due to the simplicity of taking multiple images. Then we discuss how the fulfilment of the guidelines can be checked and give an example for evaluating the camera calibration.

15.5.1 Self-calibration with an Unknown 3D Test Field

Camera calibration with self-calibrating bundle adjustment using a test field with unknown scene coordinates requires certain geometric configurations in order to yield reliable results. The following preconditions need to be fulfilled, cf. Fraser (2013):

- Take convergent photos sitting on a large enough small circle of a sphere and pointing towards the centre of the test field. This guarantees enough depth variations for determining the principal distance.
- Try to have as much 3D structure in the scene as possible. This decouples the interior orientation and the exterior orientation, see Fig. 15.20. A test field with well-identifiable natural points can also be used since the scene points are treated as new tie points in the bundle adjustment.

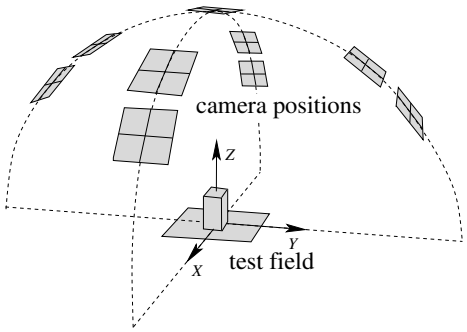


Fig. 15.20 Ideal setup of camera poses for calibration using bundle adjustment. At each position two images with 0° and 90° roll angles or four images with 0°, ± 90° and 180° roll angles are taken. The box in the middle of the test field indicates it should have some 3D structure for better separating the parameters of the exterior orientations and the interior orientations than when just relying on the obliqueness of the views onto a planar test field. Adapted from Vosselman and Förstner (1988)

- Vary the roll angle, i.e., the rotation κ around the viewing direction, by 90° (two roll angles in total), or, better, by ± 90° and 180° (four roll angles in total). This guarantees the determination of distortions of any kind of symmetry.
- Cover the complete sensor area with image points. This will not be possible for all images. But each area in the image should be covered at least a few times. This guarantees that no extrapolation of distortions is necessary.
- Keep the interior orientation fixed and especially do not zoom or vary the focus. This guarantees stable interior orientation, which can be used in a subsequent bundle adjustment.

Following these rules leads to reliable estimates for the parameters of the interior orientation.

However, they assume the ideal case where the camera is metric, i.e., stable, and the geometric configuration can be realized. Non-ideal situations require care. Such situations arise where either a zoom lens is necessary to capture relevant parts of the scene with sufficient resolution, or where the scene does not allow optimal camera positions, as in churches or caves. Therefore the following issues need further discussion.

- How to handle cameras with zoom lenses as uncalibrated metric cameras. The stability of the lenses is significantly less than that of fixed-focus lenses, which is to be expected from the existence of moving elements in zoom optics.
- How to evaluate the result of self-calibrating bundle adjustments in the case of weak configurations. Because the determination of the interior orientation, including all parameters, is uncertain in this case, the resulting scene points will be uncertain to a degree which can be expected to be much larger than the internal precision of the image measurements would suggest.

- How to evaluate the result of the self-calibration if a camera with tele-optics is used for visual odometry. In this case the pose is extrapolated from the scene and highly depends on a proper calibration.

In all these cases the user might be aware of the suboptimality and accepts a less accurate result. The loss in accuracy will not necessarily be reflected by the covariance matrix $\Sigma_{\widehat{x}\widehat{x}}$, which only reflects the precision under the constraints that the estimation model holds and that it is not underspecified. Therefore we need to also evaluate the sensitivity of the result w.r.t. wrong or uncertain additional parameters, exploiting the results of Sect. 4.6, p. 115.

15.5.2 Evaluating Additional Parameters

Let us formalize the problem within the estimation procedure. The result of a calibration usually is a set of P parameters $\widehat{\boldsymbol{s}} = [\widehat{s}_p]$ which allow the estimated corrections to the image coordinates to be written as

$$\widehat{\Delta \boldsymbol{l}} = \sum_{p=1}^{P} \widehat{s}_p \boldsymbol{h}_p(\boldsymbol{x}) \,, \tag{15.182}$$

where the vector functions $\boldsymbol{h}_p(\boldsymbol{x})$ are basis functions of the distortion model, e.g., the polynomials in Brown's model depending on the radius $r = |\boldsymbol{x} - \boldsymbol{x}_A|$, cf. (12.175), p. 506.

The parameters are estimates, therefore uncertain, $\underline{\widehat{\boldsymbol{s}}} \sim \mathcal{M}(\mathbb{E}(\underline{\widehat{\boldsymbol{s}}}), \mathbb{D}(\underline{\widehat{\boldsymbol{s}}}))$. If the calibration model is not underspecified, the uncertainty only results from the uncertainty of the measurements used for the calibration. Therefore, increasing the number of observations and improving the calibration configuration may be used to reach arbitrary precision, at least in principle.

In reality, the precision of the calibration will be limited at least for two reasons:

1. The model is always underspecified, so that there will always be distortion effects, thus systematic errors, which are not modelled, even if they are small.
2. The camera is instable. There always will be variations over time, which may be small but not zero. This is the case, for example, when they only depend on the temperature at the time the images were taken.

Both effects may be small enough to be negligible within the envisaged application. Therefore we need to have indicators which answer the following questions:

- Are there unmodelled systematic errors, and how large are they? This is only relevant if the user is interested in modelling the distortion, e.g., if he is a manufacturer of lenses.
- Do unmodelled systematic errors have a nonnegligible effect on the result? This effect needs to be adequately added to possible uncertainties in the scene coordinates in order to have a guarantee that the evaluation of the acceptability of the result is not misleading.
- Is the calibration stable and how large is the instability? Again, this is only relevant if the user is interested in the behaviour of the camera system, especially the lens system.
- Does the instability of the calibration have a tolerable effect on the result?

If the result is not acceptable, the user might ask whether unmodelled systematic errors or instabilities are the cause.

We discuss the four questions.

15.5.3 Unmodelled Systematic Errors

Unmodelled systematic errors lead to deviations between the expected values of the observations $\mathbb{E}(\underline{\boldsymbol{l}})$ and the observations $\boldsymbol{f}(\tilde{\boldsymbol{x}})$ predicted from the true parameters and the assumed model. As the estimation minimizes the residuals, these deviations will affect all estimates: the parameters $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{s}}$, but also the estimated residuals $\widehat{\boldsymbol{v}}$. Without additional information, the residuals are the only indicators for unmodelled systematic errors.

It is unlikely that the unmodelled systematic errors only have an influence on the residuals and not on the parameters. However, this would be the best case for a user, as the unmodelled systematic error will be visible and thus influence the estimated variance factor, but there will be no distortion of the envisaged scene coordinates.

Also, it is unlikely that the unmodelled systematic errors have only an influence on the estimated parameters and not on the residuals. This is the worst case, as there will be no indication of the unmodelled systematic error and the result will still be wrong. An example would be an in-plane rotation of the sensor plane within the camera body, which would influence only the rotation around the viewing direction. Situations where only the scene points are affected by an unmodelled systematic error which is not visible in the residuals are only possible if the configuration is very regular. Unmodelled systematic errors may be hardly visible in the residuals, thus not detectable, and therefore may deteriorate the result. For this reason practitioners often use *3D checkpoints* whose coordinates are taken as reference coordinates, measured independently in the scene, and which are observed in the images. These image points are treated as unknown tie points and do *not* take part in the bundle adjustment. Their scene coordinates derived after the bundle adjustment are then compared with their reference coordinates.

Exercise 15.7

check points

We discuss the case where such checkpoints are not available. We assume that unmodelled systematic errors influence both the residuals and the parameters, and in particular the sought coordinates.

As the type of unmodelled systematic errors is unknown, in a first step the user would like to have an indication as to whether there are such errors or not. The following situations indicate unmodelled systematic errors:

- The normalized residuals $\underline{z}_n = \widehat{\underline{v}}_n / \sigma_{\widehat{v}_n}$ or $\underline{X}_i = \widehat{\underline{\boldsymbol{v}}}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{\widehat{\underline{v}}_i \widehat{\underline{v}}_i}^{-1} \widehat{\underline{\boldsymbol{v}}}_i$ do not show a Gaussian or χ^2 distribution. Even if the original observations are not truly Gaussian distributed, the residuals, which are a weighted sum of many primary effects, are approximately Gaussian distributed if the assumed functional model holds. Therefore deviations from the Gaussian distribution are clear indicators of the presence of systematic errors.
- The reprojection errors $\widehat{\boldsymbol{v}}_i = \widehat{\boldsymbol{l}}_i - \boldsymbol{l}_i$ do not show a random pattern. A plot of residual vectors $\widehat{\boldsymbol{v}}_i(\boldsymbol{l}_i)$ as a function of the observed image coordinates $\boldsymbol{l}_i := \boldsymbol{x}_{it}$ aggregated from many or all images is a valuable tool for a quick check on the randomness of the residuals.
- The estimated variance factor $\widehat{\sigma}_0^2$ as well as the magnitude of the maximum residual decrease when taking only parts of the observations. Examples of such a reduction are
  - performing a free adjustment, i.e., not using control points, or
  - using only half of the images, e.g., omitting every second image in a strip with 80% endlap (overlap between neighbouring images), or
  - using only image pairs, i.e., omitting multiple view constraints.

  The reason for the effect is the following: with lower redundancy, i.e., fewer constraints, the unmodelled systematic errors – due to the smearing effect of least squares estimation – are more easily absorbed by the unknown parameters, thus the estimated variance factor decreases w.r.t. the one obtained with the complete bundle adjustment.

If there is an indication of unmodelled systematic errors, the functional model needs to be modified by extending the distortion model. In the most simple case this may be achieved by increasing the order of the basis functions (polynomials or periodic functions).

Care has to be taken that the image area is covered with observations and that these new parameters, $\boldsymbol{s}_n := \boldsymbol{s}_{\text{new}}$, are well-determinable.

15.5.4 Effect of Unmodelled Systematic Errors

Not modelling these new systematic errors, $\boldsymbol{s}_n$, will distort the result. Therefore the following analysis assumes these additional parameters to be specified. We thus assume the new systematic errors have been introduced and analyse the effect on the result if these parameters are not included.

The sensitivity of the resultant coordinates w.r.t. these new additional parameters may be evaluated similarly to evaluating the other additional parameters using the sensitivity factor,

$$\mu_{s_n k}^2 = \lambda_{\max}\left(W_{\widehat{s}_n\widehat{s}_n}^{(-k)}\Sigma_{\widehat{s}_n\widehat{s}_n} - I\right), \tag{15.183}$$

with

$$W_{\widehat{s}_n\widehat{s}_n}^{(-k)} := \overline{N}_{s_n s_n} = N_{s_n s_n} - N_{s_n p}N_{pp}^{-1}, N_{p s_n} \tag{15.184}$$

which depends on submatrices of the normal equation matrix N, referring to the transformation parameters $\boldsymbol{p}$ and the additional parameters $\boldsymbol{s}_n$, cf. Sect. 4.6.5.3, p. 134 and the discussion of (15.163), p. 694.

Then the effect of unmodelled systematic errors is bounded. With the standard deviation $\sigma_{\widehat{k}_u}$ of the scene coordinates and the test statistic $X_{s_n}^2 = \widehat{\boldsymbol{s}}_n^{\mathsf{T}}\Sigma_{\widehat{s}_n\widehat{s}_n}^{-1}\widehat{\boldsymbol{s}}_n$, we have the following inequality for the effect $\nabla_{s_n}k_u$ of not modelling the systematic errors on the coordinates:

$$\nabla_{s_n}\widehat{k}_u \le X_{s_n} \cdot \mu_{s_n k} \cdot \sigma_{\widehat{k}_u}. \tag{15.185}$$

If the user has specified tolerances d_u for the deviation of the estimated coordinates $\widehat{k}_u$ from their true values, we now arrive at the requirement

$$(k^2(P_{\max}) + X_{s_n}^2 \cdot \mu_{s_n k}^2)\,\widehat{\sigma}_0^2\,\sigma_{\widehat{k}_u}^2 \le d_u^2; \tag{15.186}$$

cf. (15.166). The second term, $X_{s_n}^2 \cdot \mu_{s_n k}^2$, may be much larger than $k^2(P_{\max})$, as in highly unstable cases $\mu_{s_n k} \gg 1$.

Visual odometry is a classical tool used in robotics. Wide angle lenses are known to yield stable results, whereas telelenses are known to yield inaccurate results. This is caused by both the weak geometry of the spatial resection and the susceptibility of the pose, especially of the position, to errors in the interior orientation. The situation is similar to estimating the scene geometry from aerial photos which are positioned using a GPS and without control points on the ground: then, errors in the interior orientation, especially the principal distance, lead to affine distortions of the 3D scene, especially an erroneous scale in the Z-direction. The sensitivity of the resultant orientation parameters $\widehat{\boldsymbol{p}}$ can easily be determined using the indicator

$$\mu_{sp}^2 = \lambda_{\max}(\Sigma_{\widehat{p}\widehat{p}}^{(-s)}W_{\widehat{p}\widehat{p}} - I) = \lambda_{\max}(W_{\widehat{s}\widehat{s}}^{(-p)}\Sigma_{\widehat{s}\widehat{s}} - I), \tag{15.187}$$

with

$$W_{\widehat{s}\widehat{s}}^{(-p)} = H^{\mathsf{T}}W_{ll}H. \tag{15.188}$$

15.5.5 Instability of Systematic Errors and Their Effect

If the systematic errors are not stable, calibration still is possible. However, the instability of the calibration will lead to residuals which are larger than is usually to be expected with a stable camera.

A reasonable model for an unstable camera would be one where the parameters are not fixed but random variables:

$$\underline{\boldsymbol{s}} \sim \mathcal{M}(\tilde{\boldsymbol{s}}, \Sigma_{ss})\,. \tag{15.189}$$

Then the observations will have the distribution

$$\underline{\boldsymbol{l}} \sim \mathcal{M}\left(\boldsymbol{f}(\tilde{\boldsymbol{k}}, \tilde{\boldsymbol{t}}, \tilde{\boldsymbol{s}}), \Sigma_{ll} + H\Sigma_{ss}H^{\mathsf{T}}\right)\,; \tag{15.190}$$

thus, they will be correlated, even if the parameters $\underline{\boldsymbol{s}}$ are uncorrelated; i.e., the instability of the interior orientation is caused by independent effects modelled by the chosen parameters.

An estimate for the instability Σ_{ss} of the additional parameters is necessary. This may be determined by repeated calibration, leading to several parameter vectors $\widehat{\boldsymbol{s}}_k$, $k = 1, ..., K$:

$$\Sigma_{ss} = \frac{1}{K-1}\sum_{k=1}^{K}(\widehat{\boldsymbol{s}}_k - \widehat{\boldsymbol{\mu}}_s)(\widehat{\boldsymbol{s}}_k - \widehat{\boldsymbol{\mu}}_s)^{\mathsf{T}} \quad \text{with} \quad \widehat{\boldsymbol{\mu}}_s^{\mathsf{T}} = \frac{1}{K}\sum_{k=1}^{K}\widehat{\boldsymbol{s}}_k\,. \tag{15.191}$$

If the self-calibrating bundle adjustment is performed with this model, specified by the matrix H, all resulting parameters are independent of the covariance matrix Σ_{ss} due to Lemma 5a in (Rao, 1967). Hence the estimated parameters are not influenced by the variation of the systematic errors.

However, the resultant covariance matrix of the parameters will be too optimistic if the second term in the covariance matrix in (15.190) is neglected, and – as is to be expected – wrong if Σ_{ss} does not correctly reflect the variations of the additional parameters. From the basic estimate of all parameters not involving the variations of additional parameters $\boldsymbol{s}$

$$\widehat{\Delta\boldsymbol{x}}_0 = N_0^{-1}A^{\mathsf{T}}W_0\Delta\boldsymbol{l} \quad \text{with} \quad W_0^{-1} = \Sigma_0 = \Sigma_{ll} \quad \text{and} \quad N_0 = A^{\mathsf{T}}W_0A\,, \tag{15.192}$$

we obtain the realistic covariance of the estimated parameters,

$$\Sigma_{\widehat{x}\widehat{x}} = N_0^{-1}A^{\mathsf{T}}W_0(\Sigma_0 + H\Sigma_{ss}H^{\mathsf{T}})W_0AN_0^{-1} = \Sigma_{\widehat{x}\widehat{x},0} + \Delta\Sigma_{\widehat{x}\widehat{x}}\,, \tag{15.193}$$

with

$$\Sigma_{\widehat{x}\widehat{x},0} = N_0^{-1} \quad \text{and} \quad \Delta\Sigma_{\widehat{x}\widehat{x}} = \Sigma_{\widehat{x}\widehat{x}} - \Sigma_{\widehat{x}\widehat{x},0} = N_0^{-1}A^{\mathsf{T}}W_0H\Sigma_{ss}H^{\mathsf{T}}W_0AN_0^{-1}\,. \tag{15.194}$$

The matrix $\Delta\Sigma_{\widehat{x}\widehat{x}}$ is positive semi-definite, and indicates how much the uncertainty of the estimated parameters increases due to the variation of the systematic errors, cf. (15.193).

15.5.6 An Example for Evaluating Calibration Parameters

We will now demonstrate the power of the statistical evaluation of a bundle adjustment by using a camera calibration as an example.

We investigate the quality of the five basic calibration parameters,

$$\boldsymbol{s} = [c, x'_h, y'_h, s, m]^{\mathsf{T}}\,, \tag{15.195}$$

of the interior orientation using a free self-calibrating bundle adjustment, taking as gauge the centroid of the approximate values of the scene points, cf. (15.88), p. 665. These five parameters are treated as additional parameters compared to a bundle adjustment without self-calibration.
We assume the following configuration, see Fig. 15.21:

- The camera has N_x columns and N_y rows, and the principal distance is c, all measures given in pixel. We assume the image size to be 768×1024. We use the principal distances $c = 1000$ pixel and 2000 pixel.
- The test field consists of I points. It is arranged in a square of side length $w = 400$ m centred at the origin of the scene coordinate system. We vary I from 16, via 25, to 36. The Z-coordinates of all points except for the inner ones are zero; the inner ones have height $Z = h$. We vary h to be 10 m and 100 m. By reinterpreting the numbers, we could imagine an indoor test field with side length 400 mm.
- The camera centres lie on a sphere around the origin at a distance D such that all scene points are visible in all images. We neglect possible occlusions, which may appear in real situations. We determine D from

$$D = 1.4\, w \frac{c}{\min(N_x, N_y)}\,, \tag{15.196}$$

  the factor 1.4 guaranteeing that all scene points are visible in the image. The distance D thus implicitly controls the viewing angle of the camera.
- Except for the centre image at $[0, 0, D]^\mathsf{T}$, we have $1, ..., N_r$ rings of cameras at different zenith angles ϕ, each ring having four positions. When using one ring, we use $\phi = 30°$; when using two rings, we use $\phi = 25°$ and $50°$.
- The camera's roll angle κ, i.e., the angle around the viewing direction, is derived from the centre camera by minimal rotation in four directions. We use several sets of angles κ: $\{0°\}$, $\{0°, 180°\}$, $\{0°, 90°\}$, and $\{0°, 90°, 180°, 270°\}$.
- The image coordinates have a standard deviation of σ_x; usually we assume $\sigma_x = 0.3$ pixel.

We use the following criteria for evaluating the calibration:

- The average precision $\overline{\sigma_k}$ of the scene coordinates.
- The standard deviations $\sigma_{\hat{s}}$ of the five parameters, the principal distance, the coordinates of the principal point in pixel, the scale and the shear, which have unit 1; these standard deviations should be small.
  The checkability parameters $\nabla_0 s = \delta_0 \sigma_{\hat{s}}$ directly depend on the standard deviation σ_s and therefore are not shown. They indicate how large the deviation of a parameter from a nominal value must be in order to be detectable using a statistical test. We use $\delta_0 = 4$ in the discussion.
- The sensitivity factor $\mu_{sx} := \mu_s$ from (4.328), p. 135 of each parameter s referring to the influence on all other parameters $\widehat{\boldsymbol{x}}$. The factor μ_{sx} indicates the maximum influence on the parameters $\widehat{\boldsymbol{x}}$ if the change of the additional parameters $\boldsymbol{s}$ is identical to their standard deviation. These and the following sensitivity factors should be small. Values below 10 are already very good. Values beyond 30 are bad and indicate a configuration which is highly sensitive to wrong additional parameters.
- The sensitivity factor μ_{sp} of the individual additional parameters $\boldsymbol{s}$ referring only to the influence on the orientation parameters $\widehat{\boldsymbol{p}}$. This is the maximum influence on the parameters $\widehat{\boldsymbol{p}}$ if the change of the additional parameters $\boldsymbol{s}$ is assumed to be identical to their standard deviation.
- The sensitivity factor μ_{sk} of the individual additional parameters $\boldsymbol{s}$ referring only to the influence on the scene coordinates $\widehat{\boldsymbol{k}}$.

We obtain the results for nine cases, which are summarized in Table 15.7.

1. We start with the configuration with the least number of images: a central image together with one ring of four images with one roll angle κ, i.e., five images. The scene

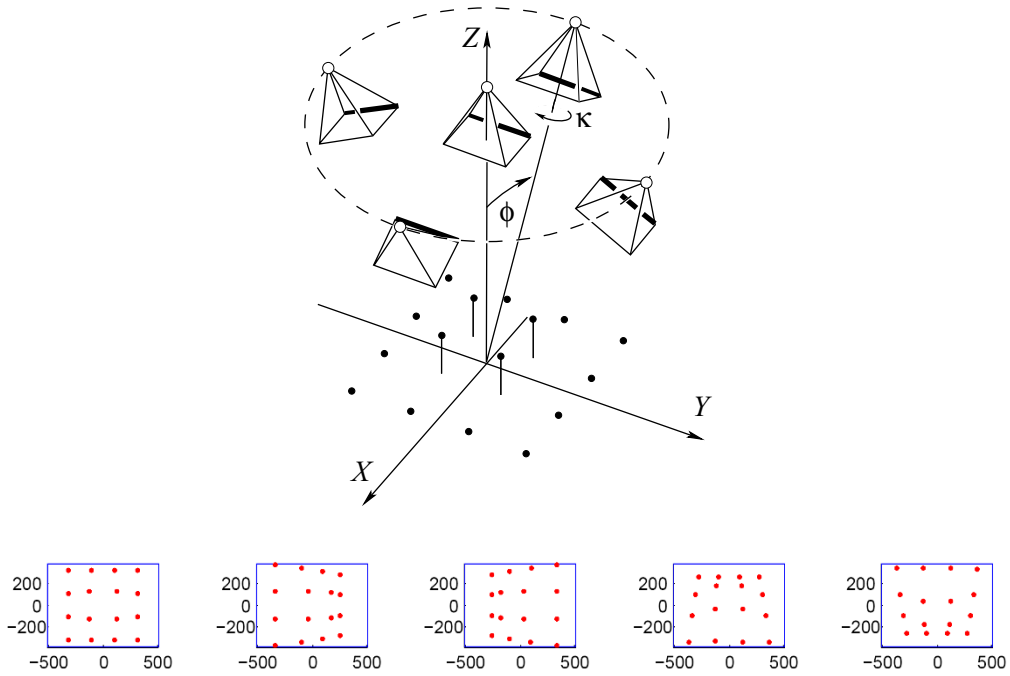


Fig. 15.21 Basic setup for camera calibration for the example. **Top:** We have one ring of four poses at a zenith distance of $\phi = 30°$. All cameras show the same roll angle κ: Therefore at least one camera should be rotated by 90° around the viewing direction. In the analysis we assume at all positions we have taken one, two or four images with different roll angles κ. **Bottom row:** The image points need to cover the complete image area

consists of 16 points, and a height of 100 m in the middle. The principal distance is $c = 1000$ pixel. As a result, the distance of the cameras from the origin is $D = 729$ m, see Fig. 15.21. This corresponds to a camera with narrow angle lens.

The standard deviations of the parameters of the interior orientation are given first: The principal distance can be determined with a relative accuracy of approximately 2.2‰. The principal point location is uncertain with a standard deviation of approximately 8.8 pixel, which is large compared to the uncertainty of the image coordinates of 0.3 pixel.

We now regard a change of the parameters of the interior orientation by values equal to the size of their individual standard deviation. The effect of such a change on some function (value) $f = f(\hat{\boldsymbol{x}})$ of the parameters, i.e., also the parameters, is up to approximately 263 times the standard deviation σ_f of that function (value). The maximum effect on the orientation with $\mu_{x_h p} = 38.6$ is approximately three times larger than the maximum effect μ_{sk} on the scene coordinates. Obviously, small changes of all interior parameters except the principal distance have a large effect on the exterior orientation.

2. We now choose the same configuration, except that at each camera position we add a pose with a roll angle κ of 180°, i.e., with ten images altogether.

   The main effect is the reduction of the instability of the configuration w.r.t. the position of the principal point. The standard deviation decreases by a factor of approximately 12, down to 0.6 pixel. Errors in the principal point have practically no effect on the scene coordinates ($\mu_{x_h k} = 0.0046$; this value for $\boldsymbol{x}_h$ appears as 0.0 in the third column from the right).
3. We now choose the same configuration, however, with camera roll angles κ varying by 90°, i.e., 20 images altogether.

   The main effect is the stabilization of the shear s and the scale difference m. They are now determinable with a standard deviation of approximately 0.1‰. Their effect on

Table 15.7 Results of calibrations. We characterize each case by the principal distance c, the height h of the centre points in the test field, the number I of scene points, the number of rings of images, the set of roll angles κ and the total number T of images taken. The resultant values used for the evaluation are the average standard deviation $\overline{\sigma_{\widehat{X}}}$ of the estimated scene points; for each of the additional parameters their standard deviation $\sigma_{\widehat{s}}$ from variance propagation; the sensitivity factors μ_{sx} w.r.t. all parameters, scene points and orientation parameters; the sensitivity factor μ_{sp} w.r.t. the orientation parameters; and the sensitivity factor μ_{sk} w.r.t. the scene coordinates for the individual additional parameters $\boldsymbol{s}$, namely the principal distance c, the coordinates $\boldsymbol{x}_h$ of the principal point, the shear, and the scale difference m

case	c [pixel]	h [m]	I	rings	κ [°]	T	$\overline{\sigma_{\widehat{X}}}$ [m]		$\boldsymbol{s}$ c	$\boldsymbol{x}_h$	shear	m
1	1000	100	16	1	0°	5	0.43	$\sigma_{\widehat{s}}$	2.2 [pixel]	8.8 [pixel]	3.3‰	2.3‰
								μ_{sx}	18.9	263.0	19.4	13.9
								μ_{sp}	1.0	38.6	3.7	1.5
								μ_{sk}	0.9	10.3	12.5	8.6
2	1000	100	16	1	**0°, 180°**	10	0.27	$\sigma_{\widehat{s}}$	1.6 [pixel]	0.6 [pixel]	2.3‰	1.7‰
								μ_{sx}	18.9	25.5	19.4	13.9
								μ_{sp}	1.0	25.5	3.7	1.5
								μ_{sk}	0.9	0.0	12.5	8.6
3	1000	100	16	1	**0°, 90°, 180°, 270°**	20	0.26	$\sigma_{\widehat{s}}$	1.0 [pixel]	0.4 [pixel]	0.13‰	0.14‰
								μ_{sx}	16.1	25.4	1.2	1.3
								μ_{sp}	0.7	25.4	0.5	0.6
								μ_{sk}	0.5	0.0	0.0	0.1
4	1000	100	16	1	**0°, 90°**	10	0.27	$\sigma_{\widehat{s}}$	1.3 [pixel]	0.8 [pixel]	0.18‰	0.19‰
								μ_{sx}	16.1	35.7	1.2	1.3
								μ_{sp}	0.7	25.5	0.5	0.6
								μ_{sk}	0.5	1.0	0.0	0.1
5	1000	**10**	16	1	0°, 90°, 180°, 270°	20	0.32	$\sigma_{\widehat{s}}$	1.0 [pixel]	0.4 [pixel]	0.17‰	0.18‰
								μ_{sx}	15.9	26.5	1.7	1.8
								μ_{sp}	0.7	26.5	1.0	1.1
								μ_{sk}	0.4	0.0	0.0	0.1
6	1000	100	**36**	1	0°, 90°, 180°, 270°	20	0.14	$\sigma_{\widehat{s}}$	0.7 [pixel]	0.3 [pixel]	0.10‰	0.11‰
								μ_{sx}	16.7	28.5	1.3	1.4
								μ_{sp}	0.6	28.5	0.6	0.7
								μ_{sk}	0.5	0.0	0.0	0.1
7	1000	100	16	**2**	0°, 90°, 180°, 270°	**36**	0.23	$\sigma_{\widehat{s}}$	0.5 [pixel]	0.3 [pixel]	0.11‰	0.11‰
								μ_{sx}	11.5	25.4	1.3	1.4
								μ_{sp}	0.6	25.4	0.7	0.7
								μ_{sk}	0.4	0.0	0.0	0.1
8	**2000**	100	16	1	0°, 90°, 180°, 270°	20	0.76	$\sigma_{\widehat{s}}$	3.4 [pixel]	1.6 [pixel]	0.14‰	0.15‰
								μ_{sx}	28.1	94.0	1.4	1.4
								μ_{sp}	0.4	94.0	0.7	0.7
								μ_{sk}	0.5	0.0	0.0	0.0
9	**700**	100	16	1	0°, 90°, 180°, 270°	20	0.15	$\sigma_{\widehat{s}}$	0.5 [pixel]	0.2 [pixel]	0.12‰	0.13‰
								μ_{sx}	13.0	13.3	1.1	1.2
								μ_{sp}	1.0	13.3	0.4	0.5
								μ_{sk}	0.6	0.0	0.0	0.1

the orientation and the scene parameter becomes negligible. This is intuitive, as the 90° roll angles κ allow their determination, in contrast to the previous setting.

4. Therefore we now investigate the effect of only two roll angles κ, but with 90° apart, thus again ten images altogether.
   The result is practically the same as in 3. Only the principal point is a bit worse in precision and has a slightly larger effect on the other parameters.

   All results have a high possible influence of the principal point on the orientation parameters in common. Taking an error of $3\sigma_{x_h}$, in the principal point its maximum effect on the orientation is up to $3\mu_{sp} \approx 2 \times 25.5 = 51$ times its standard deviation. The effect will be mainly on the angles ω and ϕ and on the position across the viewing direction.

We now start from experiment 3 and vary one of the other parameters of the configuration, while still leaving the principal distance the same, $c = 1000$ pixel.

5. Changing the height of the interior points from 100 m to 10 m does not have a large influence compared to the result of experiment 3. Viewing the practically flat scene from different directions ensures enough depth variation w.r.t. the camera.
6. We now increase the number of scene points from 16 to 36. This should decrease the standard deviations of the scene and orientation parameters by roughly a factor of $\sqrt{16/36} \approx 0.67$, since more points per image increase the precision of the orientation parameters, and this increase transfers to the triangulated scene points. Actually we observe such an improvement, but the sensitivity factors, however, do not change much. Thus increasing the number of scene points just increases the precision, not the stability.
7. If we increase the number of cameras by adding a second ring, the number of images increases from 20 to 36 and we observe a similar effect as when increasing the number of image points. The precision increases. However, the sensitivity of the orientation parameters w.r.t. changes in the principal point is still high: Errors in the principal point are compensated for a rotation of the camera. This may not be acceptable if the bundle adjustment is used for pose estimation, e.g., during ego-motion determination of a video camera.

Finally, we investigate the precision and stability when using tele or normal angle lenses.

8. We first assume a telelens with $c = 2000$ pixel, differing from the wide angle lens by a factor of 2. The standard deviations of the scene points increase to 0.48 m, approximately by a factor of 2.5. This is mainly due to the distance of the cameras from the test field, which is longer by a factor of 2. Due to the viewing angle, which is smaller by a factor of 2, the precision of both the principal distance and the principal point is worse by a factor of approximately 4, as is to be expected. However, the precisions of the shear and scale difference do not change. While the sensitivity factor μ_{cp} of the orientation parameters increases due to a change in the principal distance, the factor μ_{cs} decreases from 0.8 to 0.4, the sensitivity factor $\mu_{x_h p}$ of the orientation parameters decreases dramatically due to a change in the principal distance, and the sensitivity factor $\mu_{x_h s}$ increases from 19.2 to 69.9. Interestingly, the effect of errors in the calibration is of the same order as for normal angle lenses; it only increases due to the loss in precision.
9. We finally assume a normal angle lens with $c = 700$ pixel, which differs from the narrow angle lens only by a factor of 1.4. The precision of the scene points is now 0.1 m, smaller than for the wide angle lens by a factor of 2. Similarly, the standard deviations for the principal distance and the principal point decrease nearly by a factor of 2, while again the shear and the scale difference keep their precision. The sensitivity of the orientations parameters w.r.t. principal distance and principal point shows the opposite change, as in the previous case: The effect of errors in the principal distance are comparably larger; the factor μ_{cp} in the fourth last column changes from 0.7 to 1.1, which is still fully acceptable. The effect of errors in the principal point, however, are comparably smaller, the factor $\mu_{x_h p}$ decreasing from 19.2 to 10.2 – the first time we observe such a small sensitivity factor $\mu_{x_h p}$ for the principal point. Therefore, if the task of the bundle adjustment is ego-motion determination, larger fields of view are advisable. The sensitivity factors for the scene points are all acceptable if at least two roll angles κ with a difference of 90°are chosen.

The detailed analysis supports the basic rules, cf. Sect. 15.5.1, p. 698, for a camera configuration ensuring reliable camera calibration. Moreover, we found the following:

- The identification or definition accuracy of the image points, which need to cover the whole image area, should be comparable to the envisaged precision of the project. However, a larger number of scene points or views can compensate for a lower precision of image point identification. As a rule of thumb, the number of scene points and images taken needs to be at least four times as large if the image point identification decreases by a factor of 2 in the standard deviation. This also allows us to take natural points, not targeted ones, if their identification across the images can be performed

with sufficient precision. For example, Lowe key points empirically show a precision of approximately 0.3 to 0.5 pixel, cf. the Sect. 12.2.1.1, p. 491 and Förstner et al. (2009).

- A larger number of points only increases the precision of the result, not the stability. Even 16 points are sufficient to determine the five parameters of the calibration matrix. More may be necessary if additional parameters are to be determined (see below). The 3D coordinates of the scene points need not be known precisely, as they are determined in a self-calibrating bundle adjustment.
- The total set of all scene points needs to have a 3D structure with respect to the coordinate system of the camera. This can be realized in the following ways:
  - As shown in Fig. 15.20, p. 698, the test field may be flat and the viewing directions vary.
  - The camera is fixed, and the test field is flat and moved, leading to several tilt angles w.r.t. the camera.

  Rough approximate parameters for the principal distance and the principal point allow us to determine sufficiently accurate approximate values for the camera poses to start the bundle adjustment. Four different tilt angles of approximately 30°appear sufficient. A second ring, i.e., a second set of tilted planes with a different tilt angle, increases the precision.

  If the camera views are arranged in a planar grid and have a common viewing direction, as in classical aerotriangulation, the test field needs to be structured in 3D.
- The rotation angle κ around the viewing direction needs to vary. At least one pair should show a difference in κ of 90°.

15.6 Outlier Detection and Approximate Values

15.6.1 Outlier Detection 707
15.6.2 Sequential Solutions for Determining Approximate Values 708
15.6.3 Direct Solutions for Block Adjustment 711

Generally, there exists no direct solution for bundle adjustment. Therefore approximate values for all parameters need to be available. Their determination needs to be accompanied by outlier detection.

15.6.1 Outlier Detection

Sect. 4.7, p. 141 discusses various methods for outlier detection which can be used to advantage. In the following, we address topics specific to outlier detection in bundle blocks which have an influence on both the design of the configuration and the strategy for data cleaning.

The percentage of outliers in correspondences derived by matching procedures is between 1% and 50% approximately. In addition outliers may be (1) large, hindering convergence of iterative procedures, (2) of medium size, not hindering convergence, say significantly below 10% of the image diameter, or (3) small, i.e., hardly detectable, say in a range below 20-30 times the standard deviation of the measurements. There is no generally accepted strategy for finding all outliers, nor is there a guarantee that any strategy will find all outliers, as the complexity of the problem is in the order of 2^N, where N is the number of observations.

This is the reason why all proposed strategies start to search for outliers in small geometric units, say one, two, or three images, our motivation to discuss the various direct solutions for estimating orientation and scene parameters in the previous sections. Though

checking in small units is a necessary component of outlier detection, it is not sufficient for several reasons:

1. The detectability of outliers in small geometric units is lower than in large units, sometimes significantly lower. Take, for example, the relative orientation of two images: Outliers which happen to only have a component along the epipolar line cannot be detected at all.
2. Even if outliers are detectable, they may not be locatable, i.e., no decision can be made as to which of the involved observations actually is erroneous. Take, for example, the spatial resection: If the collinearity equation $\mathbf{x}'_{it} = \mathsf{P}_t \mathbf{X}_i$ is violated, it remains unclear whether the image point $\mathbf{x}'_{it}$, the scene point $\mathbf{X}_i$, or even the projection matrix P_t is erroneous.
3. All outlier tests implicitly assume that only one outlier exists – an unrealistic assumption. Therefore outliers of any size generally cannot be detected in the presence of outliers which are one magnitude larger if the analysis is based on the residuals. This especially holds for small outliers in the presence of systematic errors.
   This is the reason why finding inliers by some RANSAC-type procedure is frequently used, especially for eliminating medium and large outliers. However, conceptually, RANSAC does not guarantee providing a solution, especially in weak geometric configurations. Therefore statistical testing as such is not a sufficient remedy, so also additional tests on the configuration need to be performed.

outlier detection in groups of images

All three reasons provide a motivation for using larger units such as image triplets in order to check for outliers. Such larger units show a geometry which often is close to the geometry of the complete block and therefore allow a large percentage of the small outliers to be found.

This is why detecting outliers and determining approximate values for bundle adjustment are intimately linked. The process therefore consists of two steps:

1. Finding large and medium-sized outliers in small overlapping image sets.
2. Determining approximate values with observations with only small remaining outliers.

Sequential procedures, which are presented next, can be used in all situations and combined with outlier detection procedures. We close the section with direct solutions for bundle adjustments.

15.6.2 Sequential Solutions for Determining Approximate Values

Sequential solutions for determining approximate values for bundle adjustment inherently are suboptimal and depend on the chosen order of computations. There exist two strategies to approach this problem, one based on spatial resections of single images and one on relative orientations of image pairs, from which more general procedures can be derived.

15.6.2.1 Sequential Spatial Resections of Single images

Starting from an image pair or an image triplet (Reich et al., 2013), the first method builds photogrammetric models by sequentially adding one image to the existing set of images.

The orientation of the new image is determined by spatial resection. If available, control points are used to perform the absolute orientation of the photogrammetric model of all images, transforming it into the scene coordinate system.

Algorithm 22 collects the essential steps.

1-2 *Select two images and build the photogrammetric model.*
The coordinate system and scale of the photogrammetric model refer to this first image pair, say with the coordinate system, which is the one of the first camera and

Algorithm 22: Sequential spatial resections for determining approximate values for bundle adjustment
$\mathcal{M}_T$= `Sequential_Spatial_Resections` $(\{\boldsymbol{s}_t, \{\mathcal{K}'_{it}\}\})$
Input: set of images $t = 1, ..., T$ with parameters of interior orientation $\boldsymbol{s}_t$ and image points $\mathcal{K}'_{it}$.
Output: a photogrammetric model $\mathcal{M}_T$ formed by all images.

```
Initiate: τ = 2, Select two images {t1, t2};
Build photogrammetric model: M_τ = {{p_t1, p_t2}, {k_i, i ∈ I_2}};
for τ = 3, ..., T do
    Select image t_τ;
    Perform spatial resection: p_t_τ using points in model M_τ−1;
    Determine new scene points by triangulation: ΔK_T = {k_i, i ∈ ΔI_τ};
    Update photogrammetric model: M_τ = M_τ−1 ∪ {p_t_τ, ΔK_T}.
end
```

the scale is defined by the length of the basis. Therefore the relative orientation should be stable, i.e., with small variances, and the triangulation of the scene points should yield precise 3D coordinates. This can be achieved by choosing an image pair where the average roundness of the standard ellipsoids of the 3D points is above a certain threshold (see Beder and Steffen, 2006),

$$\frac{\lambda_3}{\lambda_1} \geq T\,, \tag{15.197}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the three eigenvalues of the covariance matrix of a scene point and T is an adequate threshold, e.g., T=0.1. This guarantees that the scene points have a minimum stability, independent of the scale. If possible, the number of corresponding points should be larger than 30 to ensure high enough sensitivity w.r.t. outliers. For a small number of images, all image pairs may be checked for (15.197); otherwise, the first image pair fulfilling (15.197) may be chosen. The sequence of testing image pairs may depend on the number of putative matches between two images or another qualitative measure for the overlap.

4-5 *Select a new image and determine its pose relative to the existing photogrammetric model using a spatial resection.*
Again, the selection should guarantee a stable determination of the pose. This can be achieved by selecting an image where sufficiently many image points whose 3D coordinates are available in the photogrammetric model cover a sufficiently large area of the image. In principle, we could also exploit the yet unused rays of previous images, where no scene points are available, but which correspond to image points in the new image. As there is no closed form solution for the spatial resection with scene points and scene rays, this information could be used in a refining step. This allows for outlier detection based on the coplanarity constraint as well as increasing the precision of the new scene points, derived in the next step.

6 *Determine new scene points by triangulation.*
The scene points used for the initial spatial resection may remain unchanged or may be improved by re-triangulation, i.e., using all rays referring to the scene point of concern. The 3D coordinates of all scene points observed in the new image and in previous ones are determined by triangulation.

7 *The photogrammetric model is updated by extending the orientation parameters by the one of the new image and the list of scene points by the set of new points.*
The 3D coordinates of the already existing 3D points may be updated, as mentioned before. Generalizing this idea, we could take this updating step as one in an incremental bundle adjustment, e.g., by using the sequential estimation procedure given in Sect. 4.2.7.2, p. 96. The software iSAM2 provided by Kaess et al. (2012) supports a rigorous

sequential estimation. When applied to our scenario, the updated photogrammetric model would be identical to the one obtained by a rigorous bundle adjustment using the images, up to the current step.

15.6.2.2 Sequential Similarity Transformations of Photogrammetric Models

The second method starts with determining all feasible relative orientations within the bundle block and then determining the camera poses of the photogrammetric model of all images, either in closed form or sequentially. Again, control points can be used to transform this model into the scene coordinate system.

Algorithm 23 gives the process using relative orientations.

Algorithm 23: Sequential similarity transformations for determining approximate values for bundle adjustment

$\mathcal{M}_T$= `Sequential_Similarity_Transformations` $(\{\boldsymbol{s}_t, \{\mathcal{X}'_{it}\}\})$

Input: set of images $t = 1, ..., T$ with parameters of interior orientation $\boldsymbol{s}_t$ and image points $\mathcal{X}'_{it}$.

Output: a photogrammetric model $\mathcal{M}_T$ formed by all images.

```
1  for all overlapping pairs {s,t} with s,t ∈ T of images do
2  |  Determine photogrammetric model M_st;
3  |  Check sufficiently many image triplets involving {s,t} for outliers;
4  end
5  Initiate: τ = 2, Select first image pair {t1,t2} with model M_τ ∈ {M_t1t2};
6  for τ = 3,...,T do
7  |  Select image pair {s,t_τ} with image s used in M_τ−1;
8  |  Determine scale λ_s,t_τ from some image triplet involving {s,t_τ} and similarity s → t_τ ;
9  |  Determine pose p_t_τ and new scene points ΔX_T by similarity transformation;
10 |  Update photogrammetric model: M_τ = M_τ−1 ∪ {p_t_τ, ΔX_T}.
11 end
```

1-2 Determine all available photogrammetric models $\{s, t\}$ with images $s, t \in \mathcal{T}$. Generally this requires checking all pairs of images for common points.
The relative orientations need to be determined for two purposes: outlier detection, see below, and quality evaluation. The sequencing of the image pairs depends on (1) the accuracy of the relative orientation and (2) the quality of the triangulated scene points. Given a sufficiently accurate relative orientation, the quality of a triangulated point mainly depends on the parallactic angle; thus, only these angles need to be determined in this step.

3 For each image pair, check sufficiently many image triplets containing it. This serves three purposes: Besides outlier detection and quality evaluation, triplets are necessary to transfer the scale from one image pair to an overlapping one. Therefore the ratio of the lengths of the base lines should be sufficiently accurate. The accuracy of the ratio mainly depends on the number of outlier-free corresponding points in all three images. This criterion also excludes image triplets where at least two projection centres are very close to each other.

5 The first image pair again defines the coordinate system and the scale of the photogrammetric model of all images and can be selected using the criterion above, see (15.197), p. 709.

7 The next image pair $\{s, t_\tau\}$ is used to add image t_τ to the photogrammetric model. Therefore the scale transfer needs to be reliable. This is guaranteed if there are enough points in the triplet $\{r, s, t_\tau\}$ with image pair $\{r, s\}$ in the previous model $\mathcal{M}_{\tau-1}$.

8 The scale ratio λ_{s,t_τ} can be easily determined from (14.65), p. 634. The similarity transformation from the photogrammetric model $\mathcal{M}_{st_\tau}$ of the image pair $\{s, t_\tau\}$ to

the photogrammetric model $\mathcal{M}_{\tau-1}$ of the previous set of images can be derived with the method described in Sect. 10.5.4.3, p. 408. The scene points common to both photogrammetric models may be updated using the information from the new image pair $\{s, t_\tau\}$.

9-10 The new scene points collected in the set $\Delta\mathcal{K}_\tau$ are added to photogrammetric model. The scene points common to all three images may be corrected. Again, a rigorous incremental bundle adjustment may be realized, but it needs to be based on the images rather than on the photogrammetric models. This is because the sequential estimation assumes the new information to be statistically independent of the already used information, whereas overlapping models have correlated scene points.

15.6.2.3 Reducing Drift of Approximate Values in Sequential Methods

Both sequential strategies have the disadvantage that for large sets of images the pose parameters drift during the incremental process. One remedy is to perform a bundle adjustment after several steps in order to reduce the drift effect. Klopschitz et al. (2010) therefore propose working with checked triplets, starting with stable triplets in parallel, and incrementally building the photogrammetric model of all images by adding image triplets.

Very effective is the following procedure:

1. Partition the block into sub-blocks. This may be performed sequentially, e.g., by following a sequential strategy and starting a new sub-block every T_s images. A partitioning may alternatively be performed by recursively splitting the block into sub-blocks until a certain size T_s is reached and all images within a sub-block are connected to at least two others in order to allow for scale transfer within the sub-block.
2. Determine approximate values in each sub-block using one of the sequential methods described above.
3. Perform a rigorous bundle adjustment for each sub-block. This yields a set of photogrammetric models each in a local coordinate system with arbitrary translation, rotation, and scale.
4. Fuse the sub-blocks using a spatial block adjustment, see Sect. 15.2.2.2, p. 649, based on common scene points and projection centres. Control points can be used to reach a specified coordinate system for the resulting block. The estimated transformation parameters can be used to derive approximate values for all scene points and all images.

This procedure has the advantage of simultaneously determining the approximate values of all parameters and thus avoiding drift effects.

15.6.3 Direct Solutions for Block Adjustment

We present four classical methods for a direct solution of block adjustment which are based on a specific setup, and all of which lead to a mathematical model where the observations and the unknown parameters are linked by a linear or bilinear relation.

15.6.3.1 Linear Planar Block Adjustment

There exists a *direct approximate solution for the planar model block adjustment* discussed in Sect. 15.3.1, p. 651. This solution is useful for determining approximate values for bundle adjustment if the images are vertical views, i.e., two of the rotation angles are approximately known and the scene is comparably flat, with distances from the images

varying by a few percent. The solution can easily be derived by inverting the original transformation from scene to image space.

The original observation equation, see (15.23), p. 651, now making the observed image point coordinates $\boldsymbol{l}'_{ij}$ and the residuals $\boldsymbol{v}'_{ij}$ explicit, is given by

$$\boldsymbol{l}'_{it} + \boldsymbol{v}'_{it} = \boldsymbol{r}_t + Z(\boldsymbol{s}_t)\boldsymbol{k}_i \qquad (it) \in \mathcal{E} \tag{15.198}$$

with the translation, $\boldsymbol{r}_t$, and the scale and rotation, $\boldsymbol{s}_t$, of the image t, and where $\boldsymbol{k}_i$ contains the ith scene coordinates. Multiplication with $Z^{-1}(\boldsymbol{s}_t)$ yields

$$Z^{-1}(\boldsymbol{s}_t)(\boldsymbol{l}'_{it} + \boldsymbol{v}'_{it}) = Z^{-1}(\boldsymbol{s}_t)\boldsymbol{r}_t + \boldsymbol{k}_i\,; \tag{15.199}$$

or, with the modified transformation parameters $\bar{\boldsymbol{s}}_t$,

$$Z(\bar{\boldsymbol{s}}_t) = Z^{-1}(\boldsymbol{s}_t)\,, \quad \text{or} \quad \begin{bmatrix} \bar{c}_t \\ \bar{d}_t \end{bmatrix} = \frac{1}{c_t^2 + d_t^2} \begin{bmatrix} c_t \\ -d_t \end{bmatrix}, \quad \text{and} \quad \bar{\boldsymbol{r}}_t = Z^{-1}(\boldsymbol{s}_t)\boldsymbol{r}_t\,, \tag{15.200}$$

and the modified residuals

$$\bar{\boldsymbol{v}}_{it} = Z(\bar{\boldsymbol{s}}_t)\boldsymbol{v}'_{it}\,. \tag{15.201}$$

When solving for $\bar{\boldsymbol{v}}_{it}$, we have

$$\bar{\boldsymbol{v}}_{it}(\boldsymbol{k}_i, \boldsymbol{r}_t, \boldsymbol{s}_t) = \bar{\boldsymbol{r}}_t + \boldsymbol{k}_i - Z(\boldsymbol{l}'_{it})\bar{\boldsymbol{s}}_t\,. \tag{15.202}$$

Together with the observation equations for the control points,

$$\boldsymbol{v}_{i0}(\boldsymbol{k}_i) = -\boldsymbol{l}_{i0} + \boldsymbol{k}_i\,, \tag{15.203}$$

the expressions for the residuals $\bar{\boldsymbol{v}}_{it}$ and $\boldsymbol{v}_{i0}$ are linear in all unknown parameters, namely in the coordinates $\boldsymbol{k}_i$ of the scene points and the modified transformation parameters $\bar{\boldsymbol{r}}_t$ and $\bar{\boldsymbol{s}}_t$.

Minimizing

$$\Omega(\{\boldsymbol{k}_i\}, \{(\bar{\boldsymbol{r}}, \bar{\boldsymbol{s}})_t\}) = \sum_{it \in \mathcal{E}} |\bar{\boldsymbol{v}}_{it}(\boldsymbol{k}_i, \boldsymbol{r}_t, \boldsymbol{s}_t)|^2 + \sum_{i \in \mathcal{I}_0} |\boldsymbol{v}_{i0}(\boldsymbol{k}_i)|^2 \tag{15.204}$$

w.r.t. the unknown parameters therefore leads to a linear equation system for all unknowns. The original parameters can easily be determined from[3]

$$\boldsymbol{s}_t = \begin{bmatrix} c_t \\ d_t \end{bmatrix} = \frac{1}{\bar{c}_t^2 + \bar{d}_t^2} \begin{bmatrix} \bar{c}_t \\ -\bar{d}_t \end{bmatrix}, \qquad \boldsymbol{r}_t = Z(\boldsymbol{s}_t)\bar{\boldsymbol{r}}_t\,. \tag{15.205}$$

The solution is approximate, as the weighting of the residuals $\bar{\boldsymbol{v}}_{it}$ depends on the scale parameter $|\boldsymbol{s}_t|$ of the corresponding model and the coefficients depend on the noisy observations $\boldsymbol{l}_{it}$. The more these scales are homogeneous, i.e., identical over all models, and the smaller the noise variance is, the closer the solution is to the optimal one.

15.6.3.2 Linear Spatial Bundle Adjustment for Given Rotations

Spatial bundle adjustment is linear if the cameras are calibrated and the rotation matrices are known, or, equivalently, if the infinite homography $\mathsf{H}_t = \mathsf{K}_t\mathsf{R}_t$ is known for all images. We will discuss preconditions under which this knowledge may be available.

If the calibration and the rotation matrices are known or approximately known, the model will be linear and reads

$${}^n\mathbf{x}_{it} + \boldsymbol{v}_{it} = \lambda_{it}(\boldsymbol{X}_i - \boldsymbol{Z}_t)\,, \quad \text{with} \quad {}^n\mathbf{x}_{it} = (\mathsf{K}_t\mathsf{R}_t)^{-1}\mathbf{x}_{it}\,, \tag{15.206}$$

[3] Observe: when interpreting s_t as complex number, $\bar{s}_t = s_t^{-1}$, thus $s_t = \bar{s}_t^{-1}$, see (15.200).

which is the most simple form of the collinearity equation using the model of a normalized camera, see (12.39), p. 473. By multiplication with $S(^n\mathbf{x}_{it})$, we eliminate the scale parameters λ_{it} and obtain

$$\bar{\boldsymbol{v}}_{it}(\boldsymbol{X}_i, \boldsymbol{Z}_t) = S(^n\mathbf{x}_{it})\boldsymbol{v}_{it} = [S(^n\mathbf{x}_{it}) \mid -S(^n\mathbf{x}_{it})] \begin{bmatrix} \boldsymbol{X}_i \\ \boldsymbol{Z}_t \end{bmatrix} . \tag{15.207}$$

Together with some control points, where we have the observation equation

$$\boldsymbol{v}_{i0}(\boldsymbol{X}_i) = -\boldsymbol{X}_{i0} + \boldsymbol{X}_i , \tag{15.208}$$

we obtain a linear equation system in all unknown parameters $\boldsymbol{X}_i$ and $\boldsymbol{Z}_t$ when minimizing

$$\Omega(\{\boldsymbol{X}_i\}, \{\boldsymbol{Z}_t\}) = \sum_{i \in \mathcal{I}_t, t \in \mathcal{T}} |\bar{\boldsymbol{v}}_{it}(\boldsymbol{X}_i, \boldsymbol{Z}_t)|^2 + \sum_{i \in \mathcal{I}_0} |\boldsymbol{v}_{i0}(\boldsymbol{X}_i)|^2 . \tag{15.209}$$

An efficient and robust solution is given by Goldstein et al. (2015).

There are at least four situations where this method is useful:

- The cameras are calibrated and the images are taken with known rotation, e.g., vertical views with the azimuth known from a compass.
- The cameras are calibrated, the images are approximately vertical views, and the rotation around the Z-axis is determined by the previous method for planar block adjustment.
- The cameras are calibrated and the rotations are determined with the method discussed in the next section.
- A common plane is visible in all images, which in addition have enough overlap to determine the relative pose of the images w.r.t. that plane. The idea is to geometrically transform the problem such that the given plane is the plane at infinity and derive the infinite homography from observed point correspondences. This requires knowing which points in the images belong to the same plane and at least two points in the scene which are known not to lie on the reference plane (see Rother and Carlsson, 2001, 2002; Kaucic et al., 2001).

15.6.3.3 Linear Estimation of Rotations and Scales

In order to apply the previous method for linear bundle adjustment, Martinec and Pajdla (2007) proposed a direct method for recovering the rotations R_t of T cameras, assuming their calibration is known sufficiently well.

They assume all images are mutually related by their relative orientation; thus, the relative rotations R_{st} for sufficiently many pairs of cameras is given. The task is to directly determine all rotations and relative scales. Referring to the rotation, this process is called also *rotation averaging*. *rotation averaging*

For each pair $\{s, t\}$ of cameras we have the relation $R_{st}R_s = R_t$ or the constraint

$$R_{st}R_s - R_t = 0 \qquad s, t \in \{1, ..., T\} , \tag{15.210}$$

where the rotation matrices R_s and R_t are unknown. Observe, this constraint is linear in the elements of the two unknown rotation matrices. Fixing one rotation as the unit matrix, say $R_1 = I_3$, and vectorizing the constraint

$$[I_3 \otimes R_{st} , -I_9] \begin{bmatrix} \boldsymbol{r}_s \\ \boldsymbol{r}_t \end{bmatrix} = \mathbf{0} \quad \text{with} \quad \boldsymbol{r}_s = \text{vec}(R_s) \quad \text{and} \quad \boldsymbol{r}_t = \text{vec}(R_t) , \tag{15.211}$$

we arrive at a set of constraints of the form $\mathsf{A}\boldsymbol{r} = \mathbf{0}$, with the matrix A known, and the vector $\boldsymbol{r} = [\boldsymbol{r}_t]$ containing all elements of the unknown rotation matrices $t = 2, ..., T$. These constraints only take the relative orientations into account.

Due to measurement noise in the relative orientations, these constraints will not be rigidly fulfilled. Without using orthogonality constraints for each rotation matrix, the vector $\boldsymbol{r} = [\boldsymbol{r}_t]$ can be estimated using a singular value decomposition of A. The final rotations are obtained by enforcing the orthogonality constraints on the corresponding 3×3 matrices, $[\boldsymbol{r}_{t1}, \boldsymbol{r}_{t3}, \boldsymbol{r}_{t3}]$.

The solution is suboptimal and not robust. A method optimizing the L_1-norm $\sum_{st} |\boldsymbol{d}_{st}|$ of the rotation vectors induced by $R(\boldsymbol{d}_{st}) = R_{st}R_s - R_t$ is given by Hartley et al. (2011, 2013).

The scales ${}^m\lambda_t$ of each photogrammetric model w.r.t. the scene can be easily integrated by replacing the rotation matrices R_t and R_{st} with scaled rotation matrices Q, now referring to the rotation matrices ${}^mR$ for the models,

$$Q_t = {}^m\lambda_t {}^mR_t \quad \text{and} \quad Q_{st} = {}^m\lambda_{st} {}^mR_{st} \,. \tag{15.212}$$

Thus, in addition to the relative rotations ${}^mR_{st}$ we also use the relative scale factors ${}^m\lambda_{st}$, see (14.65), p. 634 and (13.293), p. 607.

The method is not robust; thus, we need to assume the rotations have been cleaned of outliers (see Reich and Heipke, 2014).

15.6.3.4 A Solution Based on Matrix Factorization

The following method starts from the assumption that

- the projection model is orthographic, and
- all scene points are visible in all images.

Both conditions can be relaxed. Then, following Tomasi and Kanade (1992), the orientation parameters and the scene points can be derived by factorizing the matrix containing all observations. The following derivation is adapted from Lischinski (2007).

An orthographic projection is realized by a Euclidean camera with infinite principal distance. Therefore we obtain the calibration matrix $\mathsf{K} = \mathrm{Diag}([1, 1, 0])$. With the rotation matrices R_t having orthogonal row vectors $[\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t]^\mathsf{T}$, the infinite homography reads

$$\mathsf{H}_{\infty,t} = \mathsf{K}R_t = \begin{bmatrix} \boldsymbol{u}_t^\mathsf{T} \\ \boldsymbol{v}_t^\mathsf{T} \\ \mathbf{0} \end{bmatrix} . \tag{15.213}$$

The projection then reads

$$\begin{bmatrix} x_{it} \\ y_{it} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_t^\mathsf{T} \\ \boldsymbol{v}_t^\mathsf{T} \end{bmatrix} \boldsymbol{X}_i + \begin{bmatrix} c_t \\ d_t \end{bmatrix} , \tag{15.214}$$

with the unknown transformation parameters $\boldsymbol{u}_t$, $\boldsymbol{v}_t$, and $[c; d]_t$ and the unknown scene point coordinates. The observed coordinates depend linearly on both. Collecting the terms for the x- and the y-coordinates, we can write this projection relation for all images in the form

$$W = MS + T \,, \tag{15.215}$$

where

$$\underset{2T\times I}{W} = \begin{bmatrix} [x_{it}] \\ [y_{it}] \end{bmatrix} , \quad \underset{2J\times 3}{M} = \begin{bmatrix} [\boldsymbol{u}_t^\mathsf{T}] \\ [\boldsymbol{v}_t^\mathsf{T}] \end{bmatrix} , \quad \underset{3\times I}{S} = [\boldsymbol{X}_i] \,, \quad \underset{2T\times I}{T} = \begin{bmatrix} [c_t \mathbf{1}_I^\mathsf{T}] \\ [d_t \mathbf{1}_I^\mathsf{T}] \end{bmatrix} . \tag{15.216}$$

As both matrices M and S have maximal rank 3 and T has maximal rank 1, the matrix W containing all observations has maximal rank 4. Multiplying (15.215) by the projection

matrix $J = I_I - \mathbf{1}_I \mathbf{1}_I^\mathsf{T}/I$ eliminates the additive component T and we obtain

$$\overline{W} = M\overline{S}\,, \qquad \text{with} \quad \overline{W} = WJ\,, \quad \overline{S} = SJ\,. \tag{15.217}$$

Thus the reduced matrix $\overline{W}$ is the product of two matrices with maximal rank 3 and also has maximal rank 3.

We now start from $\overline{W}$, which can be derived from the observations, and determine the projection parameters and the scene coordinates. An SVD of the matrix $\overline{W}$ allows us to find a rank 3 approximation,

$$\overline{W} \approx \underset{2J\times 3}{U}\ \underset{3\times 3}{D}\ \underset{3\times I}{V^\mathsf{T}}\,. \tag{15.218}$$

Partitioning the diagonal matrix, we obtain approximations for the matrices M and $\overline{S}$,

$$M^a = U\sqrt{D} \qquad \overline{S}^a = \sqrt{D}V^\mathsf{T}\,. \tag{15.219}$$

The decomposition is not unique, as any other pair $(M^a B B^{-1} \overline{S}^a)$ would lead to the same measurement matrix W. As the matrix B is a centred affinity, the result is unique only up to an affinity.

The result can be upgraded to a Euclidean reconstruction by choosing the matrix B such that the rows of $\widehat{M} = M^a B$ are close to unit vectors and close to being pairwise perpendicular. With the symmetric matrix $C = BB^\mathsf{T}$ for each two rows $\widehat{\boldsymbol{u}}_t^\mathsf{T}$ and $\widehat{\boldsymbol{v}}_t^\mathsf{T}$ of $\widehat{M}$ corresponding to the same image we obtain the three constraints

$$\widehat{\boldsymbol{u}}_t^\mathsf{T}\widehat{\boldsymbol{u}}_t = \boldsymbol{u}_t^{a\mathsf{T}} C \boldsymbol{u}_t^a = 1\,, \quad \widehat{\boldsymbol{u}}_t^\mathsf{T}\widehat{\boldsymbol{v}}_t = \boldsymbol{u}_t^{a\mathsf{T}} C \boldsymbol{v}_t^a = 0\,, \quad \widehat{\boldsymbol{v}}_t^\mathsf{T}\widehat{\boldsymbol{v}}_t = \boldsymbol{v}_t^{a\mathsf{T}} C \boldsymbol{v}_t^a = 1\,. \tag{15.220}$$

These are $3 \times T$ linear equations for the six different entries of the matrix C. They can be determined in a least squares sense, yielding an estimate $\widehat{C}$. The matrix $\widehat{B}$ then can be determined by a Cholesky decomposition. The resultant matrix $\widehat{M}$ again is unique up to a rotation, as for any rotation matrix R we could have used $\widehat{B}' = \widehat{B}R$, also fulfilling $\widehat{B}'\widehat{B}'^\mathsf{T} = \widehat{C}$. But this just reflects the freedom in choosing the directions of the axis of the scene coordinate system.

The method can also be used in the case of occlusions (see Tomasi and Kanade, 1992). Here an initial estimate for the motion and scene parameters is determined from a small set of scene points visible in all images. This then allows us to predict the missing observations. The method has also been extended to perspective cameras. Then the factorization also needs to determine the individual scales in $\lambda_{it}\mathbf{x}_{it} = \mathsf{P}_t\mathbf{X}_i$. This leads to an iterative solution (see Triggs, 1996; Sturm and Triggs, 1996).

15.7 View Planning

15.7.1 Goals, Boundary Conditions, and Rules of Thumb 716
15.7.2 Large Flat Areas 718
15.7.3 Buildings and Indoor Environments 721

In many cases, planning the geometric camera configuration for achieving a certain goal is feasible and also necessary due to the complexity of the interplay of camera poses and scene structure. We again only address view planning for scene reconstruction, not for ego-motion determination. View planning requires pre-knowledge, which should be derivable from the task specification and which refers to the availability of hardware and software resources.

Generally, view planning can be performed by simulating the configuration and using a bundle adjustment program for determining quality measures, such as accuracy or sensitivity parameters. As a prerequisite, we need to know the achievable accuracy of image measurements. This depends on our ability to identify the objects' details in the images

relevant for the envisaged application. This is related to the image scale and to the planned measurement process, see the discussion in Sect. 12.2.1, p. 490.

The configuration then needs to specify the intended poses, the intended scene points and the assumed measuring accuracy, which allow us to simulate image measurements and to derive, for example, the expected covariance matrix $\Sigma_{\hat{x}\hat{x}} = (A^\mathsf{T}\Sigma_{ll}^{-1}A)^{-1}$ of all parameters, possibly including the additional parameters for the calibration. Especially if the measurement process is costly or cannot be repeated (e.g., when observing a scene which is expected to change or which is expected not to be accessible in the future), such a simulation is highly recommended in order to have a certain guarantee that the envisaged result fulfils the requirements. Such simulations are costly, as the space of configurations is large. Therefore the simulation package may be extended with an exploration component which tries to automatically optimize the configuration.

Often the task follows a certain standard, e.g., when reconstructing large, more or less flat, areas, complete buildings, or rooms. Then simulation studies may be used to derive rules of thumb, which, when followed, guarantee that certain requirements are fulfilled.

In the following we will first discuss possible goals, some of which have already been used in previous sections for motivating certain quality measures, boundary conditions, and general rules of thumb. Then we will give more specific rules for two typical application tasks.

15.7.1 Goals, Boundary Conditions, and Rules of Thumb

The *goal* of a scene acquisition task usually is to recover the 3D geometry with a prespecified quality.[4] There are two aspects which need to be taken into account:

1. Completeness. The reconstruction should be complete in the sense that all required details of the scene are captured. This may be easy, e.g., if the scene is flat, or very difficult, e.g., if an industrial area or a chemical plant is to be modelled. Therefore the user should specify how the completeness is measured, so that the fulfilment of this requirements can be checked. In complicated situations, this means accepting that a certain percentage or certain types of objects are not fully captured.
2. Accuracy. Accuracy depends on precision and bias, cf. Sect. (4.6.1), p. 116.
   Precision can be measured by the variances of parameters. However, this assumes that the envisaged precision of the measurements, captured in Σ_{ll}, actually holds. Therefore *after completion* of a task it is advisable to use the empirical covariance matrix $\widehat{\Sigma}_{\hat{x}\hat{x}} = \widehat{\sigma}_0^2(A^\mathsf{T}\Sigma_{ll}^{-1}A)^{-1}$, which, via the estimated variance factor $\widehat{\sigma}_0^2$, also takes the average deviation of the measurements from the assumed mathematical model into account.
   Bias may be introduced by nondetectable outliers or nonidentifiable systematic errors. In the planning phase, the expected bias can be determined using the theoretical measures of sensitivity (see the discussion in Sect. 4.6.3, p. 122 and the measures collected in Tables 4.1, p. 128 and 4.2, p. 131 for the bias caused by outliers, and Tables 4.3, p. 136 and 4.4, p. 136 for systematic errors.) Observe, these measures can be used for planning and can be determined before having access to real observations. After a task has been completed, the corresponding empirical measures can be used to document the achieved quality.
3. Ground sampling distance. The ground sampling distance is the distance between neighbouring pixels backprojected to the scene. It is usually measured in meters. Its value is identical to the image scale number, cf. (12.1), p. 457. The ground sampling distance needs to be chosen such that
   - the interpretation of the image content is possible, which highly depends on the application, and

[4] In practice we are often also interested in the surface texture, a topic we do not address here.

- the required accuracy is achievable. This usually does not lead to stricter requirements compared to the interpretability of the images.

Boundary conditions refer to all practical aspects, potentially influencing the quality of the result. They therefore require precise enough task specifications and experience by the designer of the experiment. Among other things, they refer to

- the scene type, e.g., flat or hilly terrain, urban or suburban area, or accessibility in rooms, churches, or caves;
- the characteristics of the available cameras, e.g., their resolution, focal length, and weight (relevant for choosing a platform);
- the characteristics of aerial sensor platforms, e.g., their payload, height range (at high altitudes pressurized cabins are required), and flight range;
- the characteristics of the software used, e.g., limitations on the number of images, availability of a simulation package, availability of quality measures characterizing the result, and user interface for performing final checks;
- expected weather conditions, e.g., cloud coverage (for high altitudes), wind speed (for unmanned aerial vehicles), and zenith angle of the sun for reducing shadow, and
- visibility of scene points. This includes sufficiently accurate identification, i.e., brightness variations or texture around the scene points. In the context of automatic procedures, the surface normal around points should not deviate from the viewing ray by more than a certain angle, say 45°.

These boundary conditions may be used in a simulation study, and may give some freedom in the design, but may also dictate geometric configurations which are suboptimal.

There are a few *rules of thumb* which can be used for planning:

- Precision. The scene rays should show a minimum parallactic angle of 10° to 15°.
  The precision of a scene point can be easily derived from the intersection of its viewing rays without taking into account the uncertainty of the pose of the cameras.
  As a rule of thumb the precision can be derived from the triangulation precision for the image pair (cf. 13.282 and (13.283), p. 604) with a correction factor taking into account the number T of images used to determine the 3D point. We assume that the T projection centres lie in a plane centrally above the 3D point of concern, at distance D, see Fig. 15.22. Then, using (13.282), p. 603, we have, for the precision of the position across the average viewing direction and of the distance along the average viewing direction, Exercise 15.9

$$\sigma_Q = \frac{1}{\sqrt{T}}\, D\, \frac{\sigma_{x'}}{c}\,, \qquad \sigma_D = \frac{1}{\sqrt{T}}\, \frac{D^2}{Q}\, \frac{\sigma_{x'}}{c} = \frac{D}{Q}\sigma_Q\,, \tag{15.221}$$

  with the assumed principal distance c, which may be approximated by the focal length; the assumed standard deviation $\sigma_{x'}$ of the measured image point coordinates, measured in the same units as the principal distance; and the average distance $Q = \sqrt{\sum_t (X_t^2 + Y_t^2)/T}$ of the projection centres from their centroid, measured in the same units as the distance D. For T cameras in a row and for T cameras in a square grid, we have

$$Q_{\text{row}} = \sqrt{\frac{T^2-1}{12}}\, B \quad \text{and} \quad Q_{\text{square}} = \sqrt{\frac{T-1}{6}}\, B\,. \tag{15.222}$$

  For the image pair, thus for $T = 2$ cameras in a row, the average distance Q is half the base length B.
  Observe, the relative standard deviations σ_Q/Q and σ_D/D of the position of the 3D point across and along the average viewing direction are the same. In order to achieve a depth standard deviation, which is not more than by a factor of 4 larger than the accuracy across the viewing ray, the parallactic angle should be above $1/4$ rad $\approx 15°$, since $Q/D = \sigma_Q/\sigma_D$.

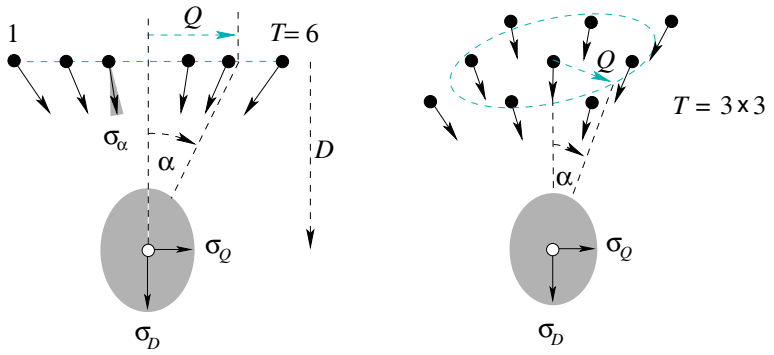


Fig. 15.22 Precision of triangulation of a scene point from T cameras. The standard deviations σ_Q and σ_D of the 3D point across and along the average viewing direction derived from T projection rays with directional uncertainty $\sigma_\alpha \approx \sigma_{x'}/c$ depend on (1) the distance D to the 3D point along the average viewing direction and (2) the average distance Q of the T projection centres from their centroid across the average viewing direction. **Left:** cameras in a row. For the image pair, as special case, we have $T = 2$. **Right:** cameras in a square (3×3) grid

Moreover, the directional uncertainty of the projection rays is $\sigma_\alpha \approx \sigma_{x'}/c$, while the variation of the viewing directions is $\alpha = \arctan(Q/D) \approx Q/D$. Therefore we have the relation

$$\frac{\sigma_D}{D} = \frac{\sigma_Q}{Q} = \frac{1}{\sqrt{T}} \frac{\sigma_\alpha}{\arctan \alpha} \approx \frac{1}{\sqrt{T}} \frac{\sigma_\alpha}{\alpha}, \tag{15.223}$$

which expresses the relative standard deviation of the 3D point across and along the average viewing direction as a function of the relative standard deviation σ_α/α of the projection rays and the number T of rays. For $T = 2$ this simplifies to the standard deviations for the image pair (13.283), p. 604, taking $\sigma_{p_x} = \sqrt{2}\,\sigma_{x'}$ into account.

- Accuracy. The measured scene points need to be visible in at least three, preferably in four, images.
  Though the minimum number of rays for determining a scene point is two, more rays are necessary for outlier detection and location. Generally, three rays guarantee the *detection* of a single outlier in the set, four rays guarantee *locating* a single outlier among these rays. If multiple outliers are to be expected per scene point, more than four rays are necessary to be able to locate them.

In the following we discuss design rules for three classical tasks.

15.7.2 Large Flat Areas

Capturing large areas which are relatively flat is a classical task of topographic mapping. The scene is observed only from above. The situation is similar to observing a large flat facade. Typically the flight path is a meander covering the area strip by strip, see Fig. 15.23: A strip leads to a sequence of camera poses which approximately lie on a straight horizontal path. If the image is rectangular with length l and width w, where $l \geq w$, then, for minimizing the flight path, the shorter side w needs to lie in flight direction. The method is called *aerotriangulation*.[5]

aerotriangulation

In order to observe each point in at least two images, the images nominally overlap by $p > 50\%$ of the width w in forward (flight) direction; the value p is called the *endlap*; often $p = 60\%$ is used. The base line b nominally is $1 - p$ times the width w.

endlap

In order that images of two neighbouring strips be properly connected, they need to overlap sufficiently, say, by q times the length l, with $q > 0$; the value q is called the

[5] It is derived from the terrestrial method of geodetic triangulation of large areas applied when determining reference points using angular measurements with theodolites started in the early nineteenth century. "Triangulation" here means the covering of an area by triangles, as in Delauney triangulation.

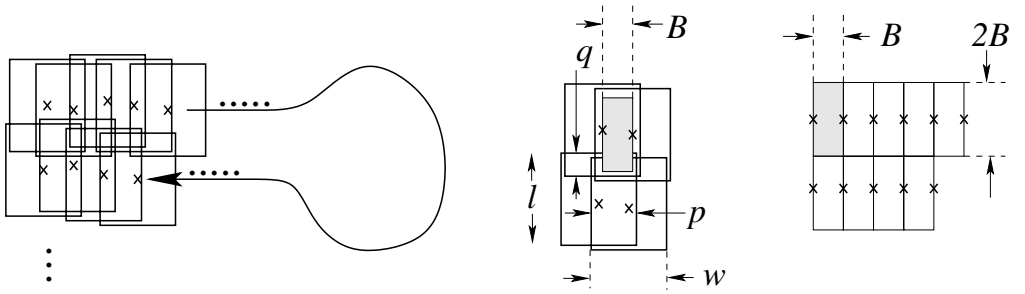


Fig. 15.23 Arrangement of camera poses of a block with $p = 60\%$ endlap and $q = 20\%$ sidelap. **Left:** images (rectangles), position of projection centres (crosses), and sketch of flight path. **Middle:** nominal photogrammetric model or net model, endlap $p = 60\%$, sidelap q =20%, length B of base line in object space ($B = b\,H_g/c$, H_g is flying height over ground). **Right:** schematic arrangement of photogrammetric models and projection centres. The figure is valid for other values for the overlap, namely $p < 67\%$ and $q < 50\%$. Then the form of the nominal model will change

sidelap; often $q = 20\%$ is used. The net area A_{net} (shaded in grey) of the overlapping area of two neighbouring images in a strip therefore is $A_{\text{net}} = (1-p)(1-q)wl$. This net area nominally is the photogrammetric model of the two neighbouring images. The projection centres lie in the middle of the longer sides of the net area of the model. Images t and $t+2$ within one strip have an endlap of $2p-1$ times the width w, 20% in our case. This guarantees that the image triplet $(t, t+1, t+2)$ allows scale transfer if there are enough scene points visible in all three images. The ratio 1:2 of the sides of the photogrammetric model only holds for specific pairs of p and q.

sidelap

Concatenating the photogrammetric models leads to strips of models as shown in the right subfigure of Fig. 15.23. Two neighbouring strips are connected by a flight loop where usually no images are taken.

The arrangement of control points is decisive for the achievable accuracy, see Fig. 15.24.

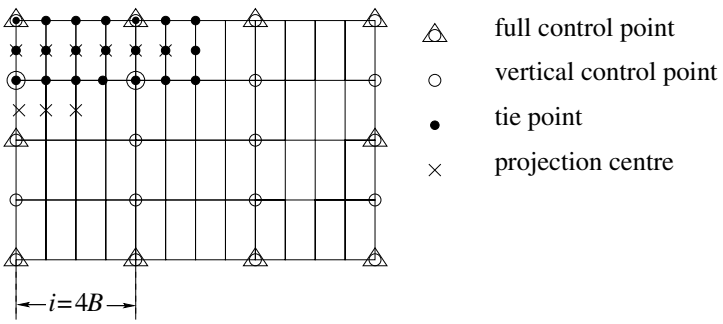


Fig. 15.24 Schematic arrangement of control points in a block with with $p = 60\%$ endlap and $q = 20\%$ sidelap. Planimetric control points (X- and Y-coordinates known) with a nominal distance of i measured in lengths B of base lines at the border of the area covered by photogrammetric models. Chains of vertical control points (Z-coordinate known). Neighbouring models are connected by tie points, which may not be of interest within the project. Here they are shown in schematic positions which yield a rectangular grid of tie points. At each position double points should be observed in order to achieve high enough detectability w.r.t. outliers. In the case of automatic detection and mensuration of tie points, the procedure should guarantee a sufficient coverage of the model boundaries with points visible in at least three images

We already discussed the theoretical precision of the planimetric coordinates X and Y of the tie points, cf. Sect. 15.3.5.2, p. 673. These results directly transfer to bundle blocks.

- Planimetric control points only are necessary at the border of the area covered by the block. The planimetric uncertainty behaves similarly to a piece of cloth which is properly fixed at the border. Therefore control points should lie really close to the

border, not further away than half a base line from the border of the area covered by photogrammetric models.

- The distance of the planimetric control points, measured in units of the base line B, influences the horizontal precision of the new points. Intervals of $i = 4$ to $i = 6$ are favourable.

The following results are taken from Ebner et al. (1977).

- The planimetric precision is very homogeneous within the area of the block, see Fig. 15.15, p. 674 and theoretically increases with the logarithm of the block size. If the control points densely cover the block boundary, the planimetric precision of new points within the block can be approximated by (for $p = 60\%$ and $q = 20\%$)

$$\sigma_X = \sigma_Y = 0.9 \cdot S \cdot \sigma_{x'} \,, \tag{15.224}$$

ground sampling distance

where $\sigma_{x'}$ is the measuring precision in the image measured in pixels and S is the scale number measured in m/pixel whose value is identical to the ground sampling distance. Obviously the measurement precision directly transfers to the planimetric precision in object space.

- If the control points only lie in the four corners of the block, the planimetric precision increases with the block size. For square-shaped blocks with n_s strips, the precision linearly increases with n_s (for p=60% and $q = 20\%$):

$$\sigma_X = \sigma_Y = 0.5 \cdot S \cdot n_s \cdot \sigma_{x'} \,. \tag{15.225}$$

The precision of the heights shows a different structure.

- If height control points are not densely positioned at the border of the block, the rotation of neighbouring strips along the flight line, the angle ω around the X-axis, which goes in flight direction, may not be determinable. To avoid the effect, which could be called *Toblerone effect*, height control points are necessary along both ends of all strips with a maximum distance of two base lines, see Fig. 15.25. Moreover,

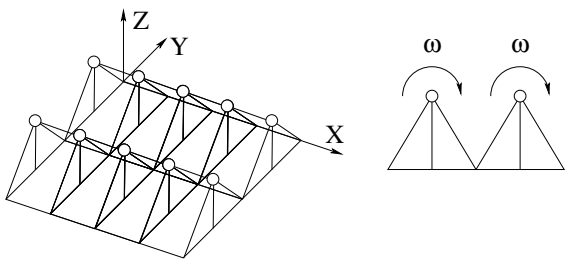


Fig. 15.25 Toblerone effect: The strips nominally have the form of a triangular prism, similarly to a Toblerone© chocolate. Each strip is stable due to the endlap of 60%. But due to the low sidelap of 20%, the strips may mutually rotate around the flight line if the tie points of two neighbouring strips lie on a straight line (parallel to the flight line). Therefore a common angle of the images in one strip may be only weakly or not at all determinable

additional height control points are necessary in order to stabilize the height of the points in the interior of the block. This behaviour is analogous to that of a horizontal piece of cloth, which needs to be more stabilized in the height than in the horizontal direction.

- The height precision σ_Z essentially depends on the interval i measured in base lines B of the height control point chains. We have

$$\sigma_Z = (1.5 + 0.3i) \cdot S \cdot \sigma_{x'} \,; \tag{15.226}$$

therefore, height control point chains with an interval of four or six are favourable.

These results refer to 60% end and 20% sidelap. This corresponds to a double coverage of the area, as each scene point is visible at least in two images. If all points need to be in at least four images, we need 80% endlap; this leads to fourfold coverage. As image storage does not pose a limitation, often 80% forward and 60% sidelap are used. Then the area is covered eightfold so that the above-mentioned theoretical standard deviations decrease significantly, approximately with the square root of the coverage, and at the same time the ability to capture systematic errors by self-calibration increases. Then a relative precision of 1 : 100 000 to 1 : 300 000 is achievable.

differential GPS

To determine the coordinates of control points for aerotriangulation, *differential GPS* is used, which yields coordinates with an accuracy below 0.1 m. Using a reference GPS station, systematic errors caused by the troposphere and the ionosphere can be captured to a large extent so that the accuracy of the coordinate determination improves. This accuracy can also be achieved for GPS antennas on aeroplanes during the flight mission. Taking the geometric and time offsets between the GPS antenna and the cameras' projection centre into account, such measurements can be used to treat projection centres as full control points, which then are available for each image. They can substitute for ground control points.

projection centres as control points

However, having no ground control points is not recommended, as especially the Z-coordinates of the terrain would fully depend on the correctness of the camera calibration, especially the principal distance. Moreover, any error in time synchronization of the GPS signal and the camera exposure time would result in systematic deviations in the control points (cf. Cramer, 1999). Therefore, at least four full control points in the corners of a block should be planned for and measured with the same setup with differential GPS. These control point coordinates then also check the consistency of the GPS coordinate system and the reference system used in the respective country.

15.7.3 Buildings and Indoor Environments

Buildings and indoor environments generally are challenging as visibility of all bounding faces may not be possible due to obstacles hindering ideal camera poses. Here, planning using a rough scene model appears indispensable; for an example, cf. Massios and Fisher (1998).

We only will discuss two basic scenarios: a rectangular building and a rectangular room which are not too high, see Fig. 15.26. In both cases we first need to determine the

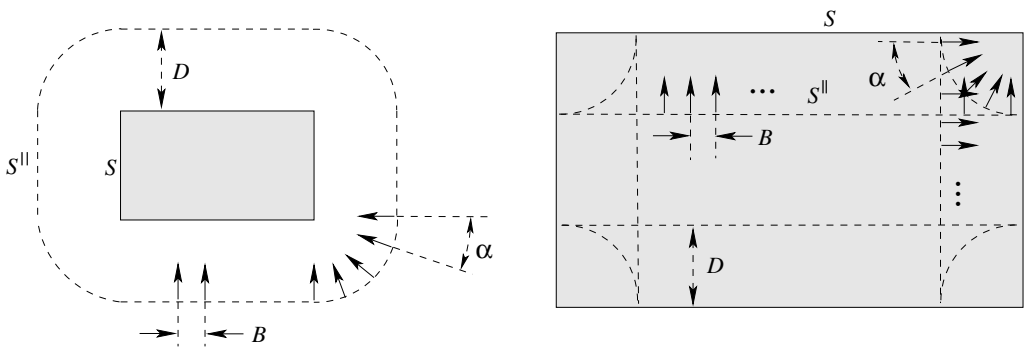


Fig. 15.26 View plan for rectangular buildings and rooms. The required maximal distances D and B follow from the accuracy requirements and the camera used. The distance between viewpoints at corners needs to guarantee (1) neighbouring view points are as far apart as possible but have a distance below the required base length B and (2) the angle between viewing directions referring to the same scene point is below $\alpha_{\max}$, approximately 45°

required distance of the camera from the walls as a function of the image resolution, or,

more specifically, the measurement accuracy $\sigma_{x'}$, the principal distance c, and the required standard deviations σ_D and σ_Q, cf. (15.221), p. 717. This also yields the required maximum distance B of the camera poses. An additional requirement is that the opening angle of the camera must guarantee at least 80% endlap. This might override the requirements for the base length B resulting from the accuracy.

In both scenarios we choose a path that is a line parallel to the object's boundary, outside and inside, respectively, and having distance D. We sample the path with a distance of base B and choose the viewing directions perpendicular to the path. The density of the camera poses at the corners needs to guarantee a maximum parallactic angle $\alpha_{\max}$ between the viewing directions of the required three or four consecutive images: then scene points can reliably be determined. This angle should be below 45°. Then the minimum parallactic angle between consecutive images should be approximately 22.5°or 15°, respectively. If possible, the angle between neighbouring images should be even between 5 and 15°, which, if the costs for taking images are low enough, may be acceptable.

Conceptually, the method can be generalized to arbitrary surfaces S. Depending on the environment, not all projection centres may be physically realizable or useful due to obstacles. In this case no general rules for planning can be given. Exploration schemes with a coarse-to-fine strategy will be necessary if the surface is completely unknown or only known to a certain extent.

15.8 Exercises

Basics

1. (1) How many aerial images do you need to cover a quadratic area of 10 km sidelength with a DMC at a flying height of 1500 m if the endlap and the sidelap are 60% and 20%? Use the specifications of the DMC from Exerc. (16), p. 538.
2. (2) The normal equation matrix is usually partitioned into a 2×2 block matrix

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{kk} & \mathbf{N}_{kp} \\ \mathbf{N}_{pk} & \mathbf{N}_{pp} \end{bmatrix} \tag{15.227}$$

   for the coordinates $\boldsymbol{k}$ and the orientation parameters $\boldsymbol{p}$. It is sparse. Assume you have a bundle block with 40 images and 2000 scene points.

   a. Sketch the nonzero structure of the normal equation matrix.
   b. What size do the matrices $\mathbf{N}_{kk}$ and $\mathbf{N}_{pp}$ have?
   c. At what positions are the elements in $\mathbf{N}_{kp}$ nonzero?
   d. In case a scene point is observed in four images on average, what is the percentage of nonzeros in the matrix $\mathbf{N}_{kp}$?
   e. Under what conditions does an elimination of the coordinate parameters from the normal equation system lead to a reduction of the processing time?
   f. Sketch the nonzero structure of the normal equation matrix if additional parameters $\boldsymbol{s}$ are to be estimated.
   g. You may augment the vector of the unknown parameters by the vector $\boldsymbol{s}$ of additional parameters in three ways: (1) $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{k}, \boldsymbol{s})$, (2) $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{k})$, or (3) $\boldsymbol{x} = (\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{k})$. Which is the best arrangement w.r.t. the expected computing time; which is the worst arrangement? Why?

3. (2) Control points $\mathcal{X}_i$ can be introduced into a bundle adjustment either as fixed, thus $\underline{\boldsymbol{X}}_i \sim \mathcal{M}(\boldsymbol{\mu}_{X_i}, \boldsymbol{0})$, or as stochastic, thus $\underline{\boldsymbol{X}}_i \sim \mathcal{M}(\boldsymbol{\mu}_{X_i}, \Sigma_{X_i,X_i})$.

   a. Is the bundle adjustment statistically optimal if the control points are treated as fixed? Explain.

b. How large is the difference in the redundancy of a bundle adjustment with fixed and one with stochastic control point coordinates?
c. What is the advantage of using the coordinates of the control points as stochastic in case they might contain outliers, e.g., caused by a wrong identification in the images?
d. The redundancy numbers (cf. (4.69), p. 88) for control points usually are quite small, often below 0.1. Assume that a control point has a covariance matrix $\Sigma_{X_iX_i} = \sigma_{X_i}^2 I_3$ with $\sigma_{X_i} = 10$ [cm]. The redundancy number of the coordinates is 0.04.
   i. What is the minimum size of an outlier in order for it to be detectable with a statistically optimal test using $z_i = \widehat{v}_{X_i}/\sigma_{\widehat{v}_{X_i}}$? *Hint:* Use (4.285), p. 125.
   ii. What is the minimum size of an outlier in order for it to be detectable with an approximate test using $z_i^* = \widehat{v}_{X_i}/\sigma_{X_i}$? *Hint:* Use (4.289), p. 126.

   Assume a noncentrality parameter $\delta_0 = 4$, which corresponds to using a critical value $k = 3.3$ for testing and requiring a minimum probability $\beta_0 \approx 0.8$ of detecting the outlier, cf. Table (3.2), p. 67.
e. Could you detect outliers in the control point if they were introduced as fixed? Explain.

Discuss the result with respect to the necessity to include control points as stochastic and to use an optimal test. Since you need the redundancy numbers of the control point coordinates, what consequences does this have on the computational complexity of the bundle adjustment?

4. (1) Give at least three reasons why the normal equation matrix in a bundle adjustment may be singular. For each of the reasons, give a remedy for achieving a regular normal equation matrix. Explain the meaning of each remedy without referring to concepts of linear algebra.

Methods

5. (2) Your task is to buy software for a free bundle adjustment. You have demo versions of three programs, which you use for testing using the same data set. The data set results from a simulation, so you have reference values for all orientation parameters and 3D points. The programs determine a free bundle adjustment, so you do not need to provide any control points. From each program you obtain as output the following entities:

   a. the distances and distance ratios between the projection centres,
   b. the rotation matrices of the camera poses,
   c. the distances and distance ratios between the scene points, and
   d. the variances of the scene point coordinates.

   Which entities do you use for evaluating the quality of the programs? Can you use the entities directly or do you need to transform them? Which entities are not usable? Give reasons.
6. (2) You obtain four images of 24 3D scene points. The correspondence between the image points and the scene points is available, see Fig. 15.27. The scene points are unknown. No control points are available. The task is to perform a free bundle adjustment.

   a. Give the numbers U of unknown parameters and N of observations.
   b. Are all parameters determinable? Explain why. If possible, mention and explain remedies. Determine the redundancy R of the estimation problem and explain all ingredients relevant for the determinability of the parameters.
   c. Give a method to determine good approximate values. Assume outlier detection has been successful.

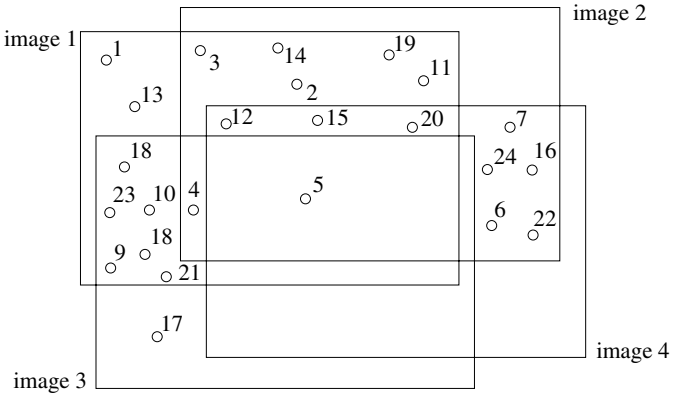


Fig. 15.27 Four overlapping images of a free bundle block with indicated image points. For example the points 12, 15, and 20 are measured in images 1, 2, and 4. Point 17 is only measured in image 3

d. Explain why the proposed method for determining the approximate values is not statistically optimal.

Proofs and Problems

7. (2) Imagine a bundle block consisting of one strip with 60% endlap and four control points in the corners of the block. Assume the image points fill the net area of the photogrammetric models. The terrain is assumed to be flat.

   a. Which of the 15 systematic errors in Table 12.6, p. 512 do not show in the residuals after a bundle block adjustment?
   b. What effect do such nonvisible systematic errors have on the scene coordinates?

   Distinguish between linear and nonlinear image distortions. Now assume the block has more than one strip.

   c. Does this change the situation?
   d. How would you change the configuration, such that there is a guarantee that all 15 parameters can be determined well?

   Discuss the flight configuration and the control point configuration separately.

   e. Why do these critical configurations w.r.t. systematic errors not lead to a singular normal equation system in practice?
   f. What indicators would you use to identify configurations close to critical ones?

8. (2) Given a sequence of independent stochastic variables $\underline{x}_n, n = 1, ...,$ with $\sigma_{x_n} = \sigma$. Show the following:

   a. For the single sum $s_N = \sum_{n=1}^{N} x_i$ of the variables x_i, the standard deviation is $\sigma_{s_N} = \sigma\sqrt{N}$.
   b. For the double sum $t_N = \sum_{i=1}^{N} s_i$ of the variables x_i, the standard deviation is $\sigma_{t_N} = \sigma\sqrt{N^3}$.
   c. Given a sequence of 2D points, defined by

$$\boldsymbol{x}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \boldsymbol{x}_i = \boldsymbol{x}_{i-1} + d \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{bmatrix}, \tag{15.228}$$

where d is a constant and the sequence α_i contains statistically independent stochastic variables with mean 0 and common variance σ_α, show that $\sigma_{y_N} \approx kN^{3/2}$. Give an explicit expression for k.

d. Why is the example from (c) a model for the statistical behaviour of the y-coordinates a long strip of images which are connected via a similarity transformation? Refer to Sect. 15.3.5.1 and Table 15.2. How would you extend the model (15.228) to mimic the statistical behaviour of the scale transfer in order to also obtain an expression for σ_{x_N}?

9. (2) Prove (15.221), p. 717. *Hint:* Assume the following: The unknown 3D point is close to the origin of the scene coordinate system. The T ideal cameras with principal distance c are at $\boldsymbol{Z}_t, t = 1, ...T$, with common $Z = Z_t$ and are given with negligible uncertainty. The image coordinates $\boldsymbol{x}'_t$ are measured with standard deviation $\sigma_{x'} = \sigma_{y'}$. Derive the covariance matrix of the unknown 3D point using a Gauss–Markov model.

Computer Experiments

10. (3) Implement Algorithm 22, p. 709.

    a. How do you select the first two images (line 1)?
    b. How do you select a new image (line 4)?

    Write a simulation program for testing the algorithm. In a first step, assume only random errors, no gross or systematic errors.

    c. Evaluate the resulting orientation parameters. Is their accuracy good enough for ensuring convergence of a bundle block adjustment?
    d. Vary the configuration (overlap between images) and evaluate the success rate, i.e., the probability of obtaining orientation parameters for all images.
    e. Is there a possibility to improve the orientation parameters after each step without performing a rigorous bundle adjustment? Implement it and check the improvement of the orientation parameters.
    f. Now augment the simulation program: allow for outliers in the image coordinates. Augment the program for spatial resection by a RANSAC procedure. Repeat the previous tests. What type and what percentage of outliers are allowed, such that the algorithm does not fail?

11. (3) Implement Algorithm 23, p. 710.

    a. How many image triplets do you choose to ensure all outliers are found (line 3)?
    b. How do you select the first and the following image pairs (lines 5 and 7)?
    c. Follow the subtasks 10c ff. of Exerc. 10.

12. (3) Implement an algorithm for the simultaneous determination of poses from relative rotations, discussed in Sects. 15.6.3.2 and 15.6.3.3. Use the first part of Algorithm 22, p. 709 for determining relative orientations.

    a. What choices do you have to fix the gauge in both cases (rotations, 3D coordinates)?
    b. Compare the accuracy of the orientation parameters with those from Exerc. 10 and 11. How does the choice of the gauge influence this comparison of the accuracy?
    c. Can you use the algorithm for the simultaneous determination of poses from relative rotations for improving the intermediate results of the algorithm in Exerc. 10? Explain.

Chapter 16
Surface Reconstruction

16.1 Introduction 727
16.2 Parametric 2½D Surfaces 733
16.3 Models for Reconstructing One-Dimensional Surface Profiles 742
16.4 Reconstruction of 2½D Surfaces from 3D Point Clouds 757
16.5 Examples for Surface Reconstruction 763
16.6 Exercises 765

This chapter addresses the problem of reconstructing the visible surface from the 3D points of the photogrammetric models derived from two or more images. We assume both problems to be solved: the matching of corresponding features in two or more images and the determination of their 3D coordinates. The solution of the first problem is a central topic of the second volume of this book; solutions can be found in recent publications e.g., in Strecha et al. (2008), Szeliski (2010, Sects. 10–12), or Haala and Rothermel (2012). The geometric reconstruction of the 3D coordinates, however, can use the methods described in the previous chapters to advantage.

We focus on methods which allow a statistical evaluation, both w.r.t. the reconstruction and the quality analysis. Therefore we only discuss the reconstruction of surfaces or surface regions which are visible from one side, what are called graph surfaces. This is no real restriction, since more complex 3D surfaces can be aggregated from patches of graph surfaces. Various quite different methods for deriving surfaces from 3D point clouds are discussed in the recent review by Berger et al. (2014).

16.1 Introduction

16.1.1 On the Definition of Surfaces 727
16.1.2 Models for Surfaces 728
16.1.3 Tasks to Be Solved 729

16.1.1 On the Definition of Surfaces

Visual reconstruction relies on the idea that the surface of the objects of interest is the boundary between material and air.

This intuitive notion is conceptually problematic. First, the boundary may not be clearly defined, as when observing clouds, see Fig. 16.1, left.

the definition of a surface is scale-dependent

Second, the definition is dependent on the scale at which the object is observable, since certain objects only occur in a certain scale range, which is true not only for fuzzy objects such as clouds but also for subjectively crisp objects (cf. the discussion in Koenderink, 1990, Chap. 2).

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4_16

Take, for example, a clay tile roof, see Fig. 16.1, right. The surface of the roof will be a different one when observed, after being cut out, under an electron microscope. We usually are not interested in that kind of detail. When the roof is observed by the human eye at reading distance, we may be interested in faults motivating a repair of the roof. This requires a resolution of a few tenths of a millimetre. Topographic mapping of a roof generally neglects the individual tiles, which is why roof reconstruction as required for topographic mapping is feasible from a height of a few hundred meters or a few kilometres.

The same problem applies to essential parts of the surface, such as edges, where the surface normal changes suddenly. Such sudden changes also depend on the scale at which they are observed. As a first example, take the brick shingle seen at reading distance: The boundaries of the shingles appear as edges, whereas the same position on the roof is perceived as flat when seen from far. The situation is different for a gable or a ridge of the roof, which appears round when seen from close, whereas it appears as an edge when seen from far away.

Fig. 16.1 Effect of scale on surface perception. **Left two:** clouds at two scales with ratio 1:2. **Right three:** Brick: electron microscopy (2000:1), terrestrial photo (1:10), aerial photo with brick roofs (1:2000). (http://cool.conservation-us.org/jaic/articles/jaic42-01-006_2.html)

digital surface model

Surfaces usually are represented as *digital surface models* (DSMs): this is a set of 3D points and a specification on how to interpolate between these points. Observe, *'model'* here is used as a (possibly scaled) simplified version of the original, similar to a city model. It is not to be confused with a generic model, for describing the surface's properties.

digital elevation model

Topographic maps generally do not contain the visible surface but the topographic surface. The topographic surface is the earth's surface without buildings and vegetation. This topographic surface often is represented by the *digital elevation model* (DEM), i.e., the model containing the elevation of each point of the topographic surface in a map above sea level.

For both surfaces, the DSM and the DEM, the user needs to specify the scale at which they need to be captured and stored. This usually is done by specifying an average distance for the 3D points of the DSM or the DEM. Figure 16.2 visualizes the differences between the DSM and the DEM and illustrates the difficulty in defining the scale.

16.1.2 Models for Surfaces

In the following we will discuss methods for surface reconstruction from 3D points derived from two or more images. We aim at using a minimum of additional assumptions about the surface. This contrasts with methods for reconstructing the objects' form using specific knowledge about these objects, which require object recognition, which is not a topic of this book.

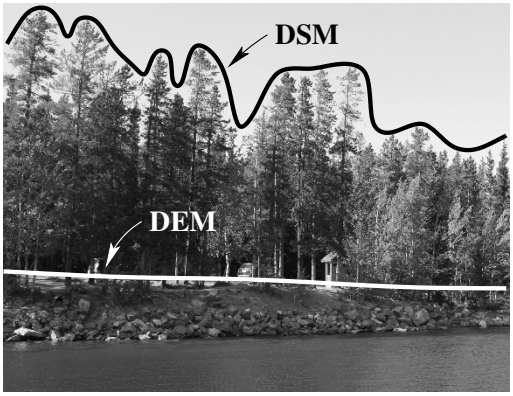


Fig. 16.2 The digital surface model (DSM) and the digital elevation model (DEM) representing the topographic surface contained in a map. When seen from above, the surface of the wood establishes the DSM. The shown undulations of the surface will only be observable from low altitudes, say less than 1000 m above ground. The topographic surface will not be visible from above at all positions, so interpolation between the neighbouring visible surface parts is required

We assume the surface to be recovered fulfills two conditions:

graph surface, 2½D surface

1. The surface can be represented as a function $z(x, y)$ where x and y are appropriate coordinates. In mathematics such a surface sometimes is called graph surface; in geosciences it often is called a $2^1/_2$D surface.
2. The surface is locally regular, e.g., piecewise flat or smooth, terms we will specify.

The first condition is fulfilled for a surface, which is observed from two images, where the distance from the surface is a function of the coordinates of one of the two images and the distances from surface points are derived by triangulation. This situation also may occur if the surface is observed using more than two images, e.g., when observing a terrain surface from an aeroplane. The condition also is fulfilled for 3D points acquired by a laser range scanner if the position of the scanner is used as the origin of a local coordinate system and x and y represent the direction of the laser beam.

The second condition can be used to interpolate between the observed surface points and at the same time to filter the given surface points. The regularity can be defined quite arbitrarily, but in order to yield acceptable results it should reflect the surface's properties. We explicitly allow for discontinuities. These may be depth discontinuities, caused by occlusions, or discontinuities of the normal directions, representing break lines.

The imposed restrictions are not too severe, as mentioned above: If the surface of interest cannot be represented as a function, then, due to its regularity, it can be partitioned such that each part is a function in a local coordinate system and, using the mutual relation between these coordinate systems, can be concatenated taking into account possible overlaps.

While the determination of the orientation parameters of the images may be achieved with a moderate number of points, say a few hundred per image, recovering the scene's surface will require a large number of points, say a few thousand per image. In the following we will therefore assume the number of available surface points is sufficiently large. The above-mentioned conditions therefore allow us to apply the methods discussed here, which can also be applied to laser range data.

16.1.3 Tasks to Be Solved

Given a set of 3D points and some information about the observation process, we want to reconstruct the surface. We assume the set of points is generated automatically, e.g., as the result of a bundle adjustment. Hence the selection of the positions of the points generally

cannot be assumed to depend on the surface structure. When derived from images, the point selection will depend on the image structure; when derived from a laser range finder, the points will be given in some grid structure. For example, image areas with no texture will not allow the selection of points, whereas highly textured image areas will lead to a dense point distribution. Interpolating between the given points, therefore, may not always allow us to capture accurately the surface between these points.

The situation can be seen in the first row of Fig. 16.3. It shows a simulated profile. Only eight points are assumed to be measured, which allow capturing the overall form of the true profile, but not all details.

Thus we face a set of problems:

- Reconstruction addresses two different types of tasks, depending on the context:
  - *Filtering* aims at eliminating the noise introduced by the observation process.
  - *Prediction* aims at deriving values $z(x, y)$ at positions which have not been observed. Prediction often is pure *interpolation*, namely if the interpolated surface $z(x, y)$ passes through the given points, i.e., if the given points are assumed to be free of measurement noise.

  As we are always faced with noisy measurements, we need to decide which method should be used for filtering and prediction.
  Among the numerous methods useful for this task, we address those which allow us to handle irregularly spaced data points, to properly handle the prior knowledge about the mensuration accuracy and the properties of the surface, and, if required, to provide information about the uncertainty of the resulting surface.
  We address mainly two methods: one specifies the properties of the surface using the statistics of derivatives of the surface function, for profiles using the concepts of autoregressive models, cf. (2.8.3), p. 52. The other is collocation, which specifies the mutual correlations between surface values, cf. (4.8.4), p. 174.
- Under what conditions can a continuous function $f(x)$ be reconstructed from samples $f(x_k), k = 1, ...$? Sampling theorems exist for different types of functions; one of such theorems we will discuss, cf. (16.10), p. 735. In all cases there is a relation between the curvature and the minimum density of the sampling. The samples in Fig. 16.3, top right, certainly are too sparse to capture the profile in the sub-figure, top left.
- How can we specify our knowledge about the surface's properties and use it for the reconstruction?
  This is the most challenging question, as the properties of surfaces vary tremendously. The profile in Fig. 16.3 is flat, i.e., approximately horizontal, in the left part ($x = 1, ..., 70$), and smooth, i.e., with low curvature, in the right part ($x = 71, ..., 200$). How can we specify the smoothness of the surface which reflects the scale at which we want to achieve the reconstruction?
  As we can expect surfaces to have different properties depending on the region, an average measure, say for smoothness, may not be proper for characterizing the structure of a complete surface.
- Can we derive this knowledge about the surface properties?
  Deriving the characteristics of the surface will require dense sampling, where the density reflects the user specifications.
  In the example in Fig. 16.3 we assume that the required density is implicitly given by the point spacing of the true profile, which is a sequence of 200 points. Obviously, we would need samples of such profiles with the same or similar characteristics in order to be able to derive measures for flatness or smoothness, see Fig. 16.4.
- How can we integrate the knowledge about the surface's structure into the reconstruction?
  Figure 16.3 shows the reconstructed profile for various assumptions about the characteristics of the profile.

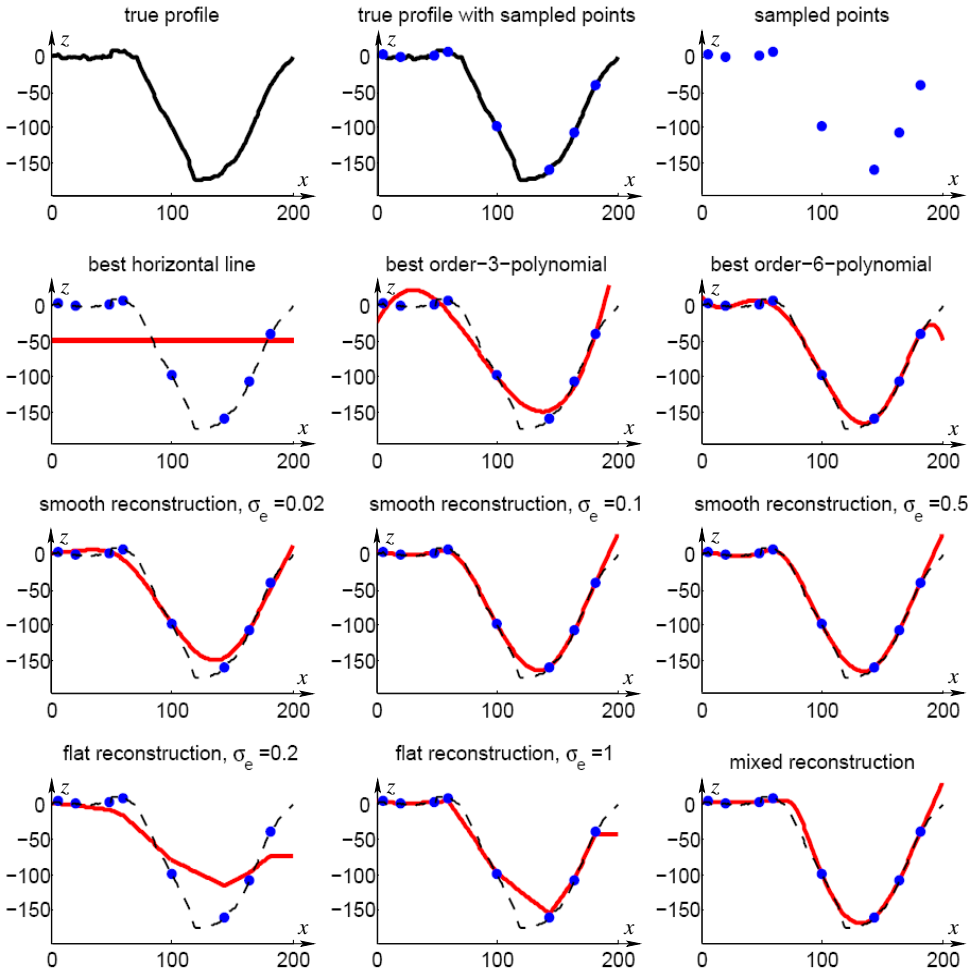


Fig. 16.3 Reconstructing a profile from given observations. The task is given in the first row, three different interpolation types are given in the next three rows with increasing adaption to the given data from left to right. **First row:** The true (discrete) profile consists of two parts: the left part (points 1-70) is flat, i.e., approximately horizontal, the right part (points 71-200) is smooth, i.e., not rough. It is sampled and observed at eight positions. The task is to reconstruct the profile from the eight observed points. **Second row:** Reconstruction using polynomials: Assuming the terrain to be horizontal (zeroth-order polynomial), assuming the terrain to be very smooth (third-order), assuming the data density to be representative for the profile (sixth-order). Obviously none of these three reconstructions with polynomials is acceptable. **Third row:** Reconstruction minimizing the curvature (second derivatives) assuming its standard deviation is σ_e: very smooth, smooth and moderately smooth. The right-hand reconstruction uses the knowledge about the degree of smoothness in the right-hand part of the true profile. It appears to be fairly good in the right part of the profile. Of course, due to the low density of the sampling details, the true surfaces are not reconstructible. **Bottom row:** The first two reconstructions assume the profile to be flat and therefore minimize the slope assuming the standard deviation of the first derivative to be σ_e: The reconstruction leads to a piecewise linear profile. The left reconstruction uses the knowledge of the flatness of the true profile in its left part. The rightmost reconstruction uses a characteristic of the true signal, namely its composition of a flat and a smooth part, with the transition at point 70. Observe: the left part of the reconstructed profile is significantly flatter than the reconstruction above

Assuming profiles to be polynomials, as in the second row, does not appear to be an adequate choice. However, assuming the profiles to be flat or smooth, as in rows 3 and 4, appears to be useful.

We will use a Bayesian setup, where we represent the assumed surface characteristics in the form of a prior probability and the characteristics of the observation process as likelihood.

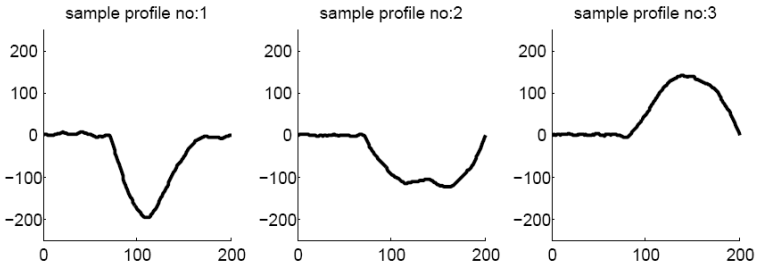


Fig. 16.4 Three sample profiles with the characteristics of the true profile in Fig. 16.3

- How accurate is the reconstruction? What does the reconstruction accuracy depend on?
  The accuracy of the reconstruction is needed at least for two purposes: (1) We want to compare the achieved accuracy with some user specification, and (2) we want to use the surface for deriving further quantities. For example, we may want to derive a slope model for checking the stability of a certain terrain region w.r.t. the potential for land slides.
  The accuracy certainly depends on the characteristics of the surface, on the point distribution, and on the measurement accuracy of the points.
  We will use the posterior probability of the reconstructed surface parameters for an analysis of the accuracy.
- How do we handle deviations from the assumed models?
  These may be outliers caused by the measurement process or violations of the regularity assumptions, e.g., at surface steps or break lines.
  We will handle surface steps and break lines as outliers in the prior knowledge. We do not address the real causes of these violations, as this would require modelling and identification of these causes using pattern recognition techniques.
- How do we represent the surface?
  We have at least two alternatives: (1) We use the $(x, y)_i$-position of the given points p_i as basis for the partitioning of the region of interest. The partitioning then might be realized by a triangular irregular network (TIN), e.g., based on a Delaunay triangulation. (2) Alternatively we partition the plane into a regular grid, e.g., with grid points $\{z(j\Delta x, k\Delta y), j, k \in \mathbb{Z}\}$ in the (x, y)-plane, see Fig. 16.5, left and centre.
  In both cases, we need an appropriate scheme for switching between the two representations, which can be based on the interpolation method provided by the specification of the surface.

triangular irregular network (TIN)

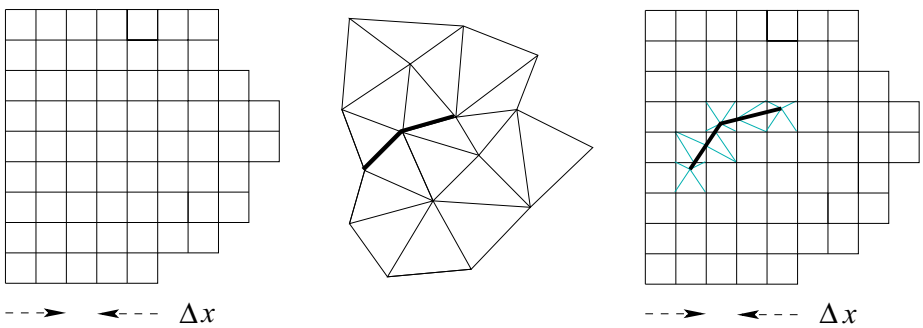


Fig. 16.5 Representations of 2 1/2-D surfaces: **Left:** grid structure with spacing Δx. **Middle:** triangular irregular network, including break lines (thick lines). **Right:** mixed representation including break lines

If break lines are provided, either from manual measurements or automatic identification, they may easily be integrated into a triangular network, but would require a mixed representation when starting with a grid structure, see Fig. 16.5 right.

This chapter first discusses means to represent surfaces which allow the integration of prior knowledge about surface characteristics.

16.2 Parametric 2½D Surfaces

16.2.1 Modelling 2½D Surfaces 733
16.2.2 Regularity Measures for Functions and 2½D Surfaces 739

This section discusses possible formalizations of the notion of smoothness for one- and two-dimensional functions and their parametric representations which are required for estimation. It is based on and follows ideas of Bosman et al. (1971), Ebner (1979), Grimson (1981), and Terzopoulos (1984). These ideas are also used by Finch et al. (2011) for interactive modelling of free form surfaces.

16.2.1 Modelling 2½D Surfaces

When reconstructing surfaces we need adequate parametrizations. Graph surfaces are most easily represented as a weighted sum of some adequate basis functions. For one- and two-dimensional functions, we then have

$$z(x) = \sum_k a_k f_k(x) = \boldsymbol{a}^\mathsf{T} \boldsymbol{f}(x)\,, \qquad z(x,y) = \sum_k a_k f_k(x,y) = \boldsymbol{a}^\mathsf{T} \boldsymbol{f}(x,y)\,, \tag{16.1}$$

where the basis functions f_k are fixed and the factors a_k parametrize the function, and where the vectors $\boldsymbol{f}(x)$ and $\boldsymbol{f}(x,y)$ collect the k basis functions.

16.2.1.1 Polynomial Basis Functions

For example, polynomial functions and 2½D surfaces use *monomials* as basis functions,

$$f_k(x) = x^k\,, \qquad f_k(x,y) = f_i(x) f_j(y)\,. \tag{16.2}$$

The parameters a_k can be derived from given data $(x,y)_n, n = 1, ..., N$, easily if we assume only the z_n-values to be uncertain, using the Gauss–Markov model,

$$\underbrace{\begin{bmatrix} z_1 \\ \dots \\ z_n \\ \dots \\ z_N \end{bmatrix}}_{\boldsymbol{z}} + \underbrace{\begin{bmatrix} v_1 \\ \dots \\ v_n \\ \dots \\ v_N \end{bmatrix}}_{\tilde{\boldsymbol{v}}} = \underbrace{\begin{bmatrix} f_0(x_1) & \dots & f_k(x_1) & \dots & f_K(x_1) \\ \dots & .. & \dots & \dots & \dots \\ f_0(x_n) & \dots & f_k(x_n) & \dots & f_K(x_n) \\ \dots & .. & \dots & \dots & \dots \\ f_0(x_N) & \dots & f_k(x_N) & \dots & f_K(x_N) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ \dots \\ a_k \\ \dots \\ a_K \end{bmatrix}}_{\tilde{\boldsymbol{a}}}, \tag{16.3}$$

which can take possibly individual variances σ_{z_n} for the z_n-values into account. This type of estimation can be transferred to all functions or 2½D surfaces which are represented as weighted sums of basis functions as long as the coefficient matrix is regular.

The determination of the K coefficients therefore requires the solution of a $K \times K$ equation system whose algorithmic complexity is of third-order. Therefore using basis

functions which lead to a full matrix A is only useful for a comparably small number of basis functions.

The estimated coefficients

$$\widehat{\boldsymbol{a}} = (\boldsymbol{A}^\mathsf{T}\boldsymbol{W}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{W}\boldsymbol{z} = \boldsymbol{A}_W^+\,\boldsymbol{z} \tag{16.4}$$

linearly depend on the given function values. This is why the resulting 2D function

$$z(x,y) = \boldsymbol{f}^\mathsf{T}(x,y)(\boldsymbol{A}^\mathsf{T}\boldsymbol{W}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{W}\boldsymbol{z} \tag{16.5}$$

also linearly depends on the given z-values, where the weight matrix is $\boldsymbol{W} = \boldsymbol{\Sigma}_{zz}^{-1} = \mathrm{Diag}([1/\sigma_{z_n}^2])$. Therefore the variance of an interpolated point $z(x,y)$ can easily be given:

$$\sigma_z^2(x,y) = \boldsymbol{f}^\mathsf{T}(x,y)(\boldsymbol{A}^\mathsf{T}\boldsymbol{W}\boldsymbol{A})^{-1}\boldsymbol{f}(x,y)\,. \tag{16.6}$$

This variance of course is realistic only if the surface can actually be represented by the assumed basis functions.

If the basis functions are orthonormal polynomials w.r.t. the given points, possibly taking the weighting into account, the normal equation matrix is a unit matrix, and therefore $\sigma_z^2(x,y) = |\boldsymbol{f}(x,y)|^2 = \sum_k f_k^2(x,y)$.

Exercise 16.6

Polynomials do not behave well at the border of the domain, and also require very high orders to represent complicated functions. Moreover, the basis functions have an infinite carrier, i.e., are nonzero in the domain $[-\infty,+\infty]$; they generally lead to a full coefficient matrix A in (16.3). Thus polynomials are only useful for small numbers of given data.

16.2.1.2 Trigonometric Basis Functions

Periodic functions are characterized by $f(x) = f(x+p)$, where p is the length of the period. They can be represented as sums of trigonometric functions. The basis functions are pairs of sine and cosine functions,

$$f_k(x) = \cos(2\pi kx/p)\,, \quad g_k(x) = \sin(2\pi kx/p)\,, \tag{16.7}$$

which are periodic and orthogonal on the interval $[0,p]$. The function f is represented by $f(x) = \sum_{k=0}^{\infty}(a_k f_k(x) + b_k g_k(x))$. The term with $k=0$ allows for functions with nonzero mean. The ratio k/p, having the unit of $1/x$, can be interpreted as a frequency, since it indicates how many cosine or sine waves cover the interval of length p.

Therefore the representation can be generalized to nonperiodic functions, leading to

$$f(x) = \int_{u=0}^{\infty} a(u)\cos(2\pi ux) + b(u)\sin(2\pi ux)\,\mathrm{d}u \quad \text{with} \quad \int_{x=-\infty}^{\infty} |f(x)|\mathrm{d}x < \infty\,. \tag{16.8}$$

The variation of $f(x)$ needs to be bounded. Here the frequency u represents the number of waves in the interval $[0,1]$.

This representation allows us to analyse under what conditions a continuous function can be reconstructed from a sample of function values. Though this result generally has no influence on the methods for sampling surface, it provides insight into the limitations of surface reconstruction. This is summarized in the following theorem.

Theorem 16.2.8: ***Sampling theorem.*** If a continuous function $f(x)$ can be represented by

$$f(x) = \int_{u=0}^{u_0} a(u)\cos(2\pi ux) + b(u)\sin(2\pi ux)\,\mathrm{d}u\,, \tag{16.9}$$

thus has an upper bound u_0 for the frequency, and if it is regularly sampled on an infinite grid $x_k = k\Delta x, k \in \mathbb{Z}$, then $f(x)$ can be reconstructed from the samples $f(x_k)$ if the sampling distance is small enough, namely

$$\Delta x \leq \frac{1}{2u_0} \tag{16.10}$$

cf. Whittaker (1915), Shannon and Weaver (1949), and Lüke (1999).

Observe, the boundary conditions for this theorem are strict: the sample needs to be infinite. However, there are generalizations to irregular sampling (cf. Aldroubi and Gröchenig, 2001) and to finite domains for finite samples (cf. Feichtinger et al., 1995).

16.2.1.3 Radial Basis Functions

A first possibility is to use basis functions with a decaying effect or even with finite support. Since in a first approximation, surface properties are independent of a rotation of the coordinate system, *radial basis functions* are useful. Then the surface is represented as *radial basis function*

$$z(x,y) = \sum_k a_k f\left(\frac{|\boldsymbol{x} - \boldsymbol{x}_k|}{h_k}\right) \tag{16.11}$$

using a set $\{\boldsymbol{x}_k\}_{k=1,\ldots,K}$ of reference points, and a function

$$f(x,y) = f(r) = f(|\boldsymbol{x}|) \tag{16.12}$$

dependent only on the length $r = |\boldsymbol{x}|$ of the vector $\boldsymbol{x}$. The set of reference points may be chosen either on a grid or irregularly, e.g., at the given points. Classical basis functions are

$$f(r) = \exp\left(-\frac{1}{2}r^2\right), \quad f(r) = \frac{1}{1+r^2} \quad \text{or} \quad f(r) = \begin{cases} (1-r^2)^4, & \text{if } r < 1 \\ 0, & \text{else} \end{cases}. \tag{16.13}$$

The parameter h_k defines the scale: large h_k lead to smooth functions, small h_k lead to rough functions. If the scale values h_k are too small compared to the distance of the reference points $\boldsymbol{x}_k$, the basis functions decay too fast to allow interpolation. Therefore the scale values h_k should be adapted to the average or to the local distance of the reference points. One suggestion is to choose h_k as a function of the radius of a circle enclosing a small number of nearest neighbours. Observe, the third basis function in (16.13) has a bounded domain, which leads to a sparse coefficient matrix $\boldsymbol{A}$ in (16.3), p. 733.

16.2.1.4 Basis Functions of Collocation

In Sect. 4.8.4, p. 174 we introduced the method of collocation, which assumes that the surface consists of a trend surface $t(x,y)$ and a random deviation $s(x,y)$ which is modelled as a stochastic process. Thus the predicted value at some arbitrary position (x,y) is given by

$$z(x,y) = \boldsymbol{a}^\mathsf{T} \boldsymbol{f}(x,y) + s(x,y)\,, \tag{16.14}$$

where the trend coefficients $\boldsymbol{a}$ and the signal component $s(x,y)$ need to be estimated from the given data.

The trend surface is usually assumed to be a low-order polynomial. The random deviations s are characterized by their covariance function $C_{ss}(d_{ii'})$, which in a first approximation only depends on the distance $d_{ii'} = |\boldsymbol{x}_i - \boldsymbol{x}_{i'}|$ between two points. If the surface is observed at K points, i.e., $(x,y,z)_k$, then the predicted height $z(x,y)$ is given by

$$z(x,y) = \widehat{\boldsymbol{a}}^\mathsf{T} \boldsymbol{f}(x,y) + \boldsymbol{c}^\mathsf{T}(x,y)\,(\boldsymbol{\Sigma}_{ss} + \boldsymbol{\Sigma}_{nn})^{-1}(\boldsymbol{z} - \boldsymbol{A}\widehat{\boldsymbol{a}})\,, \tag{16.15}$$

cf. (4.483), p. 175. Here the coefficients $\widehat{\boldsymbol{a}}$ specify the trend function. The covariances between the predicted signal $s = s(x,y)$ and the signals $s_k = s(x_k, y_k)$ of the given heights z_k are collected in the vector

$$\boldsymbol{c}(x,y) = \begin{bmatrix} c_1(x,y) \\ \cdots \\ c_k(x,y) \\ \cdots \\ c_K(x,y) \end{bmatrix} = \begin{bmatrix} \mathbb{C}\text{ov}(\underline{s},\underline{s}_1) \\ \cdots \\ \mathbb{C}\text{ov}(\underline{s},\underline{s}_k) \\ \cdots \\ \mathbb{C}\text{ov}(\underline{s},\underline{s}_K) \end{bmatrix} = \begin{bmatrix} C(d_1) \\ \cdots \\ C(d_k) \\ \cdots \\ C(d_K) \end{bmatrix}, \tag{16.16}$$

where the covariance depends only on the horizontal distance $d_k = |\boldsymbol{x} - \boldsymbol{x}_k|$ between the point to be predicted and the given points. The covariance matrices Σ_{ss} and Σ_{nn} specify the signal and the noise at the given points. The vector $\boldsymbol{z} = [z_k]$ collects the given heights and A is the coefficient matrix for determining the trend, as in (16.3), p. 733. The trend parameters, cf. (4.481), p. 175 (there called $\boldsymbol{x}$),

$$\widehat{\boldsymbol{a}} = A_W^+ \boldsymbol{z} = (A^\mathsf{T} W A)^{-1} A^\mathsf{T} W \boldsymbol{z} \quad \text{with} \quad W = (\Sigma_{nn} + \Sigma_{ss})^{-1}, \tag{16.17}$$

linearly depend on the given heights $\boldsymbol{z}$.

The predicted surface $z(x,y)$ now has the form

$$z(x,y) = [\widehat{\boldsymbol{a}}^\mathsf{T},\ (\boldsymbol{z}^\mathsf{T} - \widehat{\boldsymbol{a}}^\mathsf{T} A^\mathsf{T})(\Sigma_{ss} + \Sigma_{nn})^{-1}] \begin{bmatrix} \boldsymbol{f}(x,y) \\ \boldsymbol{c}(x,y) \end{bmatrix} \tag{16.18}$$

$$= \boldsymbol{z}^\mathsf{T} \left[(A_W^+)^\mathsf{T},\ (I_K - (AA_W^+)^\mathsf{T})(\Sigma_{ss} + \Sigma_{nn})^{-1}\right] \begin{bmatrix} \boldsymbol{f}(x,y) \\ \boldsymbol{c}(x,y) \end{bmatrix}. \tag{16.19}$$

The surface thus is a sum of basis functions, where the basis functions $\boldsymbol{f}(x,y)$ model the trend, e.g., with polynomials. The basis functions $\boldsymbol{c}(x,y)$ are covariance functions, which according to our assumption are radially symmetric. Therefore the collocation method can be seen as using a mixture of polynomials and radial basis functions. Obviously, the surface $z(x,y)$ also linearly depends on the given heights.

16.2.1.5 Splines

The previous methods are numerically complex if the number of basis functions becomes large. This is the reason why splines are regularly used for representing functions and surfaces (Piegl and Tiller, 1997). We mainly address linear splines on regular grids here, as they lead to efficient and versatile filtering and prediction schemes. Generalizations are straightforward.

The basis functions of splines are only nonzero in a limited domain. Their definition starts from a basis function, $f_0(x), x \in [-a,a]$. In the simplest case, this basis function then is shifted by integers k, yielding the kth basis function

$$f_k(x) = f_0(x-k), \quad k \in \mathbb{Z}. \tag{16.20}$$

Later we will choose a grid spacing different from 1.

hat function $\Lambda(x)$

One-Dimensional Linear Splines. For example, take piecewise linear basis functions with the hat function $\Lambda(x)$

$$f_0(x) := \Lambda(x) = \begin{cases} 1+x, & \text{for } x \in [-1,0] \\ 1-x, & \text{for } x \in [0,1] \\ 0, & \text{else} \end{cases}, \qquad f_k(x) := \Lambda_k(x) := \Lambda(x-k); \tag{16.21}$$

see Fig. 16.6. A piecewise linear function then is represented as

$$z(x) = \sum_k a_k \Lambda(x-k); \tag{16.22}$$

see Fig. 16.7.

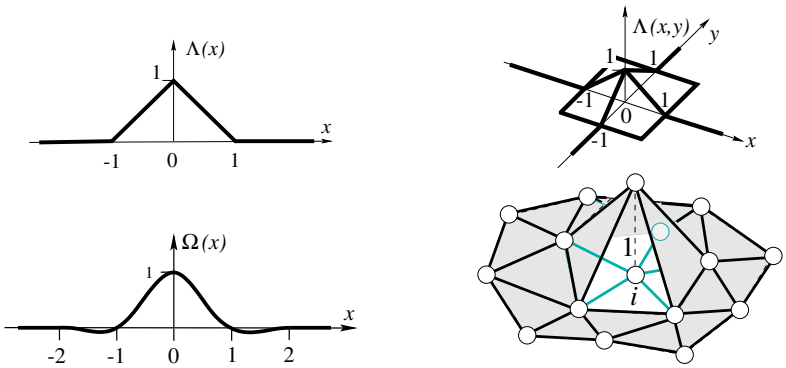


Fig. 16.6 Basis functions used for interpolation. **Top left:** One-dimensional linear interpolation with $\Lambda(x)$. **Top right:** Two-dimensional bilinear interpolation with $\Lambda(x,y) = \Lambda(x)\Lambda(y)$. **Bottom left:** One-dimensional cubic interpolation with $\Omega(x)$. **Bottom right:** Basis function $\Lambda_i(x,y)$ for linear interpolation on a triangular grid at point $\boldsymbol{x}_i$

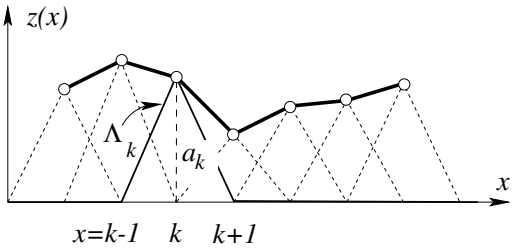


Fig. 16.7 Piecewise linear function $z(x)$, the weighted sum of regularly spaced basis functions $\Lambda_k(x)$, weights a_k, which here are identical to the function values at integer positions: $z(k) = a_k$

This representation of $z(x)$ has three important properties:

1. For integer arguments, the function value is $z(i) = a_i$, as $\Lambda(x-k)$ has value 1 at integers $x = k$. This simplifies the interpretation of the parameters a_k.
2. Between the values at integer arguments, the function is linear. Specifically, we have *linear interpolation*

$$z(x) = (1-s)z(\lfloor x \rfloor) + sz(\lceil x \rceil) \qquad s = x - \lfloor x \rfloor \,. \tag{16.23}$$

3. The function is continuous, but not differentiable at integer positions. It is called a C^0-continuous function, as only the function itself is continuous, not its first or higher derivatives.

Two-Dimensional Linear Splines. In two dimensions, we use the basis functions

$$\Lambda(x,y) = \Lambda(x)\Lambda(y) \tag{16.24}$$

and thus obtain the representation of the function on a grid with grid size 1:

$$z(x,y) = \sum_{ij} a_{ij}\Lambda_{ij}(x,y) \quad \text{with} \quad \Lambda_{ij} = \Lambda_i(x)\Lambda_j(y) = \Lambda(x-i)\Lambda(y-j)\,. \tag{16.25}$$

Again the function values at positions with integer coordinates (i,j) are $z(i,j) = a_{ij}$. The interpolation is bilinear, i.e., linear in the x- and y-directions. With the fractional parts of the coordinates $[x,y]^\mathsf{T}$,

$$s = x - \lfloor x \rfloor \,, \qquad t = y - \lfloor y \rfloor \,, \tag{16.26}$$

bilinear interpolation

we have the *bilinear interpolation* of the function $z(x, y)$, setting $a(i, j) := a_{ij} = z(i, j)$,

$$\begin{aligned} z(x,y) = (1-s)(1-t) &\cdot a(\lfloor x \rfloor, \lfloor y \rfloor) \\ +(1-s)t &\cdot a(\lfloor x \rfloor, \lceil y \rceil) \\ +s(1-t) &\cdot a(\lceil x \rceil, \lfloor y \rfloor) \\ +st &\cdot a(\lceil x \rceil, \lceil y \rceil)\,. \end{aligned} \tag{16.27}$$

The form of the surface in a grid cell is a hyperboloid of one sheet, as the weights for the four function values at the grid cell are a bilinear function of the coordinates. Specifically, we have

$$\Lambda(x,y) = (1-x)(1-y) = 1 - x - y + xy\,, \qquad x, y \in [0,1]\,. \tag{16.28}$$

If we collect the four values a_{ij} at the four corners in the matrix A and use the coefficient matrix M, namely

$$A = \begin{bmatrix} a(\lfloor x \rfloor, \lfloor y \rfloor) & a(\lfloor x \rfloor, \lceil y \rceil) \\ a(\lceil x \rceil, \lfloor y \rfloor) & a(\lceil x \rceil, \lceil y \rceil) \end{bmatrix} \quad \text{and} \quad M = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \tag{16.29}$$

Exercise 16.2

then the surface can be written as

$$z(x,y) = [1\,, s]\, \underbrace{MAM^{\mathsf{T}}}_{B} \begin{bmatrix} 1 \\ t \end{bmatrix} = \sum_{i=1}^{2} \sum_{j=1}^{2} b_{ij} s^{i-1} t^{j-1}\,. \tag{16.30}$$

This type of representation can be generalized to higher-order splines. Then we arrive at functions which are differentiable, i.e., with continuous first derivatives (C^1-continuous) or with continuous second derivatives (C^2-continuous). Generally a function is called C^n-continuous if its nth derivative is continuous.

C^n-continuous

Two-Dimensional Cubic Splines. A bicubic function within a cell $[0,1] \times [0,1]$ may be derived from the 4×4 surrounding z-values collected in a 4×4 matrix $A = [a_{ij}]$ using the coefficient matrix

$$M = \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{16.31}$$

and the vectors

$$\boldsymbol{s} = [1, s, s^2, s^3]^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{t} = [1, t, t^2, t^3] \tag{16.32}$$

Exercise 16.4

bicubic interpolation

by

$$z(x,y) = \boldsymbol{s}^{\mathsf{T}} \underbrace{MAM^{\mathsf{T}}}_{B} \boldsymbol{t} = \sum_{i=1}^{4} \sum_{j=1}^{4} b_{ij} s^{i-1} t^{j-1}. \tag{16.33}$$

The surface within a cell is defined by the heights z and the slopes z_x and z_y at the four corners of the cell. Therefore the slopes of neighbouring cells coincide and the surface is C^1-continuous. The interpolated surface may have over- or undershoots. As can be seen from the basis function $\Omega(x)$ in Fig. 16.6:

$$\Omega(x) = \begin{cases} 1 - \frac{5}{2}x^2 + \frac{3}{2}|x|^3, & \text{if } |x| < 1 \\ 2 - 4|x| + \frac{5}{2}x^2 - \frac{1}{2}|x|^3, & \text{if } |x| \in [1,2] \\ 0, & \text{else.} \end{cases} \tag{16.34}$$

This function has an undershoot at $\pm 4/3$. Specifically, it has a minimum of $\Omega(\pm 4/3) = -2/27 = -0.\overline{074}$. For example, cubic interpolation using $\Omega(x)$ for the step function $[..., 0, 0, 0, 1, 1, 1, ...]$ would lead to a C_1-continuous function with undershoots and overshoots, namely values below 0 and above 1, thus outside the range of the given data.

Observe, the simple interpretation $z(k) = a_k$ or $f(i,j) = a_{ij}$ of the parameters a_k or a_{ij}, i.e., that they represent the function values at the reference positions, does not get lost here. However, other definitions, which avoid under- and overshooting at steps, have a smoothing effect and the coefficients are not identical to the function values. This does not essentially change the estimation procedures discussed below, however.

The representation can also be generalized to irregular triangular grids. Then the linear basis functions are pointwise different, namely pyramids of height 1 over the polygon containing the triangles around a point, see Fig. (16.6), p. 737, bottom right.

16.2.2 Regularity Measures for Functions and 2½D Surfaces

The regularity of functions z may be measured in at least two different ways: by their slope or curvature variations (cf. Grimson, 1981; Terzopoulos, 1984; Blake and Zisserman, 1986; Terzopoulos, 1986) or by the width of the covariance function, cf. Sect. 2.8.2, p. 50. We start with regularity measures based on the slope and curvatures.

Measures based on Slope or Curvature Variations: A simple measure using the slope $z'(x) = \tan\alpha(x)$ of a function would be

$$F_1 = \int z'^2(x)\,\mathrm{d}x\,. \tag{16.35}$$

It measures the *flatness* of the function since F_1 only is zero if $z = a$ with some constant a, see Fig. 16.8 left.[1] The *flatness of a curve* is characterized by small values of F_1. This model is also called the *weak string model*, as the flatness measure F_1 is proportional to the energy of a string.

flatness

weak string model

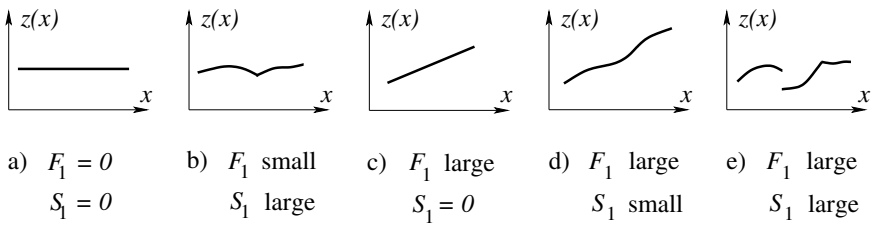


Fig. 16.8 Flat and smooth functions. The flatness is measured with F_1 integrating the squared slopes. Functions with small F_1 are flat. The smoothness is measured with S_1 integrating the squared curvatures. Functions with small S_1 are smooth. The function in b) is piecewise flat, the function in e) is piecewise smooth

Similarly, the flatness of a 2½D surface

$$z = z(x,y) \tag{16.36}$$

could be measured with the slope, e.g., via the tangent of the slope angle $\alpha(x)$,

$$|\nabla z(x)| = \tan\alpha(x)\,, \tag{16.37}$$

using the gradient

$$\nabla z(x,y) = \begin{bmatrix} \dfrac{\partial z(x,y)}{\partial x} \\ \dfrac{\partial z(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} z_x(x,y) \\ z_y(x,y) \end{bmatrix}\,, \tag{16.38}$$

[1] Actually F_1 measures the steepness. Priors on surfaces will require the surfaces to be flat, or have large flatness, therefore require to minimize F_1. This is why we further on use the terms *flat* and *flatness*.

by

$$F_2 = \int |\nabla z(x,y)|^2 \, \mathrm{d}x\mathrm{d}y = \int \left(z_x^2(x,y) + z_y^2(x,y)\right) \, \mathrm{d}x\mathrm{d}y \,. \tag{16.39}$$

weak membrane model

Again, surfaces with $F_2 = 0$ will be horizontal planes. This model also is called the *weak membrane model*. Similarly, surfaces with high flatness or smoothness are characterized by small F_2 or S_2, respectively, see Fig. 16.9.

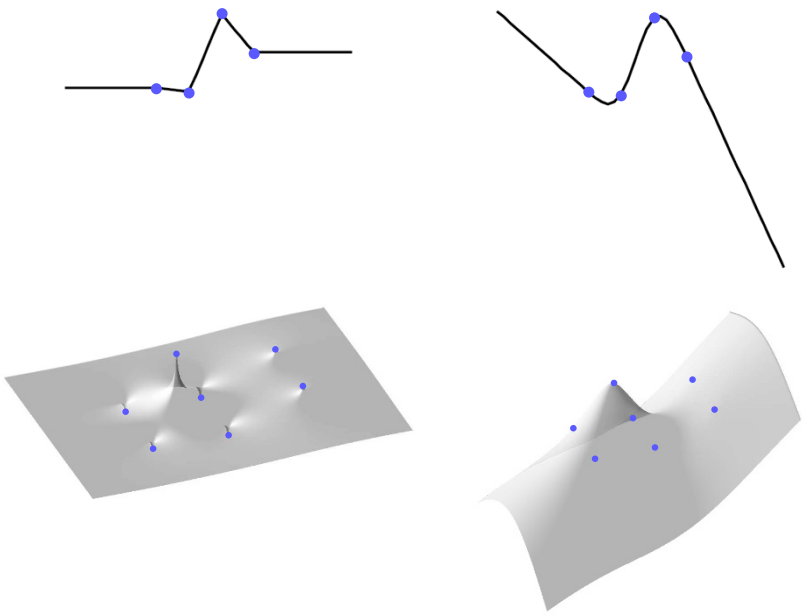


Fig. 16.9 Flat and smooth functions. **Top**: 1D functions, profiles. **Bottom**: 2D functions, surfaces. **Top left:** Best fitting 1D function through four points assuming the function to be flat, i.e., following the weak string model. **Top right:** Best fitting 1D function through the same four points assuming the function to be smooth, i.e., following the thin rod model. The two lines can be viewed as an infinite horizontal string or thin rod which need to pass through the four points. **Bottom left:** Best fitting 2D function through seven points assuming the function to be flat, i.e., following the membrane model. **Bottom right:** Best fitting 2D function through seven points assuming the function to be smooth, i.e., following the thin plate model. The surface can be viewed as an infinite horizontal membrane or thin plate which has to pass through the seven points

Flat surfaces with low values F_2 will often also be smooth. However, they may show sharp bends with high curvature, which would not be termed smooth, see Fig. 16.8, middle left. Additionally, smooth surfaces may be tilted, like roofs of buildings. The curvature $\kappa = 1/r$ is the inverse radius of the osculating circle at a point. For functions, we could measure the smoothness by[2]

smoothness

$$S_1 = \int \kappa^2(x) \, \mathrm{d}x = \int \frac{z''^2(x)}{(1 + z'^2(x))^3} \, \mathrm{d}x \,. \tag{16.40}$$

A function with $S_1 = 0$ will be a linear function, see Fig. 16.8, centre right. If the slope of a function is small, say below 0.15 or 10°, we also could neglect the first derivative in the denominator and use the approximation

thin rod model

$$S_1^a = \int z''^2(x) \, \mathrm{d}x \,. \tag{16.41}$$

[2] Again, the value S_1 actually measures the roughness of the function, but generally we will require the smoothness of a function to be high, thus low values for S_1.

This model is also called the *thin rod model*.

The smoothness of 2½D surfaces could similarly be measured using the sum of the two squared principal curvatures,

$$S_2 = \int \left(\kappa_1^2(x,y) + \kappa_2^2(x,y)\right) \, \mathrm{d}x\mathrm{d}y = \int \left(4H(x,y) - 2K(x,y)\right) \, \mathrm{d}x\mathrm{d}y \,, \tag{16.42}$$

with the Gaussian and the mean curvatures,

$$K = \frac{z_{xx}z_{yy} - z_{xy}^2}{(1+z_x^2+z_y^2)^2} \,, \qquad H = \frac{(1+z_x^2)z_{xx} - 2z_x z_y z_{xy} + (1+z_y^2)z_{yy}}{2(1+z_x^2+z_y^2)^{3/2}} \tag{16.43}$$

(cf. do Carmo, 1976). This model is also called the *thin plate model* as its bending energy is proportional to S_2. Again, for surfaces with a small slope, this smoothness measure can be approximated by

thin plate model

$$S_2^a = \int \mathrm{tr}|\boldsymbol{H}^2(x,y)| \, \mathrm{d}x\mathrm{d}y = \int \left(z_{xx}^2(x,y) + 2z_{xy}^2(x,y) + z_{yy}^2(x,y)\right) \, \mathrm{d}x\mathrm{d}y \,, \tag{16.44}$$

with the symmetric Hessian matrix

$$\boldsymbol{H}(x,y) = \begin{bmatrix} z_{xx}(x,y) & z_{xy}(x,y) \\ z_{xy}(x,y) & z_{yy}(x,y) \end{bmatrix} \tag{16.45}$$

of the 2½D surface containing the second partial derivatives of $z(x,y)$. In particular, the mixed partial derivative z_{xy} measures the degree of *torsion*. The integrand in (16.44) contains what is called the *quadratic variation* (cf. Grimson, 1981)

torsion z_{xy}

quadratic variation

$$Q = \mathrm{tr}|\boldsymbol{H}^2| = z_{xx}^2 + 2z_{xy}^2 + z_{yy}^2 \,. \tag{16.46}$$

Surfaces with $Q(x,y) = 0$ are planar. Observe, all three partial derivatives are needed.

Example 16.2.51: ***Three curvature measures.*** Figure 16.10 shows three surfaces,

$$z_1(x,y) = x^2 \,, \qquad z_2(x,y) = y^2 \,, \quad \text{and} \quad z_3(x,y) = xy \,, \tag{16.47}$$

with

$$\boldsymbol{H}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \,, \qquad \boldsymbol{H}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \,, \quad \text{and} \quad \boldsymbol{H}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \,, \tag{16.48}$$

with at least one of the three partial derivatives, which are nonzero. ◇

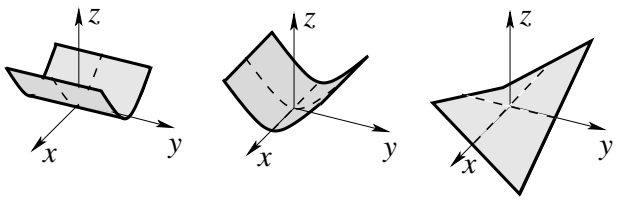


Fig. 16.10 Smooth surfaces with each second partial derivative nonzero. **Left:** $z = x^2, z_{xx} = 2$, curvature in the x-direction. **Middle:** $z = y^2, z_{yy} = 2$, curvature in the y-direction. **Right:** $z = xy, z_{xy} = 1$, torsion

Measures Based on the Width of the Covariance Function. The covariance function of a stationary stochastic process $\underline{z}(x)$ is given by, cf. (2.187), p. 49

$$C_{zz}(d) = \mathbb{C}\mathrm{ov}(\underline{z}(t), \underline{z}(t+d)) \quad \text{with} \quad \sigma_z^2 = C_{zz}(0) \geq |C_{zz}(d)| \,, \tag{16.49}$$

where $d = |\boldsymbol{x} - \boldsymbol{y}|$ is the distance between two points. Usually $C_{zz}(d)$ is a decaying function. The function is often characterized by the distance d_0 where the decay is 50% or

e^{-1}. Take as an example the exponential covariance function $C_{zz}(d) = \sigma_z^2 \exp(-\frac{1}{2}(d/d_0)^2)$ of a Gaussian process. Large values d_0 lead to smooth functions, small values d_0 lead to rough functions (see Fig. 16.11), where the transition from the Rhine valley to the Black Forest is simulated by changing the reference distance of the covariance function together with the mean and the standard deviation.

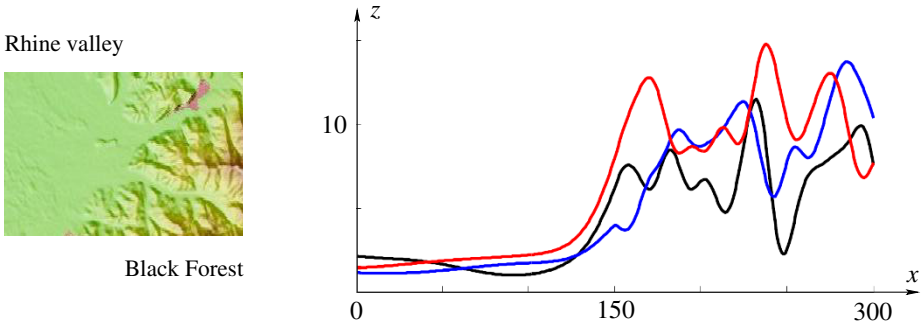


Fig. 16.11 An inhomogeneous situation where the left region is assumed to be smooth and the right region is assumed to be rough. **Left:** A real situation: Digital elevation model of a part of the Rhine valley and the Black Forest, with a smooth transition. **Right:** Three simulated samples for west–east profiles. Assumed characteristics of the generating process $x = 1, ..., 150$: $\mu_z = 1.5$, $\sigma_z = 0.5$, $d_0 = 40$, values $x = 151, ..., 300$: $\mu_z = 9$, $\sigma_z = 3$, $d_0 = 10$

We now address the problem of reconstructing a surface $z(x, y)$ from a set of given points. This problem is underdetermined, in case the number of points is less than the number of parameters necessary for describing the surface. Therefore we need to incorporate prior information. We first discuss this incorporation of prior information for one-dimensional profiles. As the discussion is largely application-independent, we refer to the profiles as signals, as in the area of signal processing.

16.3 Models for Reconstructing One-Dimensional Surface Profiles

16.3.1 Surface Reconstruction as Bayesian Estimation 742
16.3.2 The Choice of the Smoothness Prior Revisited 748
16.3.3 Changing the Grid Spacing 753
16.3.4 Violations of the Basic Model 754

16.3.1 Surface Reconstruction as Bayesian Estimation

We treat surface reconstruction as Bayesian estimation.[3] Let the surface be represented by a parametric function,

$$z = z(\boldsymbol{x}; \boldsymbol{a}) = z(x, y; \boldsymbol{a}) \,, \tag{16.50}$$

where $\boldsymbol{a}$ is a vector of parameters, e.g., of a polynomial function $z = \sum_{kj} a_{kj} x^k y^j$. Let the surface be observed by a set of M observations l_m which are noisy versions of the function at some given positions $\boldsymbol{x}_m = [x; y]_m, 1, ..., M$:

$$\underline{l}_m | \boldsymbol{a} \sim p(l_m | \boldsymbol{a}) \,, \quad m = 1, \ldots, M \,. \tag{16.51}$$

[3] We actually discuss maximum a posteriori estimates, but use the term Bayesian estimate, see the discussion in Sect. 4.1.1, p. 76.

Let us further assume the properties of the function z can be captured by the probability density function $p(\boldsymbol{a})$ of the parameters $\boldsymbol{a}$.

Then the task is to find the maximum of the a posteriori probability

$$p(\boldsymbol{a}|\boldsymbol{l}) = \frac{p(\boldsymbol{l}|\boldsymbol{a})p(\boldsymbol{a})}{p(\boldsymbol{l})}\,, \tag{16.52}$$

with the likelihood function

$$L(\boldsymbol{a}) = p(\boldsymbol{l}|\boldsymbol{a}) = \prod_{m=1}^{M} p(l_m|\boldsymbol{a}) \tag{16.53}$$

and the prior $p(\boldsymbol{a})$.

We apply several stochastical models for the observation process, including symmetric and asymmetric outlier processes. In order to discuss the form of useful priors we need to discuss the parametrization of the surfaces. For illustrating the principles, we also discuss one-dimensional surface profiles, i.e., functions $z = z(x; \boldsymbol{a})$.

Remark: We simplified the stochastical model of the observation process, assuming only the function value z to be distorted by noise, not the position (x, y). The simplification will generally lead to a nonlinear Gauss–Markov model. The simplification will be acceptable if the slope of the function is not too large. Otherwise we need to use the constraints $z(\widehat{x}_m; \widehat{\boldsymbol{a}}) = \widehat{l}_m$ and find the most probable estimated parameters $\widehat{\boldsymbol{a}}$ under these constraints. We will not follow this line here. ◇

We now make the integration of observations and prior knowledge about signals explicit. Reconstruction of a signal uses two types of information:

1. the observed signal values, and
2. prior knowledge about the true signal.

There are at least three ways to approach the inclusion of the prior information within this estimation problem, see the discussion in Sect. 4.1:

1. Maximum a posteriori estimation: Here the prior information needs to be specified using some probability density function, as already indicated at the beginning of this chapter. This is the most general setup.
2. Using pseudo-observations: Here certain properties of the signal are represented by observations in a Gauss–Markov model using only the information about the variances. This setup is conceptually simpler, and has the advantage of being general enough in our context.
3. Weighted least squares with regularization: Here the underconstrained problem is made solvable by some regularizer. This is a classical nonstatistical approach, to which we want to give a statistical interpretation

If we conceptually use the same prior information, all three approaches lead to the same estimates. We demonstrate this with a simplified setup and illustrate its potential to handle outliers and discontinuities.

16.3.1.1 The Model

In this section we assume the unknown signal $\boldsymbol{z} = [z(k)], k = 1, ..., K$ to be an equally spaced sequence. We assume some $M \leq K$ of the values on the grid to be directly observed, see Fig. 16.12. These observations are noisy versions of the unknown signal:

$$l_m = z(m) + n_m\,, \quad m = 1, ..., M\,. \tag{16.54}$$

The noise $\underline{n}_m$ is assumed to be white with mean 0 and standard deviation σ_n, independent of m. This corresponds to the fact that we assume that the observations are statistically independent and have the same weights $w_n = 1/\sigma_n^2$. The task is to estimate the unknown

signal from the given observations. This includes the estimation of function values z_k where no observations are available. Thus we simultaneously want to filter the given data, i.e., reduce the noise, and predict nonobserved function values. We subsume both tasks together under the notion reconstruction.

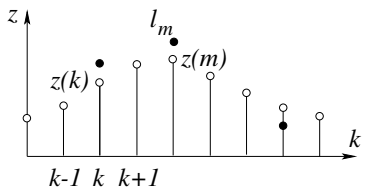


Fig. 16.12 A discrete profile $z(k), k = 1, ..., K$ (open circles), is observed at some positions leading to observations $l_m, m \in \{1, ..., K\}$ (black dots). The task is to estimate the profile from the given observations. The smoothness is measured by the discrete second derivatives of the profile

This setup, namely that only observations at grid points are available, is used to keep the derivation simple. We do not address any function values except the grid values, i.e., we do not address any interpolation between integer positions. In reality it will be advantageous to assume that the observations are at noninteger positions, which we will discuss when reconstructing $2^1/_2$D surfaces.

In addition, we assume the signal to be smooth. As we do not specify any interpolation scheme, we need to replace the curvature measure by some adequate discrete approximation. For simplicity,

1. we neglect the slope of the function. Thus we approximate the curvature by the second derivative, and
2. we replace the differential quotients by the difference quotients.

Therefore the smoothness is assumed to be characterized by (small) second derivatives,

$$e_k = z(k-1) - 2z(k) + z(k+1) = a_{k-1} - 2a_k + a_{k+1}\,, \qquad k = 2, ..., K-1\,, \quad (16.55)$$

of the unknown signal. The second derivatives can be determined at all positions except the two ends of the signal. Smoothness thus can be modelled by second derivatives with mean zero and a small standard deviation σ_e. This can be used to weight this prior information with the weight $w_e = 1/\sigma_e^2$ at each grid point, except for the two end points.

Observe, even if the underlying signal is continuous but represented by the sequence $\{a_k\}$, we can measure the smoothness by the difference quotients, as they are good enough approximations of the curvature of the underlying continuous function if it is smooth enough. We do not discuss the precise conditions of this generalization, which is related to the bandwidth of the continuous function here (cf. Oppenheim and Schafer, 1975), as this is beyond the scope of the chapter.

16.3.1.2 Maximum A Posteriori Estimation

We collect the unknown parameters in the K-vector $\boldsymbol{a}$, the M observations in the vector $\boldsymbol{l}$. Then the task is to establish a model of the form

$$p(\boldsymbol{a}|\boldsymbol{l}) \propto p(\boldsymbol{l}|\boldsymbol{a})\, p(\boldsymbol{a}) \quad (16.56)$$

and for a given set of observations $\boldsymbol{l}$ find those unknown parameters $\boldsymbol{a}$ of the signal which maximize (16.56). This then leads to the maximum a posteriori estimate. We only need to specify the likelihood function $L(\boldsymbol{a}) = p(\boldsymbol{l}|\boldsymbol{a})$ and the prior on the signal $p(\boldsymbol{a})$.

The *likelihood* can be derived as the joint probability of all observations given the parameters $\boldsymbol{a}$. As we assume the observations to be independent, their likelihood can be

partitioned into a product of M factors:

$$p(\boldsymbol{l}|\boldsymbol{a}) = \prod_{m=1}^{M} p(l_m|\boldsymbol{a})\,. \tag{16.57}$$

We now need to specify the distribution. As we have no information apart from the variance of the noise, we assume a Gaussian distribution:

$$p(l_m|\boldsymbol{a}) \propto \exp\left(-\frac{1}{2}\left(\frac{l_m - a_m}{\sigma_n}\right)^2\right)\,. \tag{16.58}$$

We do not need to specify the normalization factor as we assume σ_n to be known.

The *prior* specifies our knowledge about the signal. As we assumed that the signal's smoothness should be measured using the second derivatives e_k, their mean should be zero and their standard deviation σ_e. Thus the complete prior can be factorized as follows:

$$p(\boldsymbol{a}) = \prod_{k=2}^{K-1} p_k(\boldsymbol{a})\,, \tag{16.59}$$

where

$$p_k(\boldsymbol{a}) \propto \exp\left(-\frac{1}{2}\left(\frac{e_k}{\sigma_e}\right)^2\right) = \exp\left(-\frac{1}{2}\left(\frac{a_{k-1} - 2a_k + a_{k+1}}{\sigma_e}\right)^2\right)\,. \tag{16.60}$$

The *optimization problem* now is for given $\boldsymbol{l}$ to maximize

$$p(\boldsymbol{a}|\boldsymbol{l}) \propto \prod_{m=1}^{M} \exp\left(-\frac{1}{2}\left(\frac{l_m - a_m}{\sigma_n}\right)^2\right) \prod_{k=2}^{K-1} \exp\left(-\frac{1}{2}\left(\frac{a_{k-1} - 2a_k + a_{k+1}}{\sigma_e}\right)^2\right) \tag{16.61}$$

w.r.t. the unknown parameters $\boldsymbol{a}$. By taking the negative logarithm we see this is equivalent, for given $\boldsymbol{l}$, to minimizing

$$\Omega_{\mathrm{ML}}(\boldsymbol{a}) = -\ln p(\boldsymbol{a}|\boldsymbol{l}) - C = \sum_{m=1}^{M} \frac{1}{2}\left(\frac{l_m - a_m}{\sigma_n}\right)^2 + \sum_{k=2}^{K-1} \frac{1}{2}\left(\frac{a_{k-1} - 2a_k + a_{k+1}}{\sigma_e}\right)^2 \tag{16.62}$$

w.r.t. the unknown parameters $\boldsymbol{a}$. Observe, this expression is quadratic in the unknown parameters a_k and thus has a unique minimum. The constant C takes care of all constant factors not made explicit before.

The *solution* can be found by taking the derivatives w.r.t. all unknown parameters and requiring them to be zero. This can be simplified if we write the function Ω_{ML} as a quadratic form in the parameters $\boldsymbol{a}$. We introduce the following vectors and matrices referring to the observations $\boldsymbol{l}$,

$$\underset{M\times 1}{\boldsymbol{l}} = [l_m]\,, \quad \underset{M\times K}{\boldsymbol{A}_1} = [\delta_{mk}]\,, \quad \underset{M\times M}{\boldsymbol{W}_{11}} = \frac{1}{\sigma_n^2}\boldsymbol{I}_M\,. \tag{16.63}$$

The vector $\boldsymbol{l}$ collects all observations. The matrix $\boldsymbol{A}_1$ has value $\boldsymbol{A}_1(m,k) = 1$ if the mth observation l_m refers to unknown a_k. All other elements are zero. Finally, the matrix $\boldsymbol{W}_{11}$ is a diagonal matrix with the weights for the observations. Analogously, for the second term in (16.62), we have[4]

[4] Following our convention, the matrix $\boldsymbol{A}_2$ should have been defined as a transposed one, as it has fewer rows than columns. In order to simplify the expressions we use the definition above.

$$\underset{(K-2)\times K}{A_2} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ \cdots & & & & & \cdots \\ & & & 1 & -2 & 1 \end{bmatrix}, \quad \underset{(K-2)\times(K-2)}{W_{22}} = \frac{1}{\sigma_e^2} I_{K-2}\,. \tag{16.64}$$

Then we have all second derivatives,

$$\boldsymbol{d} = A_2 \boldsymbol{a}\,. \tag{16.65}$$

Therefore the optimization function can be written as

$$\Omega_{\mathrm{ML}}(\boldsymbol{a}) = \frac{1}{2}(\boldsymbol{l}^\mathsf{T} - \boldsymbol{a}^\mathsf{T} A_1^\mathsf{T}) W_{11}(\boldsymbol{l} - A_1\boldsymbol{a}) + \frac{1}{2}\boldsymbol{a}^\mathsf{T} A_2^\mathsf{T} W_{22} A_2 \boldsymbol{a}\,. \tag{16.66}$$

We finally have to solve the following equation system for $\boldsymbol{a}$:

$$\frac{\partial \Omega_{\mathrm{ML}}(\boldsymbol{a})}{\partial \boldsymbol{a}^\mathsf{T}} = \boldsymbol{0}\,. \tag{16.67}$$

Using the differentiation rule for quadratic forms $\partial \boldsymbol{a}^\mathsf{T} B \boldsymbol{a} / \partial \boldsymbol{a}^\mathsf{T} = 2B\boldsymbol{a}$ with symmetric matrix B, it explicitly reads as

$$\frac{\partial \Omega_{\mathrm{ML}}(\boldsymbol{a})}{\partial \boldsymbol{a}^\mathsf{T}} = -A_1^\mathsf{T} W_{11}(\boldsymbol{l} - A_1\boldsymbol{a}) + A_2^\mathsf{T} W_{22} A_2 \boldsymbol{a} = \boldsymbol{0}\,, \tag{16.68}$$

which leads to the linear equation system

$$\left(A_1^\mathsf{T} W_{11} A_1 + A_2^\mathsf{T} W_{22} A_2\right) \widehat{\boldsymbol{a}} = A_1^\mathsf{T} W_{11} \boldsymbol{l} \tag{16.69}$$

for the estimated parameters $\widehat{\boldsymbol{a}}$. They yield the global optimum of the optimization function $\Omega_{\mathrm{ML}}(\boldsymbol{a})$. The covariance matrix of the estimated parameters therefore is

$$\Sigma_{\widehat{a}\widehat{a}} = \left(A_1^\mathsf{T} W_{11} A_1 + A_2^\mathsf{T} W_{22} A_2\right)^{-1}\,. \tag{16.70}$$

The matrix A_2 in (16.64) has the rank $K-2$, thus a rank defect of 2: therefore, at least two points are necessary to arrive at a full rank equation system (16.69). If we only had two points the resultant function would be a straight line through the two points. Finally, we find the prior reads

$$p(\boldsymbol{a}) \propto \exp\left(-\frac{1}{2}\boldsymbol{a}^\mathsf{T} W_{aa} \boldsymbol{a}\right) \quad \text{with} \quad W_{aa} = A_2^\mathsf{T} W_{22} A_2\,. \tag{16.71}$$

It is a singular Gaussian distribution, since the weight matrix W_{aa} is rank deficient.

16.3.1.3 Gauss–Markov Model with Fictitious Observations

We now derive a Gauss–Markov model which leads to the same results as the maximum likelihood estimation by representing the prior knowledge as fictitious observations, in the following called *regularizing* observations.

regularizing observations

We start with the linear Gauss–Markov model for the M real observations, which simply reads as

$$\mathbb{E}(\underline{l}_m) = a_m\,, \qquad \mathbb{D}(\underline{l}_m) = \sigma_m^2\,, \quad m = 1, ..., M \tag{16.72}$$

or

$$l_m + v_{l_m} = a_m\,, \qquad \mathbb{D}(\underline{l}_m) = \sigma_m^2\,, \quad m = 1, ..., M\,. \tag{16.73}$$

As generally $M < K$, we need to make our pre-knowledge about the signal explicit. The pre-knowledge may be obtained by visual inspection or the analysis of real profiles. Here

we assume a typical signal to have small second derivatives with average deviations of σ_e. Let us denote these observations of the derivatives by δ_k. Then we obtain the following observation equation, cf. the discussion of (4.14), p. 79:

$$\mathbb{E}(\underline{\delta}_k) = a_{k-1} - 2a_k + a_{k+1}\,, \qquad \mathbb{D}(\underline{\delta}_k) = \sigma_e^2\,, \quad k = 2, ..., K-1 \tag{16.74}$$

or

$$\delta_k + v_{\delta_k} = a_{k-1} - 2a_k + a_{k+1}\,, \qquad \mathbb{D}(\underline{\delta}_k) = \sigma_e^2\,, \quad k = 2, ..., K-1\,. \tag{16.75}$$

As we visually on average observe $\delta_k = 0$, we use this as a *regularizing observation*; therefore, the vector of observations is

regularizing observations for the second derivatives

$$\boldsymbol{\delta} = \mathbf{0}\,. \tag{16.76}$$

Using the matrices A_1 and A_2, cf. (16.63), p. 745 and (16.64), p. 746, this leads to the linear Gauss–Markov model

$$\begin{bmatrix} \boldsymbol{l} \\ \boldsymbol{\delta} \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_l \\ \boldsymbol{v}_\delta \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \boldsymbol{a} \tag{16.77}$$

and

$$\mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{l}} \\ \underline{\boldsymbol{\delta}} \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{ll} & \\ & \Sigma_{ee} \end{bmatrix} = \begin{bmatrix} \sigma_n^2 I_M & \\ & \sigma_e^2 I_{K-2} \end{bmatrix}. \tag{16.78}$$

Minimizing the weighted sum of the residuals

$$\Omega(\boldsymbol{a}) = \boldsymbol{v}_l^\mathsf{T} \Sigma_{ll}^{-1} \boldsymbol{v}_l + \boldsymbol{v}_\delta^\mathsf{T} \Sigma_{ee}^{-1} \boldsymbol{v}_\delta\,, \tag{16.79}$$

which depend on the unknown parameters $\boldsymbol{a}$, yields the normal equation system, since $\boldsymbol{\delta} = \mathbf{0}$,

$$\left(A_1^\mathsf{T} W_{11} A_1 + A_2^\mathsf{T} W_{22} A_2\right) \boldsymbol{a} = A_1^\mathsf{T} W_{11} \boldsymbol{l}\,, \tag{16.80}$$

which is identical to the one of the maximum likelihood estimation, as expected.

Obviously, the prior information $p(\boldsymbol{a})$ can be integrated into the estimation process by augmenting the Gauss–Markov model for the observed function values l_m by observations made by the user. He is supposed to have observed the scene to be smooth and integrated this information into the Gauss–Markov model, as if they were observations made by an instrument. Thus conceptually we do not distinguish between observations made by some measuring device or algorithm and observations made by human inspection. This way prior information can be seen as prior observations.

16.3.1.4 Weighted Least Squares with Regularization

We finally give a derivation of the solution using a weighted least squares approach. The goal here is to find the best values a_k given the observed values l_m. Unless we have observed all function values of the sequence, this problem is underdetermined, thus not well-posed in the sense of Hadamard, see the footnote on page 82. The problem becomes well-posed by regularization, which guarantees that a unique solution exists. It changes continuously with the initial conditions, here the observed values.

In our case regularization is achieved by additionally requiring the sum of the squared second derivatives to be small. With two weighting parameters w_n and w_e, we therefore arrive at the optimization function

$$\Omega(\boldsymbol{a}) = w_n \sum_{m=1}^{M} (l_m - a_m)^2 + w_e \sum_{k=2}^{K-1} (a_{k-1} - 2a_k + a_{k+1})^2\,. \tag{16.81}$$

The first term enforces that the signal points a_m are not too far from the corresponding observations l_m. The second term is necessary to *regularize* the otherwise under-determined

problem. It enforces that the second derivatives of the unknown signal are not too large in magnitude. The two weights w_n and w_e are chosen to balance the two terms. Obviously, only their ratio $\lambda = w_e/w_n$ is relevant for the solution, which is why this problem often is written as

$$\Omega'(\boldsymbol{a}) = \sum_{m=1}^{M} (l_m - a_m)^2 + \lambda \sum_{k=2}^{K-1} (a_{k-1} - 2a_k + a_{k+1})^2 \,, \tag{16.82}$$

without specifying the meaning of the free parameter λ.

However, setting $w_n = 1/\sigma_n^2$ and $w_e = 1/\sigma_e^2$, i.e., specifying the factor λ as the ratio of the two variances,

$$\lambda = \frac{\sigma_n^2}{\sigma_e^2} \,, \tag{16.83}$$

the optimal value for the signal $\boldsymbol{a}$ is the same as in the previous two cases.

The problem of signal smoothing based on possibly sparsely observed signals can be solved in three ways, which have different interpretations. If the noise is Gaussian and the weights are chosen as inverse variances, the three approaches yield the same result. This allows the second and third solutions to be interpreted as maximum a posteriori estimates, which makes generalizations concerning the chosen distributions possible.

In the following, we use the method of regularizing observations for modelling, due to its simplicity.

16.3.2 The Choice of the Smoothness Prior Revisited

Whereas the model for the observed function values usually can be related to the mensuration process, which allows us to provide acceptable standard deviations σ_{l_m} for the observed values l_m, choosing an adequate model for the prior is application-dependent. We therefore need to revisit the discussion on how to measure smoothness and arrive at realistic priors for a given application.

We first interpret the smoothness prior in the context of a model for generating functions consistent with this prior. For this we use the concept of autoregressive models as the generation model, presented in Sect. 2.8.3, p. 52. They enable the estimation of the corresponding variance σ_e^2 from real data which are assumed to follow the generation model. This is achieved using variance component estimation as discussed in Sect. 4.2.4, p. 91. Furthermore, they can be used to generalize the priors presented up to now, which will be demonstrated using the analysis of real terrain profiles.

16.3.2.1 Modelling Surface Profiles with Autoregressive Models

Autoregressive models for sequences $\{\underline{z}_k\}$ of random variables relate the current value $\underline{z}_k$ to its P previous ones in a linear manner,

$$\underline{z}_k = \sum_{p=1}^{P} c_p \underline{z}_{k-p} + \underline{e}_k \,, \qquad \mathbb{D}(\underline{e}_k) = \sigma_e^2 \tag{16.84}$$

(cf. (2.197), p. 52), changing the notation of the coefficients to c_p. This model is useful since as it allows (1) an interpretation of the smoothness measures discussed before, and (2) a generalization to more complex smoothness terms.

First, a sequence following the weak string model from (16.35), p. 739 is shown to follow the special AR(1) model (2.200), p. 52, namely an integrated white noise process,

$$\underline{z}_k = \underline{z}_{k-1} + \underline{e}_k \,, \qquad \mathbb{D}(\underline{e}_k) = \sigma_e^2 \,, \tag{16.85}$$

since

$$\widehat{\sigma}_e^2(1) = \frac{\sum_{k=2}^{K}(z_k - z_{k-1})^2}{K-1}\,, \tag{16.86}$$

given a sequence $\{z_k\}$ and, assuming it follows (16.85), the variance of the prediction error $\underline{e}_k$. The (1) that is the argument of $\widehat{\sigma}_e^2(1)$ indicates that it refers to the AR(1) model. The variance $\widehat{\sigma}_e^2$ thus measures the flatness of the sequence $\underline{z}_k$. Hence, the AR model (16.85) is a model which generates a sequence following the weak string model.

Similarly, a sequence following the weak rod model from (16.41) follows a doubly integrated white noise process (2.218), p. 53,

$$\underline{z}_k = 2\underline{z}_{k-1} - \underline{z}_{k-2} + \underline{e}_k\,, \qquad \mathbb{D}(\underline{e}_k) = \sigma_e^2\,, \tag{16.87}$$

since the variance of the prediction error is

$$\widehat{\sigma}_e^2(2) = \frac{\sum_{k=3}^{K}(z_k - 2z_{k-1} + z_{k-2})^2}{K-2}\,, \tag{16.88}$$

which measures the squared sum of the second derivatives. Hence, the AR model (16.87) is a model which generates a sequence following the weak rod model. Hence, the variance σ_e^2 of the prediction error in a doubly integrated white noise process measures the smoothness of the function. Hence, smoothness is measured by the weight $1/\sigma_e^2$.

Usually, we do not have direct access to the process $\{z_k\}$ itself, but only to measurements $\{l_k\}$ of the process. This leads to the concept of an *observed AR-process*. It may be modelled as

observed autoregressive process

$$\underline{z}_k = \sum_{p=1}^{P} c_p \underline{z}_{k-p} + \underline{e}_k\,, \qquad \mathbb{D}(\underline{e}_k) = \sigma_e^2 \qquad k = 1, ..., K \tag{16.89}$$

$$\underline{l}_m = \underline{z}_m + \underline{n}_m\,, \qquad \mathbb{D}(\underline{n}_m) = \sigma_n^2 \qquad m \in \{1, ..., K\}\,. \tag{16.90}$$

The first equations (16.89) describe the model for the underlying unknown process, which is characterized by the P parameters c_p and the variance σ_e^2 of the driving white noise process $\underline{e}_k$. The second equations (16.90) describe the model for the observation process. Thus, the observations $\{\underline{l}_k\}$ are a noisy version of the unknown process $\{\underline{z}_k\}$, assuming the observational noise $\underline{n}_k$ to be white with variance σ_n^2.

We can simulate observed profiles based on (16.89). The generated observations can be used to reconstruct the underlying unknown profile, which then may be compared to the generated profile.

A reasonable check of the validity of the reconstruction of a discrete function as derived in the previous sections can be performed by simulation. Let us assume the model for the true sequence $\{\tilde{z}_k\}$ is a thin rod model. The check may consist of the following steps:

1. Generate a sequence of true function values $\{\tilde{z}_k\}$ using a doubly integrated white noise process,
$$\tilde{z}_k = 2\tilde{z}_{k-1} - \tilde{z}_{k-2} + e_k\,, \qquad \underline{e}_k \sim \mathcal{N}(0, \sigma_e^2)\,, \tag{16.91}$$
with starting values $\tilde{z}_1 = \tilde{z}_2 = 0$, see Fig. (2.10), p. 54, bottom row. Observe, we use a sample sequence of the AR(2)-process as the true sequence.
2. Generate observed values at selected positions,
$$l_m = \tilde{z}_m + n_m\,, \quad \underline{n}_m \sim \mathcal{N}(0, \sigma_n^2)\,. \tag{16.92}$$
3. Estimate values $\widehat{\boldsymbol{z}} = [\widehat{z}_k]$ for the complete sequence, getting $\{\widehat{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\widehat{z}\widehat{z}}\}$. Within this check we need to use the variances σ_e^2 and σ_n^2 from the simulation.
4. In the spirit of Sect. 4.6.8, p. 139 evaluate whether the difference $\widehat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}$ is biased and the covariance matrix $\boldsymbol{\Sigma}_{\widehat{z}\widehat{z}}$ actually reflects the uncertainty of the estimated sequence $\widehat{\boldsymbol{z}}$. For this we need to repeat steps 2 and 3 while fixing the positions $m \in (1, ..., K)$.

Finally, we may investigate the sensitivity of the result w.r.t. a wrong choice of the model, namely when using a wrong prior variance or even using a wrong model for the prior.

Example 16.3.52: ***Reconstructing a profile.*** We demonstrate the estimation using a simulated case. We simulated an autoregressive process AR(2), namely a doubly integrated white noise process $\{\tilde{z}_k\}$ with $\sigma_e = \sigma_n = 0.5$ having $K = 200$ samples, and enforced that the first and the last point have value 0, see the red dashed line in Fig. 16.13. We randomly selected eight positions $\{m\}$, here at $m = \{6, 12, 24, 81, 95, 124, 138, 176\}$, and generate observations $\{l_m\}$, shown as thick (blue) dots. The estimated profile $\{\widehat{z}_k\}$ is shown as (black) dotted sequence. As can be expected, the original profile cannot be recovered in large intervals between the observed points with a high precision. The standard deviation $\{\sigma_{\widehat{z}_k}\}$ of the reconstructed sequence is shown as a continuous line: The estimated profile $\{\widehat{z}_k\}$ clearly lies within the confidence interval $\{\tilde{z}_k \pm 3\sigma_{\widehat{z}_k}\}$ around the true signal $\{\tilde{z}_k\}$. ◇

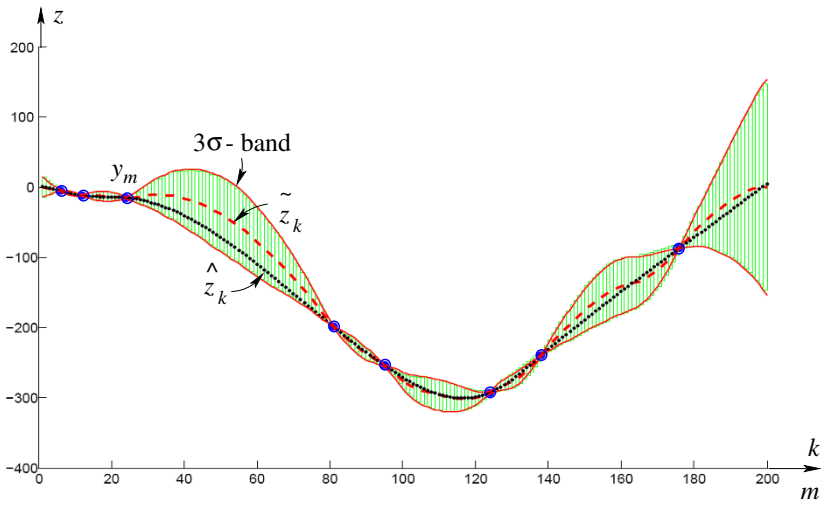


Fig. 16.13 The true values of an AR(2) process (red dashed line), observed points $\{l_m\}$ (blue dots), its reconstruction $\{\widehat{z}_k\}$ (black dotted line), and the 3σ-band around the true signal using the standard deviations $\{\sigma_{\widehat{z}_k}\}$ of the reconstructed sequence (thin continuous lines). The estimated signal does not pass through the given points, thus $\widehat{l}_m \neq l_m$, which cannot be visualized since the noise $\sigma_n = 0.5$ of the observation process is small

Autoregressive models appear to be a useful choice for modelling smooth profiles. However, we need to determine the free parameters, namely the order of the autoregressive process, the involved variances and the coefficients.

16.3.2.2 Variance Component Estimation for Observed AR Processes

In case the observational noise is small or can be neglected and the observations are dense, we can determine the variance σ_e^2 of the driving noise process $\{e_k\}$ directly from the prediction errors e_k, e.g., using (16.86), p. 749 or (16.88), p. 749. In case the observations of a sequence are not complete and their uncertainty cannot be neglected, we still are able to determine the two variances σ_e^2 and σ_n^2 by variance component estimation. Here the concept of fictitious observations shows its strength.

The complete covariance matrix Σ_{ll} of the observations can be written in the following form,

$$\Sigma_{ll} = \begin{bmatrix} \sigma_e^2 I_M & 0 \\ 0 & \sigma_n^2 I_{K-2} \end{bmatrix} = \sigma_e^2 \begin{bmatrix} I_M & 0 \\ 0 & 0 \end{bmatrix} + \sigma_n^2 \begin{bmatrix} 0 & 0 \\ 0 & I_{K-2} \end{bmatrix}, \tag{16.93}$$

which is identical to the form given in (4.91), p. 91. Therefore the iterative procedure using (4.99), p. 92 can be applied (cf. Förstner, 1985). This method can be used for arbitrary autoregressive processes.

The method has been analysed for doubly integrated white noise processes in Förstner (1985) w.r.t. its ability to determine and separate the two variances if only a single densely observed profile is available. This analysis is based on the covariance matrix

$$\mathbb{D}\left(\begin{bmatrix}\widehat{\underline{\sigma}_e^2}\\ \widehat{\underline{\sigma}_n^2}\end{bmatrix}\right) = \begin{bmatrix}\mathbb{D}(\widehat{\underline{\sigma}_e^2}) & \mathbb{C}\mathrm{ov}(\widehat{\underline{\sigma}_e^2}, \widehat{\underline{\sigma}_n^2})\\ \mathbb{C}\mathrm{ov}(\widehat{\underline{\sigma}_n^2}, \widehat{\underline{\sigma}_e^2}) & \mathbb{D}(\widehat{\underline{\sigma}_n^2})\end{bmatrix} \tag{16.94}$$

of the estimated variance components, which for long densely observed signals only depends on the ratio σ_n^2/σ_e^2.[5] The analysis yields the following results:

- The estimates $\widehat{\sigma}_e^2$ and $\widehat{\sigma}_n^2$ for the two variances are always separable, as the correlation coefficient of the two estimates always is smaller than $1/\sqrt{2} \approx 71\%$. The procedure thus will never interpret noise as signal or vice versa (see Förstner, 1985, Fig. 2).
- However, the variances are not always well-determinable, i.e., their estimates may have too low variances. If the ratio σ_n^2/σ_e^2 is far from 1, then only the larger variance can be estimated reliably.
- Especially, the variance σ_n^2 of the observation process can only be determined if it is not much smaller than the variance σ_e^2 of the driving process, which corresponds to intuition.
- Even for strongly contaminated signals, the variance σ_e^2 of the driving process can be estimated, though with lower precision.

For example, if the variances are estimated from a completely observed signal of length $K = 100$ and the unknown two noise variances are approximately equal, the standard deviation of the estimated noise variance $\widehat{\sigma_n^2}$ is approximately 28% of the variance, whereas the standard deviation of the variance of the driving process $\widehat{\sigma_e^2}$ is approximately 21% of the variance, see Fig. 2 in Förstner (1985). This demonstrates that both variances can be determined with moderate precision, and the signal should be large enough, say $K > 100$, to ensure reliable estimates. The analysis refers to densely observed signals, which is not a severe restriction.

16.3.2.3 Usefulness of Higher-Order Processes

The autoregressive model discussed so far is a first approximation, which is often adequate for the task (cf. Lemaire, 2008). Real surfaces will show deviations from a doubly integrated white noise process, which is why it may be useful to derive models which better reflect the stochastic properties of real profiles and which may be used for interpolation or reconstruction of other profiles. If we assume surface profiles to be representable by autoregressive processes, we can identify the model for the AR process, i.e., determine its order P and parameters c_p and σ_e^2.

Such an analysis has been performed by Lindenberger (1993). He used a set of densely sampled terrain profiles in eight different regions. The terrain profiles were captured from aerial images with a different scale in each project. The different image scales led to profiles with different point densities. In each region, he analysed profiles with a total of more than 1000 points. His goal was to explore the variety of models for real profiles belonging to different terrain types. We summarize the relevant results in order to demonstrate the necessity but also the possibility of using more refined models for real surface profiles.

Lindenberger (1993) found that a useful model is an integrated autoregessive process ARI(P, D), which is an autoregressive process of order P operating on the Dth derivatives, cf. Sect. 2.8.4, p. 54. Specifically, he found that it is fully sufficient to use the second derivatives, i.e., he assumed an autoregressive model ARI(P,2) for the sequence of second

[5] The covariance matrix is related to the inverse of a normal equation matrix $\mathbf{S} = [s_{ij}]; i, j \in \{e, n\}$. For example, we have $\mathbb{V}(\widehat{\sigma}_e^2) = 2(\mathbf{S}^{-1})_{ee}\widehat{\sigma}_e^4$, and $s_{ee}(\sigma_n^2/\sigma_e^2) = K\int_{-1/2}^{+1/2} 1/(\sigma_n^2/\sigma_e^2 \cdot 16\sin^4 \pi u + 1)\mathrm{d}u$, which can be determined numerically (cf. Förstner, 1985).

derivatives,

$$\underline{z}_{k-1} - 2\underline{z}_k + \underline{z}_{k+1} = \sum_{p=1}^{P} c_p \underline{z}_{k-p} + \underline{e}_k \,. \tag{16.95}$$

He identified the order P of the ARI(P,2) model, its parameters c_p and the variance σ_e^2 of the prediction error for profiles of each terrain type.

The results of his analysis are collected in the Table 16.1. Besides some properties of the different cases, it contains the estimated standard deviation of the height measurements σ_n in meters (column (6)), related to the flying height H_g over ground (column (7)), in order to compare it to the rule of thumb, which estimates the height accuracy as 0.1‰ of the flying height H_g over ground. Column (8) contains the estimated prediction error $\widehat{\sigma}_e$ when assuming the doubly integrated white noise process, thus the weak string model. Columns (9) and (10) report the estimated order P, cf. Sect. 4.10, p. 184, of the integrated autoregressive model ARI(P,2) and the resulting prediction errors.

Table 16.1 Comparison of models for terrain profiles, adapted from Lindenberger (1993). The regions 7 and 8 in Denmark (DK) are in Greenland. The image scales in the regions 5 and 6 are marked by '?' by the original author

	region	point distance [m]	image scale number	# of profiles	# of points per profile	standard dev. mensuration: $\widehat{\sigma}_n$ [m]	‰ of height H_g	standard deviation of prediction error: $\widehat{\sigma}_e$ after ARI model ARI(0,2) [m]	P	ARI(P,2) [m]	ratio (8)/(10)
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
1	Hannover D	1.0	4 000	4	215–250	0.03	0.05	0.002	4	0.001	2.0
2	Söhnstetten DK	2.5	10 000	3	210–350	0.10	0.06	0.031	5	0.009	3.4
3	Uppland S	2.5	30 000	4	208–363	0.15	0.03	0.039	5	0.007	5.6
4	Bohuslaen S	2.5	3 000	5	240–396	0.22	0.05	0.075	5	0.022	3.4
5	Drivdalen	5.0	17 000?	5	130–318	0.48	0.18	0.277	3	0.103	2.7
6	Oslo N	15.0	15 000?	7	500	0.44	0.05	0.656	1	0.259	2.5
7	Disko Island, DK	30.0	56 000	8	300	0.40	0.05	2.132	3	0.227	9.4
8	Washington, DK	50.0	56 000	6	400	0.34	0.04	2.432	2	0.315	7.7

We observe the following:

- The standard deviations (column (6) and (7)) of the measuring processes related to flying height over ground H_g are all bellow 0.1‰of H_g except in the Drivdalen region (row 5).
- For the doubly integrated white noise process, the prediction errors (column (8)) vary between 0.002 m and 2.4 m.
- When using low-order autoregressive models ($P < 10$) for the second derivatives, the prediction errors go down to a maximum of 0.3 m. The optimal order of the processes varies between 1 and 5. In most cases, the gain in standard deviation of the prediction error is in the order of 2.0 to 3.5. However, gains of factors 5.6 up to 9.4 can be achieved with low-order ($P = 2$ to 5) autoregressive processes for the second derivatives.

Summarizing, the models of flat or smooth functions using the measures F_1 (16.35), p. 739 or S_1^a (16.41), p. 740 (which are the variances of the first or second derivatives) are useful approximations if pre-knowledge about properties of the surface is weak. For natural surfaces, it appears useful to analyse representative surface profiles to arrive at more realistic prior models. The gain in prediction accuracy may be high.

16.3.3 Changing the Grid Spacing

Up to now we have always assumed the grid spacing is 1. We now want to discuss the model for the case where the grid spacing is some arbitrary value h. We only discuss the one-dimensional case.

With the linear basis function $f_0(x) = \Lambda(x)$, the unknown profile is

$$z^{(h)}(x) = \sum_{k=1}^{K} a_k^{(h)} f_0(x - kh) \tag{16.96}$$

and the observed values are $l_m = z(x_m) + n_m, m = 1, ..., M$, with arbitrary x_m and noise n_m. The superscript (h) indicates that the values refer to a sampling distance h. We now use the discrete approximation of the second derivative of $z(x)$,

$$z''^{(h)}(x) \approx \frac{a_{k-1}^{(h)} - 2a_k^{(h)} + a_{k+1}^{(h)}}{h^2} = \frac{a_k''^{(h)}}{h^2}\,, \tag{16.97}$$

using the second differences $a_k''^{(h)} = a_{k-1}^{(h)} - 2a_k^{(h)} + a_{k+1}^{(h)}$ of the coefficients $a_k^{(h)}$. Furthermore, we assume that the range of the function to be reconstructed is $[a, b]$. Then we will have

$$K = \left\lfloor \frac{b-a}{h} \right\rfloor \tag{16.98}$$

grid points. The continuous optimization function

$$\Omega_2 = \sum_{m=1}^{M} \left(\frac{l_m - z(x_m)}{\sigma_n} \right)^2 + \frac{1}{b-a} \int_{x=a}^{b} \left(\frac{z''(x)}{\sigma_{z''}} \right)^2 \mathrm{d}x \tag{16.99}$$

refers to second derivatives $z''(x)$, together with their standard deviation $\sigma_{z''}$, which we will relate to σ_e. Observe, the value Ω_2 is dimensionless. When sampling the second derivatives at $K - 2$ grid points, the optimization function Ω_2 therefore can be approximated by

$$\Omega_2^{(h)} = \sum_{m=1}^{M} \left(\frac{l_m - z(x_m)}{\sigma_n} \right)^2 + \frac{1}{b-a} \sum_{k=2}^{K-1} h \left(\frac{a_{k-1}^{(h)} - 2a_k^{(h)} + a_{k+1}^{(h)}}{h^2 \sigma_{z''}} \right)^2 \tag{16.100}$$

neglecting the boundary terms. The factor h in the regularization term is necessary, as the grid spacing is not 1. Thus we finally arrive at

$$\Omega_2^{(h)} = \sum_{m=1}^{M} \left(\frac{l_m - z(x_m)}{\sigma_n} \right)^2 + \frac{1}{b-a} \sum_{k=2}^{K-1} \left(\frac{a_{k-1}^{(h)} - 2a_k^{(h)} + a_{k+1}^{(h)}}{h^{3/2} \sigma_{z''}} \right)^2 . \tag{16.101}$$

When referring to the second differences $a_k''^{(h)}$ with their standard deviations $\sigma_e^{(h)} := \sigma_{a''}^{(h)}$, we also can write the optimization function as

$$\Omega_2^{(h)} = \sum_{m=1}^{M} \left(\frac{l_m - z(x_m)}{\sigma_n} \right)^2 + \frac{1}{b-a} \sum_{k=2}^{K-1} \left(\frac{a_k''^{(h)}}{\sigma_e^{(h)}} \right)^2 , \tag{16.102}$$

with

$$\sigma_e^{(h)} = h^{3/2} \sigma_{z''}\,. \tag{16.103}$$

For example, changing the grid spacing from h to $4h$ would require a standard deviation $\sigma_{a''}^{(4h)} = 8\sigma_{a''}^{(h)}$ for the second differences, which is larger by a factor of $8 = 4^{3/2}$, in order to obtain the same result for the optimization at the required grid positions. Apart from the

interpolation, the profiles differ due to the approximation of the second derivatives using the second differences, see Fig. 16.14.

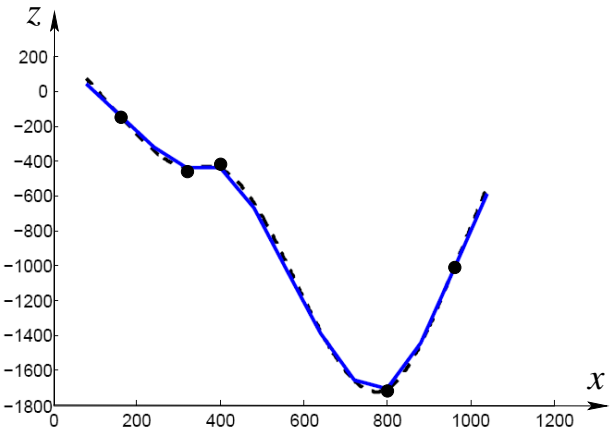


Fig. 16.14 Reconstruction with two different grid spacings h. The given points (black dots) are at integer coordinates $[160, 240, 320, 800, 960]$. **Dashed:** $h = 1$, $\sigma_e^{(1)}/\sigma_n = 1/180 \approx 0.0083$. **Solid:** $h = 80$, $\sigma_e^{(80)}/\sigma_n = 80^{3/2}\,\sigma_e^{(1)}/\sigma_n \approx 5.96$. In spite of the large difference of the grid spacings h used for the interpolation, the resulting profiles are very similar: The differences at the grid points 80 to 1040 have mean 59.4 and standard deviation 18.1, which is small when compared with the total amplitude of approximately 1800 and the lack of given points at both ends and between 400 and 800.

16.3.4 Violations of the Basic Model

The basic model assumes that the observed points and the regularizing observations, cf. Sect. 16.3.1.3, p. 746, have random deviations with zero mean and a certain standard deviation.

There are two basic types of model errors:

- The ratio σ_e/σ_n of the standard deviations for the observations deviates from its true value. The effect on the reconstruction remains small (1) for deviations below 30%, cf. Sect. 4.6.6, p. 135, and (2) if the sampled points do not lie very close to each other, as then their residuals will be small, independent of the chosen standard deviations. A change in the ratio σ_e/σ_n will have an effect only if the sampling is very dense, see the examples in Fig. (16.3), p. 731 in rows 3 and 4.
- The observed points may contain outliers specific to the measurement process. The regularizing observations for the smoothness of the surface may be erroneous at surface steps or break lines, though they show local regularity, as they do not appear in isolation but are arranged along lines. They will have a direct effect on the reconstruction and should be identified and eliminated if possible.

Therefore we only discuss the identification of outliers in the real and the regularizing observations. Both types of outliers can be handled by robustifying the estimation process. We first discuss outliers in the observed points.

16.3.4.1 Outlier Detection

Outliers ∇l_m in the observed heights may be of two types:

1. They may be two-sided random deviations: $\nabla l_m \in [-R, +R]$ with R delimiting the range of the outliers. Such outliers may be caused by wrong correspondences between two images.
2. They may be one-sided random deviations: $\nabla l_m \in [0, +R]$. Such one-sided outliers may occur when observing the terrain in a wooded area: Points above the surface, say in vegetation, are also measured by the laser range finder. When handling them as outliers their distribution is one-sided.

For symmetric outliers, we apply the results developed in Sect. 4.7, p. 141. Using the method of modified weights, we may start with a reweighting function (4.381), p. 148. With the normalized residuals $y = \widehat{v}/\sigma_{\widehat{v}}$ or $y = \widehat{v}/\sigma_l$, we have

$$w_{L_{12}}(y) = \min(1, 1/|y|)\,; \tag{16.104}$$

see Fig. 16.16, dotted curve.

Example 16.3.53: ***Symmetric outlier model.*** An example for the reconstruction of a noisy profile with outliers is shown in Fig. 16.15. The method cannot find clusters of outliers and becomes weak if the observation density is low, as then the redundancy numbers r_n become small and outliers do not show in the residuals. This is caused by the average relative redundancy of $\overline{r_n} = R/N \approx 1/2$. We can compensate for this to some extent by (1) using the normalized residuals $l_n = v_n/\sigma_{v_n}$ as argument of the weight function, instead of v_n/σ_n, and (2) by enforcing the profile to be smoother, i.e., up-weighting the regularization term or, equivalently, decreasing σ_e/σ_n in the estimation procedure. ◇

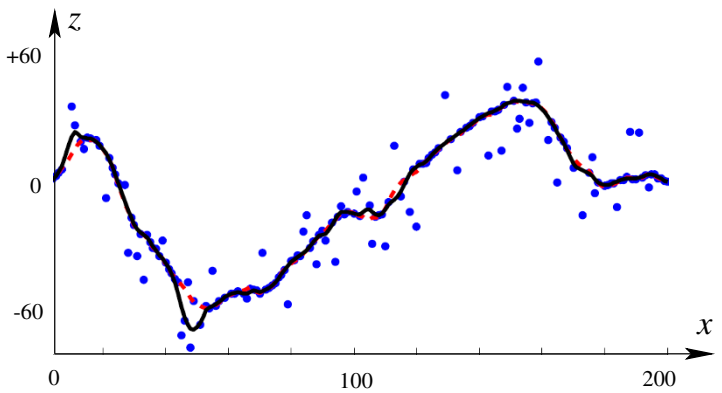


Fig. 16.15 Reconstruction of a profile with symmetric outliers. True profile: $\sigma_e = \sigma_n = 0.5$, dashed red, visible only at a few positions, e.g., around $k = 47$. Number of grid points: 200. Number of observations: 160. Outlier percentage: 40%. Outlier range: $[-25, 25]$. Reconstructed with L_{12}-norm, six iterations using σ_e/σ_n from the simulation of the true profile

For one-sided outliers, Kraus and Pfeifer (1998) propose using a one-sided weight function. The normalized residual $y = (z(x) - l)/\sigma$ is assumed to be negative if the observed point is above the surface. Kraus and Pfeifer (1998) argue that points below the ground, thus with positive residual, obtain weight 1. In a first approximation this would lead to the asymmetric weight function L'_{12}, see Fig. 16.16, blue curve. The result of an example is shown in Fig. 16.17, top: The profile has been densely sampled but with 70% of the data with outliers in the range $[0, 25]$. Obviously, the one-sided outliers have been eliminated. However, the estimated profile is slightly above the true surface on an average. This is because the transition part of the weight function between small values and 1 is at -1, and also because low negative residuals are not completely weighted down. Therefore

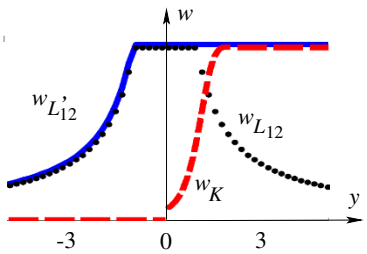


Fig. 16.16 Robust weight functions $w_{L_{12}}$ for symmetric outliers and $w_{L'_{12}}$ and w_K for asymmetric outliers, $g = 2$, $w = 2$, $a = 1$, $b = 4$ in Eq. (16.105)

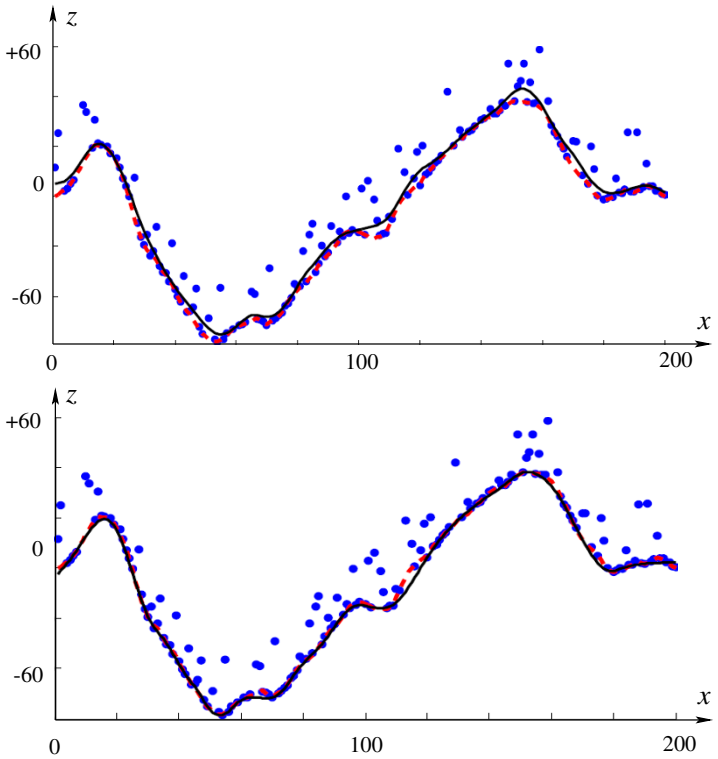


Fig. 16.17 Reconstruction of a profile with one-sided outliers. True profile: $\sigma_e = \sigma_n = 0.5$, red dashed. All grid points are observed. Outlier percentage: 70%. Outlier range: $[0, 25]$. **Upper figure:** Reconstruction with one-sided L'_{12}-weight function. **Lower figure:** Reconstruction with the weight function of Kraus and Pfeifer (1998), $g = 2$, $w = 2$, cf. (16.105), p. 756

Kraus and Pfeifer (1998) propose a weighing function which is zero for residuals below some threshold and where the transition is shifted towards positive residuals:[6]

$$w_K(v) = \begin{cases} 0 & \text{if } v < g - w \\ 1 & \text{if } v \geq g \\ \dfrac{1}{1 + a(v-g)^b} & \text{else.} \end{cases} \tag{16.105}$$

They suggest using $a = 1$ and $b = 4$ and give a scheme to determine the parameters g and w from the histogram of the residuals in each iteration. The weight function is shown in Fig. 16.16 as the dashed function, for $g = 2$ and $w = 2$. The effect of their proposal can be seen in Fig. 16.17, bottom. Due to the large percentage of outliers the prior for the

[6] The sign of the residuals $v = f(x) - l$ here is different from that used by Kraus and Pfeifer (1998). Therefore, the sign of g also is different to theirs, usually positive.

curvature has been up-weighted by a factor of 256, corresponding to reducing σ_e/σ_n by a factor of 8 compared to the simulation of the true profile. Obviously, the reconstructed profile (solid line) is significantly closer to the true profile (dashed line) than for the simple one-sided L_{12}-norm, cf Fig. 16.17, top.

16.3.4.2 Steps and Break Lines

Up to now, we have assumed that the profiles and surfaces are regular in the whole domain of interest. Violations of this model may be caused by steps or break lines of the surface or by occlusions. They can be detected and taken into account by robust estimation, now referring to the regularizing observations for the curvature, cf. Sect. 16.3.1.1, p. 743.

Ideally, at a break point, one or two regularizing observations should get weight zero, whereas at steps 2 or 3, neighbouring regularizing observations should be reweighted.

Robust estimation by reweighting does not take into account the assumption that the surface is piecewise regular, i.e., violations of the regularity occur infrequently. As neighbouring regularizing observations always refer to some common function values, their residuals will be similar. Therefore, not only one, but a larger set of regularizing observations neighbouring the break point or the step will obtain a low weight.

In order to reduce this effect we can proceed as follows:

- We start the iteration process, indexed with ν, with a weak regularization, i.e., with $\sigma_e^{(\nu=1)}/\sigma_n = 1$, and change the ratio during the iteration process as in a geometric series to arrive at the prespecified ratio σ_e/σ_n in the last iteration. This decreases the effect of break points or steps on the down-weighting of the neighbouring regularizing observations.
- As a general recommendation we use the weight function $w_{L_{12}}(y)$ in the first three iterations and the exponential reweighting in the next three iterations. However, in order to avoid weights which are numerically zero, we limit the weight factors to be larger than a small value, say 10^{-4}. The last iteration then uses weight factors of 1 for the inliers and a small value for the outliers of the regularizing observations.
- Finally, for a large number of unknowns which occur in two-dimensional reconstruction tasks, we do not necessarily have access to the standard deviations of the residuals. We thus cannot use the normalized residuals $\widehat{v}_n/\sigma_{\widehat{v}_n}$ with individual standard deviations $\sigma_{\widehat{v}_n}$. We instead partially normalize the residuals using $\widehat{v}_n/\sigma_{\widehat{v}}$, using a robust estimate of their average standard deviation $\sigma_{\widehat{v}} = 1.48\,\mathrm{med}(|\widehat{v}_n|)$, which at the same time eliminates the effect of large residuals at the break points or steps.

An example is shown in Fig. 16.18. The step and the break point are quite pronounced in this example. The minimum step height and the minimum curvature to be detected as outliers can be derived exploiting the concepts in Sect. 4.6.4.1, p. 125. They will depend on the standard deviations σ_n and σ_e and the density of the points in the neighbourhood of the step or break point. Due to the high noise level with $\sigma_n = 0.9$, the step and the break point in the example are identifiable by this method if their values are smaller by factors of 3 and 2, respectively, when compared to the situation in the figure.

16.4 Reconstruction of 2½D Surfaces from 3D Point Clouds

16.4.1 Surface Reconstruction with a General Continuous Prior 758
16.4.2 Smooth Surface Reconstruction 759
16.4.3 The Optimization 760
16.4.4 Theoretical Precision of Reconstructed Surfaces 763

We now generalize the previous method of signal reconstruction to two dimensions (cf. Ebner, 1979). We allow the given points to be at positions with noninteger coordinates;

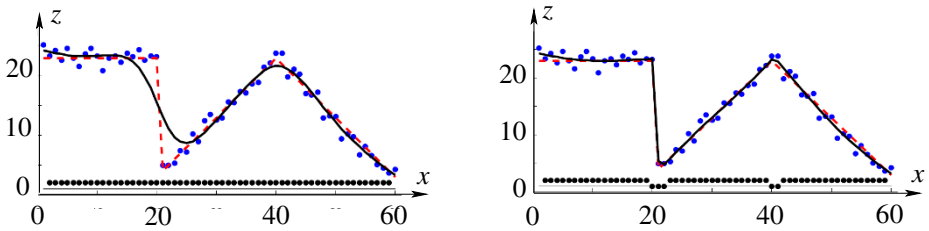


Fig. 16.18 Robust estimation for tackling steps and break points. The true profile (dashed red) is sampled (blue dots) and reconstructed (solid black) using the weighted regularizing observations (weights are 0 or 1, shown at the bottom as a sequence of dots). Based on the residuals of the regularizing observations, the weights of these observations are adapted within an ML-type robust estimation, and finally assigned the value 1 or 0. **Left:** Without robust estimation. **Right:** With robust estimation. Iterations 1 to 3 use the L_{12}-norm, iterations 4 to 6 use the exponential weight function. The prespecified standard deviations are $\sigma_e = 0.15$ and $\sigma_n = 0.9$

this allows us to interpolate at real valued positions $x, y \in \mathbb{R}$. We discuss the solution specializing to (1) bilinear interpolation and (2) a regular grid with unit grid width. We apply the method to surface reconstruction.

16.4.1 Surface Reconstruction with a General Continuous Prior

We start from observed M points, $[x, y, z]_m$. Let us assume the surface is represented as a function $z = z(x, y)$. We assume that we have observed the surface at a set of fixed, not necessarily grid positions $(x_m, y_m), m = 1, ..., M$,

$$l_m = z(x_m, y_m) + n_m \quad \text{with} \quad \mathbb{D}(l_m) = \sigma_{l_m}^2 \,. \tag{16.106}$$

We want to reconstruct the surface in a region $\mathcal{R}$. The roughness or wiggliness of the surface is assumed to be measured by the sum of the squared partial derivatives up to some order k. Then we arrive at the following optimization function,

$$\Omega_k = \sum_{m=1}^{M} \frac{(l_m - z(x_m, y_m))^2}{\sigma_{l_m}^2} + \int_{(x,y)\in\mathcal{R}} \frac{G_k^2(x, y)}{\sigma_{G_k}^2} \mathrm{d}x\mathrm{d}y \,, \tag{16.107}$$

with the regularization function (cf. Wood, 2003)

$$G_k^2(x, y) = \sum_{i+j=k} \frac{k!}{i!j!} \left(\frac{\partial z^k(x, y)}{\partial x^i \partial x^j} \right)^2 \,. \tag{16.108}$$

The regularization function $G_k^2(x, y)$ can be viewed as a generalization of the first- and second-order regularization terms used up to now, since for $k = 1$ and $k = 2$ it specializes to the squared gradient $F_2(x, y) = G_1^2(x, y) = z_x^2 + z_y^2$ and the quadratic variation $S_2(x, y) = G_2^2(x, y) = z_{xx} + 2z_{xy}^2 + z_{yy}^2$.

The order k of the model and the variance $\sigma_{G_k}^2$ need to be determined from real data, similarly to how we determined the order of the autoregressive processes and the prediction error of autoregressive models for profile reconstruction. The model for the smoothness is isotropic, i.e., rotating the coordinate system does not change the regularization term. Anisotropy may be achieved by introducing a covariance matrix for the vector of partial derivatives, which need to be estimated from real data. How to optimally estimate these parameters of the anisotropy is an open question.

16.4.2 Smooth Surface Reconstruction

We now specialize the model in two ways: First, we assume the surface is represented by a quadratic grid with unit grid size and bilinear interpolation. Second, we assume the surface smoothness can be described by the weight of the second derivatives measured at the grid points.

Thus the surface is represented as a function $z = z(x, y)$ based on a grid $a_{ij}, i \in \{i_{\min}, \dots, i_{\max}\}, j \in \{j_{\min}, \dots, j_{\max}\}$, with $I \times J$ cells, where $I = i_{\max} - i_{\min} + 1$ and $J = j_{\max} - j_{\min} + 1$. The continuous surface is defined by bilinear interpolation, cf. (16.25), p. 737,

$$z(x, y) = \sum_{ij} a_{ij} \Lambda_{ij}(x, y) \quad \text{with} \quad \Lambda_{ij} = \Lambda_i(x)\Lambda_j(y) = \Lambda(x - i)\Lambda(y - j)\,. \tag{16.109}$$

Then the corresponding observation equations follow from the linear interpolation (16.27), p. 738:

$$\begin{aligned} z_m + v_{z_m} = (1-s)(1-t) &\cdot a(\lfloor x \rfloor, \lfloor y \rfloor) \\ +(1-s)t &\cdot a(\lfloor x \rfloor, \lceil y \rceil) \\ +s(1-t) &\cdot a(\lceil x \rceil, \lfloor y \rfloor) \\ +st &\cdot a(\lceil x \rceil, \lceil y \rceil)\,, \end{aligned} \tag{16.110}$$

with

$$s = x - \lfloor x \rfloor\,, \qquad t = y - \lfloor y \rfloor\,. \tag{16.111}$$

Thus each observed z-value linearly depends on the four parameters a_{ij} of its cell corners.

For the surface to be smooth we require its second differences to be small. These are

1. the second differences d_{ii} in the x-direction,
2. the second differences d_{jj} in the y-direction, and
3. the mixed differences d_{ij} in the x- and y-directions.

We will treat these second differences independently: This allows us to individually weight the regularizing observations, but also mutual weighting in a reasonable manner.

In a first step, we use the simplest way to determine these derivatives and obtain the following regularizing observations. The observations for the second differences in the x-direction are

$$\delta_{ii} + v_{\delta_{ii}} = a(i-1, j) - 2a(i, j) + a(i+1, j)\,, \tag{16.112}$$

$$\mathbb{D}(\underline{\delta}_{ii}) = \sigma^2_{\delta_{ii}}\,, \quad i \in \{i_{\min} + 1, \dots, i_{\max} - 1\}, j \in \{j_{\min}, \dots, j_{\max}\}\,. \tag{16.113}$$

These are $(I - 2) \times J$ observations.

Similarly, we have the $I \times (J - 2)$ regularizing observations for the second differences in the y-direction:

$$\delta_{jj} + v_{\delta_{jj}} = a(i, j-1) - 2a(i, j) + a(i, j+1)\,, \tag{16.114}$$

$$\mathbb{D}(\underline{\delta}_{jj}) = \sigma^2_{\delta_{jj}}\,, \quad i \in \{i_{\min}, \dots, i_{\max}\}, j \in \{j_{\min} + 1, \dots, j_{\max} - 1\}\,. \tag{16.115}$$

The $(I - 1) \times (J - 1)$ regularizing observations for the mixed differences are

$$\delta_{ij} + v_{\delta_{ij}} = a(i-1, j-1) - a(i-1, j) - a(i, j-1) + a(i, j)\,, \tag{16.116}$$

$$\mathbb{D}(\underline{\delta}_{ij}) = \sigma^2_{\delta_{ij}}\,, \quad i \in \{i_{\min} + 1, \dots, i_{\max}\}, j \in \{j_{\min} + 1, \dots, j_{\max}\}\,. \tag{16.117}$$

The fictitious observations are

$$\delta_{ii} = \delta_{jj} = \delta_{ij} = 0 \tag{16.118}$$

in order to obtain a smooth surface. Observe, the number of observations for the regularizing second derivatives is not the same for δ_{ii}, δ_{jj}, and δ_{ij}.

16.4.3 The Optimization

The optimization function then reads as

$$\Omega(\boldsymbol{a}) = \sum_m \left(\frac{l_m - z(x_m, y_m; \boldsymbol{a})}{\sigma_{l_m}} \right)^2 + \sum_{ij} \left(\frac{\delta_{ii}(\boldsymbol{a})}{\sigma_{\delta_{ii}}} \right)^2 + \sum_{ij} \left(\frac{\delta_{ij}(\boldsymbol{a})}{\sigma_{\delta_{ij}}} \right)^2 + \sum_{ij} \left(\frac{\delta_{jj}(\boldsymbol{a})}{\sigma_{\delta_{jj}}} \right)^2 . \tag{16.119}$$

This function is quadratic in the unknown parameters $\boldsymbol{a}$. The resulting linear equation system for determining $\boldsymbol{a}$ leads to the global optimum of $\Omega(\boldsymbol{a})$ without iteration.

For a specific choice of the variances σ_{ii}, σ_{ij}, and σ_{jj}, this optimization is equivalent to the use of the quadratic variation for regularization, cf. (16.46), p. 741. So we need to choose the variances for the second derivatives as

$$\sigma_{\delta_{ii}}^2 = \sigma_{\delta_{ij}}^2 / 2 = \sigma_{\delta_{jj}}^2 = \sigma_\delta^2 . \tag{16.120}$$

If we neglect boundary effects, caused by having no regularizing observations at all border points, and assume $\sigma_{l_m} = \sigma_n$, the optimization function reads as

$$\Omega = \frac{1}{\sigma_n^2} \sum_m (l_m - a_m)^2 + \frac{1}{\sigma_\delta^2} \sum_{ij} \delta_{ii}^2 + 2\delta_{ij}^2 + \delta_{jj}^2 . \tag{16.121}$$

The regularization expression,

$$Q = \delta_{ii}^2 + 2\delta_{ij}^2 + \delta_{jj}^2 = \operatorname{tr}(\boldsymbol{H}^2) , \tag{16.122}$$

is the discrete version of the quadratic variation, cf. (16.46).

Example 16.4.54: *2½D Reconstruction.* The following small example shows the sparse structure of the design matrix $\boldsymbol{A}$ when using (16.110) to (16.117). Let the six points in Table 16.2 be given. They are assumed to determine the grid in the bounding box between $[0, 0]$ and $[6, 4]$, with a quadratic grid with grid size $\Delta x = 2$, having $I \times J = 3 \times 4 = 12$ grid points, see Fig. 16.19.

Then we obtain the following design matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} = \left[\begin{array}{cccccccccccc}
0.30 & 0.20 & & & 0.30 & 0.20 & & & & & & \\
0.24 & 0.36 & & & 0.16 & 0.24 & & & & & & \\
 & 0.04 & 0.16 & & & 0.16 & 0.64 & & & & & \\
 & & & & 0.12 & 0.28 & & & 0.18 & 0.42 & & \\
 & & & & & & 0.09 & 0.81 & & & 0.01 & 0.09 \\
 & & & & & & 0.20 & & & & 0.80 & \\
\hline
1 & -2 & 1 & & & & & & & & & \\
 & 1 & -2 & 1 & & & & & & & & \\
 & & & & 1 & -2 & 1 & & & & & \\
 & & & & & 1 & -2 & 1 & & & & \\
 & & & & & & & & 1 & -2 & 1 & \\
 & & & & & & & & & 1 & -2 & 1 \\
1 & & & & -2 & & & 1 & & & & \\
 & 1 & & & & -2 & & & 1 & & & \\
 & & 1 & & & & -2 & & & 1 & & \\
 & & & 1 & & & & -2 & & & 1 & \\
1 & -1 & & & -1 & 1 & & & & & & \\
 & 1 & -1 & & & -1 & 1 & & & & & \\
 & & 1 & -1 & & & -1 & 1 & & & & \\
 & & & & 1 & -1 & & & -1 & 1 & & \\
 & & & & & 1 & -1 & & & -1 & 1 & \\
 & & & & & & 1 & -1 & & & -1 & 1
\end{array}\right] .$$

The upper part A_1 provides the coefficients for the bilinear interpolation. Each row generally contains four nonzero entries. The lower part A_2 provides the coefficients for the regularization. It has a well-defined regular structure. The rank of the 16×12 matrix A_2 is $\mathrm{rk}(A_2) = 9$. The rank defect of 3 reflects the fact that we need at least three points to determine the surface. If we just had three observed points, we would obtain a planar surface through the given points.

Table 16.2 Coordinates of given points of the example

m	x	y	z
1	0.8	1.0	2.0
2	1.2	0.8	3.0
3	3.6	1.6	4.0
4	1.4	3.2	5.0
5	5.8	2.2	6.0
6	4.0	3.6	1.0

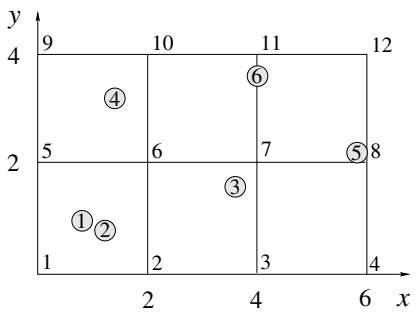


Fig. 16.19 Example for surface reconstruction. Six points (circled) are observed for the reconstruction of a quadratic 3×4-grid with spacing $\Delta x = 2$. The z-values of the grid points are the 12 unknown parameters $a_k = a_{ij}$

We chose $\sigma_n = 0.05$ and $\sigma_\delta = 0.2$. The fitted surface is shown in Fig. 16.20. The quality of the reconstructed surface can be evaluated using the standard deviations at the grid points, which lie in the range $[0.29, 3.7]$, with the largest value at the corner [6,4].

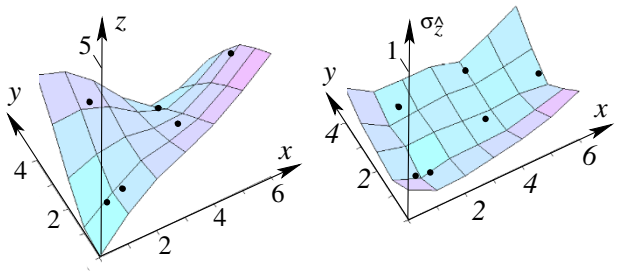


Fig. 16.20 Surface reconstruction from six points and standard deviations $\sigma_n = 0.05$ and $\sigma_\delta = 0.2$ for noise and regularizing observations. **Left:** Fitted surface **Right:** Standard deviations $\sigma_{\hat{z}}$ of the estimated grid points. The maximum standard deviation of the reconstructed points is 0.8 at the upper left corner

◇

The theoretical quality of the surface can easily be determined using the covariance matrix of the estimated parameters. The theoretical precision is given by the standard deviations $\sigma_{\hat{x}}$. Figure 16.21 shows the theoretical precision $\sigma_{\hat{z}}$ of the reconstructed surface points (left) and the sensitivity factors μ of the given points (right), assuming $\sigma_n = 1$ and $\sigma_\delta = 5$. As expected, the accuracy deteriorates with larger distances between the given points. Around the given points the standard deviations are around $\sigma_n = 1$. The

maximum standard deviation of a reconstructed point is approximately 27, the maximum sensitivity factor is approximately 83. These high values result from the assumption that the surface is very rough ($\sigma_\delta = 5$). If the reconstruction would assume a smoother surface with $\sigma_\delta = 1$, the uncertainty pattern would not change, but the standard deviations of the reconstructed points and the sensitivity factors of the given points would decrease approximately by a factor of 5; the maxima then are approximately $\sigma_{\hat{z}} = 5.5$ and $\mu = 17$.

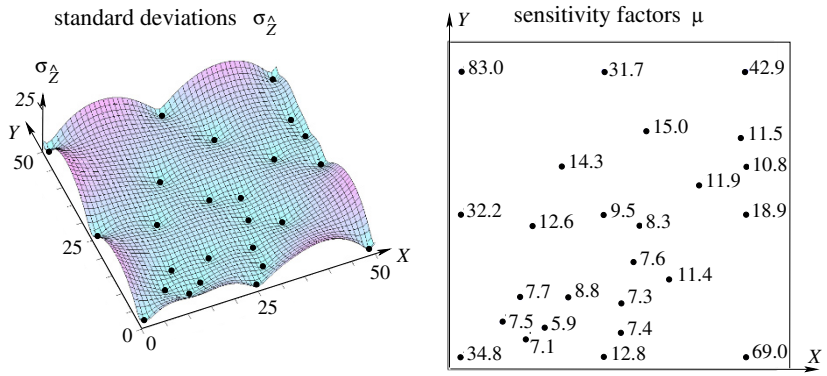


Fig. 16.21 Theoretical quality of surface reconstruction derived from $M = 25$ irregularly spaced points. The given points have a standard deviation of $\sigma_n = 1$. The fictitious curvature observations have a standard deviation of $\sigma_\delta = 5$. **Left:** Theoretical precision $\sigma_{\hat{z}}$ of interpolated points. The precision deteriorates with larger distances from the given points. They range from $\sigma_n = 1$ around the given points to a maximum of approximately 27. **Right:** The sensitivity factors $\mu_m, m = 1, ..., M$, of the given points. They range from 5.9 in the lower left area, where the point density is large, to 83.0 in the upper left corner, where the distance to the nearest given point is large. For example, in case the height z of the given point in the upper left corner were be corrupted by an outlier which is just not detectable by a statistical test, the effect of this outlier on the reconstructed points of the surface would be up to 83 times the standard deviation of these points

Finally, steps and break lines can be taken into account by robust estimation, following the same scheme as in the one-dimensional case, cf. Sect. 16.3.4.2, p. 757. The example in Fig. 16.22 starts from a densely observed surface. The step heights are 10, the noise of the observations is $\sigma_z = 1$. Reconstructing the surface, i.e., filtering the data assuming the surface to be smooth, leads to a surface where all break lines and steps are smeared out. Robust estimation preserves the steps and break lines. The peak at the right corner of the surface is interpreted as a violation of the smoothness and therefore not smoothed away. This indicates that simultaneously recovering break lines and steps together with outliers in the observations will remain challenging. For a more detailed analysis of the method and its ability to recover discontinuities, cf. Weidner (1994).

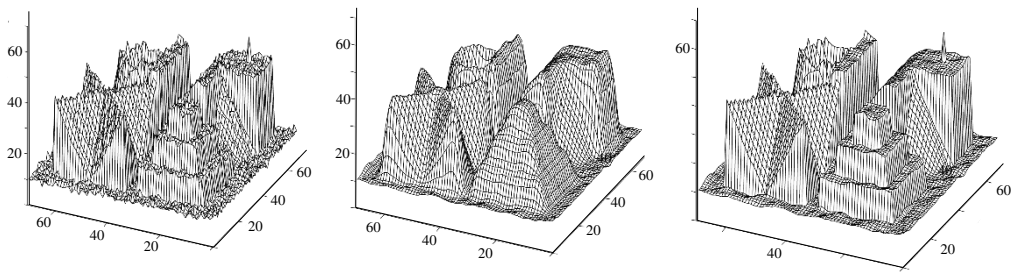


Fig. 16.22 Finding steps and break lines. **Left:** Densely observed surface, $\sigma_z = 1$, step heights 10. **Middle:** Filtered surface, assuming smoothness. **Right:** Filtered surface, assuming piecewise smoothness

16.4.4 Theoretical Precision of Reconstructed Surfaces

As in the case of the analysis of the theoretical precision of bundle adjustments, we can provide some ideas on the theoretical precision of reconstructed surfaces. We assume the surface properties are homogeneous. The precision generally depends on

- the distribution of the observed points,
- the assumed precision of the observed points,
- the assumed smoothness of the surface, and
- the validity of the previous assumptions.

At the observed positions $\boldsymbol{x}_k$, the fitted surface heights $\widehat{z}_k$ obviously are at least as precise as the observed heights. Interpolation only leads to better values if the surface is very smooth, as then the precision of the neighbouring points contribute to the height determination. In all other cases, the interpolated heights will be less accurate.

For profiles where we regularize with the second derivatives, the standard deviation between observed positions in a first approximation increases with the distance d between the points.

$$\sigma_{\widehat{z}}(d) \propto d^{3/2} \,. \tag{16.123}$$

This corresponds to the rule found for the theoretical precision of strips of images in Sect. 15.3.5, p. 670, (15.103), p. 670. The reason simply is that in both cases the second derivatives of the estimated quantities can be approximated by white noise.

For example, the standard deviations of the interpolated points in the profile in Fig. (16.13), p. 750 in the middle of observed points are given in Table 16.3.

d	$d^{3/2}$	$\sigma_{\widehat{z}}$	$\sigma_{\widehat{z}}/d^{3/2}$
6	14	1.0	0.0716
12	41	2.3	0.0564
57	430	18.1	0.0421
14	52	3.2	0.0613
33	189	7.4	0.0472
14	52	3.2	0.0608
38	234	12.9	0.0550

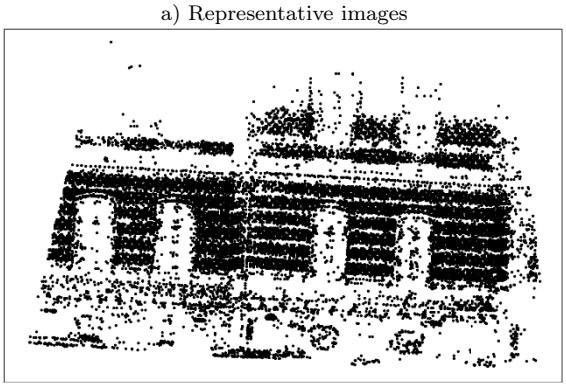
Table 16.3 Precision of interpolated points in profiles. The table shows the standard deviations $\sigma_{\widehat{z}}$ at the mid points of the intervals of length d in the profile of Fig. (16.13), p. 750. The ratio $\sigma_{\widehat{z}}/d^{3/2}$, except for the very short interval at the beginning, lies in a small range, $[0.047, 0.061]$

For surfaces, the standard deviation of interpolated points roughly is proportional to the average distance between the points, as it is comparable with the theoretical precision of the height of the bundle blocks, cf. (15.226), p. 720.

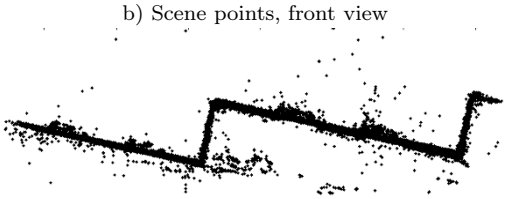
16.5 Examples for Surface Reconstruction

The following example demonstrates the use of the scene points derived from a bundle adjustment. We refer to the example in Sect. 15.4.1.3, p. 679. Figure 16.23, top, shows five representative images of a facade. The total number of 63 104 scene points was reconstructed from 70 images. They were detected using Lowe's key point detector and matched. We chose a subset of approximately 23 000 points referring to the right part of the facade. For the reconstruction, we refer to an XY-coordinate system parallel to the two major principal axes of the point cloud, such that the Z-coordinate is approximately vertical to the facade, see Fig. 16.23, centre. The robust reconstruction of the facade minimizes

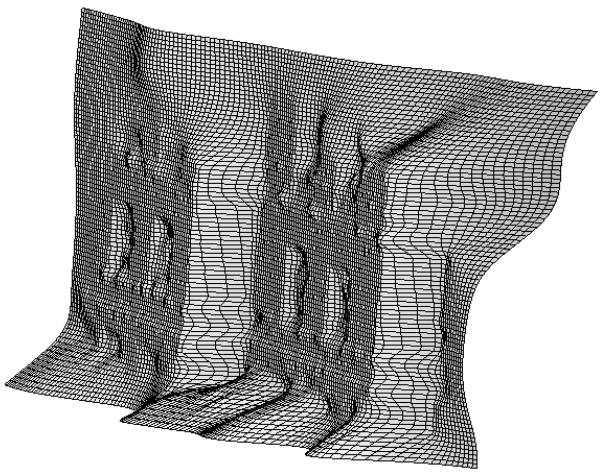
a) Representative images



b) Scene points, front view



c) Scene points, top view



d) Reconstructed surface

Fig. 16.23 Reconstruction of the surface of a facade using the scene points from a bundle adjustment with 70 images. **a)** Representative images of the facade: $[24, 18, 13, 7, 3]$ covering the part to be reconstructed. **b)** The scene points from the right side of the complete facade, 23 203 points. View onto the XY-plane and **c)** onto the XZ-plane. **d)** Reconstructed surface seen from the side. Grid size: 137×87. Assumed ratio: $\sigma_\delta/\sigma_n = 2$, cf. (16.121), p. 760. Observe, the outliers visible in c), especially those behind the facade, are eliminated by the robust surface reconstruction procedure

$$\Omega = \sum_m \rho\left(\frac{l_m - a_m}{\sigma_n}\right) + \frac{1}{\sigma_\delta^2}\sum_{ij} \delta_{ii}^2 + 2\delta_{ij}^2 + \delta_{jj}^2\,; \tag{16.124}$$

cf. (16.121), p. 760. The first three iterations again use the reweighting with $w_{L_{12}}$, the next three iterations use w_{exp} and the last iteration accepts observations with $w > 0.5$, as described in Sect. 4.7.4.2, p. 150.

The result is shown in Fig. 16.23, bottom. It indicates that already from the point cloud derived from a bundle adjustment we can reconstruct the surface to a high fidelity, which in this case is supported by the texture of the facade, which allows us to obtain a comparably large number of scene points. However, the precision of the surface can be significantly improved, mainly by increasing the density of the scene points but also by using a more accurate matching procedure, which can be the basis for structuring or interpreting the surface, topics which we do not address here.

The last example, see Fig. 16.24, shows the potential of surface reconstruction from aerial images (cf. Haala et al., 2010). Observe:

- The DSM is derived from two images using the software package MATCH-T by INPHO/Trimble (cf. Lemaire, 2008). The method for deriving the surface from a set of 3D points is the one described in this section, applying reweighting to both, the 3D points and the priors on the curvatures.
- The DSM has a grid distance of 25 cm. This allows us to the reconstruct the surface of buildings and of trees. The steps in the vineyards are clearly visible.
- The accuracy was evaluated by (1) comparing the surface using a reference surface derived from a laser scan and (2) analysing a horizontal area (sports field). The achieved root mean square errors lie in the range of 4-5 cm.

Fig. 16.24 Surface reconstruction from aerial images. **Left:** One of two aerial images of a region with buildings and vineyards in the area of Vaihingen/Enz, Germany, taken from the DGPF-Test data set. Camera: Zeiss DMC, ground sampling distance (pixel size at the ground): 8 cm. Flying height above ground approximately 1200 m. The image actually shows the aerial image rectified to the map (orthophoto) having a pixel spacing of 25 cm. **Right:** Reconstructed DSM. The software Match-T from INPHO/Trimble is used. The grid distance of the DSM is 25 cm. Reported accuracy in flat areas: 4 cm (cf. Haala et al., 2010)

16.6 Exercises

1. (2) Given are five points $(x, y)_i, 1 = 1, ..., 5$. They are assumed to lie on a parabola $y = f(x) = a + bx + cx^2$. Only the y-coordinates are assumed to be uncertain with standard deviation $\sigma_{y_i} = \sigma$. How accurately can you obtain the value $y = f(x)$, the slope $\alpha(x) = \arctan(f'(x))$, and the curvature $\kappa(x) = f''(x)/(1 + f'^2(x))^{3/2}$ at $x = 1$? Give a configuration for the x_i such that the standard deviation $\sigma_\kappa < 0.1$ if $a = b = c = 1$ and $\sigma = 1$.
2. (1) Show that the matrix M in the linear function $z(x) = [1, x]M[z(0), z(1)]^\mathsf{T}$ which passes through the points $[0, z(0)]$ and $[1, z(1)]$ is given by (16.29), p. 738.

3. (1) Show that the interpolating surface (16.30), p. 738 can be realized by first linearly interpolating in the x-direction and then, using the interpolated points, linearly interpolating in the y-direction.
4. (2) Show that the matrix M in the cubic function

$$z(x) = [1, x, x^2, x^3]\ M\ [z(-1), z(0), z(1), z(2)]^\mathsf{T},$$

which passes through the points $[0, z(0)]$ and $[1, z(1)]$ and has slopes $z'(0) = (z(1) - z(-1))/2$ and $z'(1) = (z(2) - z(1))/2$, is given by (16.31), p. 738. Show that for a function $z(i) = \delta(i)$, bicubic interpolation leads to the basis function (16.34), p. 738.
5. (2) Show that the interpolating surface (16.33), p. 738 can be realized by first cubicly interpolating in the x-direction and then, using the interpolated points, cubicly interpolating in the y-direction.
6. (2) This exercise demonstrates the precision of inter- and extrapolation when quadratically interpolating three neighbouring points of a profile. Refer to Sect. 16.2.1.1, p. 733 and the following:

   a. The standard deviation of an interpolated point $z(x)$, given three values $z(-1)$, $z(0)$, and $z(1)$ with standard deviation σ_z, is $\sigma_z(x) = 1/2 \cdot \sqrt{4 - 6\,x^2 + 6\,x^4}\ \sigma_z$.
   b. For interpolated values in the interval $x \in [-1, 1]$, we have $\sigma_z(x) \leq \sigma_z$, with a minimum of $\sigma_z(\sqrt{1/2}) = \sqrt{10}/4\ \sigma_z \approx 0.8\ \sigma_z$. Plot the function $\sigma_z(x)$ for $\sigma_z = 1$. Analyse the behaviour of $\sigma_z(x)$ outside the interval $[-1, +1]$.
   c. For equal weights, the basis functions $f_0(x) = \sqrt{1/3}$, $f_1(x) = \sqrt{1/2}\,x$, and $f_2(x) = \sqrt{3/2}\,(x^2 - 2/3)$ are orthonormal on the grid $\{-1, 0, 1\}$. Show that you obtain the same result for $\sigma_z(x)$ as when not using orthonormalized polynomials.

Appendix: Basics and Useful Relations from Linear Algebra

A.1 Inner Product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_A = \boldsymbol{x}^\mathsf{T} \mathsf{A} \boldsymbol{y}. \tag{A.1}$$

The index, written with font *Times*, indicates the matrix used in the bilinear form. In the case of homogeneous vectors we have $\langle \mathbf{x}, \mathbf{y} \rangle_\mathrm{A} = \mathbf{x}^\mathsf{T} \mathsf{A} \mathbf{y}$. We omit the index when it is clear from the context.

A.2 Determinant

A.2.1 Definition of the Determinant

The determinant of an $N \times N$ matrix is a scalar function $D = \det(\mathsf{A}) : \mathbb{R}^{N \times N} \to \mathbb{R}$ with the following properties

1. The determinant is linear in the columns (or rows) of the matrix. That is, if the nth column is $\boldsymbol{a}_n = \alpha \boldsymbol{x} + \beta \boldsymbol{y}$ for any vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and some constants α, β, then

$$|(\boldsymbol{a}_1, ..., \alpha \boldsymbol{x} + \beta \boldsymbol{y}, ..., \boldsymbol{a}_N)| = \alpha |(\boldsymbol{a}_1, ..., \boldsymbol{x}, ..., \boldsymbol{a}_N)| + \beta |(\boldsymbol{a}_1, ..., \boldsymbol{y}, ..., \boldsymbol{a}_N)| \tag{A.2}$$

2. When exchanging two rows or two columns, the sign of the determinant changes.
3. If $N = 1$, $\det([1]) = 1$.

We also write

$$\det \mathsf{A} = |\mathsf{A}| \,. \tag{A.3}$$

For $N = 2$, we have

$$\det \left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \right) = a_{11} a_{22} - a_{12} a_{21} \,. \tag{A.4}$$

A.2.2 Laplacian Development of a Determinant

The following theorem allows us to write the determinant of a matrix A as a sum of products of sub-determinants of a matrix. Let $\boldsymbol{r} = \{r_1, ..., r_K\}$ with $r_1 < ... < r_K$ be a set of row indices $r_k \in \mathcal{N}$, and $\boldsymbol{c} = \{c_1, ..., c_K\}$ with $c_1 < ... < c_K$ be a set of column indices $c_k \in \mathcal{N}$. The $K \times K$ submatrix only keeping the indices $\boldsymbol{r}$ and $\boldsymbol{c}$ is written as

$$\mathsf{S}(\mathsf{A}, \boldsymbol{r}, \boldsymbol{c}) \,. \tag{A.5}$$

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4

The complementary $(N-K)\times(N-K)$ submatrix removing these indices is

$$S'(A,\boldsymbol{r},\boldsymbol{c})\,. \tag{A.6}$$

Then we have the

Theorem A.2.9: ***Laplacian development theorem.*** Given the $N\times N$ matrix A and two lists $\boldsymbol{r}=\{r_k\}$ and $\boldsymbol{c}=\{c_k\}$ of K row and column indices with $1\le r_1<...<r_K<N$ and $1\le c_1<...<c_K<N$, the determinant can be expressed as

$$|A|=(-1)^{|\boldsymbol{c}|}\sum_{\boldsymbol{r}}(-1)^{|\boldsymbol{r}|}|S(A,\boldsymbol{r},\boldsymbol{c})|\,|S'(A,\boldsymbol{r},\boldsymbol{c})|\,, \tag{A.7}$$

where $|\boldsymbol{r}|=r_1+...+r_K$ and $|\boldsymbol{c}|=c_1+...+c_K$, and the summation is taken over all possible combinations of $\boldsymbol{c}$ with $1\le c_1<...<c_k<...<c_K<N$.

Clearly, if the properties of the determinant hold for the submatrices $S(A,\boldsymbol{r},\boldsymbol{c})$ and $S'(A,\boldsymbol{r},\boldsymbol{c})$, they also hold for the determinant of the matrix A, which allows the theorem to be proven by induction, as it holds for $N=2$.

The determinant of a quadratic submatrix is also called *minor*. Thus the Laplacian development theorem expresses the determinant of the matrix as a sum of products of minors. Two cases are of special interest.

An important example is the development of a 4×4 matrix by the first two columns. Thus we fix $\boldsymbol{c}=(1,2)$ and obtain

$$\det A=(-1)^{1+2}\sum_{\boldsymbol{r}}(-1)^{r_1+r_2}|S(A,\boldsymbol{r},\boldsymbol{c})|\,|S'(A,\boldsymbol{r},\boldsymbol{c})| \tag{A.8}$$

$$\begin{aligned}
&=+|S(A,(1,2),(1,2))|\,|S'(A,(1,2),(1,2))|\\
&\quad-|S(A,(1,3),(1,2))|\,|S'(A,(1,3),(1,2))|\\
&\quad+|S(A,(1,4),(1,2))|\,|S'(A,(1,4),(1,2))|\\
&\quad+|S(A,(2,3),(1,2))|\,|S'(A,(2,3),(1,2))|\\
&\quad-|S(A,(2,4),(1,2))|\,|S'(A,(2,4),(1,2))|\\
&\quad+|S(A,(3,4),(1,2))|\,|S'(A,(3,4),(1,2))|
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
&=+\begin{vmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{vmatrix}\begin{vmatrix}a_{33}&a_{34}\\a_{43}&a_{44}\end{vmatrix}\\
&\quad-\begin{vmatrix}a_{11}&a_{12}\\a_{31}&a_{32}\end{vmatrix}\begin{vmatrix}a_{23}&a_{24}\\a_{43}&a_{44}\end{vmatrix}\\
&\quad+\begin{vmatrix}a_{11}&a_{12}\\a_{41}&a_{42}\end{vmatrix}\begin{vmatrix}a_{23}&a_{24}\\a_{33}&a_{34}\end{vmatrix}\\
&\quad+\begin{vmatrix}a_{21}&a_{22}\\a_{31}&a_{32}\end{vmatrix}\begin{vmatrix}a_{13}&a_{14}\\a_{43}&a_{44}\end{vmatrix}\\
&\quad-\begin{vmatrix}a_{21}&a_{22}\\a_{41}&a_{42}\end{vmatrix}\begin{vmatrix}a_{13}&a_{14}\\a_{33}&a_{34}\end{vmatrix}\\
&\quad+\begin{vmatrix}a_{31}&a_{32}\\a_{41}&a_{42}\end{vmatrix}\begin{vmatrix}a_{13}&a_{14}\\a_{23}&a_{24}\end{vmatrix}\,.
\end{aligned} \tag{A.10}$$

As the minors referring to a set $\boldsymbol{c}$ of columns of a square matrix can be interpreted as the Plücker coordinates of the join of the points $\mathcal{X}_c$ in $\mathbb{P}^{N-1}$ in these columns, the determinant of a matrix is the sum of the products of the Plücker coordinates of the columns $\boldsymbol{c}$ and of the columns not $\boldsymbol{c}$, taking the correct signs into account.

The second application of (A.7) is the following lemma.

Lemma A.2.1: ***Development of a determinant by row.*** The determinant of an $N\times N$ matrix can be expressed as

$$|A| = \sum_{n=1}^{N} (-1)^{1+n} a_{1,n} \, |S'(A, 1, \{2, ..., n\})| \,. \tag{A.11}$$

This results from (A.7) by setting $r = 1$ and $c = 2 : n$. For example, take the determinant of a 3×3 matrix:

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} . \tag{A.12}$$

A.2.3 Determinant of a Block Matrix

The determinant of a block matrix is given by

$$\begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} = |A_{11}| \, |A_{22} - A_{21} A_{11}^{-1} A_{12}| = |A_{22}| \, |A_{11} - A_{12} A_{22}^{-1} A_{21}| \,. \tag{A.13}$$

A.3 Inverse, Adjugate, and Cofactor Matrix

The inverse A^{-1} of a regular square matrix A fulfils $A^{-1}A = AA^{-1} = I$.

We have the Woodbury identity, with correctly related matrices A, B, C,

$$(A \pm CBC^{\mathsf{T}})^{-1} = A^{-1} - A^{-1}C(C^{\mathsf{T}}A^{-1}C \pm B^{-1})^{-1}C^{\mathsf{T}}A^{-1} \tag{A.14}$$

(see Petersen and Pedersen, 2012). We also have

$$A^{-1} + B^{-1} = A^{-1}(A + B)B^{-1} \,; \tag{A.15}$$

(see Petersen and Pedersen, 2012, (144)).

The inverse of a symmetric 2×2 block matrix is given by

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}C_2^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}C_2^{-1} \\ -C_2^{-1}A_{21}A_{11}^{-1} & C_2^{-1} \end{bmatrix} \tag{A.16}$$

$$= \begin{bmatrix} C_1^{-1} & -C_1^{-1}A_{12}A_{22}^{-1} \\ -A_{22}A_{21}C_1^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}C_1^{-1}A_{12}A_{22}^{-1} \end{bmatrix} , \tag{A.17}$$

with

$$C_1 = A_{11} - A_{12}A_{22}^{-1}A_{21} \,, \qquad C_2 = A_{22} - A_{21}A_{11}^{-1}A_{12} \,, \tag{A.18}$$

assuming at least one of the two submatrices A_{ii} to be regular.

The *cofactor matrix* A° of a square, not necessarily regular, matrix is the matrix of the determinants of its submatrices

$$A^{\circ} = [(-1)^{i+j} |A^{(ij)}|] \,, \tag{A.19}$$

where $A^{(ij)}$ is the matrix with row i and column j deleted. For a 2×2 matrix we have

$$A^{\circ} = \begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix} . \tag{A.20}$$

For a general 3×3 matrix $A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3]$ with column vectors $\boldsymbol{a}_i$, it can be shown that

$$A^{\circ} = [\boldsymbol{a}_2 \times \boldsymbol{a}_3, \boldsymbol{a}_3 \times \boldsymbol{a}_1, \boldsymbol{a}_1 \times \boldsymbol{a}_2] \,. \tag{A.21}$$

The *adjugate matrix* A^* of a square matrix, which is not necessarily regular, is the transpose of the cofactor matrix,

$$A^* = (A^\circ)^\mathsf{T} = [(-1)^{i+j}|A^{(ji)}|]\,. \tag{A.22}$$

It is closely related to the inverse by

$$A^* = |A|A^{-1}\,, \tag{A.23}$$

and thus is proportional to the inverse, if A is regular.

The determinant therefore can be written as

$$|A| = \frac{1}{N}\mathrm{tr}(A^*A) = \frac{1}{N}\mathrm{tr}((A^\circ)^\mathsf{T}A)\,, \tag{A.24}$$

where tr$A$ is the trace of the matrix A. Finally, we observe for regular $n \times n$ matrices,

$$(A^*)^* = |A|^{n-2}A \quad \text{and} \quad (A^\circ)^\circ = |A|^{n-2}A\,, \tag{A.25}$$

due to $(A^*)^* = (|A|{\cdot}A^{-1})^* = |A|^{n-1}{\cdot}|A|^{-1}A = |A|^{n-2}A$.

A.4 Skew Symmetric Matrices

Skew matrices play a central role when representing rotations. An $N \times N$ skew symmetric matrix S has properties:

$$S = -S^\mathsf{T}\,, \tag{A.26}$$

$$\mathrm{tr}S = 0\,. \tag{A.27}$$

A.4.1 2×2 *Skew Matrix*

For a scalar x, we obtain the 2×2 skew-symmetric matrix

$$S_x = S(x) = \begin{bmatrix} 0 & -x \\ x & 0 \end{bmatrix} \tag{A.28}$$

with the following properties:

- It is regular with determinant
$$\det(S(x))^2 = x^2 \tag{A.29}$$
and eigenvalues
$$\lambda_1 = ix \quad \lambda_2 = -ix \qquad \text{with} \qquad i = \sqrt{-1}\,. \tag{A.30}$$
- Its square, its cube, and its fourth power are
$$S^2(x) = -x^2 I_2\,, \qquad S^3(x) = -x^3 S(x)\,, \qquad S^4(x) = x^4 I_2\,. \tag{A.31}$$
- If $x = 1$, then $S(1)$ rotates a 2-vector
$$\begin{bmatrix} -b \\ a \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = R_{90^\circ} \begin{bmatrix} a \\ b \end{bmatrix} \tag{A.32}$$
by 90° anti-clockwise.

- We have the rotation matrix

$$R(x) = \exp(S_x) = \cos(x)\ I_2 + \sin(x)\ S(1) = \begin{bmatrix} \cos x & -\sin x \\ \sin x & \cos x \end{bmatrix} \tag{A.33}$$

using the matrix exponential, see Sect. (A.13), p. 781, which can be proven by using the definition of the matrix exponential and collecting the odd and even terms.

A.4.2 3×3 *Skew Matrix*

For a 3-vector $\boldsymbol{x} = [x, y, z]^\mathsf{T}$, the 3×3 skew symmetric matrix is defined as

$$S_x = S(\boldsymbol{x}) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} . \tag{A.34}$$

The matrix $S(\boldsymbol{x})$ has the following properties:

- The product with a 3-vector is identical to the anti-symmetric cross product of two vectors:

$$S(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y} = -\boldsymbol{y} \times \boldsymbol{x} = -S(\boldsymbol{y})\boldsymbol{x} . \tag{A.35}$$

  Therefore, often $S(\boldsymbol{x})$ is denoted by $[\boldsymbol{x}]_\times$, leading to the intuitive relation $\boldsymbol{x} \times \boldsymbol{y} = [\boldsymbol{x}]_\times \boldsymbol{y}$. We do not follow this notation since the vector product does not immediately generalize to higher dimensions.
- Its right null space is $\boldsymbol{x}$ as $\boldsymbol{x} \times \boldsymbol{x} = \boldsymbol{0}$.
- If $\boldsymbol{x} \neq 0$, the matrix has rank 2. Its eigenvalues are

$$\lambda_1 = i|\boldsymbol{x}| \quad \lambda_1 = -i|\boldsymbol{x}| \quad \lambda_3 = 0 . \tag{A.36}$$

- The matrix $S(\boldsymbol{x})$ and its square $S^2(\boldsymbol{x})$ are related to the dyad

$$D_x = \boldsymbol{x}\boldsymbol{x}^\mathsf{T} \qquad \text{with} \qquad \mathrm{tr} D_x = |\boldsymbol{x}|^2 \tag{A.37}$$

  by

$$S_x\, D_x \quad = \quad 0 \tag{A.38}$$

  and

$$S_x^2 = \boldsymbol{x}\boldsymbol{x}^\mathsf{T} - |\boldsymbol{x}|^2 I_3 \qquad \text{with} \qquad \mathrm{tr}(S_x^2) = -2|\boldsymbol{x}|^2 . \tag{A.39}$$

- The third and the fourth powers are

$$S_x^3 = -|\boldsymbol{x}|^2 S_x \qquad \text{and} \qquad S(\boldsymbol{x})^4 = |\boldsymbol{x}|^4 (I_3 - \frac{\boldsymbol{x}\boldsymbol{x}^\mathsf{T}}{|\boldsymbol{x}|^2}) . \tag{A.40}$$

- Therefore we have the relation, for any 3×3 skew matrix,

$$S_x S_x^\mathsf{T} S_x = \frac{1}{2} \mathrm{tr}(S_x S_x^\mathsf{T}) S_x . \tag{A.41}$$

- The following relations hold for unit vectors $\boldsymbol{r}$ with $|\boldsymbol{r}| = 1$:

$$D_r^2 = D_r \tag{A.42}$$
$$S_r^2 = -(I_3 - D_r) \tag{A.43}$$
$$S_r^3 = -S_r \tag{A.44}$$
$$S_r^4 = I_3 - D_r . \tag{A.45}$$

The following relations between a skew-symmetric matrix and a regular matrix are useful.

Lemma A.4.2: ***Product of skew symmetric matrix with regular matrix.*** For each regular matrix M and all vectors $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^3$, we have

$$M\boldsymbol{x} \times M\boldsymbol{y} = M^{\circ}(\boldsymbol{x} \times \boldsymbol{y}) \tag{A.46}$$

and

$$S(M\boldsymbol{x})M = M^{\circ}S(\boldsymbol{x})\,, \tag{A.47}$$

with the cofactor matrix

$$M^{\circ} = |M|\; M^{-\mathsf{T}}\,. \tag{A.48}$$

For a rotation matrix, due to $R^{\circ} = R$, we thus have

$$R\boldsymbol{x} \times R\boldsymbol{y} = R(\boldsymbol{x} \times \boldsymbol{y}) \qquad \text{and} \qquad S(R\boldsymbol{x})R = R^{\mathsf{T}}S(\boldsymbol{x})\,, \tag{A.49}$$

and also

$$S(\boldsymbol{x})R = RS(R^{\mathsf{T}}\boldsymbol{x})\,. \tag{A.50}$$

Proof: For proving (A.46) we start from

$$< \boldsymbol{x} \times \boldsymbol{y}, \boldsymbol{z} >= (\boldsymbol{x} \times \boldsymbol{y})^{\mathsf{T}}\boldsymbol{z} = |\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}|\,. \tag{A.51}$$

Thus we have, for arbitrary $\boldsymbol{z}$,

$$\begin{aligned}
(\boldsymbol{x} \times \boldsymbol{y})^{\mathsf{T}}\boldsymbol{z} &= < M\boldsymbol{x} \times M\boldsymbol{y}, \boldsymbol{z} > && \text{(A.52)}\\
&= |M\boldsymbol{x}, M\boldsymbol{y}, MM^{-1}\boldsymbol{z}| && \text{(A.53)}\\
&= |M|\;|\boldsymbol{x}, \boldsymbol{y}, M^{-1}\boldsymbol{z}| && \text{(A.54)}\\
&= |M|\;< \boldsymbol{x} \times \boldsymbol{y}, M^{-1}\boldsymbol{z} > && \text{(A.55)}\\
&= |M|\;(\boldsymbol{x} \times \boldsymbol{y})M^{-1}\boldsymbol{z} && \text{(A.56)}\\
&= |M|\;< M^{-\mathsf{T}}(\boldsymbol{x} \times \boldsymbol{y}), \boldsymbol{z} > && \text{(A.57)}\\
&= < |M|\; M^{-\mathsf{T}}(\boldsymbol{x} \times \boldsymbol{y}), \boldsymbol{z} > && \text{(A.58)}\\
&= < M^{\circ}(\boldsymbol{x} \times \boldsymbol{y}), \boldsymbol{z} > && \text{(A.59)}\\
&= (M^{\circ}(\boldsymbol{x} \times \boldsymbol{y}))^{\mathsf{T}}\boldsymbol{z}\,. && \text{(A.60)}
\end{aligned}$$

◇

A.5 Eigenvalues

The characteristic polynomial of an $N \times N$ matrix A is given by

$$P(\lambda) = |A - \lambda I_N|\,. \tag{A.61}$$

Its zeros are the eigenvalues. The eigenvectors result from the homogeneous equation system

$$(A - \lambda I_N)\boldsymbol{x} = \boldsymbol{0}\,. \tag{A.62}$$

A symmetric matrix A can be decomposed into the following product

$$A = X\boldsymbol{\Lambda}X^{\mathsf{T}} = \sum_{n=1}^{N} \lambda_n \boldsymbol{x}_n \boldsymbol{x}_n^{\mathsf{T}} \tag{A.63}$$

where the orthogonal matrix $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n, ..., \boldsymbol{x}_N]$ contains the eigenvectors $\boldsymbol{x}_n$ as columns and the real eigenvalues are collected in the diagonal matrix $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, ..., \lambda_N)$. The eigenvectors are unique if all eigenvalues are distinct. The eigenvalue decomposition requires $2/3\,N^3 + PN^2$ for determining the eigenvalues and P eigenvectors, thus $5/3\,N^3$ if

all eigenvectors are to be determined (see Bathe and Wilson, 1973, Table 1). It is realized in MATLAB as $[X, \Lambda]) = \texttt{eig}(A)$.

A.5.1 Eigenvalues of Matrix Products

The nonzero eigenvalues of the product AB of an $m \times n$ matrix A and an $n \times m$ matrix B are invariant to the sequence of the product:

$$\lambda_i(AB) = \lambda_i(BA)\,, \qquad i = 1, \ldots, \min(m, n)\,. \tag{A.64}$$

Proof: We have from the determinant of the special matrix,

$$\begin{vmatrix} \lambda I_m & A \\ B & \lambda I_n \end{vmatrix} = |\lambda I_m|\,|\lambda I_n - \lambda^{-1} BA| = |\lambda I_n|\,|\lambda I_m - \lambda^{-1} AB|\,, \tag{A.65}$$

which leads to

$$\lambda^{m-n}|\lambda^2 I_n - BA| = \lambda^{n-m}|\lambda^2 I_m - AB|\,, \tag{A.66}$$

or, with $\mu = \lambda^2$,

$$\mu^{m-n}|\mu I_n - BA| = |\mu I_m - AB|\,. \tag{A.67}$$

The characteristic equations $c(\mu) = 0$ for AB and BA differ by a factor μ^{m-n}; thus, the first $\min(n, m)$ eigenvalues of the two matrix products are the same. Bathia (2002) gives seven different proofs, assuming A and B having the same size, each revealing a different aspect. ◇

A.5.2 Eigenvalues of Sub-blocks of a Matrix and Its Inverse

Given a symmetric positive definite 2×2 block matrix

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xs} \\ \Sigma_{sx} & \Sigma_{ss} \end{bmatrix} \tag{A.68}$$

and its inverse, $W = \Sigma^{-1}$,

$$W = \begin{bmatrix} W_{xx} & W_{xs} \\ W_{sx} & W_{ss} \end{bmatrix}\,. \tag{A.69}$$

The two block matrices can be interpreted as the covariance matrix and the normal equation matrix for the U unknowns $\boldsymbol{x}$ and the P additional parameters $\boldsymbol{s}$, in all cases omitting the hats on the estimated parameters. The block diagonals of these two matrices are closely related.

Theorem A.5.10: ***Relative difference of covariance and weight matrices.*** The largest $\min(P, U)$ eigenvalues of the two matrices

$$\underset{U\times U}{K} = (\Sigma_{xx} - W_{xx}^{-1})W_{xx}\,, \qquad \underset{P\times P}{L} = (W_{ss} - \Sigma_{ss}^{-1})\Sigma_{ss} \tag{A.70}$$

coincide, i.e., assuming the eigenvalues sorted in decreasing order, we have

$$\lambda_i\,(K) = \lambda_i\,(L) \quad i = 1, \ldots, \min(U, P)\,. \tag{A.71}$$

This theorem can be exploited to determine the eigenvalues of K by calculating the eigenvalues of L, which is numerically less complex if $U \gg P$.

Proof: Using

$$W_{xx} = D_1^2\,, \quad W_{ss} = D_2^2\,, \quad \Sigma_{xx} = X_1^2\,, \quad \Sigma_{ss} = X_2^2\,, \quad W_{xs} = B\,, \tag{A.72}$$

let us simplify the notation and indicate the diagonal blocks of the inverse, see (A.18), p. 769,

$$W = \begin{bmatrix} D_1^2 & B \\ B^\mathsf{T} & D_2^2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} X_1^2 & . \\ . & X_2^2 \end{bmatrix} = \begin{bmatrix} (D_1^2 - BD_2^{-2}B^\mathsf{T})^{-1} & . \\ . & (D_2^2 - B^\mathsf{T}D_1^{-2}B)^{-1} \end{bmatrix} . \tag{A.73}$$

Thus we need to compare the eigenvalues of the two matrices

$$K = (X_1^2 - D_1^{-2})D_2^2 := (\Sigma_{xx} - W_{xx}^{-1})W_{xx} \tag{A.74}$$

and

$$L = (D_2^2 - X_2^{-2})X_2^2 := (W_{ss} - \Sigma_{ss}^{-1})\Sigma_{ss} . \tag{A.75}$$

We use the relation for the $U \times P$ matrix B from the Woodbury identity (A.14), p. 769,

$$(I_U - BB^\mathsf{T})^{-1} = I_U + B(I_P - B^\mathsf{T}B)^{-1}B^\mathsf{T} , \tag{A.76}$$

and the eigenvalue relation from (A.64), p. 773,

$$\lambda_i(UV) = \lambda_i(VU) . \tag{A.77}$$

Let us now express both matrices as a function of the elements of W using the $\cong$ sign to indicate the largest eigenvalues are identical:

$$\begin{aligned}
\underset{U\times U}{K} &= (X_1^2 - D_1^{-2})D_1^2 && \text{(A.78)}\\
&\overset{(A.77)}{\cong} D_1(X_1^2 - D_1^{-2})D_1 && \text{(A.79)}\\
&\overset{X_1^2=(D_1^2-BD_2^{-2}B^\mathsf{T})^{-1}}{=} D_1((D_1^2 - BD_2^{-2}B^\mathsf{T})^{-1} - D_1^{-2})D_1 && \text{(A.80)}\\
&= (I_U - D_1^{-1}BD_2^{-2}B^\mathsf{T}D_1^{-1})^{-1} - I_U && \text{(A.81)}\\
&\overset{\overline{B}=D_1^{-1}BD_2^{-1}}{=} (I_U - \overline{B}\,\overline{B}^\mathsf{T})^{-1} - I_U && \text{(A.82)}\\
&\overset{(A.76)}{=} \underbrace{\overline{B}(I_P - \overline{B}^\mathsf{T}\overline{B})^{-1}\overline{B}^\mathsf{T}}_{U\times U} . && \text{(A.83)}
\end{aligned}$$

Similarly, we obtain

$$\begin{aligned}
\underset{P\times P}{L} &= (D_2^2 - X_2^{-2})X_2^2 && \text{(A.84)}\\
&\overset{X_2^{-2}=(D_2^2-B^\mathsf{T}D_1^{-2}B)}{=} (D_2^2 - (D_2^2 - B^\mathsf{T}D_1^{-2}B))(D_2^2 - B^\mathsf{T}D_1^{-2}B)^{-1} && \text{(A.85)}\\
&= B^\mathsf{T}D_1^{-2}B(D_2^2 - B^\mathsf{T}D_1^{-2}B)^{-1} && \text{(A.86)}\\
&\overset{(A.77)}{\cong} \underbrace{D_1^{-1}B(D_2^2 - B^\mathsf{T}D_1^{-2}B)^{-1}B^\mathsf{T}D_1^{-1}}_{U\times U} && \text{(A.87)}\\
&= D_1^{-1}BD_2^{-1}(I_P - D_2^{-1}B^\mathsf{T}D_1^{-2}BD_2^{-1})^{-1}D_2^{-1}B^\mathsf{T}D_1^{-1} && \text{(A.88)}\\
&\overset{\overline{B}=D_1^{-1}BD_2^{-1}}{=} \underbrace{\overline{B}(I_P - \overline{B}^\mathsf{T}\overline{B})^{-1}\overline{B}^\mathsf{T}}_{U\times U} . && \text{(A.89)}
\end{aligned}$$

Therefore the largest eigenvalues of K and L are identical. ◇

A.6 Idempotent Matrices

A matrix P is called idempotent in case $P^2 = P$. We have the following properties:

- The eigenvalues λ_n of an $N \times N$ idempotent matrix are

$$\lambda_i \in \{0, 1\} , \tag{A.90}$$

  which can easily be proven using the eigenvalue decomposition.
- For an idempotent matrix P, we have

$$\mathrm{tr}P = \mathrm{rk}P , \tag{A.91}$$

where rk$P$ is the rank of the matrix P.
- If P is idempotent, also $I - P$ is idempotent.
- In case the $N \times U$ matrix A and the $N \times N$ matrix W have full rank, the matrices

$$P = A(A^\mathsf{T} W A)^{-1} A^\mathsf{T} W \qquad \text{and} \qquad Q = I - P \tag{A.92}$$

are idempotent with ranks U and $N - U$.

A.7 Kronecker Product, vec(·) Operator, vech(·) Operator

The Kronecker product and the vec(•)-operator are important for deriving trilinear relations between geometric entities and their transformations. For symmetric matrices, it is also useful to know the vech(•) operator. The Kronecker product collects all products of the elements of two matrices in one matrix.

Definition A.7.28: *Kronecker product.* Let $A = (a_{ij})$ be an $m \times n$ matrix and $B = (b_{ij})$ be a $p \times q$ matrix. Then the Kronecker product $A \otimes B$ of A and B yields the $mp \times nq$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \cdots & \ddots & \cdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} . \tag{A.93}$$

◇

The vec(•)-operator transforms a matrix into a vector by stacking its column vectors.

Definition A.7.29: *vec*(·) *operator.* Let $A = (a_{ij})$ be an $m \times n$ matrix, then vec(A) is a $mn \times 1$ vector:

$$\text{vec}A := (a_{11}, \cdots, a_{m1}, a_{12}, \cdots, a_{mn})^\mathsf{T} .$$

◇

Especially, we have for two $m \times n$ matrices A and B,

$$\text{tr}(A^\mathsf{T} B) = \text{vec}^\mathsf{T} A \, \text{vec}B = \sum_{ij} a_{ij} b_{ij} , \tag{A.94}$$

which is the sum of the products of corresponding elements of both matrices.

The vec(•)-operator and the Kronecker product are intimately related. The basic relation refers to vectorizing the product of three matrices (see Koch, 1999, Eq. (1.147)). Given are three matrices, the $m \times n$ matrix A, the $n \times p$ matrix B and the $p \times s$ matrix C. Then we have

$$\text{vec}(ABC) = (C^\mathsf{T} \otimes A)\text{vec}B . \tag{A.95}$$

From this expression we can find a set of useful relations by assuming the matrices to have a special forms. For example, if either A or C is a vector, due to vec $\boldsymbol{x} =$ vec $\boldsymbol{x}^\mathsf{T}$, we obtain the relation

$$\text{vec}(\boldsymbol{a}^\mathsf{T} B C) = (C^\mathsf{T} \otimes \boldsymbol{a}^\mathsf{T})\text{vec}B = (\boldsymbol{a}^\mathsf{T} \otimes C^\mathsf{T})\text{vec}B^\mathsf{T} . \tag{A.96}$$

The vech(•)-operator assumes a symmetric matrix and stacks the columns of the lower left triangular matrix into a vector.

Definition A.7.30: *vech*(·)*-operator.* Let $A = (a_{ij})$ be a symmetric $n \times n$ matrix, then vech$A$ is a $\binom{n}{2} \times 1$ vector:

$$\text{vech}A := (a_{11}, \cdots, a_{n1}, a_{22}, \cdots, a_{n2}, a_{33}, \cdots, a_{nn})^\mathsf{T} .$$

◇

A.8 Hadamard Product

The Hadamard product is the elementwise multiplication of two matrices having the same size, $M \times N$:

$$A \odot B = [a_{ij} b_{ij}] \,. \tag{A.97}$$

It forms an Abelian i.e., commutative group, with the unit element $\mathit{1} = [1_{MN}]$.

We need the following result: Given are diagonal matrices $U = \mathrm{Diag}(\boldsymbol{u})$ and $V = \mathrm{Diag}(\boldsymbol{v})$ and square matrices A and B, all of the same size; then,

$$\mathrm{tr}(UAVB) = \boldsymbol{u}^{\mathsf{T}}(B^{\mathsf{T}} \odot A)\boldsymbol{v} \,. \tag{A.98}$$

Proof: With $P = UAVB$ or

$$p_{im} = \sum_{jkl} u_{ij} a_{jk} v_{kl} b_{lm} \,, \tag{A.99}$$

we have

$$\mathrm{tr}P = \sum_i p_{ii} = \sum_{ijkl} u_{ij} a_{jk} v_{kl} b_{li} = \sum_{ik} u_{ii} a_{ik} v_{kk} b_{ki} \tag{A.100}$$

$$= \sum_{ik} u_i a_{ik} v_k b_{ki} = \sum_{ik} u_i b_{ki} a_{ik} v_k = \boldsymbol{u}^{\mathsf{T}}(B^{\mathsf{T}} \odot A)\boldsymbol{v} \,. \tag{A.101}$$

◇

A.9 Cholesky and QR Decomposition

Cholesky Decomposition. The Cholesky decomposition of a symmetric positive definite $N \times N$ matrix A is unique and given by

$$A = C^{\mathsf{T}} C \,, \tag{A.102}$$

where C is an upper triangular matrix, with $c_{ij} = 0, i > j$, and positive diagonal elements. The Cholesky matrix $C = C^{-\mathsf{T}} A$ can be determined efficiently with approximately $N^3/3$ operations (see Golub and van Loan, 1996). It can be used to solve the linear equation system $A\boldsymbol{x} = \boldsymbol{b}$ by first calculating C, then solving $C^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$, and finally solving $C\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$.

QR Decomposition. The QR decomposition of an $M \times N$ matrix is defined as

$$\underset{M\times N}{A} = \underset{M\times M}{Q}\ \underset{M\times N}{R} \,, \tag{A.103}$$

with the orthonormal matrix Q, with $Q^{\mathsf{T}} = Q^{-1}$, and the upper triangular matrix R, with $r_{ij} = 0, i > j$. Since Q is regular, the rank of R is identical to the rank of A.

The QR decomposition is only unique (up to a multiplication with a diagonal matrix with entries ± 1) if A has full rank and $M \le N$. If we require the diagonal elements of R to be positive, the decomposition is unique. If $M > N$ we have the partition

$$\underset{M\times N}{A} = [\ \underset{M\times N}{Q_1}\ ,\ \underset{M\times(M-N)}{Q_2}\ ] \begin{bmatrix} \underset{N\times N}{R_1} \\ \underset{(M-N)\times N}{0} \end{bmatrix} = Q_1 R_1 \,, \tag{A.104}$$

with some arbitrary matrix Q_2 fulfilling $Q_2^{\mathsf{T}} Q_2^{-1} = I_{M-N}$ and $Q_1^{\mathsf{T}} Q_2 = 0$, since Q is orthonormal. Calculating the QR decomposition requires $4N(M^2 - MN + N^2/3)$ operations for (Q, R) and $2N^2(M - N/3)$ operations if only R_1 is required (see Patel, 2002, Lemma 1.11.1).

The QR decomposition can be used for solving a linear equation system. This is a factor two slower than with Cholesky decomposition but is numerically more stable. It also can be used for determining the null space of a matrix, see below.

Observe, if we have the QR decomposition of the design matrix $A = QR$ of a linear Gauss–Markov model (with R having positive diagonal elements) and the Cholesky decomposition of the corresponding normal equation matrix $A^\mathsf{T}A = C^\mathsf{T}C$, we see that $R = C$, i.e., we can directly determine the Cholesky matrix C from A, without having to build the normal equation matrix.

A.10 Singular Value Decomposition

The singular value decomposition of an $M \times N$ matrix A, with $M \geq N$, is defined as

$$\underset{M\times N}{A} = \underset{M\times M}{U}\ \underset{M\times N}{S}\ \underset{N\times N}{V^\mathsf{T}} = \sum_{n=1}^{N} s_n \boldsymbol{u}_n \boldsymbol{v}_n^\mathsf{T}\,, \tag{A.105}$$

with the orthogonal matrices

$$U^\mathsf{T} = U^{-1} \qquad V^\mathsf{T} = V^{-1} \tag{A.106}$$

not necessarily having determinant $+1$, and the rectangular matrix

$$S = \begin{bmatrix} \underset{N\times N}{\mathrm{Diag}(s_n)} \\ \underset{(M-N)\times N}{0} \end{bmatrix}. \tag{A.107}$$

This partitioning requires $4\,M^2N + 22\,N^2$ operations (see Golub and van Loan, 1996, p. 254) and is generated in MATLAB using $[\mathtt{U},\mathtt{S},\mathtt{V}] = \mathtt{svd}(\mathtt{A})$.

The often very large matrix U may be split into two parts,

$$\underset{M\times M}{U} = [\ \underset{M\times N}{U_1}\ |\ \underset{M\times(M-N)}{U_2}\ ]\,, \tag{A.108}$$

where only the left part is relevant, as

$$A = U_1 \mathrm{Diag}(s_n) V^\mathsf{T}\,. \tag{A.109}$$

This more economic partitioning is generated in MATLAB using $[\mathtt{U1},\mathtt{S},\mathtt{V}] = \mathtt{svd}(\mathtt{A},'\mathtt{econ}')$. It requires $6\,MN^2 + 20\,N^3$ operations (see Golub and van Loan, 1996, p. 254).

A.11 The Null Space and the Column Space of a Matrix

Null space. Given an $M \times N$ matrix A with rank R its null space, also called kernel, is defined as the set of vectors $\boldsymbol{x}$ with $A\boldsymbol{x} = \boldsymbol{0}$:

$$\mathrm{kernel}(A) = \{\boldsymbol{x} \in \mathbb{R}^N \mid A\boldsymbol{x} = \boldsymbol{0}\}\,. \tag{A.110}$$

The dimension of the null space is $N - R$. The null space may be empty.

The null space usually is given by an $N \times (N - R)$ matrix $\mathsf{null}(A) = N$ such that any vector $\boldsymbol{x}$ in the null space is a linear combination of its column vectors:

$$\boldsymbol{x} = N\boldsymbol{b} \qquad \text{for some arbitrary } \boldsymbol{b}\,; \tag{A.111}$$

therefore

$$AN = A\,\mathsf{null}(A) = 0\,. \tag{A.112}$$

In particular, we have for any N-vector $\boldsymbol{y}$ the null space of its transposed vector,

$$\mathsf{null}(\boldsymbol{y}^\mathsf{T}) = \underset{N\times(N-1)}{J} \quad \text{with} \quad \boldsymbol{y}^\mathsf{T}J = 0^\mathsf{T}\,, \quad \text{or} \quad J^\mathsf{T}\boldsymbol{y} = \mathbf{0}\,. \tag{A.113}$$

The null space of a nonzero column vector is empty: $\mathsf{null}(\boldsymbol{y}) = \emptyset$. We always interpret $\mathsf{null}(\bullet)$ as an orthonormal matrix with basis vectors as columns, see (A.106).

Relation to SVD. The null space is closely related to the singular value decomposition of the matrix A. Let

$$\underset{M\times N}{A} = \begin{bmatrix} \underset{M\times R}{U_1} \mid \underset{M\times(N-R)}{U_2} \end{bmatrix} \begin{bmatrix} \underset{R\times R}{\mathrm{Diag}(s_n)} & 0_{R\times(N-R)} \\ 0_{(M-R)\times R} & 0_{(M-R)\times(N-R)} \end{bmatrix} \begin{bmatrix} \underset{N\times R}{V_1} \mid \underset{N\times(N-R)}{V_2} \end{bmatrix}^\mathsf{T}. \tag{A.114}$$

The rank of the matrix is R and the null space is

$$\mathsf{null}(A) = V_2\,. \tag{A.115}$$

Relation to QR Decomposition and Efficient Computation. The null space of an $M \times N$ matrix A with $M < N$ and full row rank is also related to the QR decomposition of its transpose. Let the QR decomposition of the transpose of the $M \times N$ matrix be

$$\underset{N\times M}{A^\mathsf{T}} = \underset{N\times N}{Q}\,\underset{N\times M}{R} = [\,\underset{N\times M}{Q_1}\,,\ \underset{N\times(N-M)}{Q_2}\,] \begin{bmatrix} \underset{M\times M}{R_1} \\ \underset{(N-M)\times M}{0} \end{bmatrix} = Q_1R_1\,, \tag{A.116}$$

where $Q^\mathsf{T} = Q^{-1}$, and R is an upper triangular matrix. We partition the two matrices Q and R after their first M columns and rows, respectively. Then we have $AQ_2 = R_1^\mathsf{T}Q_1^\mathsf{T}Q_2 = 0$ since Q is orthogonal. Thus the null space is identical to

$$\mathsf{null}(A) = Q_2\,. \tag{A.117}$$

For nearly square matrices this procedure for determining the null space is approximately 20 times faster than using the SVD.

Efficient Computation of $\mathsf{null}(\boldsymbol{x}^\mathsf{T})$. The null space $\mathsf{null}(\boldsymbol{x}^\mathsf{T})$ of a N-vector $\boldsymbol{x}^\mathsf{T}$ efficiently can be determined from a partitioning of the rotation matrix representing the minimal rotation $R = R_{ab}\left(\boldsymbol{e}_N^{(N)}, \boldsymbol{x}\right) = [J, \boldsymbol{x}]$ (see (8.76), p. 340) from $\boldsymbol{e}_N^{(N)}$ to the vector $\boldsymbol{x}$. Then the null space $\mathsf{null}(\boldsymbol{x}^\mathsf{T})$ is the left $N \times (N-1)$ submatrix J of R. This follows from $R^\mathsf{T}\boldsymbol{x} = \boldsymbol{e}_N^{(N)}$, thus $\boldsymbol{x}^\mathsf{T}R = [0, ..., 0, x_N]$. If $\boldsymbol{x} = -\boldsymbol{e}_N^{(N)}$, the rotation matrix $R\left((\boldsymbol{e}_N^{(N)}, \boldsymbol{x}\right)$ is not defined. Therefore we use the negative vector $-\boldsymbol{x}$ if $x_N < 0$ thus the rotation matrix

$$R = R_{ab}\left(\boldsymbol{e}_N^{(N)}, \pm\boldsymbol{x}\right) = \left[\underset{N\times(N-1)}{J}\ ,\ \underset{N\times 1}{\pm\boldsymbol{x}}\right], \tag{A.118}$$

and obtain the null space as

$$\mathsf{null}(\boldsymbol{x}^\mathsf{T}) = J\,. \tag{A.119}$$

The rotation matrix can be given explicitly if the vector is spherically normalized and partitioned $\boldsymbol{x}^s = [\boldsymbol{x}_0^\mathsf{T}, x_N]^\mathsf{T}$.

$$J(x_N > 0) = \begin{bmatrix} I_{N-1} - \boldsymbol{x}_0\boldsymbol{x}_0^\mathsf{T}/(1+x_N) \\ -\boldsymbol{x}_0^\mathsf{T} \end{bmatrix}, \quad J(x_N \le 0) = \begin{bmatrix} I_{N-1} - \boldsymbol{x}_0\boldsymbol{x}_0^\mathsf{T}/(1-x_N) \\ \boldsymbol{x}_0^\mathsf{T} \end{bmatrix}. \tag{A.120}$$

This explicit way of determining the null space of a row vector is faster than when using the QR decomposition, since it only requires $2(N-1)^2$ operations. For example, the normalized 3-vector $\boldsymbol{x}^s = [x_1^s, x_2^s, x_3^s]^\mathsf{T}$ with $x_3^s > 0$ has null space

$$J(x_3 > 0) = \begin{bmatrix} 1 - \dfrac{x_1^s x_1^s}{1+x_3^s} & -\dfrac{x_1^s x_2^s}{1+x_3^s} \\ -\dfrac{x_1^s x_2^s}{1+x_3^s} & 1 - \dfrac{x_2^s x_2^s}{1+x_3^s} \\ -x_1^s & -x_2^s \end{bmatrix} . \tag{A.121}$$

Column space The column space of a matrix is the set of all vectors spanned by its columns:

$$\mathsf{span}(A) = \{\boldsymbol{x} \mid \boldsymbol{x} = A\boldsymbol{b} \text{ for some } \boldsymbol{b}\} . \tag{A.122}$$

The column space also is called the *image* of A. It can be given by an $N \times R$ matrix $\mathsf{span}(A)$, whose columns form a basis of the column space of A. The column space can be determined using the singular value decomposition of A:

$$\mathsf{span}(A) = V_1 . \tag{A.123}$$

Therefore, also the matrix $\mathsf{span}(A)$ is orthonormal, see (A.106).

A.12 The Pseudo-inverse

The pseudo-inverse A^+ of a possibly rectangular $M \times N$ matrix A is an $N \times M$ matrix which satisfies the following relations:

$$AA^+A = A , \quad A^+AA^+ = A^+ , \quad (AA^+)^\mathsf{T} = AA^+ , \quad (A^+A)^\mathsf{T} = A^+A . \tag{A.124}$$

Thus AA^+ and A^+A are symmetric idempotent matrices, see A.6. A matrix fulfilling these constraints is unique and also called the *Moore–Penrose inverse*, see (Penrose, 1954; Moore, 1920).

A.12.1 The Pseudo-inverse of a Rectangular Matrix

For an $M \times N$ matrix A with singular value decomposition $A = USV^\mathsf{T}$, its pseudo-inverse is given by

$$A^+ = VS^+U^\mathsf{T} = \sum_{n=1}^{N} s_n^+ \boldsymbol{v}_n \boldsymbol{u}_n^\mathsf{T} , \tag{A.125}$$

with $M \geq N$ and

$$\underset{N\times M}{S^+} = [\mathrm{Diag}(s_n^+) \mid 0_{N\times(M-N)}] \qquad s_n^+ = \begin{cases} 1/s_n, & \text{if } s \neq 0 \\ 0, & \text{else} \end{cases} . \tag{A.126}$$

Thus S^+ is the transpose of S with all nonzero elements replaced by their inverses.

A.12.2 Pseudo-inverse of a Singular Symmetric Matrix

Let the $U \times U$ matrix A be symmetric and singular with rank $Q < U$ and null space $\underset{U\times Q}{N}$, thus

$$AN = 0\,, \qquad N^{\mathsf{T}}N = I_{U-Q}\,. \tag{A.127}$$

Then the pseudo-inverse may be determined from

$$\begin{bmatrix} A^{+} & N \\ N^{\mathsf{T}} & 0 \end{bmatrix} = \begin{bmatrix} A & N \\ N^{\mathsf{T}} & 0 \end{bmatrix}^{-1}\,. \tag{A.128}$$

It explicitly reads as

$$A^{+} = (A + NN^{\mathsf{T}})^{-1} - NN^{\mathsf{T}}\,. \tag{A.129}$$

Proof: Let the singular value decomposition of A be

$$A = [U \mid N]\ \mathrm{Diag}([S, 0])\ [U \mid N]^{\mathsf{T}} \quad \text{with} \quad [U \mid N][U \mid N]^{\mathsf{T}} = I_U\,. \tag{A.130}$$

Then the pseudo-inverse is given by

$$A^{+} = [U \mid N]\ \mathrm{Diag}([S^{+}, 0])\ [U \mid N]^{\mathsf{T}}\,, \tag{A.131}$$

which allows us to prove (A.128).

$$\begin{bmatrix} A & N \\ N^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} A^{+} & N \\ N^{\mathsf{T}} & 0 \end{bmatrix} = \begin{bmatrix} AA^{+} + NN^{\mathsf{T}} & AN \\ N^{\mathsf{T}}A^{+} & N^{\mathsf{T}}N \end{bmatrix} = \begin{bmatrix} I_U & 0 \\ 0 & I_{U-Q} \end{bmatrix}\,. \tag{A.132}$$

◇

A.12.3 The Pseudo-inverse of a Rectangular Matrix Having Full Rank

If the rectangular $M \times N$ matrix with $M > N$ has rank N, its pseudo-inverse is given by

$$A^{+} = (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}\,. \tag{A.133}$$

If $M < N$ then

$$A^{+} = A^{\mathsf{T}}(AA^{\mathsf{T}})^{-1}\,. \tag{A.134}$$

A.12.4 The Weighted Pseudo-inverse of a Rectangular Matrix Having Full Rank

With the positive symmetric $M \times M$ weight matrix W and V, the weighted pseudo-inverse of a rectangular $M \times N$ matrix A with $M > N$ and $\mathrm{rk}A = N$ is defined as

$$A_W^{+} = (A^{\mathsf{T}}WA)^{-1}A^{\mathsf{T}}W\,. \tag{A.135}$$

It fulfills the following relations:

$$AA_W^{+}A = A\,, \quad A_W^{+}AA_W^{+} = A_W^{+}\,, \quad (WAA_W^{+})^{\mathsf{T}} = WAA_W^{+}\,, \quad (A_W^{+}A)^{\mathsf{T}} = I\,. \tag{A.136}$$

This is a special case of the doubly weighted Moore–Penrose inverse of an arbitrary matrix used in Pepić (2010). The weighted pseudo-inverse can be used to determine the minimum $\widehat{\boldsymbol{x}} = A_W^{+}\boldsymbol{b}$ of $(A\boldsymbol{x} - \boldsymbol{b})^{\mathsf{T}}W(A\boldsymbol{x} - \boldsymbol{b})$ when A has full column rank.

A.13 Matrix Exponential

The exponential of a matrix occurs naturally with rotation matrices and is frequently used when updating linear transformations.

The exponential of an $n \times n$ matrix A is defined as

$$e^{A} = I_n + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \dots . \tag{A.137}$$

In the case of a diagonal matrix $D = \mathrm{Diag}(d_i)$, we have

$$e^{D} = \mathrm{Diag}\left(e^{d_i}\right) . \tag{A.138}$$

Taking the decomposition of a symmetric A,

$$A = UDU^{-1} , \tag{A.139}$$

we therefore obtain the following relations:

$$e^{A} = Ue^{D}U^{-1} , \tag{A.140}$$

$$e^{A}e^{-A} = I_n , \tag{A.141}$$

$$e^{\alpha A}e^{\beta A} = e^{(\alpha+\beta)A} , \tag{A.142}$$

$$e^{\left(A^{\mathsf{T}}\right)} = \left(e^{A}\right)^{\mathsf{T}} , \tag{A.143}$$

$$\det\left(e^{A}\right) = e^{\mathrm{tr}A} . \tag{A.144}$$

For skew symmetric matrices S, see Sect. 8.1.1, we get

$$R = e^{S} , \tag{A.145}$$

a proper rotation matrix with $|R| = 1$.

In general, the product is not commutative,

$$e^{A}e^{B} \neq e^{B}e^{A} , \tag{A.146}$$

but if the matrices commute, we have

$$AB = BA \quad \Rightarrow \quad e^{A}e^{B} = e^{B}e^{A} = e^{(A+B)} , \tag{A.147}$$

which can be shown by expanding the product series and collecting terms of the same order.

As the multiplication of a matrix with a scalar e^{λ} is the multiplication with $e^{\lambda}I_n$, which commutes with all matrices, we have, from (A.147),

$$e^{\lambda}e^{A} = e^{\lambda I_n + A} . \tag{A.148}$$

The inverse relation to the matrix exponential is called the matrix logarithm. For regular, symmetric A we have, with (A.139),

$$\ln A = U \ln D U^{-1} , \tag{A.149}$$

with

$$\ln \mathrm{Diag}(d_i) = \mathrm{Diag}(\ln d_i) . \tag{A.150}$$

As for any complex number $z = re^{i\theta}$ we have

$$\ln z = \ln r + i(\theta + 2\pi k)\,, \tag{A.151}$$

with some arbitrary integer $k \in \mathbb{Z}$; the logarithm of matrices is not unique.

A.14 Tensor Notation

Tensor notation simplifies the derivation of multilinear forms. The elements of vectors and matrices are represented using their elements together with the indices. The indices are assumed to run through a fixed sequence.

Coordinates of points $\mathcal{X}$ are written with upper-script indices: x^i, the index $i \in \{1, 2, 3\}$. The coordinates of lines $\mathcal{l}$ are written with lower-indices, thus l_i. They are tensors of first-order. The inner product $w = \mathbf{x}^\mathsf{T}\mathbf{l}$ is written as

$$w = x^i l_i = l_i x^i \doteq \sum_{i=1}^{3} x^i l_i\,. \tag{A.152}$$

The sum is taken over indices with the same name, one index being an upper one, the other a lower one.

Matrices are represented by two indices. They are second-order tensors. For example, the homography matrix H is represented by h_i^j, which allows us to write the projective transformation as

$$x^j = h_i^j x^i \doteq \sum_{i=1}^{3} h_i^j x^i \quad j = 1, 2, 3\,. \tag{A.153}$$

We also have matrices with two lower or two upper indices, e.g., when determining the quadratic form

$$s^2 = x^i x^j w_{ij} \doteq \mathbf{x}^\mathsf{T}\mathsf{W}\mathbf{x}\,, \tag{A.154}$$

with the weight matrix w_{ij}. The Jacobian $\boldsymbol{a} = (\partial s/\partial \boldsymbol{x})$ of a scalar s and a vector $\boldsymbol{x}$ has indices which must satisfy the relation

$$\mathrm{d}s = a_i \mathrm{d}x^i\,. \tag{A.155}$$

However, in case we want to express $\boldsymbol{b} = \partial t/\partial \boldsymbol{l}$, we have

$$\mathrm{d}t = b^i \mathrm{d}l_i\,. \tag{A.156}$$

The index of a vector may be exchanged by multiplication with the unit matrix:

$$x^j = \delta_i^j x^i\,. \tag{A.157}$$

The transpose of a matrix a_{lm} is the matrix a_{ml}. The inverse b^{ij} of a matrix a_{jk} must fulfill

$$b^{ij} a_{jk} = \delta_k^i\,. \tag{A.158}$$

The skew symmetric matrix $\mathsf{S}(\mathbf{x})$ depending on a vector $\mathbf{x}$ uses the fully anti-symmetric third-order tensor

$$\varepsilon_{ijk} = \begin{cases} 1, & \text{if } (ijk) \text{ is an even permutation} \\ -1, & \text{if } (ijk) \text{ is an odd permutation} \\ 0, & \text{if } (ijk) \text{ is no permutation, thus contains an index at least twice,} \end{cases} \tag{A.159}$$

as

$$s_{jk} = \varepsilon_{ijk} x^i\,. \tag{A.160}$$

A.15 Variance Propagation of Spectrally Normalized Matrix

Given an uncertain $n \times n$-homography $(\mathsf{H}, \Sigma_{\mathrm{hh}})$ with $\Sigma_{\mathrm{hh}} = \mathbb{D}(\mathrm{vec}\mathsf{H})$, spectral normalization leads to

$$\mathsf{M} = \frac{\mathsf{H}}{\mathrm{abs}|\mathsf{H}|^{1/n}} \qquad \mathrm{sign}(|\mathsf{M}|) = \mathrm{sign}(|\mathsf{H}|)\,. \tag{A.161}$$

This section shows that the covariance matrix Σ_{mm} of $\mathrm{vec}\mathsf{M}$ is

$$\Sigma_{\mathrm{mm}} = J_{\mathrm{mh}} \Sigma_{\mathrm{hh}} J_{\mathrm{mh}}^{\mathsf{T}}\,, \tag{A.162}$$

with the Jacobian

$$J_{\mathrm{mh}} = \frac{1}{\mathrm{abs}|\mathsf{H}|^{1/n}} \left(I_n - \frac{1}{n} \mathbf{h}\mathbf{i}^{\mathsf{T}} \right) \tag{A.163}$$

and

$$\mathbf{h} = \mathrm{vec}\mathsf{H} \qquad \mathbf{m} = \mathrm{vec}\mathsf{M} \qquad \mathbf{i} = \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})\,. \tag{A.164}$$

Proof: In the following we assume the determinants to be positive. We use the derivative of the determinant of a general regular matrix X (see Petersen and Pedersen, 2012),

$$\mathrm{d}|X| = |X| \mathrm{tr}(X^{-1} \mathrm{d}X)\,. \tag{A.165}$$

We have

$$\mathrm{tr}(AB) = \mathrm{vec}(A^{\mathsf{T}})^{\mathsf{T}}\, \mathrm{vec}B\,; \tag{A.166}$$

thus, we obtain

$$\mathrm{d}|X| = |X| \mathrm{vec}(X^{-\mathsf{T}})^{\mathsf{T}}\, \mathrm{vec}(\mathrm{d}X)\,. \tag{A.167}$$

We can now determine the Jacobian J_{mh} of $\mathbf{m}$ w.r.t. $\mathbf{h}$. From

$$|\mathsf{H}|^{1/n}\, \mathsf{M} = \mathsf{H}\,, \tag{A.168}$$

we obtain the differential

$$\mathrm{d}(|\mathsf{H}|^{1/n})\, \mathsf{M} + |\mathsf{H}|^{1/n}\, \mathrm{d}\mathsf{M} = \mathrm{d}\mathsf{H}\,. \tag{A.169}$$

For $y = x^a$ we have $\mathrm{d}y = \mathrm{d}(x^a) = a x^{a-1} \mathrm{d}x$. Thus the differential is

$$\frac{1}{n} |\mathsf{H}|^{-(n-1)/n} \mathrm{d}|\mathsf{H}|\, \mathsf{M} + |\mathsf{H}|^{1/n}\, \mathrm{d}\mathsf{M} = \mathrm{d}\mathsf{H}\,, \tag{A.170}$$

and therefore, using M from (A.161),

$$\frac{1}{n} |\mathsf{H}|^{-(n-1)/n} |\mathsf{H}| \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})^{\mathsf{T}}\, \mathrm{vec}(\mathrm{d}\mathsf{H})\, \mathsf{M} + |\mathsf{H}|^{1/n}\, \mathrm{d}\mathsf{M} = \mathrm{d}\mathsf{H}\,. \tag{A.171}$$

Vectorization yields

$$\frac{1}{n} \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})^{\mathsf{T}}\, \mathrm{vec}(\mathrm{d}\mathsf{H})\, \mathrm{vec}\mathsf{H} + |\mathsf{H}|^{1/n}\, \mathrm{vec}(\mathrm{d}\mathsf{M}) = \mathrm{vec}(\mathrm{d}\mathsf{H})\,. \tag{A.172}$$

Solving for $\mathrm{vec}(\mathrm{d}\mathsf{M})$ gives

$$\mathrm{vec}(\mathrm{d}\mathsf{M}) = |\mathsf{H}|^{-1/n} \left(I_n - \frac{1}{n} \mathrm{vec}\mathsf{H}\, \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})^{\mathsf{T}} \right) \mathrm{vec}(\mathrm{d}\mathsf{H})\,. \tag{A.173}$$

With

$$\mathbf{h} = \mathrm{vec}\mathsf{H} \qquad \mathbf{m} = \mathrm{vec}\mathsf{M} \qquad \mathbf{i} = \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})\,, \tag{A.174}$$

this finally leads to

$$\mathrm{d}\mathbf{m} = J_{\mathrm{mh}} \mathrm{d}\mathbf{h}\,, \quad \text{with} \quad J_{\mathrm{mh}} = |\mathsf{H}|^{-1/n} \left(I_n - \frac{1}{n} \mathbf{h}\mathbf{i}^{\mathsf{T}} \right)\,. \tag{A.175}$$

This proves the claim. ◇

Observe

$$n = \mathrm{tr} I_n = \mathrm{tr}(\mathsf{H}^{-1}\mathsf{H}) = \mathrm{vec}(\mathsf{H}^{-\mathsf{T}})^{\mathsf{T}} \mathrm{vec}\mathsf{H} = \mathbf{i}^{\mathsf{T}} \mathbf{h}\,. \tag{A.176}$$

Therefore

$$\mathbf{i}^{\mathsf{T}} J_{\mathrm{mh}} = \mathbf{0} \qquad J_{\mathrm{mh}} \mathbf{h} = \mathbf{0}\,, \tag{A.177}$$

and thus the null space of Σ_{mm} is $\mathbf{i} = \lambda \mathrm{vec}(\mathsf{H}^{\circ})$, where H° is the cofactor matrix of H.

References

The numbers at the end of each reference are the pages where it is cited.

Abadir, K. R. and J. R. Magnus (2002). Notation in econometrics: a proposal for a standard. *Econometrics Journal 5*, 76–90. 16

Abdel-Aziz, Y. I. and H. M. Karara (1971). Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range Photogrammetry*, Falls Church, VA, USA, pp. 1–18. American Society of Photogrammetry. 472

Abraham, S. and W. Förstner (2005). Fish-eye-stereo calibration and epipolar rectification. *ISPRS J. of Photogrammetry & Remote Sensing 59*(5), 278–288. 444, 486

Abraham, S. and T. Hau (1997). Towards Autonomous High-Precision Calibration of Digital Cameras. In *Videometrics V, Proceedings of SPIE Annual Meeting*, 3174, San Diego, pp. 82–93. 512

Absil, P.-A., R. Mahony, and R. Sepulchre (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press. 370, 416

Ackermann, F. (1966). On the Theoretical Accuracy of Planimetric Block Triangulation. *Photogrammetria 21*, 145–170. 670, 673

Ackermann, F., H. Ebner, and H. Klein (1970). Ein Programmpaket für die Aerotriangulation mit unabhängigen Modellen. *Zeitschrift für Bildmessung und Luftbildwesen 38*, 218–224. 649

Ackermann, F., H. G. Jerie, and K. Kubik (1972). 129: Räumliche Aerotriangulation. In W. Jordan, O. Eggert, and M. Kneissl (Eds.), *Handbuch der Vermessungskunde*, Volume III a/3: Photogrammetrie. Metzelersche Verlagsbuchhandlung. 450

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics 21*, 243–247. 184

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Trans. on Automatic Control 19*, 716–723. 138

Al-Sharadqah, A. and N. Chernov (2009). Error analysis for circle fitting algorithms. *Electronic Journal of Statistics 3*, 886–911. 182

Albertz, J. (2001). Albrecht Meydenbauer - Pioneer of Photogrammetric Documentation of the Cultural Heritage. In *Proc. 18th Int. Symposium CIPA*, Potsdam. 1

Aldroubi, A. and K. Gröchenig (2001). Nonuniform Sampling and Reconstruction in Shift-Invariant Spaces. *SIAM Rev. 43*(4), 585–620. 735

Ansar, A. and K. Daniilidis (2002). Linear Pose Estimation from Points or Lines. In *Proc. ECCV*. 521

Antone, M. and S. Teller (2002). Scalable Extrinsic Calibration of Omni-Directional Image Networks. *International Journal of Computer Vision 49*, 143–174. 402

Antoniou, A. and W.-S. Lu (2007). *Practical Optimization: Algorithms and Engineering Applications* (1st ed.). Springer. 103

Arun, K. S., T. S. Huang, and S. B. Blostein (1987). Least-Squares Fitting of Two 3D Point Sets. *IEEE T-PAMI 9*(5), 698–700. 340, 407

Ashdown, M. (1998). Geometric algebra. http://geometry.mrao.cam.ac.uk/, last visited 1.12.2015. 236

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4

Åström, K. (1998). Using Combinations of Points, Lines and Conics to Estimate Structure and Motion. In O. Eriksson (Ed.), *Proceedings SSAB Symposium on Image Analysis*, pp. 61–64. Swedish Society for Automated Image Analysis. 369

Avidan, S. and A. Shashua (1998). Threading Fundamental Matrices. In H. Burkhardt and B. Neumann (Eds.), *Proc. ECCV 98*, Volume 1 of *LNCS 1406*, pp. 124–140. Springer. 634

Avidan, S. and A. Shashua (2000). Trajectory Triangulation: 3D Reconstruction of Moving Points from a Monocular Image Sequence. *PAMI 22*(4), 348–357. 303

Baarda, W. (1967). *Statistical Concepts in Geodesy*, Volume 2/4 of *Publication on Geodesy, New Series*. Delft: Netherlands Geodetic Commission. 66, 72, 75, 123, 125, 126, 127, 134

Baarda, W. (1968). *A Testing Procedure for Use in Geodetic Networks*, Volume 2/5 of *Publication on Geodesy, New Series*. Delft: Netherlands Geodetic Commission. 72, 75, 123, 125, 126, 127, 134

Baarda, W. (1973). *S-Transformations and Criterion Matrices*, Volume 5/1 of *Publication on Geodesy, New Series*. Netherlands Geodetic Commission. 75, 109, 110, 112, 120

Baker, S. and S. K. Nayar (1999). A Theory of Single-Viewpoint Catadioptric Image Formation. *International Journal of Computer Vision 35*(2), 1–22. 488

Bartoli, A. and P. Sturm (2005). Structure-from-motion using lines: Representation, triangulation and bundle adjustment. *Computer Vision and Image Understanding 100*, 416–441. 381

Bathe, K.-J. and E. L. Wilson (1973). Solution Methods for Eigenvalue Problems in Structural Mechanics. *Int. J. for Numerical Methods in Engineering 8*, 213–226. 773

Bathia, R. (2002). Eigenvalues of AB and BA. *Resonance 7*(1), 88–93. 773

Beder, C. and R. Steffen (2006). Determining an initial image pair for fixing the scale of a 3D reconstruction from an image sequence. In K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer (Eds.), *Pattern Recognition*, LNCS 4174, pp. 657–666. Springer. 709

Begelfor, E. and M. Werman (2005). How to Put Probabilities on Homographies. *IEEE Trans. Pattern Anal. Mach. Intell. 27*(10), 1666–1670. 384

Berger, M., A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva (2014). State of the Art in Surface Reconstruction from Point Clouds. In S. Lefebvre and M. Spagnuolo (Eds.), *Eurographics 2014 - State of the Art Reports*. The Eurographics Association. 727

Berliner, A. (1928). *Lehrbuch der Physik*. Salzwasser Verlag. 666

Bickel, D. and R. Fruehwirth (2006). On a Fast, Robust Estimator of the Mode: Comparisons to Other Robust Estimators with Applications. *Computational Statistics and Data Analysis 12*, 3500–3530. 146

Binford, T. O. (1981). Inferring Surfaces from Images. *Artificial Intelligence 17*, 205–244. 448

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer. 43, 78, 83, 93

Blake, A. and A. Zisserman (1986). Invariant surface reconstruction using weak continuity constraints. In *Proc. Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, pp. 62–67. 739

Blazquez, M. and I. Colomina (2010). On the Role of Self-Calibration Functions in Integrated Sensor Orientation. In *Proc. of the International Calibration and Orientation Workshop (EuroCOW)*. 510, 684

Bloch, A. (1978). *Murphy's Law and Other Reasons Why Things Go Wrong*. Price/Stern/Sloan Publ. Inc. 142

Bookstein, F. (1979). Fitting conic sections to scattered data. *CGIP 9*(1), 56–71. 177, 182

Bosman, E. R., D. Eckhart, and K. Kubik (1971). The Application of Piecewise Polynomials to Problems of Curve and Surface Approximation. Technical report, Rijkswaterstaat, The Hague, Netherlands. 733

Box, G. and G. Jenkins (1976). *Time Series Analysis*. Holden-Day. 52, 184

Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press. 150, 161

Brand, L. (1947). *Vector and Tensor Analysis*. John Wiley & Sons, Inc. 193

Brand, L. (1966). *Vector and Tensor Analysis*. John Wiley & Sons, Inc. Tenth printing. 199, 206, 309

Bräuer-Burchardt, C. and K. Voss (2000). Automatic lens distortion calibration using single views. In *Mustererkennung 2000*, pp. 187–194. Springer. 507

Brown, D. (1976). The Bundle Adjustment – Progress and Prospects. In *International Archives of Photogrammetry, Comm. III*. XIIIth ISP Congress, Helsinki. 508

Brown, D. C. (1971). Close-range Camera Calibration. *Photogrammetric Engineering 37*(8), 855–866. 506

Brown, M., R. I. Hartley, and D. Nister (2007). Minimal Solutions for Panoramic Stitching. In *Conf. on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, pp. 1–8. IEEE Computer Society. 323

Browne, J. (2009). Grassmann Algebra. https://sites.google.com/site/grassmannalgebra/, last visited 1.12.2015. 226, 232, 234, 258

Buchanan, T. (1988). The Twisted Cubic and Camera Calibration. *CVGIP 42*(1), 130–132. 521
Busemann, H. and P. J. Kelley (1953). *Projective Geometry and Projective Metrics*. Academic Press, NY. 221
Castleman, K. R. (1996). *Digital Image Processing*. Prentice Hall Inc. 42
Chen, C.-H. and P. G. Mulgaonkar (1990). Robust Vision-Programs Based on Statistical Feature Measures. In *Proc. of IEEE Workshop on Robust Computer Vision, Seattle*. IEEE Computer Society Press. 159
Chen, C.-S. and W.-Y. Chang (2004). On Pose Recovery for Generalized Visual Sensors. *IEEE Trans. on Pattern Analysis and Machine Intelligence 26*(7), 848–861. 452
Chernoff, H. (1964). Estimation of the mode. *Annals of the Institute of Statistical Mathematics 15*(1), 31–41. 146
Chin, T.-J., P. Purkait, A. Eriksson, and D. Suter (2015). Efficient Globally Optimal Consensus Maximisation With Tree Search. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2413–2421. 143, 157
Chum, O. and J. Matas (2005). Matching with PROSAC – Progressive Sample Consensus. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Volume 1, Washington, DC, USA, pp. 220–226. IEEE Computer Society. 154
Chum, O., J. Matas, and J. Kittler (2003). Locally Optimized RANSAC. In *Lecture Notes in Computer Science*, Volume 2781, pp. 236–243. 154
Chumerin, N. and M. M. Van Hulle (2008). Ground Plane Estimation Based on Dense Stereo Disparity. In *Fifth International Conference on Neural Networks and Artificial Intelligence*. 602
Coleman, T. F. and D. C. Sorensen (1984). A Note on the Computation of an Orthonormnal Basis for the Null Space of a Matrix. *Mathematical Programming 29*, 234–242. 179
Collins, R. (1993). *Model Acquisition Using Stochastic Projective Geometry*. Ph. D. thesis, Department of Computer Science, University of Massachusetts. Also published as UMass Computer Science Technical Report TR95-70. 359
Cook, R. D. (1977). Detection of Influential Observation in Linear Regression. *Technometrics 19*(1), 15–18. 127
Cook, R. D. and S. Weisberg (1982). *Residuals and Influence in Regression*. Chapman and Hall. 123, 127
Cooper, M. C. (1993). Interpretation of line drawings of complex objects. *Image and Vision Computing 11*(2), 82–90. 523
Courant, R., H. Robbins, and I. Stewart (1996). *What Is Mathematics? An Elementary Approach to Ideas and Methods*. Oxford University Press. 303
Cover, T. and J. A. Thomas (1991). *Elements of Information Theory*. John Wiley & Sons. 81
Cramer, M. (1999). Direct Geocoding – is Aerial Triangulation Obsolete? In D. Fritsch and R. Spiller (Eds.), *Photogrammetric Week*, pp. 59–70. Wichmann Verlag, Heidelberg. 721
Cramer, M. (2010). The DGPF-Test on Digital Airborne Camera Evaluation Overview and Test Design. *Z. f. Photogrammetrie, Fernerkundung, Geoinformation 2*, 73–82. 682, 683
Cremona, L. (1885). *Elements of Projective Geometry*. Clarendon Press Oxford. 277, 284
Criminisi, A. (1997). Modelling and Using Uncertainties in Video Metrology. Technical report, University of Oxford. 359
Criminisi, A. (2001). *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer. 33, 523
Cuthill, E. and J. McKee (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proc. 24th Nat. Conf. ACM*, pp. 157–172. 662
Das, G. B. (1949). A Mathematical Approach to Problems in Photogrammetry. *Empire Survey Review X*(73), 131–137. 472, 473
Dellaert, F. and M. Kaess (2006). Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research 25*(12), 1181–1203. 654
Demazure, M. (1988). Sur deux problèmes de reconstruction. Technical Report 882, INRIA. 577
Dempster, A. P. (1969). *Elements of Continuous Multivariate Analysis*. Addison-Wesley. 33
Dhome, M., J. T. Lapreste, M. Richetin, and G. Rives (1989). Determination of the Attitude of 3-D Objects from a Single Perspective View. *IEEE T-PAMI 11*, 1265–1278. 513
Dhome, M., J. T. Lapreste, G. Rives, and M. Richetin (1990). Spatial Localization of Modelled Objects of Revolution in Monocular Perspective Vision. In *Proceedings of the First European Conference on Computer Vision*, ECCV 90, New York, NY, USA, pp. 475–485. Springer. 534

Dickscheid, T., T. Läbe, and W. Förstner (2008). Benchmarking Automatic Bundle Adjustment Results. In *Int. Archives for Photogrammetry and Remote Sensing*, Volume XXXVII, part B3a. 114, 668

do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces*. Prentice Hall. 741

Dorst, L., D. Fontijne, and S. Mann (2009). *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann. 236

Draper, N. and H. Smith (1998). *Applied Regression Analysis* (3rd ed.). Wiley Series in Probability and Statistics. New York: Wiley. 83

Duda, R. O. and P. E. Hart (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM 15*(1), 11–15. 158

Eade, E. (2014). Lie Groups for 2D and 3D Transformations. http://ethaneade.com/lie_groups.pdf, last visited 2.6.2016. 382

Ebner, H. (1976). Selfcalibrating Block Adjustment. *Zeitschrift für Bildmessung und Luftbildwesen 44*, 128–139. 508, 509, 510, 511, 512, 683

Ebner, H. (1979). Zwei neue Interpolationsverfahren und Beispiele für ihre Anwendung. *BuL 47*, 15–27. 733, 757

Ebner, H., K. Krack, and E. Schubert (1977). Genauigkeitsmodelle für die Bündelblocktriangulation. *Bildmessung und Luftbildwesen 5*, 141–148. 720

Eggert, D. W., A. Lorusso, and R. B. Fisher (1997). Estimating 3D rigid body transformations: A comparison of four major algorithms. *Mach. Vis. Appl. 9*(5/6), 272–290. 407

Fackler, P. L. (2005). Notes on matrix calculus. Technical report, North Carolina State University, http://www4.ncsu.edu/~pfackler/MatCalc.pdf. 84

Fallat, S. M. and M. J. Tsatsomeros (2002). On the Cayley Transform of Positivity Classes of Matrices. *Electronic Journal of Linear Algebra 9*, 190–196. 336

Faugeras, O. (1992). What Can Be Seen in Three Dimensions with an Uncalibrated Stereo Rig? In G. Sandini (Ed.), *Computer Vision - ECCV '92*, Volume 588 of *LNCS*, pp. 563–578. Springer. 552

Faugeras, O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA, USA: The MIT Press. 245, 495, 578

Faugeras, O. and Q.-T. Luong (2001). *The Geometry of Multiple Images*. MIT Press. with contributions from T. Papadopoulo. 457, 464, 465, 521, 555, 571, 625

Faugeras, O. and T. Papadopoulo (1998). Grassmann-Cayley Algebra for Modeling Systems of Cameras and the Algebraic Equations of the Manifold of Trifocal Tensors. In *Trans. of the Royal Society A, 365*, pp. 1123–1152. 193, 234

Faugeras, O. D. and F. Lustman (1988). Motion and Structure from Motion in a Piecewise Planar Enviroment. *International Journal of Pattern Recognition and Artificial Intelligence 2*(3), 485–508. 578

Faugeras, O. D. and S. J. Maybank (1990). Motion from Point Matches: Multiplicity of Solutions. *International Journal of Computer Vision 4*(3), 225–246. 575, 577

Feferman, S. (2006). Turing's Thesis. *Notices of the American Mathematical Society 53*, 1190–1199. 142

Feichtinger, H. G., K. Gröchenig, and T. Strohmer (1995). Efficient Numerical Methods in Non-uniform Sampling Theory. *Numer. Math. 69*(4), 423–440. 735

Ferraz, L., X. Binefa, and F. Moreno Noguer (2014). Leveraging Feature Uncertainty in the PnP Problem. In *BMVC14*. 518, 519

Ferraz, L., X. Binefa, and F. Moreno-Noguer (2014). Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. 518, 519

Finch, M., J. Snyder, and H. Hoppe (2011). Freeform Vector Graphics with Controlled Thin-plate Splines. *ACM Trans. Graph. 30*(6), 166:1–166:10. 733

Finsterwalder, S. (1903). Eine Grundaufgabe der Photogrammetrie und ihre Anwendung auf Ballonaufnahmen. In *Sebastian Finsterwalder zum 75. Geburtstage*. Abhandlung Königlich-Bayerische Akademie der Wissenschaften, II. Klasse, XXII. Band, II. Abteilung. 515

Fischler, M. A. and R. C. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM 24*(6), 381–395. 144, 153, 155, 316, 515

Fisher, R. A. (1922). On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society of London. Series A 222*, 309–368. 63

Fitzgibbon, A. (2001). Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Volume 1, pp. I–125–I–132 vol.1. 508

Fitzgibbon, A. W., M. Pilu, and R. B. Fisher (1999). Direct Least Squares Fitting of Ellipses. *IEEE T-PAMI 21*(5), 476–480. 183

Forkert, G. (1994). *Die Lösung photogrammetrischer Orientierungs- und Rekonstruktionsaufgaben mittels allgemeiner kurvenförmiger Elemente*. Ph. D. thesis, Institut für Photogrammetrie und Fernerkundung, Wien. 570

Förstner, W. (1979). Das Programm TRINA zur Ausgleichung und Gütebewertung geodätischer Lagenetze. *Zeitschrift für Vermessungswesen 104*(2), 61–72. 88

Förstner, W. (1980). Zur Prüfung zusätzlicher Parameter in Ausgleichungen. *Zeitschrift für Vermessungswesen 105*(11), 510–519. 133

Förstner, W. (1983). Reliablitiy and Discernability of Extended Gauss–Markov Models. In F. Ackermann (Ed.), *Seminar "Mathematical Models of Geodetic/Photogrammetric Point Determiation with regard to Outliers and Systematic Errors"*, pp. 79–103. Deutsche Geodätische Kommission, A98, München. 69, 125, 130

Förstner, W. (1984). Quality Assessment of Object Location and Point Transfer Using Digital Image Correlation Techniques. In *Intl. Archives of Photogrammetry and Remote Sensing*, Volume XXV, Comm. III, Part A3a, pp. 197–219. XVth ISPRS Congress, Rio de Janeiro. 569

Förstner, W. (1985). Determination of the Additive Noise Variance in Observed Autoregressive Processes Using Variance Component Estimation Technique. *Statistics and Decisions Supplement Issue 2*, 263–274. 750, 751

Förstner, W. (1987). Reliability Analysis of Parameter Estimation in Linear Models with Applications to Mensuration Problems in Computer Vision. *CVGIP 40*, 273–310. 67, 87, 122, 126, 127

Förstner, W. (1993). A future of photogrammetric research. *NGT Geodesia 93*(8), 372–383. 7, 442

Förstner, W. (2001). Generic Estimation Procedures for Orientation with Minimum and Redundant Information. In A. Gruen and T. S. Huang (Eds.), *Calibration and Orientation of Cameras in Computer Vision*, Number 34 in Series in Information Sciences. Springer. 130

Förstner, W. (2010a). Minimal Representations for Uncertainty and Estimation in Projective Spaces. In *Proc. of Asian Conference on Computer Vision, Queenstown*. 193, 370

Förstner, W. (2010b). Optimal Vanishing Point Detection and Rotation Estimation of Single Images of a Legoland scene. In *Int. Archives of Photogrammetry and Remote Sensing*. ISPRS Symposium Comm. III, Paris. 359

Förstner, W. (2012). Minimal Representations for Testing and Estimation in Projective Spaces. *Z. f. Photogrammetrie, Fernerkundung und Geoinformation 3*, 209–220. 370

Förstner, W. (2013, 08). Graphical Models in Geodesy and Photogrammetry. *Zeitschrift für Photogrammetrie, Fernerkundung, Geoinformation 2013*(4), 255–267. 654

Förstner, W. (2016). On the equivalence of S-transformations and reducing coordinates. Technical report, University of Bonn, Institute for Geodesy and Geoinformation. 112

Förstner, W., A. Brunn, and S. Heuel (2000). Statistically Testing Uncertain Geometric Relations. In G. Sommer, N. Krüger, and C. Perwass (Eds.), *Mustererkennung 2000*, Informatik aktuell, pp. 17–26. 22. DAGM Symposium, Kiel: Springer. 193

Förstner, W., T. Dickscheid, and F. Schindler (2009). Detecting Interpretable and Accurate Scale-Invariant Keypoints. In *12th IEEE International Conference on Computer Vision (ICCV'09)*, Kyoto, Japan, pp. 2256–2263. 707

Förstner, W. and E. Gülch (1987). A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Circular Features. In *Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data*, Interlaken, pp. 281–305. 402

Förstner, W. and B. Moonen (1999). A Metric for Covariance Matrices. In *Quo vadis geodesia ...? Festschrift for Erik W. Grafarend on the occasion of his 60th birthday*, Schriftenreihe der Institute des Studiengangs Geodaesie und Geoinformatik, pp. 113–128 Part 1. Geodaetisches Institut der Universitaet Stuttgart. 121

Fraser, C. (1997). Digital camera self-calibration. *Photogrammetry and Remote Sensing 52*(4), 149–159. 507

Fraser, C. (2013). Automatic Camera Calibration in Close Range Photogrammetry. *Photogrammetric Engineering & Remote Sensing 79*(4), 381–388. 698

Fraundorfer, F., P. Tanskanen, and M. Pollefeys (2010). A Minimal Case Solution to the Calibrated Relative Pose Problem for the Case of Two Known Orientation Angles. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pp. 269–282. Springer. 581

Galton, F. (1890). Kinship and Correlation. *North American Review 150*, 419–431. 81

Gao, X., X. Hou, J. Tang, and H. Cheng (2003). Complete Solution Classification for the Perspective-Three-Point Problem. *IEEE T-PAMI 25*(8), 930–943. 515

Gauss, C. F. (1903). Zur Hannoverschen Triangulation. In K. G. der Wissenschaften zu Göttingen (Ed.), *Carl Friedrich Gauss – Werke*, pp. 343–434. Springer. 3

Gebken, C. (2009). *Conformal Geometric Algebra in Stochastic Optimization*. Ph. D. thesis, Christian-Albrechts-University of Kiel, Institute of Computer Science. 236

Gillard, J. W. (2006). An Historical Overview of Linear Regression with Errors in both Variables. Technical report, School of Mathematics, Cardiff University. 161

Goldstein, T., P. Hand, C. Lee, V. Voroninski, and S. Soatto (2015). ShapeFit and ShapeKick for Robust, Scalable Structure from Motion. In *Proc. of European Conference on Computer Vision*. arXiv:1608.02165v1 [cs.CV]. 713

Golub, G. H. and C. F. van Loan (1996). *Matrix Computations* (3rd ed.). Johns Hopkins Studies in the Mathematical Sciences. Baltimore, MD: The Johns Hopkins University Press. 84, 86, 161, 286, 776, 777

Grimson, W. E. L. (1981). *From Images to Surfaces: A Computational Study to the Human Early Visual System*. Cambridge, MA: MIT Press. 733, 739, 741

Griva, I., S. Nash, and A. Sofer (2009). *Linear and Nonlinear Optimization: Second Edition*. Society for Industrial and Applied Mathematics. 105

Gros, P. and L. Quan (1992). Projective Invariants for Vision. Technical Report RT 90 IMAG - 15 LIFIA, LIFIA Institut IMAG, Grenoble. 266

Grossberg, M. D. and S. K. Nayar (2001). A general imaging model and a method for finding its parameters . In *Int. Conference on Computer Vision*. 446

Grün, A. (1978). Experiences with Self-Calibrating Bundle Adjustment. In *Proceedings of the American Conference on Surveying and Mapping and the American Society of Photogrammetry*, Washington, DC, USA. 510, 512, 683

Gründig, L. (1975). Ein Verfahren zur Auswertung und strengen Ausgleichung von großräumigen Polaraufnahmen. *Zeitschrift für Vermessungswesen 9*, 453–457. 150

Grunert, J. A. (1841). Das Pothenot'sche Problem in erweiterter Gestalt nebst über seine Anwendungen in der Geodäsie. *Grunerts Archiv für Mathematik und Physik 1*, 238–248. 513

Haala, N., H. Hastedt, K. Wolff, C. Ressl, and S. S. Baltrusch (2010). Digital Photogrammetric Camera Evaluation - Generation of Digital Elevation Models. *Zeitschrift für Photogrammetrie, Fernerkundung und Geoinformation 2*, 98–115. 765

Haala, N. and M. Rothermel (2012). Dense Multi-Stereo Matching for High Quality Digital Elevation Models. *Photogrammetrie Fernerkundung Geoinformation (PFG) 4*, 331–343. 727

Hampel, F. R., E. M. Ronchetty, P. J. Rousseeuw, and W. A. Stahel (1986). *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley. 143, 144, 147

Haralick, R. and L. G. Shapiro (1992). *Computer and Robot Vision*, Volume II. Reading, MA: Addison-Wesley. 569

Haralick, R. M., C. Lee, K. Ottenberg, and M. Nölle (1994). Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *International Journal of Computer Vision 13*(3), 331–356. 515

Hartley, R. I. (1992). Estimation of Relative Camera Positions for Uncalibrated Cameras. In G. Sandini (Ed.), *Computer Vision–ECCV'92*, Volume 588 of *LNCS*, pp. 579–587. Proc. 2nd European Conf. on Computer Vision, Santa Margherita, Ligure, Italy: Springer. 552, 581

Hartley, R. I. (1997a). In Defense of the Eight-Point Algorithm. *IEEE T-PAMI 19*(6), 580–593. 286, 287, 603

Hartley, R. I. (1997b). Lines and points in three views and the trifocal tensor. *IJCV 22*(2), 125–140. 636

Hartley, R. I., K. Aftab, and J. Trumpf (2011). L1 Rotation Averaging Using the Weiszfeld Algorithm. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, Washington, DC, USA, pp. 3041–3048. IEEE Computer Society. 714

Hartley, R. I. and P. Sturm (1997). Triangulation. *Computer Vision and Image Understanding 68*(2), 146–157. 596

Hartley, R. I., J. Trumpf, Y. Dai, and H. Li (2013). Rotation Averaging. *International Journal of Computer Vision 103*(3), 267–305. 714

Hartley, R. I. and A. Zisserman (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press. 46, 156, 238, 245, 258, 321, 355, 357, 359, 371, 406, 426, 457, 464, 465, 466, 473, 495, 521, 555, 571, 594, 636

Hedborg, J., P.-E. Forssen, M. Felsberg, and E. Ringaby (2012). Rolling shutter bundle adjustment. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, Washington, DC, USA, pp. 1434–1441. IEEE Computer Society. 452

Heikkila, J. (2000). Geometric Camera Calibration Using Circular Control Points. *T-PAMI 22*(10), 1066–1077. 213

Helmert, F. R. (1872). *Die Ausgleichungsrechnung nach der Methode der Kleinsten Quadrate*. Leipzig: Teubner. 160

Hestenes, D. and R. Ziegler (1991). Projective Geometry with Clifford Algebra. *Acta Applicandae Mathematicae 23*, 25–63. 236

Heuel, S. (2004). *Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction*, Volume 3008 of *Lecture Notes in Computer Science*. Springer. PhD. Thesis. 193

Hofmann-Wellenhof, B., H. Lichtenegger, and E. Wasle (2008). *GNSS - Global Navigation Satellite Systems*. Springer. 493

Holt, R. J. and A. N. Netravali (1995). Uniqueness of Solutions to Three Perspective Views of Four Points. *IEEE Trans. Pattern Anal. Mach. Intell. 17*(3), 303–307. 636

Horn, B. K. P. (1990). Relative Orientation. *IJCV 4*(1), 59–78. 588, 593

Hotelling, H. (1931). The Generalization of Student's Ratio. *The Annals of Mathematical Statistics 2*(3), 360–378. 70

Howell, T. D. and J.-C. Lafon (1975). The Complexity of the Quaternion Product. Technical Report TR75-245, Cornell University. 334

Huber, P. J. (1981). *Robust Statistics*. New York: Wiley. 144

Huber, P. J. (1991). Between Robustness and Diagnostics. In W. Stahel and S. Weisberg (Eds.), *Directions in Robust Statistics and Diagnostics*, pp. 121–130. Springer. 115, 142

Huber, P. J. (2009). *Robust Statistics* (2nd ed.). New York: John Wiley. 142

Illingworth, J. and J. Kittler (1988). A Survey of the Hough Transform. *CVGIP 44*(1), 87–116. 158

Ilson, C. F. (1997). Efficient pose clustering using a randomized algorithm. *Int. Journal on Computer Vision 23*, 131–147. 158

Jacobi, W. G. (2005). Regression III: Advanced Methods. `http://polisci.msu.edu/jacoby/icpsr/regress3/`, last visited 1.12.2015. 115

Jacobsen, K., M. Cramer, R. Ladstädter, C. Ressl, and V. Spreckels (2010). DGPF-Project: Evaluation of digital photogrammetric camera systems geometric performance. *Z. f. Photogrammetrie, Fernerkundung und Geoinformation 2*, 83–97. 682, 683

Jin, H. (2008). A three-point minimal solution for panoramic stitching with lens distortion. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. 323

Jolion, J.-M., P. Meer, and S. Bataouche (1991, 8). Robust Clustering with Applications in Computer Vision. *IEEE T-PAMI 13*(8), 791–801. 146

Jones, M. (2000). Introduction to Plücker Coordinates. `http://www.flipcode.com/archives/Introduction_To_Plcker_Coordinates.shtml`, last visited 1.12.2015. 245

Julier, S. J. and J. K. Uhlmann (1997). A new extension of the Kalman filter to nonlinear systems. In *11th International Symposium on Aerospace/Defense Sensing (AeroSense), Simulations and Controls*. 47

Kabanikhin, S. I. (2008). Definitions and examples of inverse and ill-posed problem. *Journal of Inverse and Ill-posed Problems 16*(4), 317–357. 82

Kaess, M., H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert (2012). iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *International Journal of Robotics Research 31*, 217–236. 709

Kager, H., K. Kraus, and K. Novak (1985). Entzerrung ohne Passpunkte. *Bildmessung und Luftbildwesen 53*, 43–53. 578

Kahl, F. and A. Heyden (1998). Using conic correspondences in two images to estimate the epipolar geometry. In *IEEE Proc. International Conference on Computer Vision*, pp. 761–766. 570

Kanatani, K. (1990). *Group Theoretical Methods in Image Understanding*. New York: Springer. 326, 340

Kanatani, K. (1991). Hypothesizing and Testing Geometric Properties of Image Data. *CVGIP: Image Understanding 54*(3), 349–357. 359

Kanatani, K. (1993). *Geometric Computation for Machine Vision*. Oxford Engineering Science Series. Oxford: Clarendon Press. 193, 362, 534

Kanatani, K. (1996). *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science. 33, 359

Kanatani, K., A. Al-Sharadqah, N. Chernovand, and Y. Sugaya (2012). Renormalization Returns: Hyper-renormalization and Its Applications. In *Proc. European Conf. Computer Vision*, pp. 384–397. 183

Kanatani, K., Y. Sugaya, and H. Niitsuma (2008). Triangulation from two views revisited: Hartley-Sturm vs. optimal correction. In *British Machine Vision Conference*, pp. 173–182. 597, 600

Kaseorg, A. (2014). How do I find the side of the largest cube completely contained inside a regular tetrahedron of side s? https://www.quora.com/How-do-I-find..., last visited 10/2015. 538

Kaucic, R., R. I. Hartley, and N. Y. Dano (2001). Plane-based Projective Reconstruction. In *ICCV*, pp. 420–427. 713

Kiryati, N., Y. Eldar, and A. M. Bruckstein (1990). A Probabilistic Hough Transform. Technical Report 746, Technion Israel Institute of Technology, Dept. of Electrical Engineering, Haifa, Israel. Also submitted to IEEE Workshop on Robust Computer Vision, 1990. 158

Kúkelová, Z. (2013). *Algebraic Methods in Computer Vision*. Ph. D. thesis, Faculty of Electrical Engineering, Prague. 575

Kúkelová, Z., M. Byröd, K. Josephson, T. Pajdla, and K. Aström (2010). Fast and robust numerical solutions to minimal problems for cameras with radial distortion. *Computer Vision and Image Understanding 114*(2), 234–244. 323, 508

Kúkelová, Z. and T. Pajdla (2007). A minimal solution to the autocalibration of radial distortion. In *Conference on Computer Vision and Pattern Recognition, 2007*. 323

Klopschitz, M., A. Irschara, G. Reitmayr, and D. Schmalstieg (2010). Robust Incremental Structure from Motion. In *Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*. 711

Klumpp, A. R. (1976). Singularity-free extraction of a quaternion from a direction-cosine matrix. *Journal of Spacecraft and Rockets 13*, 754–755. 328

Kneip, L., D. Scaramuzza, and R. Siegwart (2011). A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *Proc. of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 515

Koch, K.-R. (1996). Robuste Parameterschätzung. *AVN 103*(1), 1–17. 143

Koch, K.-R. (1999). *Parameter Estimation and Hypothesis Testing in Linear Models* (2nd ed.). Springer. 34, 72, 79, 81, 90, 97, 99, 100, 111, 113, 137, 141, 181, 775

Koenderink, J. J. (1990). *Solid Shape*. Cambridge/London: MIT Press. 442, 727

Krames, J. (1941). Zur Ermittlung eines Objektes aus zwei Perspektiven. (Ein Beitrag zur Theorie der „Gefährlichen Örter".). *Monatshefte für Mathematik und Physik 49*, 327–354. 571

Krarup, T., J. Juhl, and K. Kubik (1980). Götterdämmerung over Least Squares Adjustment. In *Intl. Archives of Photogrammetry and Remote Sensing*, Volume XXIII, pp. 369–378. Proc. XIVth ISPRS Congress, Hamburg, Germany. 149

Kraus, K. (1993). *Photogrammetry*. Bonn: Dümmler. 467, 477

Kraus, K. and N. Pfeifer (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *Photogrammetry and Remote Sensing 53*, 193–203. 755, 756

Kschischang, F., B. Frey, and H.-A. Loeliger (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on 47*(2), 498–519. 654

Läbe, T., T. Dickscheid, and W. Förstner (2008). On the Quality of Automatic Relative Orientation Procedures. In *ISPRS Archives*, Volume XXXVII Part B3b, pp. 37–42. 491

Lam, T. Y. (2002). Hamilton's quaternions. Technical report, University of California, Berkeley. 333

Lee, G. H., B. Li, M. Pollefeys, and F. Fraundorfer (2013). Minimal Solutions for Pose Estimation of a Multi-Camera System. In *Robotics Research - The 16th International Symposium ISRR, 16-19 December 2013, Singapore*, pp. 521–538. 452

Lee, K.-M., P. Meer, and R.-H. Park (1998). Robust Adaptive Segmentation of Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 200–205. 146

Lemaire, C. (2008). Aspects of the DSM production with high resolution images. In *Int. Archives of Photogrammetry and Remote Sensing*, Volume XXXVII, Part B4, pp. 1143–1146. 751, 765

Lenz, R. and D. Fritsch (1990). Accuracy of Videometry with CCD-sensors. *ISPRS Journal of Photogrammetry and Remote Sensing 45*, 90–110. 508

Leotta, M., P. Moulon, S. Agarwal, F. Dellaert, and V. Rabaud (2015). CVPR tutorial: Open Source Structure-from-Motion. https://midas3.kitware.com/midas/community/46. 645

Lepetit, V., F. Moreno-Noguer, and P. Fua (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision (IJCV) 81*(2), 155–166. 519

Lhuillier, M. (2006). Effective and Generic Structure from Motion Using Angular Error. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 01*, ICPR '06, Washington, DC, USA, pp. 67–70. IEEE Computer Society. 491

Li, H. (2009). Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, pp. 1074–1080. IEEE. 143

Li, H. (2010). Multi-view structure computation without explicitly estimating motion. In *CVPR*, pp. 2777–2784. 637

Li, S. Z. (2000). *Markov random field modeling in computer vision*. Springer. 77

Lindenberger, J. (1993). *Laser-Profilmessungen zur topographischen Geländeaufnahme*. Number 400 in C. Bayerische Akademie der Wissenschaften, München: Deutsche Geodätische Kommission. 751, 752

Lischinski, D. (2007). Structure from Motion: Tomasi-Kanade Factorization. http://www.cs.huji.ac.il/~csip/CSIP2007-sfm.pdf, last visited 1.12.2015. 714

Lüke, H. D. (1999). The Origins of the Sampling Theorem. *IEEE Communications Magazine 37*(4), 106–108. 735

Lourakis, M. I. A. and A. A. Argyros (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software 36*(1), 1–30. 649

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*, 91–110. 491, 679, 680

Luxen, M. and W. Förstner (2001). Optimal Camera Orientation from Points and Straight Lines. In B. Radig and S. Florczyk (Eds.), *Mustererkennung 2001*, Volume 2191 of *LNCS*, pp. 84–91. Proc. 23. DAGM Symposium, Muenchen: Springer. 521

Mardia, K. V. and P. E. Jupp (1999). *Directional Statistics*. Wiley. 368

Martinec, D. and T. Pajdla (2007). Robust Rotation and Translation Estimation in Multiview Reconstruction. In *Conf. on Computer Vision and Pattern Recognistion*. IEEE Computer Society. 713

Massios, N. A. and R. B. Fisher (1998). A Best Next View Selection Algorithm Incorporating a Quality Criterion. In *Proc. British Machine Vision Conference BMVC98*, pp. 780–789. 721

Matas, J. and O. Chum (2005). Randomized RANSAC with Sequential Probability Ratio Test. In *Int. Conf. on Computer Vision*, Volume 2, pp. 1727–1732. 155

McGlone, C. J., E. M. Mikhail, and J. S. Bethel (2004). *Manual of Photogrammetry* (5th ed.). Maryland, USA: American Society of Photogrammetry and Remote Sensing. 81

McGlone, J. C. (2013). *Manual of Photogrammetry* (6th ed.). Maryland, USA: American Society of Photogrammetry and Remote Sensing. 461, 590, 601, 644

Meidow, J., C. Beder, and W. Förstner (2009). Reasoning with Uncertain Points, Straight Lines, and Straight Line Segments in 2D. *International Journal of Photogrammetry and Remote Sensing 64*, 125–139. 359, 369, 376

Meissl, P. (1972). A Theoretical Random Error Propagation Law for Anblock-Networks with Constrained Boundary. *Österreichische Zeitschrift für Vermessungswesen 60*, 61–65. 670

Mendonça, P. R. S., K.-Y. K. Wong, and R. Cipolla (2001). Epipolar Geometry from Profiles Under Circular Motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence 23*, 604–616. 279

Miccoli, S. (2003). Efficient Implementation of a Generalized Cholesky Factorization for Symmetric Galerkin Boundary Element Methods. *Computational Mechanics 32*(4–6), 362–369. 100

Mičušík, B. (2004). *Two-View Geometry of Omnidirectional Cameras*. Ph. D. thesis, Czech Technical University, Centre for Machine Perception. 447

Mikhail, E. M. (1962). Use of Triplets for Analytical Aerotriangulation. *Photogr. Eng. 28*, 625–632. 628

Mikhail, E. M. (1963). Use of Two-Directional Triplets in a Sub-Block Approach for Analytical Aerotriangulation. *Photogr. Eng. 29*, 1014–1024. 628

Mikhail, E. M. and F. Ackermann (1976). *Observations and Least Squares*. University Press of America. 81, 97

Mikhail, E. M., J. S. Bethel, and J. C. McGlone (2001). *Introduction to Modern Photogrammetry*. Wiley. 335

Mirzaei, F. M. and S. I. Roumeliotis (2011). Globally optimal pose estimation from line correspondences. In *IEEE International Conference on Robotics and Automation*. 521

Molenaar, M. (1981). *A further inquiry into the theory of S-transformations and criterion matrices*. New Series 26. Netherlands Geodetic Commission NCG: Publications on Geodesy, Delft, Rijkscommissie voor Geodesie. 109

Montiel, J. M. M. (2006). Unified Inverse Depth Parametrization for Monocular SLAM. In *Proceedings of Robotics: Science and Systems*, pp. 16–19. 257

Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society 26*, 394–395. 101, 779

Moreno-Noguer, F., V. Lepetit, and P. Fua (2007). Accurate Non-Iterative O(n) Solution to the PnP Problem. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 518, 519

Mühlich, M. and R. Mester (1999). Subspace methods and equilibration in computer vision. In *Proc. Scandinavian Conference on Image Analysis*. 161, 185

Mulawa, D. (1989). *Estimation and Photogrammetric Treatment of Linear Features*. Ph. D. thesis, Purdue University, West Lafayette, IN, USA. 216

Mundy, J. L. and A. P. Zisserman (1992). *Geometric Invariance in Computer Vision*. MIT Press. 266

Nayar, S. K. (1997). Catadioptric Omnidirectional Camera. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, pp. 482–488. IEEE Computer Society. 488

Nayar, S. K. (2006). Computational Cameras: Redefining the Image. *IEEE Computer Magazine, Special Issue on Computational Photography 39*(8), 30–38. 444

Neyman, J. and E. S. Pearson (1933). On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Phil. Trans. of the Royal Society, Series A 231*, 289–337. 61

Niini, I. (2000). *Photogrammetric Block Adjustment Based on Singular Correlation*. Ph. D. thesis, Helsinki University of Technology, Espoo, Finland. 564

Nistér, D. (2003). An efficient solution to the five-point relative pose problem. In *CVPR '03, Madison, Wisconsin, USA*, Volume II, pp. 195–202. 575, 613

Nistér, D. and F. Schaffalitzky (2006). Four points in two or three calibrated views: Theory and practice. *International Journal of Computer Vision 67*(2), 211–231. 636

Nocedal, J. and S. J. Wright (1999). *Numerical Optimization* (2nd ed.). New York: Springer. 107

Nurutdinova, I. and A. Fitzgibbon (2015). Towards Pointless Structure from Motion: 3D Reconstruction and Camera Parameters from General 3D Curves. In *IEEE Proc. International Conference on Computer Vision*. 570

Ochoa, B. and S. Belongie (2006). Covariance Propagation for Guided Matching. In *3rd Workshop on Statistical Methods in Multi-Image and Video Processing*. CD-ROM. 369

Oishi, T., R. Kurazume, A. Nakazawa, and K. Ikeuchi (2005). Fast simultaneous alignment of multiple range images using index images. In *3DIM05*, pp. 476–483. 649

Oppenheim, A. V. and R. W. Schafer (1975). *Digital Signal Processing*. Prentice Hall. 744

Papoulis, A. (1965). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill. 21, 30, 33

Papoulis, A. and S. U. Pillai (2002). *Probability, Random Variables and Stochastic Processes* (4th ed.). McGraw-Hill. 21, 24, 40, 41, 48

Patel, H. (2002). *Solving the Indefinite Least Squares Problems*. Ph. D. thesis, Faculty of Science and Engineering, Manchester. 776

Pennec, X. (2006). Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *J. Math. Imaging Vis. 25*(1), 127–154. 383

Pennec, X. and J.-P. Thirion (1997). A Framework for Uncertainty and Validation of 3-D Registration Methods based on Points and Frames. *Int. Journal of Computer Vision 25*, 203–229. 383

Penrose, R. (1954). A Generalized Inverse for Matrices. *Proceedings of the Cambridge Philosophical Society 51*, 406–413. 101, 779

Pepić, S. H. (2010). Weighted Moore-Penrose Inverse: PHP vs. Mathematica. *Ser. Math. Infrom. 25*, 35–45. 780

Perwass, C. (2009). *Geometric Algebra with Applications in Engineering*, Volume 4 of *Geometry and Computing*. Springer. 236

Peternell, M. and H. Pottmann (2001). Approximation in the space of planes — Applications to geometric modeling and reverse engineering. Technical Report TR 87, Institute of Geometry, Vienna Univ. of Technology. 374

Petersen, K. B. and M. S. Pedersen (2012). The Matrix Cookbook. Version 20121115. 769, 783

Philip, J. S. (1997). An algorithm for determining the position of a circle in 3D from its perspective 2D projection. Technical Report TRITA-MAT-1997-MA-1, Department of Mathematics, Royal Institute of Technology, Stockholm. 534, 536

Piegl, L. and W. Tiller (1997). *The Nurbs Book* (2nd ed.). Springer. 484, 736

Pless, R. (2003). Using Many Cameras as One. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 587–593. 446

Pollefeys, M., R. Koch, and L. Van Gool (1999). A simple and efficient rectification method for general motion. In *IEEE Proc. International Conference on Computer Vision*, Volume 1, pp. 496–501. 566

Pottmann, H. and J. Wallner (2010). *Computational Line Geometry*. Springer. 227, 245, 284

Pratt, V. (1987). Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph. 21*(4), 145–152. 182

Quan, L. and Z. Lan (1999). Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell. 21*(8), 774–780. 518

Raguram, R., O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm (2013). USAC: A Universal Framework for Random Sample Consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*(8), 2022–2038. 153, 157

Raguram, R., J.-M. Frahm, and M. Pollefeys (2009). Exploiting uncertainty in random sample consensus. In *International Conference on Conmputer Vision*, pp. 2074–2081. IEEE Computer Society. 154, 588

Rao, R. C. (1967). Least squares theory using an estimated dispersion matrix and its application to measurement of signals. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*, pp. 355–372. Univ. of Calif. Press. Lemma 5a. 138, 492, 702

Rao, R. C. (1973). *Linear Statistical Inference and Its Applications*. New York: Wiley. 81, 86, 118

Rasmussen, C. E. and C. K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, USA: The MIT Press. 50

Reich, M. and C. Heipke (2014). A Global Approach for Image Orientation Using Lie Algebraic Rotation Averaging and Convex L_∞ Minimization. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 40. 714

Reich, M., J. Unger, F. Rottensteiner, and C. Heipke (2013). On-Line Compatible Orientation of a Micro-UAV Based on Image Triplets. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II*(2), 37–42. 708

Reisert, M. and H. Burkhardt (2007). Learning equivariant functions with matrix valued kernels. *J. Mach. Learn. Res. 8*, 385–408. 267

Ressl, C. (2003). *Geometry, Constraints and Computation of the Trifocal Tensor*. Ph. D. thesis, Universität Wien, Institut für Photogrammtrie und Fernerkundung. 258, 284, 603, 625, 629, 631, 632, 636, 639

Rhudy, M., Y. Gu, J. Gross, and M. R. Napolitano (2011). Evaluation of Matrix Square Root Operations for UKF within a UAV GPS/INS Sensor Fusion Application. *International Journal of Navigation and Observation ID 416828*, 11 p. 47

Rinner, K. (1963). Studien über eine allgemeine, voraussetzungsfreie Lösung des Folgebildanschlusses. ÖZfV, Sonderheft 23. 556, 557

Rodriguez, O. (1840). Des lois géometriques qui regissent les deplacements d'un système solide independament des causes qui peuvent les produire. *Journal de mathématiques pures et appliquées 1*(5), 380–440. 327, 335

Rosebrock, D. and F. Wahl (2012). Complete generic camera calibration and modeling using spline surfaces. In *11th Asian Conference on Computer Vision*, LNCS 7724–7727. Springer. 512

Rother, C. and S. Carlsson (2001). Linear Multi View Reconstruction and Camera Recovery. In *Proceedings of 8th ICCV*, Vancouver, Canada. 713

Rother, C. and S. Carlsson (2002). Linear Multi View Reconstruction with Missing Data. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen (Eds.), *ECCV (2)*, Volume 2351 of *Lecture Notes in Computer Science*, pp. 309–324. Springer. 713

Rousseeuw, P. J. and A. M. Leroy (1987). *Robust Regression and Outlier Detection*. New York: Wiley. 143, 146

Scaramuzza, D. (2008). *Omnidirectional Vision: From Calibration to Robot Motion Estimation,*. Ph. D. thesis, ETH Zürich. 487, 507, 686, 687

Schaffalitzky, F. and A. Zisserman (2000). Planar Grouping for Automatic Detection of Vanishing Lines and Points. *Image and Vision Computing 18*, 647–658. 271

Schaffrin, B. and K. Snow (2010). Total Least-Squares Regularization of Tykhonov Type and an Ancient Racetrack in Corinth. *Linear Algebra and its Applications 432*, 2061–2076. 161, 185

Scherer-Negenborn, N. and R. Schaefer (2010). Model Fitting with Sufficient Random Sample Coverage. *International Journal on Computer Vision 89*(1), 120–128. 156

Schewe, H. (1988). Automatische photogrammetrische Erfassung von Industrieoberflächen. Technical report, Inpho GmbH, Stuttgart. 5, 6

Schilcher, M. (1980). *Empirisch-statistische Untersuchungen zur Genauigkeitsstruktur des photogrammetrischen Luftbildes*. Number 262 in C. Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissensch., München. 505

Schmid, H. H. (1958). Eine allgemeine analytische Lösung für die Aufgabe der Photogrammetrie. *Bildmessung und Luftbildwesen 26/27*, 103–113, 1959: 1–12. 644, 649, 676

Schneider, J. and W. Förstner (2013). Bundle adjustment and system calibration with points at infinity for omnidirectional camera systems. *Zeitschrift für Photogrammetrie, Fernerkundung und Geoinformation 4*, 309–321. 467, 488, 687

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics 6*(2), 461–464. 139

Schwidefsky, K. and F. Ackermann (1976). *Photogrammetrie*. Stuttgart: Teubner. 605

Sebbar, A. and A. Sebbar (2012). Equivariant functions and integrals of elliptic functions. *Geometriae Dedicata 160*(1), 373–414. 267

Semple, J. G. and G. T. Kneebone (1998). *Algebraic Projective Geometry*. Oxford Classic Texts in the Physical Sciences. Oxford University Press. 564

Sester, M. and W. Förstner (1989). Object Location Based on Uncertain Models. In H. Burkhardt, K.-H. Höhne, and B. Neumann (Eds.), *Mustererkennung 1989*, Volume 219 of *Informatik Fachberichte*. 11. DAGM Symposium, Hamburg: Springer. 158

Shannon, C. E. and W. Weaver (1949). *The Mathematical Theory of Communication*. Urbana, Illinois: The University of Illinois Press. 735

Sinha, S. N. and M. Pollefeys (2010). Camera Network Calibration and Synchronization from Silhouettes in Archived Video. *Int. J. Comput. Vision 87*(3), 266–283. 635

Slama, C. C. (1980). *Manual of Photogrammetry*. American Society of Photogrammetry. 457

Smith, R., M. Self, and P. Cheeseman (1991). A Stochastic Map for Uncertain Spatial Relationships. In S. S. Iyengar and A. Elfes (Eds.), *Autonomous Mobile Robots: Perception, Mapping, and Navigation (Vol. 1)*, pp. 323–330. Los Alamitos, CA: IEEE Computer Society Press. 110

Snay, R. A. (1976). Reducing the Profile of Sparse Symmetric Matrices. *Bull. Geod. 50*, 341–352. 662

Sokolnikov, I. S. (1956). *Mathematical Theory of Elasticity*. McGraw-Hill. 256

Stefanovic, P. (1973). Relative Orientation – A New Approach. *ITC Journal 3*, 417–448. 557

Stefanovic, P. (1978). Blunders and Least Squares. *ITC Journal 1*, 122–157. 129

Steger, C. (2012). Estimating the fundamental matrix under pure translation and radial distortion. *Photogrammetry and Remote Sensing 74*(1), 202–217. 508

Steinke, N. S. (2012). Simultaneous Localization and Mapping (SLAM) mittels einer Microsoft Kinect. Master's thesis, Freie Universität Berlin, Fachbereich Mathematik und Informatik, Lehrstuhl für Künstliche Intelligenz. 6

Stewart, J. E. (1996). *Optical principles and technology for engineers*. Marcel Dekker. 379

Stewénius, H. (2005). *Gröbner basis methods for minimal problems in computer vision*. Ph. D. thesis, Lund Inst. of Technology, Centre for Math. Sciences. 575

Stewénius, H., C. Engels, and D. Nistér (2006). Recent developments on direct relative orientation. *International Journal of Photogrammetry and Remote Sensing 60*, 284–294. 575, 577

Stewénius, H., D. Nistér, M. Oskarsson, and K. Åstrom (2005). Solutions to minimal generalized relative pose problems. https://www.inf.ethz.ch/personal/pomarc/pubs/sm26gen.pdf, last visited 1.12.2015. 581

Stocker, J. and Schmid (1966). *A. Dürer: Underweysung der Messung, mit dem Zirckel und Richtscheyt, in Linien, Ebenen und ganzen Corporen, Nürnberg 1525, Faksimile Nachdruck*. Dietikon. 457

Stockman, G. C. (1987). Object Recognition and Localization via Pose Clustering. *Computer Vision, Graphics and Image Processing 40*(3), 361–387. 158

Stolfi, J. (1991). *Oriented Projective Geometry: A Framework for Geometric Computations*. San Diego: Academic Press. 245, 344

Strecha, C., W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen (2008). On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*. 727

Stuelpnagel, J. (1964). On the Parametrization of the Three-Dimensional Rotation Group. *SIAM-Review 6*(4), 422–430. 333

Sturm, P., S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto (2011). Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Foundations and Trends in Computer Graphics and Vision 6*(1-2), 1–183. 488

Sturm, P. and B. Triggs (1996). A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In B. Buxton and R. Cipolla (Eds.), *Computer Vision–ECCV'96, Vol. II*, Volume 1065 of *LNCS*, pp. 709–720. 715

Sugihara, K. (1986). *Machine Interpretation of Line Drawings*. MIT Press, Cambridge, MA. 523, 534

Swaminathan, R. and S. K. Nayar (2000). Nonmetric Calibration of Wide-Angle Lenses and Polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(10), 1172–1178. 445

Sweeney, C., V. Fragoso, T. Höllerer, and M. Turk (2014). gDLS: A Scalable Solution to the Generalized Pose and Scale Problem. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, 2014, Proceedings, Part IV*, pp. 16–31. 452

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer. 727

Takahashi, K., J. Fagan, and M.-S. Chen (1973). Formation of a sparse bus impedance matrix and its application to short circuit study. In *IEEE Power Engineering Society*, Volume 7. 86, 663

Tang, R., D. Fritsch, and M. Cramer (2012). New rigorous and flexible Fourier self-calibration models for airborne camera calibration. *International Journal on Photogrammetry and Remote Sensing 71*(1), 76–85. 512, 684

Taubin, G. (1991). Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*(11), 1115–1138. 179

Taubin, G. (1993). An Improved Algorithm for Algebraic Curve and Surface Fitting. In *Fourth International Conference on Computer Vision, Berlin*, pp. 658–665. 179

Teller, S. and M. Hohmeyer (1999). Determining the lines through four lines. *ACM Journal of Graphics Tools 4*(3), 11–22. 303

Terzopoulos, D. (1984). *Multiresolution computation of visible-surface representations*. Ph. D. thesis, MIT. 733, 739

Terzopoulos, D. (1986). Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE T-PAMI 8*(4), 413–423. 739

Tomasi, C. and T. Kanade (1992). Shape and Motion from Image Streams under Orthography: A Factorization Method. *IJCV 9*, 137–154. 714, 715

Tordoff, B. and D. W. Murray (2002). Guided Sampling and Consensus for Motion Estimation. In *Computer Vision–ECCV'02, Vol. I*, Volume 2350 of *LNCS*, pp. 82–98. Proc. 7th European Conf. on Computer Vision, Copenhagen: Springer. 156

Torr, P. H. S. and A. Zisserman (1997). Robust Parameterization and Computation of the Trifocal Tensor. *Image and Vision Computing 15*(8), 591–605. 636

Torr, P. H. S. and A. Zisserman (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78*, 138–156. 155

Triggs, B. (1996). Factorization Methods for Projective Structure and Motion. In *International Conference on Computer Vision & Pattern Recognition (CVPR '96)*, San Francisco, USA, pp. 845–851. IEEE Computer Society. 715

Triggs, B., P. McLauchlan, R. Hartley, and A. Fitzgibbon (2000). Bundle Adjustment – A Modern Synthesis. In B. Triggs, A. Zisserman, and R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice*, Volume 1883

of *LNCS*, pp. 298–372. Proc. of the Intl. Workshop on Vision Algorithms: Theory and Practice: Springer. 359, 649, 663

Tsai, R. Y., T. H. Huang, and W. Zhu (1982). Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch, II: Singular Value Decomposition. *IEEE Transactions on Acoustics, Speech and Signal Processing 30*(4), 525–534. 578

Tschupik, J. P. and F. Hohenberg (1972). Die geometrischen Grundlagen der Photogrammetrie. In W. Jordan, O. Eggert, and M. Kneissl (Eds.), *Handbuch der Vermessungskunde*, Volume II a/3, Chapter IV, pp. 2223–2295. Stuttgart: Metzlersche Verlagsbuchhandlung. 564, 571

Turing, A. M. (1939). Systems of logic based on ordinals. *Proc. London Math. Soc. 2*, 161–228. 142

Uhlmann, J. K. (1995). *Dynamic map building and localization: new theoretical foundations*. Ph. D. thesis, University of Oxford. 98

Vanhatalo, J. and A. Vehtari (2008). Modelling local and global phenomena with sparse Gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*, pp. 571–578. 86, 663

Varley, P. A. C. and R. R. Martin (2002). Estimating Depth from Line Drawing. In *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, SMA '02, New York, NY, USA, pp. 180–191. ACM. 523

Vaseghi, S. V. (2000). *Advanced Digital Signal Processing and Noise Reduction*. Wiley. 77

Ventura, J., C. Arth, G. Reitmayr, and D. Schmalstieg (2014, 06/2014). A Minimal Solution to the Generalized Pose-and-Scale Problem. In *CVPR*. 452

Vinicius, M., A. Andrade, and J. Stolfi (2001). Exact Algorithms for Circles on the Sphere. *Intl. J. of Computational Geometry and Applications 11*(3), 267–290. 344

Vo, M., S. G. Narasimhan, and Y. Sheikh (2016). Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction. In *Proc. of Conference on Computer Vision and Pattern Recognition*. 645, 675

von Gruber, O. (1938). Kniffe und Pfiffe bei der Bildorientierung in Stereoauswertegeräten. *Bildmessung und Luftbildwesen 13*, 17–26, 73–77. 590

von Sanden, H. (1908). *Die Bestimmung der Kernpunkte in der Photogrammetrie*. Ph. D. thesis, Universität Göttingen. 571

Vosselman, G. and W. Förstner (1988). The Precision of a Digital Camera. In *Intl. Archives of Photogrammetry*, Volume 27, Part B1, pp. 148–157. Proc. XVIth ISPRS Congress, Comm. III, Kyoto. 698

Walker, G. (1931). On Periodicity in Series of Related Terms. *Proceedings of the Royal Society of London, Ser. A 131*, 518–532. 184

Watzlawick, P. (1978). *Wie wirklich ist die Wirklichkeit?* Piper. 11

Weber, M. (2003a). Quadric through three lines (in German). Personal communication. 303

Weber, M. (2003b). Rotation between two vectors. Personal communication, Bonn. 340

Weidner, U. (1994). Parameterfree Information-Preserving Surface Restauration. In J. O. Eklundh (Ed.), *Computer Vision–ECCV '94 vol. II*, Volume 801 of *LNCS*, pp. 218–224. Proc. 3rd European Conf. on Computer Vision, Stockholm: Springer. 762

Weinberg, S. L. and S. K. Abramowitz (2006). *Statistics Using SPSS: An Integrative Approach*. Cambridge University Press. 115

Werner, T. and T. Pajdla (2001). Oriented matching constraints. In T. Cootes and C. Taylor (Eds.), *British Machine Vision Conference 2001*, London, UK, pp. 441–450. British Machine Vision Association. 606, 627

Whittaker, J. M. (1915). On the functions which are represented by the expansions of the interpolation-theory. *Proc. Roy. Soc. 35*, 181–194. 735

Wikipedia (2015). Matrix Calculus. `http://en.wikipedia.org/wiki/Matrix_calculus`, last visited 1.12.2015. 84

Willson, R. G. and S. A. Shafer (1994). What is the centre of the image? *Journal of the Optical Society of America 11*, 2946–2955. 462

Wirth, N. (1978). *Algorithms + Data Structures = Programs*. Upper Saddle River, NJ, USA: Prentice Hall. 146

Wolff, K. and W. Förstner (2000). Exploiting the Multi View Geometry for Automatic Surfaces Reconstruction Using Feature Based Matching in Multi Media Photogrammetry. In *Intl. Archives of Photogrammetry and Remote Sensing*, Volume XXXIII, Part B 5/2, Amsterdam, pp. 900–907. Proc. XIXth ISPRS Congress, Amsterdam. 603

Wood, S. N. (2003). Thin plate regression splines. *J. R. Statist. Soc. B 65, Part 1*, 95–114. 758

Wrobel, B. P. (2001). Minimum Solutions for Orientation. In A. Gruen and T. S. Huang (Eds.), *Calibration and Orientation of Cameras in Computer Vision*, Volume 34 of *Springer Series in Information Sciences*, pp. 7–62. Berlin/Heidelberg: Springer. 495, 521

Wrobel, B. P. (2012). Kreismarken in perspektiver Abbildung – im Bild und im Bündelblock. *PFG Photogrammetrie, Fernerkundung, Geoinformation 3*, 221–236. 534, 535

Wunderlich, W. (1982). Rechnerische Rekonstruktion eines ebenen Objektes aus zwei Photographien. In *Mitteilungen des Geodätischen Instituts*, Volume 40, pp. 265–377. Technischen Universität Graz. 578

Yang, J., H. Li, and Y. Jia (2014). Optimal Essential Matrix Estimation via Inlier-Set Maximization. In *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pp. 111–126. 143

Yule, G. U. (1927). On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers. *Philosophical Transactions of the Royal Society of London, Ser. A 226*, 267–298. 184

Zach, C. (2014). Robust Bundle Adjustment Revisited. In *Proc. of Int. Conference on Computer Vision*. 150

Zeisl, B., P. F. Georgel, F. Schweiger, E. Steinbach, and N. Navab (2009). Estimation of location uncertainty for scale invariant feature points. In *Proc. BMVC*, pp. 57.1–57.12. doi:10.5244/C.23.57. 491

Zheng, Y., Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi (2013). Revisiting the PnP Problem: A Fast, General and Optimal Solution. In *Int. Conference on Computer Vision*, pp. 2344–2351. 518

Index

C^n-continuous, 738
L_1-norm minimization, 150
    ML-type estimation as L_1., 148
L_{12}-norm minimization, **148**, 755, 764
S^1, 214, 215
S^2, 199, 200, 242, 243
S^3, 242, 333
S^5, 243
$\mathsf{\Pi}$, **300**
$\mathsf{\Pi}(\mathbf{x})$, independent rows and columns, 319
$\overline{\mathsf{\Pi}}$, **301**
$\mathsf{\Gamma}$, 219
$\mathsf{\Gamma}(\mathbf{x})$, independent rows, 319
$\overline{\mathsf{\Gamma}}$, 233
Ω, **84**
χ distribution, 34
χ^2-square distribution, 33
δ_0, 131
    for multi-dimensional test, 68, 130
    for one-dimensional test, 66, 128
$\ell(\ell'', \ell''')$, **625**
D, 227
$\mathbb{P}$, 231
$\mathbb{P}^0$, 215, 216, 231
$\mathbb{P}^1$, 214
$\mathbb{P}^2$, 200, 203, **206**, 207, 211, 215, 231
$\mathbb{P}^3$, **210**, 231
$\mathbb{P}^5$, 231
$\mathbb{P}^n$, **215**
$\mathbb{P}^{*2}$, **209**, 209
$\mathbb{P}^{*3}$, **212**, 231
$\mathbb{T}^n$, **345**
$\mathbb{T}^{*2}$, 346
$\mathbb{T}^{*3}$, 347
$\mathsf{S}(\mathbf{x})$, independent rows, 318
ρ-function, 144
    table of ρ., 149
$\wp_2(\mathcal{X}', \ell'''))$, **631**
$\wp_3(\mathcal{X}', \ell''))$, **631**
x-parallax, 566, **589**, 601–604
y-parallax, 561, 566, **589**, 590, 592, 602
$2^1/_2$D surface, 729
1D homography, 257
    fixed entities of h., 276
1D point, 214
    at infinity, 214
    homogeneous coordinates of p., 214
2D affinity, 252
    fixed line of a., 274
2D autocollineation, perspective a., 277
2D block adjustment
    free b., 663
    functional model of b., 651
    gauge constraints in b., 665
    gauge transformation in b., 668
    inner precision of b., 666
    mathematical model of b., 651
    sparsity of matrices, 655
    stochastical model of b., 652
    theoretical quality of b., 670–674
2D circle, 237
2D homography, 250, **253**
    algebraic solution for h., 389
    closed form solution of h., 406
    degrees of freedom of h., 321
    fixed entities of h., 274
    from uncertain point pairs, 425
    minimal parametrization of h., 385
    minimal solution of h., 321
    orientedness of transformed entities, 355
    uncertainty of points mapped with h., 387
2D line, 207
    at infinity, **207**
    centroid representation of l., 374
    direction of l., 294, 348
    distinct points of l., 318
    from two points, 292
    from uncertain centroid to Hessian form of l., 374
    Hessian normal form of l., **207**, 374
    Hessian normal form of uncertain l., 374
    homogeneous coordinates of l., 207
    null space of covariance matrix of l., 375, 392
    optimal closed form solution for l., 397
    oriented l., 346
    Plücker coordinates of l., 223
    point at infinity of l., 209, 294
    point-direction form of l., 209
    through two points, 221
    transformation of l., 258
    uncertain Hessian form to homogeneous coordinates of l., 375
    uncertain Hessian parameters of l., 376
    uncertain homogeneous coordinates to Hessian form, 377
    uncertain l., 373–377
2D model block adjustment, 650–674
2D motion, 251
    fixed entities of m, 274
2D point, 206
    at infinity, 206
    closest to conic, 295
    covariance matrix of spherically normalized p., 393
    degrees of freedom of p., 207
    direct least squares solution of intersection p., 401
    direct solution for intersection p., 401
    distinct lines of p., 318
    dual of p., 204
    homogeneous coordinates of p., 206
    null space of covariance matrix of p., 368, 393
    oriented p., 345
    Plücker coordinates of p., 223
    reduced coordinates of p., 370
    sign of intersection p., 353
    spherical normalization of uncertain p., 368
    uncertain p., 366–372
2D rotation, 251
    fixed entities of r., 274
2D scaling, 251
2D shear, 252
2D similarity, 252
2D translation, 251
    fixed entities of t., 274
3D affinity, 255
3D autocollineation, 280
3D circle, 241
3D conics, 241

© Springer International Publishing Switzerland 2016

W. Förstner and B.P. Wrobel, *Photogrammetric Computer Vision*, Geometry and Computing 11,
DOI 10.1007/978-3-319-11550-4

3D homography, 256
  closed form solution of h., 406
  fixed entities of h., 275
  minimal solution of h., 322
3D line
  and spatial triangle, 351
  approximating 6-vector of l., 381
  at infinity, 219
  coplanarity of two l., 304
  covariance matrix of reduced coordinates of l., 381
  degrees of freedom of l., 216, 227
  direct solution of l., 412
  directed l., 606
  direction of l., 348, 353
  distinct planes of l., 319
  distinct points of l., 319
  dual l., 233
  from two planes, 220
  moment vector of l., **218**, 219, 220, 227
  oriented l., 348
  parameters of l., 216
  Plücker coordinates of l., **218**, 226
  point at infinity of l., 220
  point-direction form of l., 220
  projection, 480
  reconstruction from two images, 605
  reduced coordinates of l., 380–381
  spherical projection of l., 481
  through four 3D lines, 302
  through two points, 300
  transformation, 259
  two-point form of l., 220
  uncertain l., 379–381
3D model block adjustment, 649
3D motion, 255
  fixed entities of m., 276
3D point, 210
  at infinity, 210
  degrees of freedom of p., 211
  direct least squares solution of intersection p., 402
  direct solution for line intersection p., 401
  distinct planes of p., 319
  from several images, 602
  from two rays, normal case of image pair, 601
  homogeneous coordinates of p., 210
  oriented p., 346
  Plücker coordinates of p., 225
  quality of p., 526
  triangulation of p. from several images, 602
  uncertain p., 372–373
3D points, collinearity of p., 306
3D rotation, *see* rotation
3D similarity, 255
3D translation, 255
  fixed entities of t., 275

a posteriori probability density, 77
a priori probability density, 76
absolute orientation, 549, **552**, 607, 609
  direct solution of a. with three points, 515
  of calibrated cameras, 552
  of uncalibrated cameras, 552
  redundancy of a., 613
  within two-step procedure, 612
absolute points, 241
acceptability
  of a covariance matrix, 120
  of bundle adjustment result, 688
  of configuration, 495, 516
  of precision, 117, 120
accuracy, **116**, 490
  empirical a., 117
  identification a., 117, **490**, 706
  of the mean, 116
adaptive least Kth order squares, 146
additional parameters, 123, 464
  choice of a., 509, 684
  elimination, 695
  evaluation, 693
  evaluation of a., 699
adjacency graph, 654, 660
adjugate matrix, 770
adjustment, *see* estimation
adjustment, block a., 643
ADS 80, Leica, 443, 445, 446
aerotriangulation, 707, **718**, 721
affinity
  2D a., 252
  3D a., 255
  chirality of transformed configurations, 357
  minimal solution of a., 320
  sign of entities transformed with a., 357
algebraic solution, 178
  covariance matrix of a., 180
  for 2D homography, 389
  for 2D line, 396
  for 2D line intersection, 401
  for 3D line intersection, 401
  for plane, 396
  for projection matrix, 494
  with eigenvalue decomposition, 179
  with SVD, 179
algorithm
  $\mathsf{K}, R, \boldsymbol{Z}$ from P, 500
  $\mathbf{b}, R$ from essential matrix, 583
  3D circle from its image, 536
  direct LSE 2D line from points, 401
  direct LSE 3D line from points, 414
  direct LSE 3D similarity from point pairs, 411
  direct LSE mean axis, 405
  direct LSE mean direction, 404
  direct LSE point from lines, 403
  direct LSE rotation from point pairs, 408
  DLT for projection matrix, 496
  Gauss–Helmert model with reduced coordinates, 416
  Gauss–Markov model with constraints, 108
  Gauss–Markov model, linear, 91
  homography from point pairs, 389
  model with constraints between the observations only, 171
  optimal P from image points, 498
  RANSAC, 156
  reweighing constraints, 169
  robust a. for Gauss–Helmert model with constraints, 168
  sequential similarity transformations, 710
  sequential spatial resection, 709
  triangulation, 600
algorithmic complexity, 452
analysis model, 7, 448
angle
  between 2D direction and coordinate axes, 206
  between 3D direction and coordinate axes, 210
  between two 2D lines, 298
  direction a, 298
  parallactic a., 420, 421
  tilt a. of plane, 212
  zenith a., 210
antiparallel, 343
antipodal
  line, 348
  plane, 347
  point, 344–346
approximate residuals, 164
approximate values, 145, 452
  for bundle adjustment, 707–715
  for bundle adjustment, direct solutions, 711
  for bundle adjustment, sequential solutions, 708
  for relative orientation, 583
  for relative orientation, normal case, 589
AR-process
  AR(P), 52
  AR(1), 53
  AR(2), 53
  for modelling profiles, 748
  integrated A., 54, 752
  observed A., 749
area camera, 444
area of triangle, 222
astigmatism, 379
asymmetric weight function, 756
attitude of camera, 456
autocollineation, 248, 256
  2D a., 277
  fixed elements of a., 272
  perspective a., **277**
automatic image matching, 563
autoregressive process, *see* AR process
axis and angle
  from rotation matrix, 331
axis, closed form solution of mean a., 405

back projection, 482

bandwith of matrix, 662
barycentric coordinates, 213, 349
base line, 551, 563
base vector, 553, 578, 582
  estimation from 2 points, given rotation, 578
base-to-height ratio, 604
basis functions
  monomials as b., 733
  of collocation, 735
  radial b., 735
  splines as b., 736
  trigonometric b., 734
Bayer pattern, 444
Bayesian estimation, 76–78
  in Gauss–Markov model, 93
  surface reconstruction as B., 742
Bayesian factor, 64
Bayesian information criterion, 139, 686
Bertrand's paradox, 22
best linear unbiased estimate, 79
best unbiased estimate, 79
bi-partite graph, 646
bias, **116**
  induced by linearization, 44
  of estimated variance factor, 137
  of estimates, 79, 141
  of normalization, 45
  of product, 44
  of the mean, 45
  of the variance, 45
  of variance of image points and lines, 493
bicubic interpolation, 738
bilinear interpolation, 738
bilinearity of coplanarity constraint, 553
binomial distribution, 28
bivector, 235
block, 643
block adjustment, 643
  2D model b., 651
  adjacency graph in b., 660
  bundle adjustment as b., 648
  free b., 663
  linear planar b., 711–714
  mathematical model of b., 647
  model b., 645
block matrix, inverse of b., 769
blunder, *see* outlier
boundary of region, 442
break down point, 145
bundle adjustment, 450, 609, **648**
  acceptability of accuracy of b., 692
  acceptability of design of b., 691
  acceptance criteria for results of b., 688
  approximate values, 707–715
  as block adjustment, 648
  Cramer–Rao bound of b., 683
  empirical accuracy of b., 682
  Euclidean b., 649
  evaluation of b., 687
  factorization method for b., 714
  for camera calibration, 696
  for image pair, 610
  for image triplet, 638
  for relative orientation, 585
  functional model of b., 675
  gauge constraints in b., 665
  linear estimation of rotations in b., 713
  linearized model of b., 676
  MAP estimate for b., 648
  nonlinear model of b., 675
  outlier detection, 707–715
  projective, 676
  projective b., 649
  redundancy of image pair, 611
  self-calibrating b., 450, 674–696
  sensitivity of b., 701
  spherical camera b., 686
  variance component estimation in b., 679
  view planning for b., 715–722
  with lines, 676
  with points, 610, 649
bundle of rays, 461, 559

calibrated camera, **460**, 555–557, 607, 622
  for the image pair, 556
calibration
  laboratory c., 697
  of camera, 449
  self-c., 697
  stability of c., 702
  test field c., 697
  with unknown testfield, 698
calibration matrix, 471
  differential c., 501
  from projection matrix, 499
camera, 460
  absolute orientation of calibrated c., 552
  absolute orientation of uncalibrated c., 552
  affine c., **464**
  area c., 444
  calibrated c., 452, **460**, 607, 622
  calibrated perspective c., 555–557
  calibration, 449
  calibration, bundle adjustment for c., 696
  calibration, laboratory c., 697
  calibration, test field c., 697
  catadioptric c., 487
  central c., **446**, 456, 465, 622
  computational c., 444
  coordinate system, **463**, 465, 602
  coplanarity constraint for images of calibrated c., 555–557
  coplanarity constraint for images of spherical c., 556–557
  digital c., 465
  distortions of perspective c., 505
  essential matrix of normalized c. , 559
  Euclidean c., **464**
  fish-eye c., 485
  generic c., **446**, 460
  geometric c. model, 443
  ideal c., **465**, 561
  ideal perspective c., 468
  ideal unit c., 465
  line c., 444
  matrix, 281
  metric c., **460**, 460
  model, 441, **445**, 479
  model of central c., 468
  model of perspective c., 462, 464, 470
  model of real c., 461
  model of spherical c., 462, 468
  moving c., 568
  normalized c., **465**, 472, 713
  orientation, 449
  partially calibrated c., **460**
  perspective c., 248, **446**, 456, 460, 464, 607
  perspective c. for the image pair, 550
  perspective c. for the image triplet, 622
  perspective c. with distortions, **464**
  pinhole c., 253, 281, 464, 465
  planes, 554
  point c., 444
  pose, 456
  principal planes of c., 474
  ray, 469
  real c., 456
  relative orientation, 552
  spherical, 582
  spherical c., **446**
  spherical c. for the image pair, 555, 556
  spherical c. for the image triplet, 622
  stellar calibration of perspective c., 496
  systems, 488
  uncalibrated, **461**
  uncalibrated c., 452, 490, 550, 622
  with affine sensor, 470
  with distortion, 462
  with Euclidean sensor, 531
Canon PowerShot A630, 443, 445
catadioptric camera, 487
catadioptric optics, 446
caustic, 445, 446
Cayley representation of rotation, 336
central camera, **446**, 456, 465
  model of, 468
central limit theorem, 30
central projection, **456**, 467, 481, 485, 487, 490
centroid representation, 490, 492
  of 2D line, 374
  of plane, 377
Chasles' theorem, 272
check of linearization, 104
check points, 683, 700
checkability, 117, 453
  of coordinates in absolute orientation, 411
  of parameters, 133

checking the implementation, 139
chirality, 357
  effect of affinity, 357
  effect of homography on c., 356
  of 2D line and point, 349
  of four 3D points, 350
  of plane and point, 350
  of three 2D points, 349
  of two 3D lines, 350, 355
choice of additional parameters, 509, 684
Cholesky decomposition, 86, 661, 776
circle, 237
  2D, 237
  3D, 241
circle fitting, 177
close range applications, 452
closed form estimation, 176–183
closed form solution, *see* direct solution
clustering, 157
cofactor matrix, 369, 556, **769**
  as dual transformation matrix, 259
  for line conic, 239
  for plane quadric, 240
  for polarity, 238
  for transformation of hyperplanes, 258
collinearity
  equations, 470
  of 3D points, 306, 602
  of projection centres, 622
  of three 2D points, 296
  projective c. equations, 472
collineation, 247, **277**
  1D c., 257
  perspective c., 248, 277, 278
  projective c., 248
collocation, 174–176
  basis functions of c., 735
colour image, 569
complanarity, *see* coplanarity
complete search, 151
complete space U, 231
complex numbers, 651
computational camera, 444
concatenation
  of displacements, 262
  of homographies, 261
  of transformations, 261
concurrence
  of planes, 307
  of three 2D lines, 296
condition number, **118**, 286, 537, 574, 659
conditional probability, 23
conditioning, **286**
  effect of c. on normal equations, 657
  of point coordinates, 321, 465, 494, 571, 603, 606
  of projection matrix, 537
confidence
  ellipse, 32, 369
  hyperbola, 374
conformal geometric algebra, 236
conic, 236
  central form of c., 237
  closed form solution for c., 182
  dual c., 239
  general form of c., 236
  orientation of c., 348
  parametric form of c., 237
  point closest to c., 295
  point of symmetry of c., 237
  tangent at c., 238
  transformation of c., 260
conjugate
  rotation, 281–282, 321
  transformation, 278
  translation, 279
consensus set maximization, 143, 157
constraints
  between corresponding image features, 451
  coplanarity c. for image of calibrated cameras, 555
  coplanarity c. for images from uncalibrated cameras, 552
  coplanarity c. for normal case, 562
  crisp c., 96
  epipolar c. for image triplet, 639
  for essential matrix, 557
  for fundamental matrix, 553
  for groups of observations, 167
  for three image points, 623
  gauge c., 110
  weak c., 102
constructions
  in 2D, 292–295
  in 3D, 300–304
continuous random variables, 26
continuous, C^n-c., 738
control
  feature, 647
  full c. point, 608
  horizontal c. point, 609
  line, 493, 548, 609
  plane, 609
  planimetric c. point, 609
  point, 450, 493, 527, 548
  points of image pair, 608
  stochastic c. point, 610
convolution, 42
Cook's distance, 127
coordinate axes, 244
coordinate system
  camera c., 461, **463**, 465, 602
  elements of c., 243
  image c., 463
  normalized camera c., 466
  object c., **462**, 465
  of photogrammetric model, 559
  scene c., 462
  sensor c., 463
coordinate transformation, 109, 262–266
  interpretation of c., 249
coordinates
  barycentric c., 213, **349**
  homogeneous c., 45, **195**, 205
  image c., 469
coplanarity
  condition, 228
  of 3D points, 602
  of four 3D points, 307
  of two 3D lines, 304
coplanarity constraint, **550**, 564
  for normal case, 561, 562
  from projection rays, 554
  of images of calibrated cameras, 555–557
  of images of spherical cameras, 556–557
  of images of uncalibrated cameras, 552, 553
  table, 562
corrections, 82
correctness, 116
  of covariance matrix, 140
  of variance factor, 140
correlation, 248
  coefficient, 31, 37
  function, 50
  matrix, 38
  of parameters of spatial, 522
  projective c., 282
  singular, 564
correspondence, 548
correspondence problem, 9
corresponding
  image lines, 568
  image points, 561, 563, 568, 569
  points and lines, 621
covariance, 37
  function, **49**, 121, 736, 739, 741
  intersection, 98
  operator, 38
covariance matrix, 37
  acceptability of c., 120
  correctness of c., 140
  effect of wrong c., 135
  eigenvalue decomposition of c., 667
  empirical c., 118
  evaluation of c., 669
  metric for c., 121
  of algebraic solution, 180
  of centroid, 410
  of estimated observations, 86
  of estimated parameters, 86, 96
  of estimated residuals, 87
  of five-point solution, 588
  of homography from four points, 388
  of image coordinates, 569
  of matrix, 32
  of mean direction, 403
  of parallaxes, 569
  of parameters, relative orientation, 591
  of projection matrix, 495
  of quaternion, 383
  of quaternion from directions, 408
  of reduced 2D line coordinates, 376
  of reduced coordinates $\boldsymbol{x}_r$, 371

of reduced coordinates of 3D line, 381
of rotation, 435
of rotation from directions, 407
of rotation matrix, 435
of the residuals, relative orientation, 591
reference c., 517
singular c., 33
sparsity of c., 663
specification of c., 121
theoretical c., 517
Cramer–Rao bound, **86**, 118, 648
of bundle adjustment, 683
crisp constraints, 96
criterion matrix, 120
critical configuration, **452**, 452
DLT with points, 495
estimation of trifocal tensor, 636
image pair, 614
line prediction, 626
prediction in image triplet, 635
spatial resection, 515, 521
critical cylinder of spatial resection, 515, 517
critical surface
estimation of essential matrix, 588
estimation of fundamental matrix, 571
relative orientation, 588
cross ratio
of collinear points, 268
of concurrent lines, 270
cumulative distribution, 26
cumulative distribution, inverse c., 40
curvature
as fictitious observations, 747
as prior, 745
for regularization, 747
curve, flatness of c., 739
Cuthill–McKee algorithm, 662

d.o.f., *see* degrees of freedom
datum, *see* gauge
decision theory tasks, 19
decomposition
Cholesky d., 661, 776
LU d., 661
of projection matrix, 498
QR d., 776
definition accuracy, 117
degenerate configuration, 517
degrees of freedom
of 2D point, 207
degrees of freedom, 360
3D rotation, **327**
in χ^2-distribution, 33
in t-distribution, 35
in Fisher distribution, 35
in noncentral χ^2 distribution, 34
in Wishart distribution, 34
of 2D elation, 278
of 2D homography, **251**, 253, 321
of 2D line, 375
of 2D perspectivity, 278
of 2D point, 368
of 3D elation, 281
of 3D homography, 255
of 3D homology, 281
of 3D line, **216**, 227, 264
of 3D point, 211, 373
of 3D rotation, 327, 382
of collineations, 285
of essential matrix, 556
of fundamental matrix, 553
of general homography, 249
of general rotation, 326
of image pair, 550
of plane, 212
of projection matrices from fundamental matrix, 594
of projection matrix, 472
of test w.r.t. ground truth, 119
of test on bias, 141
of test on correctness of covariance matrix, 141
of test on groups of outliers, **129**, 129, 131
of test on noise level, 140
of test on systematic errors, 134
of tests on geometric relations, 393, 395
of transformations, 253
of trifocal tensor, 622
of variance factor, 90, 98
Delaunay triangulation, 732
delta function, 26
Denavit–Hartenberg parameters, 264
density
a posteriori d., 77
function, 26
dependent images, 558, 595
general parametrization, 558
design matrix, 82
reduced d., 95
detectability
ellipse, 130
factor, 125, 129, 131
of gross errors, relative orientation, 591
of groups of gross errors, 129
of single gross errors, 125
detectable gross error, 125
detectable outliers during relative orientation, 592
diagnostics, 142
external d., **115**, 119
internal d., **115**, 115, 117, 118
differential
angles, 337, 338
calibration matrix, 501
rotation, 336
rotation vector, 337
similarity transformation, 111
differential GPS, 683, 721
DigiCAMx4, IGI, 683
digital elevation model, 728
digital surface model, 728
direct linear transformation (DLT), 247, **249**
direct LS estimation
2D lines from points, 401
3D line from points, 414
3D similarity from point pairs, 411
mean axis, 405
mean direction, 404
point from lines, 403
rotation from point pairs, 408
direct solution, 176, 178, 452
minimal d., 178
minimum norm solution, 179
of 2D intersection point, 401
of 3D homography, 406
of 3D intersection point, 401, 402
of 3D line, 411
of absolute orientation with three points, 515
of algebraic surface, 183
of best fitting 2D line, 397
of best fitting mean axis, 405
of best fitting mean direction, 403
of best fitting plane, 400
of best fitting rotation from directions, 406
of best fitting similarity, 408
of bundle adjustment, 711
of circle fitting, 177
of conic, 182
of ellipse fitting, 182
of estimation a 3D line, 412
of homography, 406
of quadric fitting, 183
of spatial resection, 513
of spatial resection with > 3 points, 518
of trifocal tensor, 636
directed
3D line, 606
prediction of d. image lines in image triplet, 627
direction
angle, 298
angle between 2D d. and coordinate axes, 206
angle between 3D d. and coordinate axes, 210
closed form solution for mean d., 403
cosine matrix for rotation, 328
interpolation of d., 341
of 2D line, 294
of camera ray, 467, **469**
of intersection of two planes, 353
of join of two points, 352
of line segment, 352
of lines and planes, 346–348
vector of image point, **469**, 553, 556
discrete random variables, 26
dispersion operator, 38
displacements, 262
concatenation of d., 262
distance
between two 2D points, 298
between two 3D lines, 310
between two 3D points, 309
from origin in 2D, 297
Mahalanobis d., 84

of 2D entities, 297–298, 310
of 2D point from 2D line, 298
of 2D point from line, 298
of 3D entities, 308
of 3D line from origin, 308
of 3D line to origin, 218
of 3D point from line, 309
of 3D point from origin, 308
of 3D point from plane, 310
of plane from origin, 308
of two covariance matrices, 121
signed d., 354
distinct entities defining
2D line, 318
2D point, 318
3D line, 319
3D point, 319
plane, 319
distortion
lens d., 464, 507
nonlinear d., 452, 477
of perspective mapping, 479
radial, 506
radial d., 506–508
tangential d., 506
distortion model, 476
phenomenological d., 508
physical d., 506
distribution, 24, 28–35
χ d., 34
χ^2-square d., 33
t-d., 35
binomial d., 28
cumulative d., 26
exponential d., 29
Fisher d., 35
Gaussian d., 29
inverse cumulative d., 40
Laplace d., 29
mixed d., 143
multi-dimensional normal d., 31
normal d., 29
quantiles of d., 40
Rayleigh d., 29
Student's t-d., 35
uniform d., 28
Wishart d., 34
DLT, 247, **249**, 480, 622
algebraic solution for d., 494
direct estimation of d., 494
explicit form of D., 472
for uncalibrated cameras, 472
from 3D lines, 504
precision of d. compared to spatial resection, 523
theoretical precision of d., 522
two d. for image pair, 611
DMC, Intergraph, 683
double points, 523, 590, 592
doubly integrated white noise process, 53
driving process, 52, 749
dual
3D line, 233
conic, 239
entities, 231
of 2D point, 204
oriented projective plane, 346
oriented projective space, 347
Plücker coordinates, 233
Plücker matrix, 233
projective plane $\mathbb{P}^{*2}$, **209**, 209
projective space $\mathbb{P}^{*3}$, **212**, 231
transformation, 259
duality, 203, 229–236, 283
of 2D point and 2D line, 234
of 3D lines, 235
of 3D point and plane, 234
of transformations, 259
dualizing matrix, 227

effect
of intrinsics and extrinsics on image coordinates, 502
of random errors on estimation, 117
of wrong covariance matrix, 135
ego-motion determination, 644
eigenvalue
decomposition of covariance matrix, 667
generalized e. problem, 517
eigenvalues, 773–774
elation
definition of e., 278
elementary rotation, 328
angles from rotation matrix, 330
concatenation of e., 329
ellipse, 237
confidence e., 32
detectability e., 130
fitting, 182
sensitivity e., 130
standard e., **31**, 366, 369
empirical
accuracy, 117, **118**
accuracy of bundle adjustment, 682
covariance matrix, 118
precision, 117, **118**
sensitivity, 126, 130, 134
standard deviation, 118
empty projective space $\mathbb{P}^0$, 231
endlap, 700, **718**, 721
epipolar
axis, 563, 564
constraints, 639
geometry, 562–565
line, 248, 563–565, 623
line, curved e., 564
line, oriented e., 564
plane, 563
epipolar line, 573–574
epipole, 563, 565, 594
equidistant projection, 487
equisolid projection, 487
equivalence of uncertain homogeneous vectors, 390
equivariant function, 267
error in variables model, 161
error propagation, *see* variance propagation
error, quasi-systematic, 667
essential matrix, 556–557, 562, 575–583, 613, 623
degrees of freedom of e., 556
dependent images, 558
from ≥ 7 points, 575
from 2 points, given rotation, 578
from 4 coplanar points, 577
from 5 points, 575
normalized cameras, 559
parametrizations of e., 557
projection matrices from e., 595
singular values of e., 557
estimable quantities, 109, 666
estimate
Bayesian e., 76–78, 93
best linear unbiased e., 79
best unbiased e., 79
least squares e., 79
maximum a posteriori e., 77
maximum likelihood e., 78
estimated
covariance matrix of e. observations, 86
covariance matrix of e. parameters, 86
covariance matrix of e. residuals, 87
observations, 86
parameters, 84
residuals, 84
size of gross errors, 131
size of group of gross errors, 128
size of single gross error, 124
variance factor, 89
estimation
Bayesian e., 93
Bayesian e. in Gauss–Markov model, 93
bias of e., 141
evaluation of e., 117
in Gauss–Helmert model with constraints, 163–170
in Gauss–Markov model with constraints, 99–102
in linear Gauss–Markov model, 81–102
in model with constraints between observations only, 170
in nonlinear Gauss–Markov model, 102–107
of 2D intersection point, 417
of 3D similarity transformation, 607
of variance components, 91–93, 493
on curved manifolds, 415
robust e., 141
sequential e., 96
statistically optimal e., 452
stochastical model of e., 76, 83
with implicit functions, 160
with reduced coordinates, 415
with two group, 96
estimation theory, 75–81
tasks, 19

Euclidean
bundle adjustment, 649
camera, 464
normalization, 196, 198
normalization of matrix, 285
normalization of vector, 199
Euclidean part
of 1D point, 214
of 2D line, 207
of 2D point, 206
of 3D line coordinates, 218
of 3D point, 210
of plane, 211
Euler's rotation theorem, 326
evaluation
w.r.t. groups of gross errors, 128
w.r.t. single gross errors, 124
w.r.t. systematic errors, 133
of acceptability of precision, 120
of additional parameters, 699
of block adjustment, 662
of bundle adjustment, 687
of calibration model, 684
of checkability of parameters, 133
of covariance matrix of block adjustment, 669
of detectability of groups of gross errors, 129
of detectability of single gross errors, 125
of effect of errors, 122
of empirical accuracy, 118
of empirical precision, 118
of estimation, 115
of theoretical precision, 117
of uncertain relations, 393
expectation, 36, 38
of function of stochastic vector, 44
operator, 38
exponential ρ-function, 149
exponential distribution, 29
exterior orientation, **460**, 460, 629, 634
model of e., 465
of camera systems, 488
of image pair, 550, 610
of image triplet, 622, 623
exterior parameters, 460
external diagnostics, **115**, 119
external precision, 116
extrapolation during transformation, 389
extrinsic parameters, *see* exterior parameters
from spatial resection, 513

factor graph, **654**, 659, 661
factorization
of matrix, *see* decomposition
factorization for bundle adjustment, 714
feature
control f., 647
image f., 646
scene f., 646
fictitious observations, 78, 750
for profile reconstruction, 746
field of view, 445, 446, **458**, 459, 468, 484
field-based representation, 8
field-based scene description, 442
fill-in, 662–665
filtering, 730, 736, 762
Kalman f., 96, 98
Wiener f., 93
FinePix REAL 3D W1, Fuji, 443, 445
fish-eye, 444
camera, 485
lens, 459, 485
optics, 478
Fisher
distribution, 35
information matrix, 86
fitted observations, 86
fixed entities, 272–277
of 1D homography, 276
of 2D affinity, 274
of 2D homography, 274
of 2D motion, 274
of 2D rotation, 274
of 2D translations, 274
of 3D affinity, 276
of 3D homography, 275
of 3D motion, 276, 282
of 3D translation, 275
of planar motion, 275
of spatial motion, 276
flatness
of a curve, 739
of a surface, 739–741
flight direction, 558
flight plan, 452
flying height, 605
focal length, 256, 461
focal point, 461
foot point
of 2D origin on line, 295
of origin on 3D line, 323
forward motion, 589
image pair, quality, 593
free adjustment, 109
inner precision of f., 666
minimum trace solution of f., 111
of block, 663
reduced normal equations of f., 114
free network, 109
Fuji FinePix REAL 3D W1, 443, 445
function
ρ-f., 144
delta f., 26
density f., 26
multi-dimensional probability f., 27
of a random variable, 40
of two random variables, 42
separable f., 28
step f., 25
functional model, 75, 490
algebraic structure of f., 161
invertibility of f., 144
nonlinear – linear f., 161
of 2D block adjustment, 651
of block adjustment, 647
of bundle adjustment, 675
table with f., 171
fundamental matrices of image triplet, 623
fundamental matrix, 553–555, 570–574, 612, 622, 629
as singular correlation, 564
degrees of freedom of f., 553
from ≥ 7 points, 571
from ≥ 8 points, 570
from camera planes, 554
from projection matrices, 554
from projection matrix for lines, 564
singular values of f., 554

Gamma-matrix, 219
dual G., 233
gauge, **108**, 703
constraints, 110
constraints in bundle adjustment, 665
definition of g. of coordinate system, 109
definition of g. of covariance matrix, 109
in bundle block, 645
linear g. constraints, 112
minimal control in block adjustment for g., 664
nonlinear g. constraints, 111
transformation, 108–114
transformation in block adjustment, 668
transformation, regular g., **112**, 121
transformation, singular g., 112
unspecified g., 669
Gauss–Helmert model, 160, **162**, 162, 163, 174, 414, 415
for 2D intersection point, 418
for homography estimation, 424, 425
for relative orientation, 586
for total least squares, 161
Gauss–Helmert model with constraints, **163**, 174
estimation in G., 163–170
linear G., 163
nonlinear G., 163
normal equations of G., 165
redundancy of G., 165
robust algorithm for G., 168
Gauss–Markov model, 162, 173, 414, 415, 497
Bayesian estimation in G., 93
for homography estimation, 424, 427
for image triplet, 638
for self-calibrating bundle adjustment, 675, 678
for surface reconstruction, 743, 746
linear G., 81–102
nonlinear G., 102–107
nonlinear G. with constraints, 104

Gauss–Markov model with constraints, 99–102, 162, 173
estimation in G., 100
linear G., 162
nonlinear G., 162
Gauss–Newton method, 103, 105
Gaussian distribution, 29
general weighted least squares, 80
generative model, 83
generic camera, **446**, 460
relative pose, 581
geometric algebra, 236
geometric image model, 447
geometric relations
in 2D, 295–299
in 3D, 304–308
geometry
epipolar g., 562–565
image pair, 549–568
image triplet, 622–632
of single image, 456
single image, 488
global test, 90
bundle adjustment, 689
GPS, 450, 452, 493, 647, 653
differential G., 683, 721
Gram–Schmidt orthogonalization, 511
graph
adjacency g., 654, 660
bi-partite g., 646
factor g., **654**, 659, 661
graph surface, **729**, 739
Grassmann–Cayley algebra, 234
gross error, *see* outlier
detection, 452
estimated size of g., 124, 131
lower bound for detectable g., 125, 131
model of g., 123
test statistic for g., 131
gross errors, 452
ground sampling distance, **457**, 590, 683, 716, 720
ground truth, 115, 119, 429, 683
groups of observations, 86
constraints for g., 167
detectable gross errors of g., 129
diagnostics of g., 130
evaluation of g., 128
in sequential estimation, 96
in variance component estimation, 91, 92
normal equations for g., 96
outlier model of g., 123
sensitivity factor of g., 130
Gruber
points, 590
position, 590
GSD, *see* ground sampling distance

Hadamard product, 137, **776**
harmonic homology, 280
harmonic points, 270
hat function, 736
hat matrix, 86
Helmert point error, 366
Hessian matrix, 44, 106
of log-likelihood function, 78
of surface function, 741
Hessian normal form, 207
uncertain, 374
hierarchy of transformations, 285
homogeneous
uncertainty, 396
homogeneous entities, notation of h., 196
homogeneous coordinates, 45, **195**, 490
of 1D point, 214
of 2D line, 207
of 2D point, 206
of 3D point, 210
of plane, 211
homogeneous part
of 1D point coordinates, 214
of 2D line coordinates, 207
of 2D point coordinates, 206
of 3D line coordinates, 218
of 3D point coordinates, 210
of plane coordinates, 211
homogeneous representation, 195
homogeneous stochastic process, 51
homogeneous uncertain vectors, equivalence of h., 390
homogeneous uncertainty, 121, 371
homogeneous vectors, normalization of h., 198, 241
homography, **249**, 247–249, 253, 254, 256
1D h., 257
2D h., 250, **253**
2D h. between images, 567, 578
3D h., 255, 256, 613, 622
concatenation of h., 261
constraint of two 3D lines, 316
cross ratio, 268
depth and parallax map, 602
effect of h. on chirality , 356
fixed entities of 1D h., 276
fixed entities of 2D h., 274
fixed entities of 3D h., 275
from point pairs, 389
from uncertain point pairs, 425
general h., 248
image to map plane, 526
image to scene plane, 524, 525
invariants of h., 268
minimal parametrization of uncertain h., **384**, 426
quasi-affine h., 357
table of constraints with h., 316
uncertain h., 384–386
uncertainty of h. from uncertain points, 387
uncertainty of points mapped with h., 387
vector form of h., 315
homologeous, *see* corresponding
homology, **277**, 280, 284, 568
between images, 567
harmonic h., 280
singular values of h., 568
homology, see also perspective autocollineation, 277
horizon, 208, **458**, 468, 474, 482
as 3D control line, 494, 504
horizon line, 458, 459
horizontal view, 456
horopter, 521
Hough transformation, 158, **283**
Huber estimator, **148**, 150
human stereo vision system, 561
hyperbola, 237
standard h., 374
hyperplane, 221, 224, 226, 283
at infinity, 215
transformation of h., 258

ideal
camera, 465
image point, 463
lens, 256
perspective camera, 468
unit camera, 465
ideal point, *see* point at infinity
identification accuracy, 117, 706
of features, 490
identity
of two 2D entities, 296
of two 3D entities, 306
IGI DigiCAMx4, 683
ill-posed, 82
image
coordinate system, 463
coordinates, 469
coplanarity constraint of i. from uncalibrated cameras, 552
distortion model, 476
distortions, 505
feature, 646
geometry, 456
geometry of nadir i., 459
geometry of slanted i., 459
matching, 402, 563
model, 7, 441
orientation of single image, 489
oriented i. line, 482
pair, *see* image pair
perspective i., 456
point, *see* image point
pyramid, 491
rectified i., 477
reduced i. coordinates, **468**, 470
relative orientation, 552
scale, 457
sequence, 647, *see* image strip
straight line-perturbing i. errors, 464
theoretical precision of i. block, 673
theoretical precision of i. strip, 670–673
triplet, *see* image triplet
two-step procedure of i. orientation, 549
image pair
bundle solution for i., 610
comparison of procedures for i., 614

control points of i., 608
critical configuration of i., 614
degrees of freedom of i., 550
exterior orientation of i., 550
geometry of i., 549
interior orientation of i., 550
normal case of i., 465, 561
object points, 608
orientation of i., 549, 608
triangulation for normal case of i., 601
triangulation for perspective i., 600
two-step procedure for i., 612
image point
direction, 556
ideal i., **463**, 469
observable i., 461, **463**
uncertainty of i., 491
image triplet
geometry of i., 622
nonlinear observation equations for i., 638
orientation of i., 632
predicting directed lines in i., 627
relative orientation of i., 633, **636**
images, dependent i., 595
implicit functions, estimation with i., 160
implicit variance propagation, **43**, 516
IMU, 653
incidence
of 2D entities, 295
of 2D line and 2D point, 295
of 3D line and plane, 305
of 3D point and line, 306
of 3D point and plane, 304
of two 3D lines, 304
independence
stochastic i., 28
independent events, 23
independent images, **558**, 581, 589
independent random variables, 31
influence function, 147
table of i., 149
information matrix, 86
inner
geometry, 110
precision, 110, 666
precision of free block, 667
inner product, 767
INS, 452, 647
integrated AR-process, 54
integrated white noise process, 53
Intergraph DMC, 683
interior and exterior orientation
from projection matrix, 500
interior orientation, **460**, 464, 610, 629
of image pair, 550
of image triplet, 622
interior parameters of camera, **460**
interior parameters of camera system, 488
internal
diagnostics, **115**, 115, 117, 118
precision, 116
interpolation, 730
bicubic i., 738
bilinear i., 738
linear i., 737
precision of i., 734
interpolation during transformation, 389
interpolation of directions, 341
interpolation of rotations, 341
interpretation, 9
interpretation model, 7, 448
interpretation of line drawings, 523
intersection, 549
of 2D line and conic, 293
of 3D line and plane, 301
of three planes, 302
of two 2D lines, 202, 292
of two planes, 220, 301
intersection of planes, direct LS solution of i., 403
intrinsic parameters, *see* interior parameters
intrinsic parameters of a camera, 463
invariant, 266
number of independent i., 271
of affinities, 268
of five 2D points, 272
of homography, 268
of perspective mappings, 268
of polygon, 267
of projective mapping, 266
of rectangle, 266, 271
inverse cumulative distribution, 40
inverse depth, 257
inverse perspective, 489, 523
inversion of transformation, 261
invertibility of functional model, 144
isocentre, 459
isometric parallel, 459
isotropic stochastic process, 51
isotropic uncertainty, 121
of directions, 367, **371**, 403, 413
of points, **368**, 396, 399, 406, 408, 412
iterative estimation, 92, **103**, 414, 452
of spatial resection, 520
on curved manifolds, 415
iterative solution, 452

Jacobian
of DLT, 501
of DLT for nadir view, 501
of Hessian form to homogeneous 2D line, 375
of homogeneous to Euclidean 3D coordinates, 373
of homogeneous to Euclidean coordinates, 371
of homogeneous to Hessian form of 2D line, 377
of reduction of 3D line coordinates, 380
of reduction of point coordinates, 370
of spherical normalization, 368, 376
of spherical normalization of 3D point, 373
within estimation, 161
within variance propagation, 43
join
of 3D point and line, 302
of three 3D points, 302
of two 2D points, 202, 293
of two 3D points, 300

K-transformation, 109
Kalman filter, 96, 98
keypoint detector
uncertainty, 491
Kronecker product, 137, 555, 775

laboratory calibration, 697
Ladybug 3, Pointgrey, 443–445
lag, 49
Laplace distribution, 29
Laplacian development theorem, 768
law of cosines, 514
least squares
estimate, 79
general weighted l., 80
ordinary l., 80
weighted l., 79
with regularization for profile reconstruction, 747
leave-one-out test, 124, 128
Legoland scene, 529
Leica ADS 80, 443, 445, 446
lens
distortion, 461, 464, **507**
fish-eye l., 459
narrow-angle l., 459
normal l., 459
thin l. projection, 256
ultra-wide angle l., 459
wide-angle l., 459
zoom l., 459
levels of reasoning, 7
leverage point, 127
Lie group, 284
likelihood function, 77
line
antipodal l., 348
at infinity, 203, 345
direction of l. segment, 352
segment, 352
vanishing l., 529
line at infinity, 208
of plane, **208**, 212, 220
line camera, 444
line drawing interpretation, 523
line segment, uncertainty of l., 492
linear substitute model, 103
linearization, check of l. within Gauss–Markov model, 104
linearized model of bundle adjustment, 676
linearized models, table with l., 171
loop closing, 672
lower bound
for detectable deviation, 66
for detectable gross error, 125, 131
LS, *see* least squares

LU-decomposition, 661

M-estimation, 609
MAD, **40**, 146
Mahalanobis distance, **69**, 84, 361
Manhattan scene, 529
MAP estimate, 77
  bundle adjustment, 648
  profile reconstruction, 744
Maple, 522
mapping, 644
  affine m., 357
  as coordinate transformation, 249
  as displacement, 249
  general m. 3D to 2D, 479
  of general lines, 484
  of quadrics, 484
  perspective m., 277
  quasi-affine m., 357
matching, 9
matrix
  block, inverse of b., 769
  correlation m., 38
  covariance m., 37
  Euclidean normalization of m., 285
  exponential, 781
  exponential for homography, 384
  exponential for motion and similarity, 384
  exponential for rotation, **326**, 326, 337, 338, 382
  exponential for transformations, 382
  normally distributed m., 32
  precision m., 83
  product, eigenvalues of m., 773
  random m., 27
  representation of 2D entities, 312
  representation of 3D entities, 313
  skew symmetric m., 336
  sparse m., 86
  sparse structure of m., 655
  spectral normalization of m., 286
  Toeplitz m., 53
  weight coefficient m., 89
  weight m., 83, 89
maximum a posterior estimate, 77
maximum likelihood estimation, 78–79
maximum likelihood type estimation, 147
mean, 36
  accuracy of the m., 116
  bias of m., 45
  of ratios, 46
  operator, 38
  precision of the m., 116
  vector, 38
median, 40
median absolute difference, 40, 146
meta model, 6, 441
method of modified weights, 147
metric camera, **460**, 696
metric for covariance matrices, 121
minimal parametrization
  of 2D homography, 385
  of uncertain homography, 384
  of uncertain motion, 383
  of uncertain quaternions, 383
  of uncertain rotation, 382
  of uncertain similarity, 383
  of uncertain transformations, 381
minimal representation
  of 2D uncertain point, 369
minimal solution, 178
  of 2D affinity, 320
  of 2D homography, 321
  of 3D homography, 322
  of basis of image pair, 578
  of essential matrix, 575
  of fundamental matrix, 571
  of image orientation, 489
  of projection matrix, 494
  of relative orientation from mirror images, 579
  of relative orientation of three images, 636
  of relative orientation, iterative m., 585
  of spatial resection, 513
  of trifocal tensor, 636
  with QR decomposition, 179
minimum norm solution, 179
minimum trace solution, 111
minimum-volume estimate, 146
mirroring
  at y-axis, 251, 279
  at a plane, 281
  transformation in 2D, 279
mixed distribution, 143
ML-type estimation, 147
  L_1-norm minimization as, 148
model
  functional m., 75
  generative m., 83
  geometric m. of camera, 443
  geometric m. of scene, 442
  linear substitute m., 103
  mathematical m., 75
  notion of m., 7
  of analysis, 7
  of camera, 441, **445**
  of constraints between observations only, **162**, 162
  of distortion, 476
  of image, 7, 441
  of interpretation, 7
  of projection, 449
  of scene, 7, 441
  of sensor, 7
  of world, 7
  phenomenological m. of distortion, 505
  photogrammetric, *see* photogrammetric model
  physical m. of distortion, 505
  stochastical m., 76, 83
  thin plate m., 741
  thin rod m., 741
  weak membrane m., 740
  weak string m., 739
  with constraints between the observations only, 173
model block adjustment
  2D, 650
  2D m., 651–674
  3D m., 649
  projective m., 649
modified weights, 147
moment vector of a 3D line, **218**, 219, 220, 227
moments, 36
  central m., 37
  general m., 36
  of normal distribution, 39
mono-plotting, 526
monomials, 733
motion
  2D m., 260
  3D m., 260
  fixed entities of planar m., 275
  fixed entities of spatial m., 276
  forward m., 589
  from object to camera, 466
  from structure, 449
  planar, 251
  rigid body m., 255
  rotational m., 337
  sideward m., 588
  spatial m., 255
  uncertain m., 383
moving object, 568
moving camera, 568
multiple solutions, 452
  of relative orientation, E-matrix, 582
  of relative orientation, F-matrix, 571
  of spatial resection, 515

nadir
  direction, 528
  point, 346, 458
  view, 456, 521
narrow-angle lens, 459
negative point, 344
net area of model, 719
Newton–Raphson method, 105
noncentral χ'^2 distribution, 34
noncentrality parameter, 65, 131
nonlinear
  distortions, 452, 477
  Gauss–Markov model, 102
  Gauss–Markov model with constraints, 104
  model of bundle adjustment, 675
nonmetric camera, uncalibrated n., 697
nonrejection region, 63
normal case
  of image pair, 561
  of image pair, iterative relative orientation, 588
  of single image, 465
  triangulation for n. of image pair, 601
normal cases, 453
normal distribution, 29
  in best unbiased estimation, 81

multi-dimensional n., 31
normal equation
components, 85
for groups, 96
for system for two groups of observations, 96
in Gauss–Helmert model with constraints, 165
in Gauss–Markov model, 84
in Gauss–Markov model with constraints, 100
partitioned n., 94
profile of n. matrix, 662
reduced n., **94**, 660
reduced n. of free adjustment, 114
sparsity of n. matrix, 657–661
table with n., 171
normal lens, 459
normal line
through 2D point, 294
through the 2D origin, 294
normalization
bias of n., 45
Euclidean n., 242
Euclidean n. of matrix, 285
Euclidean n. of vector, 199
of homogeneous matrices, 285
of homogeneous vectors, 198, 241
spectral n. of matrix, 286
spherical n., 45, 199, **242**
spherical n. of matrix, 286
normalized
camera, **465**, 472, 713
camera coordinate system, 466
definition of n. residuals, 144
residuals, 152, 170
trifocal tensor, 628
variance of n. residuals, 145
notation of homogeneous entities, 196
null space
for estimating homography, 389, 496
of covariance matrix of 2D line, 375, 392
of covariance matrix of 2D point, 368, 393
using QR decomposition, 179
numerical differentiation, 453
NURBS, mapping of N., 484

object
coordinate system, **462**, 465
moving o., 568
point, 563
points, image pair, 608
object-based representation, 8
object-based scene description, 442
oblique view, 456
observable image point, 463
observation equations, 82
linearized o. for normal case of relative orientation, 589
nonlinear o., 173
nonlinear o. for image triplet, 638
observation process, 25
observational errors, 83
observations
fictitious o., 78
uncertain o., 490
observed AR-process, 749
One Shot 360, 443, 445, 446
operator
covariance o., 38
dispersion o., 38
mean o., 38
optical axis, 461
optical ray, 528
optics, 456
catadioptric o., 446
optimal estimation
of intrinsics and extrinsics, 501
of projection matrix, 496
oracle, robust estimation as o., **142**, 167
ordinary least squares, 80
orientation
absolute o., 549, **552**, 607
absolute o. within two-step procedure, 612
also see direction, 352
exterior o., **465**, 610
exterior o. of camera systems, 488
exterior o. of image pair, 550
interior o., 610
interior o. of image pair, 550
of cameras, 449
of conics and quadrics, 348
of image pair, 549, 608
of image triplet, 632
of join of 3D line and point, 353
of plane, 354
parameters from essential matrix, 581
quality of o. procedures, 453
relative o., 450, 551
relative o. of image triplet, 633
relative o. within two-step procedure, 612
orientation-preserving transformation, 355
oriented
2D line, 346
2D point, 345
3D line, 348
3D point, 346
epipolar line, 564
image line, 482
plane, 347
point, 344
projective geometry, 343
projective plane, 345
projective space, **345**
orthogonal
projection, 487
orthogonality
of 2D lines, 297
of 3D line and plane, 308
of 3D lines, 307
of planes, 307
outlier, *see* gross error, 609
asymmetric distribution of o., 755
detection, 142
detection in bundle adjustment, 707–715
model, 143

P3P problem, 513–518
panorama, 644
Panoscan Mark III, Rollei, 443, 445, 446
parabola, 237
parallactic angle, 420, 421, **548**, 549, 550, 596, 598, 599, 601, 604, 717
parallax
x-p., 569, **589**, 601–604
y-p., 561, **589**, 590, 592, 602
accuracy of p., 569
covariance matrix, 569
map, 602
vertical p., 561
parallel line
through 2D point, 294
through the 2D origin, 294
parallel projection, 545
parallelepiped, 553
parallelism
of 2D lines, 297
of 3D line and plane, 308
of 3D lines, 307
of planes, 307
parameters
additional p., 123, 464
extrinsic p., 460
interior p. of camera system, 488
intrinsic p. of a camera, 460
parametrization of relative orientation, 557–559
singularity, 559
partially calibrated camera, 460
partitioning of normal equation matrix, 94
PCA of covariance matrix, 667
pencil of planes in epipolar geometry, 563
percentile, 40
perspective
2D autocollineation, 277
3D autocollineation, 280
autocollineation, 277
calibrated p. camera, 555–557
camera, 248, **446**, 456, 460, 464, 607, 622
camera for the image pair, 550
camera model, 462, 470
collineation, 248, **277**, 277, 278
distortion, 503
distortions of p. camera, 505
image, 456
inverse p., 523
mapping with distortions, 479
model of p. camera, 464
projection, 467, **470**, 486
projection of 3D line, 480
perspectivity, 277, 278
phenomenological distortion model, 505, 508
photo collection, 644

photogrammetric model, **549**, 612–613, 708–711, 719
  absolute orientation of p., 607
  coordinate system definition, 559
  coplanarity constraint, 551
  net area of p., 719
  of image triplet, 622–633
  of images of calibrated cameras, 552
  of images of uncalibrated cameras, 552
  parameters of p. for given projection matrix P', 560
  scale of p., 559, 607, 634, 714
photogrammetric models
  mutual scale of p., 621
physical distortion model, 505, 506
Pi-matrix, 300
Pi-matrix, dual P., 301
pinhole camera, 253, 257, 281, 464, 465
pixel distance, 465
Plücker
  constraint, 218
  enforcing P. constraint, 381
  matrix, 219
Plücker constraint, 227
Plücker coordinates, 221–229, 768
  definition of P., 223
  dual P., 233
  Euclidean part of P. of 3D line, 218
  homogeneous part of P. of 3D line, 218
  of 2D line, 223
  of 2D points, 223
  of 3D line, **218**, 226
  of 3D line from points, 217
  of 3D points, 225
  of plane, 225
Plücker matrix, 227
  dual P., 233
planar homography
  fixed entities, 274
planar motion, 251
planar object
  critical configuration of DLT, 495
plane, 211
  antipodal p., 347
  at infinity, 212
  centroid representation of p., 377
  degrees of freedom of p., 212
  distinct points of p., 319
  homogeneous coordinates of p., 211
  horizon of p., 208
  intersection, 301
  joining 3D point and line, 302
  line at infinity of p., **208**, 212, 220
  optimal direct solution of p., 400, 436
  orientation of p., 348, 354
  oriented p., 347
  parameters of p. through three points, 225
  Plücker coordinates of p., 225
  point-direction form of p., 213
  quadric, 240
  slope of p., 212
  three-point representation of p., 213
  transformation of p., 258
  uncertain p., 377–379, 403
planes, concurrent p., 306
PnP problem, 513–521
point
  antipodal p., 344–346
  control p. of image pair, 608
  in tetrahedron, 351
  negative p., 344
  of symmetry, 462
  oriented p., 344
  positive p., 344
  tie p. of image pair, 608
  uncertainty of transformed p., 387
  vanishing p., 210, 529–534
point at infinity, 472, 493, 550
  1D p., 214
  2D p., 206
  3D p., 210
  of 2D line, 209, 294
  of 3D line, 220
point camera, 444
point of symmetry
  of conic, 237
  of quadric, 240
Pointgrey Ladybug 3, 443–445
polar of a point, 233
polarity, 233, 283, 285
  at conics, 238
  at the unit circle, 233
  on the sphere, 200
pole of a line, 233
polycamera, 445
pose, 456
  of camera, 6, 456, **460**
  theoretical precision of p., 522, 523
positive definite function, 50
positive point, 344
power function, 65
power of test, 62
PowerShot A630, Canon, 443, 445
pre-image of line at infinity, 355
precision, **116**, 453
  acceptability of p., 117
  and accuracy, 116
  empirical p., 117
  external p., 116
  inner p., 110
  internal p., 116
  matrix, **43**, 83, 367
  of interpolation, 734
  of the mean, 116
  singular p. matrix, 367
  theoretical p., 117
prediction, 730
  in image pair, image point, 562–565
  in image triplet, points and lines, 623–625
  of points, lines and planes, 451
prediction errors, 97
prediction operator
  $\ell(l'', l''')$, 625
  $\wp_2(x', l'''))$, 631
  $\wp_3(x', l''))$, 631
principal
  distance, 457, **462**, 464, 465, 471
  distance from two vanishing points, 531
  line, 458
  plane of optics, 256, **461**
  planes of camera, 474
  point, 458, **462**, 464, 465, 470, 471
  point from projection matrix, 475
  point from three vanishing points, 532
prior
  for profiles and surfaces, 745–748
  variance component estimation of p., 750
prior, see also a priori, 76
probability
  a posteriori p. density, 77
  a priori p. density, 76
  axiomatic definition of p., 22
  axioms of p., 23
  conditional p., definition, 23
  density function, 26
  distribution, 24
  notion of p., 21
  total p., 23
  von Mises' definition, 22
process
  doubly integrated white noise p., 53
  integrated white noise p., 53
  stochastic p., 49
profile of normal equation matrix, 662
profile reconstruction
  fictitious observations for p., 746
  LS with regularization, 747
  MAP estimate for p., 744
  outlier detection in p., 755
projection
  central p., 481, 485, 487
  equidistant p., 487
  equisolid p., 487
  line, **483**, 564
  matrix, *see* projection matrix
  model, 449
  not straight line-preserving p., 564
  of 3D lines, 480
  of lines, 564
  orthogonal p., 487
  parallel p., 545
  perspective p., 467, **470**, 486
  perspective p. of 3D line, 480
  plane, 482, 483
  ray, 445
  spherical p., 467
  spherical p. of 3D line, 481
  stereographic p., 346, 487
  thin lens p., 256
  uncertainty of p. ray, 524
projection centre, 248, 254, **457**, 460–466, 474–475, 563
  collinear, 622

from projection matrix, 498
quality of p. from spatial resection, 516, 517
projection matrix, **472**, 607, 629
algebraic solution for p., 494
covariance matrix of p., 495
decomposition of p., 498
direct estimation of p., 494
DLT with p., 494
for 3D lines, 480, 626
for 3D points, 626
from essential matrix, 595
from fundamental matrix, 594
general p., 479
interior and exterior orientation from, 500
optimal estimation of p., 496
proper p., 468, 474, 482, 483
properties of p., 473
properties of p. for 3D lines, 481
uncertainty of p., 475
projective
bundle adjustment, 649
collination, 248
correlation, 282
line $\mathbb{P}$, 231
model, 480
model block adjustment, 649
oriented dual p. plane, 346
oriented p. plane, 345
oriented p. space, 345
plane $\mathbb{P}^2$, 200, 203, **206**, 231
3D points at infinity, 211
partioning of p., 215
representation as unit sphere, 215
point $\mathbb{P}^0$, 215, 216, 231
space $\mathbb{P}^1$, 214
space $\mathbb{P}^3$, **210**, 231
space $\mathbb{P}^n$, 215
transformation, 611
projective bundle adjustment, 676
projectivity
2D p., 253
3D p., 256
pure p., 254
singular p., 473
propagation of uncertainty, 386
proper
projection matrix, 468, 474, 482, 483
rotation matrix, 499
pseudo-inverse, 101, 779
rectangular matrix, 779
symmetric matrix, 779
pseudo-likelihood function, 144

QR decomposition, 776
for minimal solution, 179
for null space, 179
quadratic variation, **741**, 760
quadric, 239
fitting, 183
mapping of q., 484
orientation of q., 348
point of symmetry of q., 240
tangent plane at q., 240
transformation of q., 260
quality
checkability of the observations, 609
criteria, 609
of 3D point from two images, 603
of 3D points, 526
of parameters, 452
of relative orientation, 590
precision, 609
quantile, 40
quasi-affine projective mapping, 357
quasi-systematic errors, 667
quaternion, 332–335
as hyper-complex numbers, 333
covariance matrix of q., 383
uncertain q., 383

R (software package), 115
radial basis function, 735
radial distortion, 506–508
random
matrix, 27
number generation, 55
vector, 27
random perturbations, 452
random sample consensus, *see* RANSAC
random variables, 24–28
continuous r., 26
discrete r., 26
independent r., 31
transformation of r., 41
uncorrelated r., 31
RANSAC, 153–157, 609
ray direction, 467, **469**, 514
sign of r., 492
uncertainty of r., 492
Rayleigh distribution, 29
reasoning levels, 7
reconstruction
of point on plane, 524
of points and lines, 596–606
quality of r. procedures, 453
rectangle, invariants of r., 271
reduced
design matrix, 95
normal equations, 660
reduced coordinates, 393
covariance matrix of r., 371
of 2D point, 370
of 3D line, 380–381
redundancy, **82**, 609
matrix, **87**, 669
numbers, **88**, 145
numbers, relative orientation, 592
of absolute orientation, 613
of bundle adjustment image pair, 611
of Gauss–Helmert model with constraints, 165
of Gauss–Markov model, 82
of Gauss–Markov model with constraints, 100
of two DLTs, 611
of two spatial resections, 612
relative r., 145
table with r., 171
reference covariance matrix, 517
refraction, 477
regression model, 81
regularization, 82, 747
regularizing observations, 746
rejection region, 63
relative orientation, 450, 551, 634
epipolar geometry of r., 562
iterative r., 585–594
iterative r. for ideal forward motion, 593
iterative r. for normal case, 588
of image pair, 622
of image triplet, 636
of images of calibrated cameras, 552
of images of generic cameras, 581
of images of uncalibrated cameras, 552
planar object, 567
quality of r., 590
theoretical precision of r., 590
theoretical reliability of r., 590
with known plumb line, 581
within two-step procedure, 612
relative redundancy, 145
relief displacement, 459
representation of uncertain points and lines, minimal r., 369–371
residual of coplanarity constraint, 555, 562
residuals, 82
approximate r., 164
covariance matrix of r., 87
normalized r., 144, 152, 170
standardized r., 125
variance of normalized r., 145
resolution, 442
reverse Cuthill–McKee algorithm, 662
rho-function, *see* ρ-function
rigid body motion, 255
RMSE, 119
robust estimate
of standard deviation of normalized residuals, 146
of variance factor, 146
robust estimation, 141–185
as oracle, 142
maximum likelihood-type estimation, 147
of variance factor, 145–146
strategies, 158
with L_1-norm minimization, 150
with clustering, 157
with complete search, 151
with RANSAC, 153
robustness, 142
evaluation of r. with influence function, 147
Rollei Panoscan Mark III, 443, 445, 446
root mean square error, 119
rotating slit camera, 485
rotation, 325, 460
2D r., 251

3D r., 255
averaging, 713
axis and angle from r. matrix, 331
axis angle representation of r., 331
Cayley representation of r., 336
closed form solution for r. from directions, 406
concatenation of elementary r., 329
concatenation of r. with quaternions, 337
conjugate r., 281, 321
differential equation for r., 337
differential r. vector, 337
direction between planes, 353
direction cosine matrix for r., 328
eigenvalues of r. matrix, 327
elementary r., 328
Euler's theorem for r., 326
interpolation, 341
matrix, *see* rotation matrix
minimal r. between two vectors, 340
quaternion representation of r., 332, 335
relations between representations for r., 338
representations in 3D: overview, 326
Rodriguez representation of r., 335
singularity of r., 330
skew-symmetric matrix representation of r., 336
uncertain r., 382–383
vector, 336–338
with Euler angles, 328
rotation matrix, 326, 466
as local coordinate system, 378
direction cosine r., 328
exponential form of r., 326, 337
from corresponding vectors, 339
from projection matrix, 499
from three image lines, 531
representation, 327
rotational motion, 337
rounding error, 26, 29, 39

S-matrix, 112
S-transformation, 109, 113
in block adjustment, 668
regular S-, **112**, 121
singular S-, 112
sampling
in RANSAC, 154
of distributions, 56
scale
factor, 604
mutual s. of photogrammetric models, 621, 639
number, **457**, 591, 604, 720
of photogrammetric model, 551, **559**, 607
scale difference, 464
of image coordinate system, 471
scale transfer, 634, 639
scaling in 2D, 251
scene
coordinate system, 462
feature, 646
field-based s. description, 442
geometric model of s., 442
model, 7, 441
object-based s. description, 442
reconstruction, 450
Schur complement, 660
search, complete s., 151
segment, line s., 352
selecting independent constraints, 317
self-calibrating bundle adjustment, 450, 674–696
self-calibration, 492, 697–699
self-diagnosis, 452
sensitivity, 115
w.r.t. groups of outliers, 130
w.r.t. single outliers, 126–128
w.r.t. systematic errors, 134–135
analysis, 592, 609, 695, 699
ellipse, 130
factor, 126–128, 130, 132, 692–694, 701, 705–706
of bundle adjustment, 691, 693, 701
of relative orientation, 589
sensor coordinate system, 463
sensor model, 7
separable function, 28
sequential estimation, 96
shear, 464, 470, 471
shear in 2D, 252
sidelap, **718**, 721
sideward motion, 588
sign
constraints for points in image triplet, 624
effect of affinity on s. of entity, 357
of distance, 354
of intersection of 3D line and plane, 353
of intersection point of two 2D lines, 353
of ray direction, 492
significance level, 62
significance number, 62
similarity
2D s., 252
3D s., 255, 613, 622
closed form solution of s., 408
minimal solution of s., 320
transformation, 611
uncertain s., 383
simulating data, 55
single image, normal case of, 465
single viewpoint, 445
singular
correlation, 564
dual conic, 241
line conic, 241
precision matrix, 367
projectivity, 473
value decomposition, 777
values of essential matrix, 557
values of fundamental matrix, 554
values of homology, 568
vector, right s., 571
skew matrix, 336, **770**
product with matrix, 772
properties, 770
SLERP, *see* spherical linear interpolation
slope of curve, 739
slope of plane, 212
smoothness
of a function, 740
of a surface, 740–742
SO(n), 326
solution
direct s., 452
iterative s., 452
minimal direct s., 178
space
complete s., 231
empty s., 231
sparse
covariance matrix, 86, 663
design matrix, 656, 657
matrix, 86
normal equation matrix, 657–661
reduced normal equation matrix, 660
structure of matrices, 655
spatial resection, 513–521, 533, 623
critical configuration of s., 515
direct solution of s., 513
direct solution with > 3 points, 518
iterative solution, 520
quality of projection centre from s., 516
theoretical precision of s., 523
two s. for image pair, 612
with observed scene lines, 521
spectral normalization
of matrix, 286
variance propagation, 783
spherical
linear interpolation, 341
normalization, 45, 198, 199, **242**
normalization of matrix, 286
normalization of uncertain 2D point, 368
projection, 467
projection of 3D line, 481
spherical camera, **446**, 456, 462, 468, 555, 582
bundle adjustment, 686
calibrated s. for the image pair, 556
triangulation for s., 597
spherically normalized homogeneous coordinates, 490
spline, 736
one-dimensional s., 736
two-dimensional s., 737
SPSS (software package), 115
stable configuration, 452
standard deviation, 37
empirical s., 118

of exterior orientation with spatial resection, 523
of height, 604
of image coordinates, 527
of parameters
relative orientation, 591
robust estimator of s. of normalized residuals, 146
standard ellipse, **31**, 366, 369
standard hyperbola, 374
standardized residuals, 125
stationary, 49
statistically best fitting
2D line, 397
mean axis, 405
mean direction, 403
plane, 400
rotation from directions, 406
similarity, 408
Steiner's theorem, **37**, 116
stellar calibration, **496**, 533
step function, 25
stereo vision, human s., 561
stereographic projection, **243**, 346, 487
stitching, 644
stochastic independence, 28
stochastic process, 48–55
homogeneous s., 51
isotropic s., 51
stationary s., 49
stochastical model, 76
for surface reconstruction, 743
of 2D block adjustment, 652
of block adjustment, 647
of estimation, 83
straight line segments, 480
straight line-preserving, 249, 470
structure from motion, 450, 568
structure tensor, 402, 569
Student's t-distribution, 35
suboptimal test for gross errors, 126
substitute model, 103
substitute parameters, 177
subsymbolic level of real world, 7
surface, 442
$2^1/_2$D s., 729, 733–742
flatness of s., 739–741
graph s., **729**, 739
reconstruction, 727, 730
reconstruction as Bayesian estimation, 742
reconstruction, outlier detection in s., 755
smoothness of s., 740–742
surfel, 442
SVD
for algebraic solution, 179
for consistency of essential matrix, 575
for consistency of rotation matrix, 531
for estimation of 3D line, 606
for estimation of 3D point, 603
for estimation of base vector, 579
for estimation of fundamental matrix, 571
for estimation of projection matrix, 495
for partitioning of essential matrix, 581
for partitioning of fundamental matrix, 595
of essential matrix, 557
of fundamental matrix, 553
swing angle, 459
symbolic image description, 448
symbolic level of world model, 7
systematic error, 116
in bundle adjustment, 683
model of s., 123
quasi-s., 667
systems of cameras, 488

taking position, **457**, 469, 550, 563
tangent
line at conic, 238
plane at quadric, 240
tangent space, 370
tangential distortion, 485, 506
telelens, 459
tensor notation, 782
test
for 3D line through triangle, 351
for coplanarity constraint, 555
for estimated parameters, 133
for gross error, 131
for point in tetrahedron, 351
of point in triangle, 349
of prediction errors, 98
one-sided t., 67
suboptimal t. for gross errors, 126
two-sided t., 65
within sequential estimation, 98
test field calibration, 697, 698
testability, 66
of mean, 65
of mean vector, 67
of observations, relative orientation, 592
testing, 393
approximate t., 393
geometric relations in 2D and 3D, 393
uncertain relations, 393
tetrahedron
chirality of t., 350
point in t., 351
sign of volume of t., 225
volume of t., 225
theoretical
precision, 117
sensitivity, 127, 135
theoretical covariance matrix, 517
theoretical precision
of 3D point, 526
from two images, 603
of bundle adjustment, 683
of DLT, 522
of image blocks, 673
of image strips, 671
of point on horizontal plane, 527
of pose estimation, 521
of relative orientation, 590
of spatial resection, 523
of strips and blocks, 670–674
of surface interpolation, 761
theoretical reliability
of absolute orientation, 411
of relative orientation, 590
thin plate model, 741
thin rod model, 741
tie points of image pair, 608
tilt angle, 459
tilt angle of a plane, 212
Toeplitz matrix, 53
torsion, 741
total least squares, 161
total probability, 23
traffic light decision, 63
transformation, 247
1D t., 257
2D affinity, 252
2D homography, 253
2D mirroring at y-axis, 251
2D projectivity, 253
2D rotation, 251
2D scaling, 251
2D shear, 252
2D translation, 251
3D affinity, 255
3D homography, 256, 552, 613
3D homography, image pair, 611
3D projectivity, 256
3D rotation, 255
3D similarity, 255, 551, 613, 622
3D similarity, image pair, 611
3D translation, 255
concatenation of t., 261
conjugate t., 278
coordinate t., 262
direct solution of similarity t., 408
dual t., 259
extrapolation during t., 389
group, 284
hierarchy, 285
interpolation during t., 389
inversion of t., 261
of 2D line, 258
of 2D point, 250
of 3D line, 259
of 3D point, 255
of conic, 260
of coordinates w.r.t. fixed reference, 263
of coordinates w.r.t. transformed reference, 264
of hyperplane, 258
of object w.r.t. fixed reference, 262
of object w.r.t. transformed reference, 263
of oriented entities, 355
of plane, 258
of quadric, 260
of random variable, 41
orientation-preserving t., 355
planar motion, 251
similarity t., 252
spatial motion, 255

translation
2D t., 251
3D t., 255
conjugate t., 279
triangle
area of t., 222
chirality of t., 349
spatial t. and 3D line, 351
triangular irregular network, 732
triangulation, 549, 552, 595, **596**
algebraic t. from multiple images, 602
Delaunay t., 732
for normal case of image pair, 601
for perspective image pair, 600
for spherical camera, 597
trifocal matrices, 625–626
trifocal plane, 623
trifocal tensor, 622, **625**, 625, 629
degrees of freedom of t., 622
direct estimation for t., 636
from projection matrices, efficient computation, 626
iterative solution for t., 637
minimal solution of t., 636
normalized t., 628
trigonometric basis functions, 734
true value, 79
truncated L_2-norm minimization, 148
Tschebyscheff polynomials, 510
twisted cubic, 495

ultra-wide lens, 459
Ultracam, Vexcel, 443–445, 683
uncalibrated camera, **461**, 490, 622
uncertain
2D line, 373–377
2D point, 366–372
3D line, 379–381
3D point, 372–373
Euclidean coordinates, 371
geometric entities, 359
geometric relations, 359
Hessian parameters of 2D line, 376
homogeneous coordinates, 367, 375
homography, 384–386
matrix, 32
minimal representation of u. 2D point, 369
motion, 383
observations, 490
plane, 377–379, 403
quaternion, 383
rotation, 382–383
rotation matrices, 382
scene points and lines, 493
similarity, 383
spherical normalization of u. 2D point, 368
uncertainty
homogeneous u., 121, 371, 396
isotropic u., 121, 396
isotropic u. of directions, 367, **371**, 403, 413
isotropic u. of points, **368**, 369, 399, 406, 408, 412
of directions in the camera, 492
of estimated 3D line, 413
of extrinsics and intrinsics from projection matrix, 500
of feature identification, 490
of image points, 491
of key point, 491
of line segments, 492
of projection matrix, 475
of projection ray, 524
of ray direction, 492
of transformed points, 387
propagation, 386
uncorrelated random variables, 31
uniform distribution, 28
unit camera, 465
unit circle
S^1 in $\mathbb{R}^2$, 214, 215
unit sphere
S^2 in $\mathbb{R}^3$, 199, 200, 242, 243
S^3 in $\mathbb{R}^4$, 242
S^5 in $\mathbb{R}^6$, 243
unscented transformation for variance propagation, 47

vanishing line, 529
vanishing point, 208, 210, 459, 529–534
estimation, 417
variance, 37
bias of v., 45
of normalized residuals, 145
of ratio, 46
of residual of coplanarity constraint, 555
of scale of similarity from points, 410
variance component estimation, 91–93, 493
bundle adjustment, 679
profile reconstruction, 750
variance factor, 716
correctness of v., 140
estimated v., **89**, 680–685, 700–701
initial, 89
interpretation of v., 498
robust estimation of v., 145–146
test of v., 90
variance propagation, **42**, 42–48
implicit v., **43**, 154, 180, 516
nonlinear v., 43
of bilinear forms, 387
of linear functions, 42
of nonlinear functions, 43
of spectral normalization, 783
unscented transformation, 47
with weight matrices, 43
vec operator, 775
vech operator, 775
vector representation
of conics and quadrics, 316
of transformations, 315
vertical parallax, 561
vertical view, 456
Vexcel Ultracam, 443–445, 683
view
horizontal v., 456
nadir v., 456
oblique v., 456
perspective v., 459
vertical v., 456
zenith v., 456
view planning, 715–722
for flat areas, 718
of buildings and rooms, 721
rules of thumb for v., 716
viewing angle, 371, **458**
viewing direction, 458, 462
from projection matrix, 475
normal case of image pair, 561
viewing position, **457**, 469, 550
viewing sphere, 423, 447
viewline, 445
viewpoint of camera, 445
visual odometry, 644
volume of tetrahedron, 225

weak configuration, 698
weak constraints, 102
weak membrane model, 740
weak string model, 739
weight, 83
coefficient matrix, 89
matrix, **43**, 83, 89
table of w. functions, 149
weight function
asymmetric w., 756
weighted
least squares, 79
sum of the squared residuals, 84
weights
in least squares estimation, 81
modified w., 147
whitening, 41
wide-angle lens, 459
Wiener filter, 93
Wishart distribution, 34
Woodbury identity, 769
world model, 7

Yule–Walker equations, 184

zenith
angle, 210
point, 346
view, 456
zoom lens, 459