

TEXTS IN COMPUTER SCIENCE

Computer Vision

Algorithms and Applications



Richard Szeliski

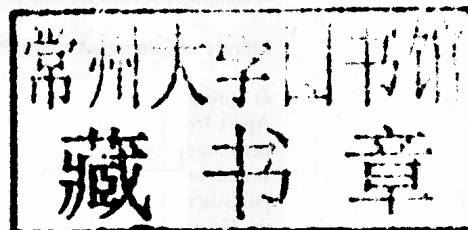


Springer

Richard Szeliski

Computer Vision

Algorithms and Applications



 Springer

Dr. Richard Szeliski
Microsoft Research
One Microsoft Way
98052-6399 Redmond
Washington
USA
szeliski@microsoft.com

Series Editors

David Gries
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

Fred B. Schneider
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

ISSN 1868-0941 e-ISSN 1868-095X
ISBN 978-1-84882-934-3 e-ISBN 978-1-84882-935-0
DOI 10.1007/978-1-84882-935-0
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2010936817

© Springer-Verlag London Limited 2011

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Texts in Computer Science

Editors

David Gries

Fred B. Schneider

For further volumes:

www.springer.com/series/3191

Book overview • Sample syllabus • Notation • What is computer vision? • A brief history



Geometric primitives and transformations • Electromagnetic image formation • The digital camera



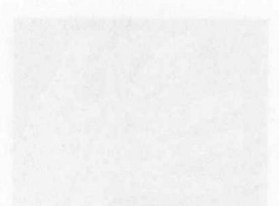
Point operators • Linear filtering • More neighborhood operators • Fourier transforms • Pyramids and wavelets • Geometric transformations



*This book is dedicated to my parents,
Zdzisław and Jadwiga,
and my family,
Lyn, Anne, and Stephen.*



Active contours • Split and merge • Mean shift and mode finding • Normalized cuts • Graph cuts and energy-based methods



2D and 3D feature-based alignment • Pose estimation • Geometric multiview calibration



Triangulation • Two-frame structure from motion • Factorization • Bundle adjustment • Constrained structure and motion



Preface

The seeds for this book were first planted in 2001 when Steve Seitz at the University of Washington invited me to co-teach a course called “Computer Vision for Computer Graphics”. At that time, computer vision techniques were increasingly being used in computer graphics to create image-based models of real-world objects, to create visual effects, and to merge real-world imagery using computational photography techniques. Our decision to focus on the applications of computer vision to fun problems such as image stitching and photo-based 3D modeling from personal photos seemed to resonate well with our students.

Since that time, a similar syllabus and project-oriented course structure has been used to teach general computer vision courses both at the University of Washington and at Stanford. (The latter was a course I co-taught with David Fleet in 2003.) Similar curricula have been adopted at a number of other universities and also incorporated into more specialized courses on computational photography. (For ideas on how to use this book in your own course, please see Table 1.1 in Section 1.4.)

This book also reflects my 20 years’ experience doing computer vision research in corporate research labs, mostly at Digital Equipment Corporation’s Cambridge Research Lab and at Microsoft Research. In pursuing my work, I have mostly focused on problems and solution techniques (algorithms) that have practical real-world applications and that work well in practice. Thus, this book has more emphasis on basic techniques that work under real-world conditions and less on more esoteric mathematics that has intrinsic elegance but less practical applicability.

This book is suitable for teaching a senior-level undergraduate course in computer vision to students in both computer science and electrical engineering. I prefer students to have either an image processing or a computer graphics course as a prerequisite so that they can spend less time learning general background mathematics and more time studying computer vision techniques. The book is also suitable for teaching graduate-level courses in computer vision (by delving into the more demanding application and algorithmic areas) and as a general reference to fundamental techniques and the recent research literature. To this end, I have attempted wherever possible to at least cite the newest research in each sub-field, even if the technical details are too complex to cover in the book itself.

In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects even turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended

problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative variants on the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

- **Scientific:** build detailed models of the image formation process and develop mathematical techniques to invert these in order to recover the quantities of interest (where necessary, making simplifying assumption to make the mathematics more tractable).
- **Statistical:** use probabilistic models to quantify the prior likelihood of your unknowns and the noisy measurement processes that produce the input images, then infer the best possible estimates of your desired quantities and analyze their resulting uncertainties. The inference algorithms used are often closely related to the optimization techniques used to invert the (scientific) image formation processes.
- **Engineering:** develop techniques that are simple to describe and implement but that are also known to work well in practice. Test these techniques to understand their limitation and failure modes, as well as their expected computational costs (run-time performance).

These three approaches build on each other and are used throughout the book.

My personal research and development philosophy (and hence the exercises in the book) have a strong emphasis on *testing* algorithms. It's too easy in computer vision to develop an algorithm that does something *plausible* on a few images rather than something *correct*. The best way to validate your algorithms is to use a three-part strategy.

First, test your algorithm on clean synthetic data, for which the exact results are known. Second, add noise to the data and evaluate how the performance degrades as a function of noise level. Finally, test the algorithm on real-world data, preferably drawn from a wide variety of sources, such as photos found on the Web. Only then can you truly know if your algorithm can deal with real-world complexity, i.e., images that do not fit some simplified model or assumptions.

In order to help students in this process, this book comes with a large amount of supplementary material, which can be found on the book's Web site <http://szeliski.org/Book>. This material, which is described in Appendix C, includes:

- pointers to commonly used data sets for the problems, which can be found on the Web
- pointers to software libraries, which can help students get started with basic tasks such as reading/writing images or creating and manipulating images
- slide sets corresponding to the material covered in this book
- a BibTeX bibliography of the papers cited in this book.

The latter two resources may be of more interest to instructors and researchers publishing new papers in this field, but they will probably come in handy even with regular students. Some of the software libraries contain implementations of a wide variety of computer vision algorithms, which can enable you to tackle more ambitious projects (with your instructor's consent).

Acknowledgements

I would like to gratefully acknowledge all of the people whose passion for research and inquiry as well as encouragement have helped me write this book.

Steve Zucker at McGill University first introduced me to computer vision, taught all of his students to question and debate research results and techniques, and encouraged me to pursue a graduate career in this area.

Takeo Kanade and Geoff Hinton, my Ph. D. thesis advisors at Carnegie Mellon University, taught me the fundamentals of good research, writing, and presentation. They fired up my interest in visual processing, 3D modeling, and statistical methods, while Larry Matthies introduced me to Kalman filtering and stereo matching.

Demetri Terzopoulos was my mentor at my first industrial research job and taught me the ropes of successful publishing. Yvan Leclerc and Pascal Fua, colleagues from my brief interlude at SRI International, gave me new perspectives on alternative approaches to computer vision.

During my six years of research at Digital Equipment Corporation's Cambridge Research Lab, I was fortunate to work with a great set of colleagues, including Ingrid Carlbom, Gudrun Klinker, Keith Waters, Richard Weiss, Stéphane Lavallée, and Sing Bing Kang, as well as to supervise the first of a long string of outstanding summer interns, including David Tonnesen, Sing Bing Kang, James Coughlan, and Harry Shum. This is also where I began my long-term collaboration with Daniel Scharstein, now at Middlebury College.

At Microsoft Research, I've had the outstanding fortune to work with some of the world's best researchers in computer vision and computer graphics, including Michael Cohen, Hugues Hoppe, Stephen Gortler, Steve Shafer, Matthew Turk, Harry Shum, Anandan, Phil Torr, Antonio Criminisi, Georg Petschnigg, Kentaro Toyama, Ramin Zabih, Shai Avidan, Sing Bing Kang, Matt Uyttendaele, Patrice Simard, Larry Zitnick, Richard Hartley, Simon Winder, Drew Steedly, Chris Pal, Nebojsa Jojic, Patrick Baudisch, Dani Lischinski, Matthew Brown, Simon Baker, Michael Goesele, Eric Stollnitz, David Nistér, Blaise Aguera y Arcas, Sudipta Sinha, Johannes Kopf, Neel Joshi, and Krishnan Ramnath. I was also lucky to have as interns such great students as Polina Golland, Simon Baker, Mei Han, Arno Schödl, Ron Dror, Ashley Eden, Jinxiang Chai, Rahul Swaminathan, Yanghai Tsin, Sam Hasinoff, Anat Levin, Matthew Brown, Eric Bennett, Vaibhav Vaish, Jan-Michael Frahm, James Diebel, Ce Liu, Josef Sivic, Grant Schindler, Colin Zheng, Neel Joshi, Sudipta Sinha, Zeev Farbman, Rahul Garg, Tim Cho, Yekeun Jeong, Richard Roberts, Varsha Hedau, and Dilip Krishnan.

While working at Microsoft, I've also had the opportunity to collaborate with wonderful colleagues at the University of Washington, where I hold an Affiliate Professor appointment. I'm indebted to Tony DeRose and David Salesin, who first encouraged me to get involved with the research going on at UW, my long-time collaborators Brian Curless, Steve Seitz, Maneesh Agrawala, Sameer Agarwal, and Yasu Furukawa, as well as the students I have had the privilege to supervise and interact with, including Frédéric Pighin, Yung-Yu Chuang, Doug Zongker, Colin Zheng, Aseem Agarwala, Dan Goldman, Noah Snavely, Rahul Garg, and Ryan Kaminsky. As I mentioned at the beginning of this preface, this book owes its inception to the vision course that Steve Seitz invited me to co-teach, as well as to Steve's encouragement, course notes, and editorial input.

I'm also grateful to the many other computer vision researchers who have given me so many constructive suggestions about the book, including Sing Bing Kang, who was my infor-

mal book editor, Vladimir Kolmogorov, who contributed Appendix B.5.5 on linear programming techniques for MRF inference, Daniel Scharstein, Richard Hartley, Simon Baker, Noah Snavely, Bill Freeman, Svetlana Lazebnik, Matthew Turk, Jitendra Malik, Alyosha Efros, Michael Black, Brian Curless, Sameer Agarwal, Li Zhang, Deva Ramanan, Olga Veksler, Yuri Boykov, Carsten Rother, Phil Torr, Bill Triggs, Bruce Maxwell, Jana Košecká, Eero Simoncelli, Aaron Hertzmann, Antonio Torralba, Tomaso Poggio, Theo Pavlidis, Baba Vemuri, Nando de Freitas, Chuck Dyer, Song Yi, Falk Schubert, Roman Pflugfelder, Marshall Tapen, James Coughlan, Sammy Rogmans, Klaus Strobel, Shanmuganathan, Andreas Siebert, Yongjun Wu, Fred Pighin, Juan Cockburn, Ronald Mallet, Tim Soper, Georgios Evangelidis, Dwight Fowler, Itzik Bayaz, Daniel O'Connor, and Srikrishna Bhat. Shena Deuchers did a fantastic job copy-editing the book and suggesting many useful improvements and Wayne Wheeler and Simon Rees at Springer were most helpful throughout the whole book publishing process. Keith Price's Annotated Computer Vision Bibliography was invaluable in tracking down references and finding related work.

If you have any suggestions for improving the book, please send me an e-mail, as I would like to keep the book as accurate, informative, and timely as possible.

Lastly, this book would not have been possible or worthwhile without the incredible support and encouragement of my family. I dedicate this book to my parents, Zdzisław and Jadwiga, whose love, generosity, and accomplishments have always inspired me; to my sister Basia for her lifelong friendship; and especially to Lyn, Anne, and Stephen, whose daily encouragement in all matters (including this book project) makes it all worthwhile.

Lake Wenatchee

August, 2010

Contents

Preface	ix
1 Introduction	1
1.1 What is computer vision?	3
1.2 A brief history	10
1.3 Book overview	17
1.4 Sample syllabus	23
1.5 A note on notation	25
1.6 Additional reading	25
2 Image formation	27
2.1 Geometric primitives and transformations	29
2.1.1 Geometric primitives	29
2.1.2 2D transformations	33
2.1.3 3D transformations	36
2.1.4 3D rotations	37
2.1.5 3D to 2D projections	42
2.1.6 Lens distortions	52
2.2 Photometric image formation	54
2.2.1 Lighting	54
2.2.2 Reflectance and shading	55
2.2.3 Optics	61
2.3 The digital camera	65
2.3.1 Sampling and aliasing	69
2.3.2 Color	71
2.3.3 Compression	80
2.4 Additional reading	82
2.5 Exercises	82
3 Image processing	87
3.1 Point operators	89
3.1.1 Pixel transforms	91
3.1.2 Color transforms	92
3.1.3 Compositing and matting	92
3.1.4 Histogram equalization	94

3.1.5	<i>Application: Tonal adjustment</i>	97
3.2	Linear filtering	98
3.2.1	Separable filtering	102
3.2.2	Examples of linear filtering	103
3.2.3	Band-pass and steerable filters	104
3.3	More neighborhood operators	108
3.3.1	Non-linear filtering	108
3.3.2	Morphology	112
3.3.3	Distance transforms	113
3.3.4	Connected components	115
3.4	Fourier transforms	116
3.4.1	Fourier transform pairs	119
3.4.2	Two-dimensional Fourier transforms	123
3.4.3	Wiener filtering	123
3.4.4	<i>Application: Sharpening, blur, and noise removal</i>	126
3.5	Pyramids and wavelets	127
3.5.1	Interpolation	127
3.5.2	Decimation	130
3.5.3	Multi-resolution representations	132
3.5.4	Wavelets	136
3.5.5	<i>Application: Image blending</i>	140
3.6	Geometric transformations	143
3.6.1	Parametric transformations	145
3.6.2	Mesh-based warping	149
3.6.3	<i>Application: Feature-based morphing</i>	152
3.7	Global optimization	153
3.7.1	Regularization	154
3.7.2	Markov random fields	158
3.7.3	<i>Application: Image restoration</i>	169
3.8	Additional reading	169
3.9	Exercises	171
4	Feature detection and matching	181
4.1	Points and patches	183
4.1.1	Feature detectors	185
4.1.2	Feature descriptors	196
4.1.3	Feature matching	200
4.1.4	Feature tracking	207
4.1.5	<i>Application: Performance-driven animation</i>	209
4.2	Edges	210
4.2.1	Edge detection	210
4.2.2	Edge linking	215
4.2.3	<i>Application: Edge editing and enhancement</i>	219
4.3	Lines	220
4.3.1	Successive approximation	220
4.3.2	Hough transforms	221

4.3.3	Vanishing points	224
4.3.4	<i>Application: Rectangle detection</i>	226
4.4	Additional reading	227
4.5	Exercises	228
5	Segmentation	235
5.1	Active contours	237
5.1.1	Snakes	238
5.1.2	Dynamic snakes and CONDENSATION	243
5.1.3	Scissors	246
5.1.4	Level Sets	248
5.1.5	<i>Application: Contour tracking and rotoscoping</i>	249
5.2	Split and merge	250
5.2.1	Watershed	251
5.2.2	Region splitting (divisive clustering)	251
5.2.3	Region merging (agglomerative clustering)	251
5.2.4	Graph-based segmentation	252
5.2.5	Probabilistic aggregation	253
5.3	Mean shift and mode finding	254
5.3.1	K-means and mixtures of Gaussians	256
5.3.2	Mean shift	257
5.4	Normalized cuts	260
5.5	Graph cuts and energy-based methods	264
5.5.1	<i>Application: Medical image segmentation</i>	268
5.6	Additional reading	268
5.7	Exercises	270
6	Feature-based alignment	273
6.1	2D and 3D feature-based alignment	275
6.1.1	2D alignment using least squares	275
6.1.2	<i>Application: Panography</i>	277
6.1.3	Iterative algorithms	278
6.1.4	Robust least squares and RANSAC	281
6.1.5	3D alignment	283
6.2	Pose estimation	284
6.2.1	Linear algorithms	284
6.2.2	Iterative algorithms	286
6.2.3	<i>Application: Augmented reality</i>	287
6.3	Geometric intrinsic calibration	288
6.3.1	Calibration patterns	289
6.3.2	Vanishing points	290
6.3.3	<i>Application: Single view metrology</i>	292
6.3.4	Rotational motion	293
6.3.5	Radial distortion	295
6.4	Additional reading	296
6.5	Exercises	296

7	Structure from motion	303
7.1	Triangulation	305
7.2	Two-frame structure from motion	307
7.2.1	Projective (uncalibrated) reconstruction	312
7.2.2	Self-calibration	313
7.2.3	<i>Application: View morphing</i>	315
7.3	Factorization	315
7.3.1	Perspective and projective factorization	318
7.3.2	<i>Application: Sparse 3D model extraction</i>	319
7.4	Bundle adjustment	320
7.4.1	Exploiting sparsity	322
7.4.2	<i>Application: Match move and augmented reality</i>	324
7.4.3	Uncertainty and ambiguities	326
7.4.4	<i>Application: Reconstruction from Internet photos</i>	327
7.5	Constrained structure and motion	329
7.5.1	Line-based techniques	330
7.5.2	Plane-based techniques	331
7.6	Additional reading	332
7.7	Exercises	332
8	Dense motion estimation	335
8.1	Translational alignment	337
8.1.1	Hierarchical motion estimation	341
8.1.2	Fourier-based alignment	341
8.1.3	Incremental refinement	345
8.2	Parametric motion	350
8.2.1	<i>Application: Video stabilization</i>	354
8.2.2	Learned motion models	354
8.3	Spline-based motion	355
8.3.1	<i>Application: Medical image registration</i>	358
8.4	Optical flow	360
8.4.1	Multi-frame motion estimation	363
8.4.2	<i>Application: Video denoising</i>	364
8.4.3	<i>Application: De-interlacing</i>	364
8.5	Layered motion	365
8.5.1	<i>Application: Frame interpolation</i>	368
8.5.2	Transparent layers and reflections	368
8.6	Additional reading	370
8.7	Exercises	371
9	Image stitching	375
9.1	Motion models	378
9.1.1	Planar perspective motion	379
9.1.2	<i>Application: Whiteboard and document scanning</i>	379
9.1.3	Rotational panoramas	380
9.1.4	Gap closing	382

9.1.5	<i>Application: Video summarization and compression</i>	383
9.1.6	Cylindrical and spherical coordinates	385
9.2	Global alignment	387
9.2.1	Bundle adjustment	388
9.2.2	Parallax removal	391
9.2.3	Recognizing panoramas	392
9.2.4	Direct vs. feature-based alignment	393
9.3	Compositing	396
9.3.1	Choosing a compositing surface	396
9.3.2	Pixel selection and weighting (de-ghosting)	398
9.3.3	<i>Application: Photomontage</i>	403
9.3.4	Blending	403
9.4	Additional reading	406
9.5	Exercises	407
10	Computational photography	409
10.1	Photometric calibration	412
10.1.1	Radiometric response function	412
10.1.2	Noise level estimation	415
10.1.3	Vignetting	416
10.1.4	Optical blur (spatial response) estimation	416
10.2	High dynamic range imaging	419
10.2.1	Tone mapping	427
10.2.2	<i>Application: Flash photography</i>	434
10.3	Super-resolution and blur removal	436
10.3.1	Color image demosaicing	440
10.3.2	<i>Application: Colorization</i>	442
10.4	Image matting and compositing	443
10.4.1	Blue screen matting	445
10.4.2	Natural image matting	446
10.4.3	Optimization-based matting	450
10.4.4	Smoke, shadow, and flash matting	452
10.4.5	Video matting	454
10.5	Texture analysis and synthesis	455
10.5.1	<i>Application: Hole filling and inpainting</i>	457
10.5.2	<i>Application: Non-photorealistic rendering</i>	458
10.6	Additional reading	460
10.7	Exercises	461
11	Stereo correspondence	467
11.1	Epipolar geometry	471
11.1.1	Rectification	472
11.1.2	Plane sweep	474
11.2	Sparse correspondence	475
11.2.1	3D curves and profiles	476
11.3	Dense correspondence	477

11.3.1	Similarity measures	479
11.4	Local methods	480
11.4.1	Sub-pixel estimation and uncertainty	482
11.4.2	<i>Application: Stereo-based head tracking</i>	483
11.5	Global optimization	484
11.5.1	Dynamic programming	485
11.5.2	Segmentation-based techniques	487
11.5.3	<i>Application: Z-keying and background replacement</i>	489
11.6	Multi-view stereo	489
11.6.1	Volumetric and 3D surface reconstruction	492
11.6.2	Shape from silhouettes	497
11.7	Additional reading	499
11.8	Exercises	500
12	3D reconstruction	505
12.1	Shape from X	508
12.1.1	Shape from shading and photometric stereo	508
12.1.2	Shape from texture	510
12.1.3	Shape from focus	511
12.2	Active rangefinding	512
12.2.1	Range data merging	515
12.2.2	<i>Application: Digital heritage</i>	517
12.3	Surface representations	518
12.3.1	Surface interpolation	518
12.3.2	Surface simplification	520
12.3.3	Geometry images	520
12.4	Point-based representations	521
12.5	Volumetric representations	522
12.5.1	Implicit surfaces and level sets	522
12.6	Model-based reconstruction	523
12.6.1	Architecture	524
12.6.2	Heads and faces	526
12.6.3	<i>Application: Facial animation</i>	528
12.6.4	Whole body modeling and tracking	530
12.7	Recovering texture maps and albedos	534
12.7.1	Estimating BRDFs	536
12.7.2	<i>Application: 3D photography</i>	537
12.8	Additional reading	538
12.9	Exercises	539
13	Image-based rendering	543
13.1	View interpolation	545
13.1.1	View-dependent texture maps	547
13.1.2	<i>Application: Photo Tourism</i>	548
13.2	Layered depth images	549
13.2.1	Impostors, sprites, and layers	549

13.3	Light fields and Lumigraphs	551
13.3.1	Unstructured Lumigraph	554
13.3.2	Surface light fields	555
13.3.3	<i>Application: Concentric mosaics</i>	556
13.4	Environment mattes	556
13.4.1	Higher-dimensional light fields	558
13.4.2	The modeling to rendering continuum	559
13.5	Video-based rendering	560
13.5.1	Video-based animation	560
13.5.2	Video textures	561
13.5.3	<i>Application: Animating pictures</i>	564
13.5.4	3D Video	564
13.5.5	<i>Application: Video-based walkthroughs</i>	566
13.6	Additional reading	569
13.7	Exercises	570
14	Recognition	575
14.1	Object detection	578
14.1.1	Face detection	578
14.1.2	Pedestrian detection	585
14.2	Face recognition	588
14.2.1	Eigenfaces	589
14.2.2	Active appearance and 3D shape models	596
14.2.3	<i>Application: Personal photo collections</i>	601
14.3	Instance recognition	602
14.3.1	Geometric alignment	603
14.3.2	Large databases	604
14.3.3	<i>Application: Location recognition</i>	609
14.4	Category recognition	611
14.4.1	Bag of words	612
14.4.2	Part-based models	615
14.4.3	Recognition with segmentation	620
14.4.4	<i>Application: Intelligent photo editing</i>	621
14.5	Context and scene understanding	625
14.5.1	Learning and large image collections	627
14.5.2	<i>Application: Image search</i>	630
14.6	Recognition databases and test sets	631
14.7	Additional reading	631
14.8	Exercises	637
15	Conclusion	641
A	Linear algebra and numerical techniques	645
A.1	Matrix decompositions	646
A.1.1	Singular value decomposition	646
A.1.2	Eigenvalue decomposition	647

A.1.3	QR factorization	649
A.1.4	Cholesky factorization	650
A.2	Linear least squares	651
A.2.1	Total least squares	653
A.3	Non-linear least squares	654
A.4	Direct sparse matrix techniques	655
A.4.1	Variable reordering	656
A.5	Iterative techniques	656
A.5.1	Conjugate gradient	657
A.5.2	Preconditioning	659
A.5.3	Multigrid	660
B	Bayesian modeling and inference	661
B.1	Estimation theory	662
B.1.1	Likelihood for multivariate Gaussian noise	663
B.2	Maximum likelihood estimation and least squares	665
B.3	Robust statistics	666
B.4	Prior models and Bayesian inference	667
B.5	Markov random fields	668
B.5.1	Gradient descent and simulated annealing	670
B.5.2	Dynamic programming	670
B.5.3	Belief propagation	672
B.5.4	Graph cuts	674
B.5.5	Linear programming	676
B.6	Uncertainty estimation (error analysis)	678
C	Supplementary material	679
C.1	Data sets	680
C.2	Software	682
C.3	Slides and lectures	689
C.4	Bibliography	690
	References	691
	Index	793