Computer Vision
A Modern Approach

David A. Forsyth
University of California at Berkeley

Jean Ponce
University of Illinois at Urbana-Champaign

An Alan R. Apt Book

Prentice Hall
Hall

Prentice Hall
Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data

CIP data on file.

Vice President and Editorial Director, ECS: *Marcia J. Horton*
Publisher: *Alan Apt*
Associate Editor: *Toni D. Holm*
Editorial Assistant: *Patrick Lindner*
Vice President and Director of Production and Manufacturing, ESM: *David W. Riccardi*
Executive Managing Editor: *Vince O'Brien*
Assistant Managing Editor: *Camille Trentacoste*
Production Editor: *Leslie Galen*
Director of Creative Services: *Paul Belfanti*
Creative Director: *Carole Anson*
Art Director: *Wanda Espana*
Art Editor: *Greg Dulles*
Manufacturing Manager: *Trudy Pisciotti*
Manufacturing Buyer: *Lynda Castillo*
Marketing Manager: *Pamela Shaffer*
Marketing Assistant: *Barrie Reinhold*

About the cover: Image courtesy of Gamma/Superstock

Prentice Hall

Pearson Education, Inc.
Upper Saddle River, NJ 07458

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-085198-1

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educacíon de Mexico, S.A. de C.V.
Pearson Education–Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, *Upper Saddle River, New Jersey*

To my family—DAF

To Camille and Oscar—JP

Contents

Preface

Computer vision as a field is an intellectual frontier. Like any frontier, it is exciting and disorganised; there is often no reliable authority to appeal to—many useful ideas have no theoretical grounding, and some theories are useless in practice; developed areas are widely scattered, and often one looks completely inaccessible from the other. Nevertheless, we have attempted in this book to present a fairly orderly picture of the field.

We see computer vision—or just "vision"; apologies to those who study human or animal vision—as an enterprise that uses statistical methods to disentangle data using models constructed with the aid of geometry, physics and learning theory. Thus, in our view, vision relies on a solid understanding of cameras and of the physical process of image formation (part I of this book) to obtain simple inferences from individual pixel values (part II), combine the information available in multiple images into a coherent whole (part III), impose some order on groups of pixels to separate them from each other or infer shape information (part IV), and recognize objects using geometric information (part V) or probabilistic techniques (part VI). Computer vision has a wide variety of applications, old (e.g., mobile robot navigation, industrial inspection, and military intelligence) and new (e.g., human computer interaction, image retrieval in digital libraries, medical image analysis, and the realistic rendering of synthetic scenes in computer graphics). We discuss some of these applications in part VII.

WHY STUDY VISION?

Computer vision's great trick is extracting descriptions of the world from pictures or sequences of pictures. This is unequivocally useful. Taking pictures is usually non-destructive and sometimes discreet. It is also easy and (now) cheap. The descriptions that users seek can differ widely between applications. For example, a technique known as structure from motion makes it possible to extract a representation of what is depicted and how the camera moved from a series of pictures. People in the entertainment industry use these techniques to build three-dimensional (3D) computer models of buildings, typically keeping the structure and throwing away the motion. These models are used where real buildings cannot be; they are set fire to, blown up, etc. Good, simple, accurate and convincing models can be built from quite small sets of photographs. People who wish to control mobile robots usually keep the motion and throw away the structure. This is because they generally know something about the area where the robot is working, but don't usually know the precise robot location in that area. They can determine it from information about how a camera bolted to the robot is moving.

There are a number of other, important applications of computer vision. One is in medical imaging: One builds software systems that can enhance imagery, or identify important phenomena or events, or visualize information obtained by imaging. Another is in inspection: One takes pictures of objects to determine whether they are within specification. A third is in interpreting satellite images, both for military purposes—a program might be required to determine what militarily interesting phenomena have occurred in a given region recently; or what damage was caused by a bombing—and for civilian purposes—what will this year's maize crop be? How much rainforest is left? A fourth is in organizing and structuring collections of pictures. We know how to search and browse text libraries (though this is a subject that still has difficult open questions) but don't really know what to do with image or video libraries.

Computer vision is at an extraordinary point in its development. The subject itself has been around since the 1960s, but it is only recently that it has been possible to build useful computer systems using ideas from computer vision. This flourishing has been driven by several trends: Computers and imaging systems have become very cheap. Not all that long ago, it took tens of thousands of dollars to get good digital color images; now it takes a few hundred, at most. Not all that long ago, a color printer was something one found in few, if any, research labs; now they are in many homes. This means it is easier to do research. It also means that there are many people with problems to which the methods of computer vision apply. For example, people would like to organize their collection of photographs, make 3D models of the world around them, and manage and edit collections of videos. Our understanding of the basic geometry and physics underlying vision and, what is more important, what to do about it, has improved significantly. We are beginning to be able to solve problems that lots of people care about, but none of the hard problems have been solved and there are plenty of easy ones that have not been solved either (to keep one intellectually fit while trying to solve hard problems). It is a great time to be studying this subject.

What Is in This Book?

This book covers what we feel a computer vision professional ought to know. However, it is addressed to a wider audience. We hope that those engaged in computational geometry, computer graphics, image processing, imaging in general, and robotics will find it an informative reference. We have tried to make the book accessible to senior undergraduates or graduate students with a passing interest in vision. Each chapter covers a different part of the subject, and, as a glance at Table 1 will confirm, chapters are relatively independent. This means that one can dip into the book as well as read it from cover to cover. Generally, we have tried to make chapters run from easy material at the start to more arcane matters at the end. Each chapter has brief notes at the end, containing historical material and assorted opinions. We have tried to produce a book that describes ideas that are useful, or likely to be so in the future. We have put emphasis on understanding the basic geometry and physics of imaging, but have tried to link this with actual applications. In general, the book reflects the enormous recent influence of geometry and various forms of applied statistics on computer vision.

A reader who goes from cover to cover will hopefully be well informed, if exhausted; there is too much in this book to cover in a one-semester class. Of course, prospective (or active) computer vision professionals should read every word, do all the exercises, and report any bugs found for the second edition (of which it is probably a good idea to plan buying a copy!). While the study of computer vision does not require deep mathematics, it does require facility with a lot of different mathematical ideas. We have tried to make the book self contained, in the sense that readers with the level of mathematical sophistication of an engineering senior should be comfortable with the material of the book, and should not need to refer to other texts. We have also tried to keep the mathematics to the necessary minimum—after all, this book is about computer vision, not applied mathematics—and have chosen to insert what mathematics we have kept in the main chapter bodies instead of a separate appendix.

Generally, we have tried to reduce the interdependence between chapters, so that readers interested in particular topics can avoid wading through the whole book. It is not possible to make each chapter entirely self contained, and Table 1 indicates the dependencies between chapters.

TABLE 1 Dependencies between chapters: It will be difficult to read a chapter if you don't have a good grasp of the material in the chapters it "requires." If you have not read the chapters labeled "helpful," you may need to look one or two things up.

Part	Chapter	Requires	Helpful
I	1: Cameras		
	2: Geometric camera models	1	
	3: Geometric camera calibration	2	
	4: Radiometry—measuring light		
	5: Sources, shadows and shading		4, 1
	6: Color		5
II	7: Linear filters		
	8: Edge detection	7	
	9: Texture	7	8
III	10: The geometry of multiple views	3	
	11: Stereopsis	10	
	12: Affine structure from motion	10	
	13: Projective structure from motion	12	
IV	14: Segmentation by clustering		9, 6, 5
	15: Segmentation by fitting a model		14
	16: Segmentation and fitting using probabilistic methods		15,10
	17: Tracking with linear dynamic models		
V	18: Model-based vision	3	
	19: Smooth surfaces and their outlines	2	
	20: Aspect graphs	19	
	21: Range data		20, 19, 3
VI	22: Finding templates using classifiers		9, 8, 7, 6, 5
	23: Recognition by relations between templates		9, 8, 7, 6, 5
	24: Geometric templates from spatial relations	2, 1	16, 15, 14
VII	25: Application: Finding in digital libraries		16, 15, 14, 6
	26: Application: Image-based rendering	10	13, 12, 11, 6, 5, 3, 2, 1

What Is Not in This Book

The computer vision literature is vast, and it was not easy to produce a book about computer vision that can be lifted by ordinary mortals. To do so, we had to cut material, ignore topics, and so on. We cut two entire chapters close to the last moment: One is an introduction to probability and inference, the other an account of methods for tracking objects with non-linear dynamics. These chapters appear on the book's web page `http://www.cs.berkeley.edu/~daf/book.html`.

We left out some topics because of personal taste, or because we became exhausted and stopped writing about a particular area, or because we learned about them too late to put them in, or because we had to shorten some chapter, or any of hundreds of other reasons. We have tended to omit detailed discussions of material that is mainly of historical interest, and offer instead some historical remarks at the end of each chapter. Neither of us claims to be a fluent intellectual archaeologist, meaning that ideas may have deeper histories than we have indicated. We just didn't get around to writing up deformable templates and mosaics, two topics of considerable practical importance; we will try to put them into the second edition.

ACKNOWLEDGMENTS

In preparing this book, we have accumulated a significant set of debts. A number of anonymous reviewers have read several drafts of the book and have made extremely helpful contributions. We are grateful to them for their time and efforts. Our editor, Alan Apt, organized these reviews with the help of Jake Warde. We thank them both. Leslie Galen, Joe Albrecht, and Dianne Parish, of Integre Technical Publishing, helped us over numerous issues with proofreading and illustrations. Some images used herein were obtained from IMSI's Master Photos Collection, 1895 Francisco Blvd. East, San Rafael, CA 94901-5506, USA. In preparing the bibliography, we have made extensive use of Keith Price's excellent computer vision bibliography, which can be found at `http://iris.usc.edu/Vision-Notes/bibliography/contents.html`.

Both the overall coverage of topics and several chapters were reviewed by various colleagues, who made valuable and detailed suggestions for their revision. We thank Kobus Barnard, Margaret Fleck, David Kriegman, Jitendra Malik and Andrew Zisserman. A number of our students contributed suggestions, ideas for figures, proofreading comments, and other valuable material. We thank Okan Arikan, Sébastien Blind, Martha Cepeda, Stephen Chenney, Frank Cho, Yakup Genc, John Haddon, Sergey Ioffe, Svetlana Lazebnik, Cathy Lee, Sung-il Pae, David Parks, Fred Rothganger, Attawith Sudsang, and the students in several offerings of our vision classes at U.C. Berkeley and UIUC. We have been very lucky to have colleagues at various universities use (often rough) drafts of our book in their vision classes. Institutions whose students suffered through these drafts include, besides ours, Carnegie-Mellon University, Stanford University, the University of Wisconsin at Madison, the University of California at Santa Barbara and the University of Southern California; there may be others we are not aware of. We are grateful for all the helpful comments from adopters, in particular Chris Bregler, Chuck Dyer, Martial Hebert, David Kriegman, B.S. Manjunath, and Ram Nevatia, who sent us many detailed and helpful comments and corrections. The book has also benefitted from comments and corrections from Aydin Alaylioglu, Srinivas Akella, Marie Banich, Serge Belongie, Ajit M. Chaudhari, Navneet Dalal, Richard Hartley, Glen Healey, Mike Heath, Hayley Iben, Stéphanie Jonquières, Tony Lewis, Benson Limketkai, Simon Maskell, Brian Milch, Tamara Miller, Cordelia Schmid, Brigitte and Gerry Serlin, Ilan Shimshoni, Eric de Sturler, Camillo J. Taylor, Jeff Thompson, Claire Vallat, Daniel S. Wilkerson, Jinghan Yu, Hao Zhang, and Zhengyou Zhang. If you find an apparent typographic error, please email DAF (daf@cs.berkeley.edu) with the details, using the phrase "book typo" in your email; we will try to credit the first finder of each typo in the second edition.

We also thank P. Besl, B. Boufama, J. Costeira, P. Debevec, O. Faugeras, Y. Genc, M. Hebert, D. Huber, K. Ikeuchi, A.E. Johnson, T. Kanade, K. Kutulakos, M. Levoy, S. Mahamud, R. Mohr, H. Moravec, H. Murase, Y. Ohta, M. Okutami, M. Pollefeys, H. Saito, C. Schmid, S. Sullivan, C. Tomasi, and M. Turk for providing the originals of some of the figures shown in this book.

DAF acknowledges a wide range of intellectual debts, starting at kindergarten. Important figures in the very long list of his creditors include Gerald Alanthwaite, Mike Brady, Tom Fair, Margaret Fleck, Jitendra Malik, Joe Mundy, Mike Rodd, Charlie Rothwell and Andrew Zisserman. JP cannot even remember kindergarten, but acknowledges his debts to Olivier Faugeras, Mike Brady, and Tom Binford. He also wishes to thank Sharon Collins for her help. Without her, this book, like most of his work, probably would have never been finished. Both authors would also like to acknowledge the profound influence of Jan Koenderink's writings on their work at large and on this book in particular.

SAMPLE SYLLABI

The whole book can be covered in two (rather intense) semesters, by starting at the first page and plunging on. Ideally, one would cover one application chapter—probably the chapter on image-based rendering—in the first semester, and the other one in the second. Few departments will experience heavy demand for so detailed a sequence of courses. We have tried to structure this book so that instructors can choose areas according to taste. Sample syllabi for busy 15-week semesters appear in Tables 2 to 6, structured according to needs that can reasonably be expected. We would encourage (and expect!) instructors to rearrange these according to taste.

Table 2 contains a suggested syllabus for a one-semester introductory class in computer vision for seniors or first-year graduate students in computer science, electrical engineering, or other engineering or science disciplines. The students receive a broad presentation of the field, including application areas such as digital libraries and image-based rendering. Although the

TABLE 2 A one-semester introductory class in computer vision for seniors or first-year graduate students in computer science, electrical engineering, or other engineering or science disciplines.

Week	Chapter	Sections	Key topics
1	1, 4	1.1, 4 (summary only)	pinhole cameras, radiometric terminology
2	5	5.1–5.5	local shading models; point, line and area sources; photometric stereo
3	6	all	color
4	7, 8	7.1–7.5, 8.1–8.3	linear filters; smoothing to suppress noise; edge detection
5	9	all	texture: as statistics of filter outputs; synthesis; shape from
6	10, 11	10.1, 11	basic multi-view geometry; stereo
7	14	all	segmentation as clustering
8	15	15.1–15.4	fitting lines, curves; fitting as maximum likelihood; robustness
9	16	16.1,16.2	hidden variables and EM
10	17	all	tracking with a Kalman filter; data association
11	2, 3	2.1, 2.2, all of 3	camera calibration
12	18	all	model-based vision using correspondence and camera calibration
13	22	all	template matching using classifiers
14	23	all	matching on relations
15	25, 26	all	finding images in digital libraries; image based rendering

TABLE 3 A syllabus for students of computer graphics who want to know the elements of vision that are relevant to their topic.

Week	Chapter	Sections	Key topics
1	1, 4	1.1, 4 (summary only)	pinhole cameras, radiometric terminology
2	5	5.1–5.5	local shading models; point, line and area sources; photometric stereo
3	6.1–6.4	all	color
4	7, 8	7.1–7.5, 8.1–8.3	linear filters; smoothing to suppress noise; edge detection
5	9	9.1–9.3	texture: as statistics of filter outputs; synthesis
6	2, 3	2.1, 2.2, all of 3	camera calibration
7	10, 11	10.1, 11	basic multi-view geometry; stereo
8	12	all	affine structure from motion
9	13	all	projective structure from motion
10	26	all	image-based rendering
11	15	all	fitting; robustness; RANSAC
12	16	all	hidden variables and EM
13	19	all	surfaces and outlines
14	21	all	range data
15	17	all	tracking, the Kalman filter and data association

TABLE 4 A syllabus for students who are primarily interested in the applications of computer vision.

Week	Chapter	Sections	Key topics
1	1, 4	1.1, 4 (summary only)	pinhole cameras, radiometric terminology
2	5, 6	5.1,5.3, 5.4, 5.5, 6.1–6.4	local shading models; point, line and area sources; photometric stereo; color—physics, human perception, color spaces
3	2, 3	all	camera models and their calibration
4	7, 9	all of 7; 9.1–9.3	linear filters; texture as statistics of filter outputs; texture synthesis
5	10, 11	all	multiview geometry, stereo as an example
6	12,13	all	affine structure from motion; projective structure from motion
7	13, 26	all	projective structure from motion; image-based rendering
8	14	all	segmentation as clustering, particular emphasis on shot boundary detection and background subtraction
9	15	all	fitting lines, curves; robustness; RANSAC
10	16	all	hidden variables and EM
11	25	all	finding images in digital libraries
12	17	all	tracking, the Kalman filter and data association
13	18	all	model-based vision
14	22	all	finding templates using classifiers
15	20	all	range data

hardest theoretical material is omitted, there is a thorough treatment of the basic geometry and physics of image formation. We assume that students will have a wide range of backgrounds, and can be assigned background readings in probability (we suggest the chapter on the book's web page) around week 2 or 3. We have put off the application chapters to the end, but many may prefer to do chapter 20 around week 10 and chapter 21 around week 6.

Table 3 contains a syllabus for students of computer graphics who want to know the elements of vision that are relevant to their topic. We have emphasized methods that make it possible to recover object models from image information; understanding these topics needs a working knowledge of cameras and filters. Tracking is becoming useful in the graphics world, where it is particularly important for motion capture. We assume that students will have a wide range of backgrounds, and have some exposure to probability.

Table 4 shows a syllabus for students who are primarily interested in the applications of computer vision. We cover material of most immediate practical interest. We assume that students will have a wide range of backgrounds, and can be assigned background reading on probability around week 2 or 3.

Table 5 is a suggested syllabus for students of cognitive science or artificial intelligence who want a basic outline of the important notions of computer vision. This syllabus is less aggressively paced, and assumes less mathematical experience. Students will need to read some material on probability (e.g., the chapter on the book's web page) around week 2 or 3.

Table 6 shows a sample syllabus for students who have a strong interest in applied mathematics, electrical engineering or physics. This syllabus makes for a very busy semester; we move fast, assuming that students can cope with a lot of mathematical material. We assume that students will have a wide range of backgrounds, and can be assigned some reading on probability around week 2 or 3. As a break in a pretty abstract and demanding syllabus, we have inserted a brief review of digital libraries; the chapter on image-based rendering or that on range data could be used instead.

TABLE 5 For students of cognitive science or artificial intelligence who want a basic outline of the important notions of computer vision.

Week	Chapter	Sections	Key topics
1	1, 4	1, 4 (summary only)	pinhole cameras; lenses; cameras and the eye; radiometric terminology
2	5	all	local shading models; point, line and area sources; photometric stereo; interreflections; lightness computations
3	6	all	color: physics, human perception, spaces; image models; color constancy
4	7	7.1–7.5, 7.7	linear filters; sampling; scale
5	8	all	edge detection
6	9	all	texture; representation, synthesis, shape from
7	10.1,10.2	all	basic multiple view geometry
8	11	all	stereopsis
9	14	all	segmentation by clustering
10	15	all	fitting lines, curves; robustness; RANSAC
11	16	all	hidden variables and EM
12	18	all	model-based vision
13	22	all	finding templates using classifiers
14	23	all	recognition by relations between templates
15	24	all	geometric templates from spatial relations

TABLE 6 A syllabus for students who have a strong interest in applied mathematics, electrical engineering or physics.

Week	Chapter	Sections	Key topics
1	1, 4	all	cameras, radiometry
2	5	all	shading models; point, line and area sources; photometric stereo; interreflections and shading primitives
3	6	all	color:—physics, human perception, spaces, color constancy
4	2, 3	all	camera parameters and calibration
5	7, 8	all	linear filters and edge detection
6	8, 9	all	finish edge detection; texture: representation, synthesis, shape from
7	10, 11	all	multiple view geometry, stereopsis as an example
8	12, 13	all	structure from motion
9	14, 15	all	segmentation as clustering; fitting lines, curves; robustness; RANSAC
10	15, 16	all	finish fitting; hidden variables and EM
11	17, 25	all	tracking: Kalman filters, data association; finding images in digital libraries
12	18	all	model-based vision
13	19	all	surfaces and their outlines
14	20	all	aspect graphs
15	22	all	template matching

NOTATION

We use the following notation throughout the book: points, lines, and planes are denoted by Roman or Greek letters in italic font (e.g., P, Δ, or Π). Vectors are usually denoted by Roman or Greek bold-italic letters (e.g., $\boldsymbol{v}$, $\boldsymbol{P}$, or $\boldsymbol{\xi}$), but the vector joining two points P and Q is often denoted by $\overrightarrow{PQ}$. Lower-case letters are normally used to denote geometric figures in the image plane (e.g., p, $\boldsymbol{p}$, δ), and upper-case letters are used for scene objects (e.g., P, Π). Matrices are denoted by Roman letters in calligraphic font (e.g., $\mathcal{U}$).

The familiar three-dimensional Euclidean space is denoted by $\mathbb{E}^3$, and the vector space formed by n-tuples of real numbers with the usual laws of addition and multiplication by a scalar is denoted by $\mathbb{R}^n$, with $\mathbf{0}$ being used to denote the zero vector. Likewise, the vector space formed by $m \times n$ matrices with real entries is denoted by $\mathbb{R}^{m \times n}$. When $m = n$, Id is used to denote the identity matrix—that is, the $n \times n$ matrix whose diagonal entries are equal to 1 and nondiagonal entries are equal to 0. The transpose of the $m \times n$ matrix $\mathcal{U}$ with coefficients u_{ij} is the $n \times m$ matrix denoted by $\mathcal{U}^T$ with coefficients u_{ji}. Elements of $\mathbb{R}^n$ are often identified with column vectors or $n \times 1$ matrices, e.g., $\boldsymbol{a} = (a_1, a_2, a_3)^T$ is the transpose of a 1×3 matrix (or *row vector*), i.e., an 3×1 matrix (or *column vector*), or equivalently an element of $\mathbb{R}^3$.

The *dot product* (or *inner product*) of two vectors $\boldsymbol{a} = (a_1, \dots, a_n)^T$ and $\boldsymbol{b} = (b_1, \dots, b_n)^T$ in $\mathbb{R}^n$ is defined by

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_1 b_1 + \cdots + a_n b_n,$$

and it can also be written as a matrix product, i.e., $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = \boldsymbol{b}^T \boldsymbol{a}$. We denote by $|\boldsymbol{a}|^2 = \boldsymbol{a} \cdot \boldsymbol{a}$ the square of the Euclidean norm of the vector $\boldsymbol{a}$ and denote by d the distance function induced by the Euclidean norm in $\mathbb{E}^n$, i.e., $d(P, Q) = |\overrightarrow{PQ}|$. Given a matrix $\mathcal{U}$ in $\mathbb{R}^{m\times n}$, we generally use $|U|$ to denote its *Frobenius norm*, i.e., the square root of the sum of its squared entries.

When the vector $\boldsymbol{a}$ has unit norm, the dot product $\boldsymbol{a}\cdot\boldsymbol{b}$ is equal to the (signed) length of the projection of $\boldsymbol{b}$ onto $\boldsymbol{a}$. More generally,

$$\boldsymbol{a}\cdot\boldsymbol{b} = |\boldsymbol{a}|\,|\boldsymbol{b}|\,\cos\theta,$$

where θ is the angle between the two vectors, which shows that a necessary and sufficient condition for two vectors to be orthogonal is that their dot product be zero.

The *cross product* (or *outer product*) of two vectors $\boldsymbol{a} = (a_1, a_2, a_3)^T$ and $\boldsymbol{b} = (b_1, b_2, b_3)^T$ in $\mathbb{R}^3$ is the vector

$$\boldsymbol{a}\times\boldsymbol{b} \stackrel{\text{def}}{=} \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix}.$$

Note that $\boldsymbol{a}\times\boldsymbol{b} = [\boldsymbol{a}_\times]\boldsymbol{b}$, where

$$[\boldsymbol{a}_\times] \stackrel{\text{def}}{=} \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}.$$

The cross product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{R}^3$ is orthogonal to these two vectors, and a necessary and sufficient condition for $\boldsymbol{a}$ and $\boldsymbol{b}$ to have the same direction is that $\boldsymbol{a}\times\boldsymbol{b} = \boldsymbol{0}$. If θ denotes as before the angle between the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, it can be shown that

$$|\boldsymbol{a}\times\boldsymbol{b}| = |\boldsymbol{a}|\,|\boldsymbol{b}|\,|\sin\theta|.$$

PROGRAMMING ASSIGNMENTS AND RESOURCES

The programming assignments given throughout the book sometimes require routines for numerical linear algebra, singular value decomposition, and linear and nonlinear least squares. An extensive set of such routines is available in MATLAB as well as in public-domain libraries such as LINPACK, LAPACK, and MINPACK, which can be downloaded from the Netlib repository (`http://www.netlib.org/`). We offer some pointers to other software on the book's web page `http://www.cs.berkeley.edu/~daf/book.html`. Datasets—or pointers to datasets—for the programming assignment are also available there.