

---

Wilhelm Burger · Mark J. Burge

# Digital Image Processing

An Algorithmic Introduction

Third Edition

 Springer

Wilhelm Burger  
University of Applied Sciences  
Upper Austria  
Hagenberg, Austria

Mark J. Burge  
Federal Bureau of Investigation  
Quantico, VA, USA

ISSN 1868-0941                      ISSN 1868-095X (electronic)  
Texts in Computer Science  
ISBN 978-3-031-05743-4              ISBN 978-3-031-05744-1 (eBook)  
<https://doi.org/10.1007/978-3-031-05744-1>

1<sup>st</sup> & 2<sup>nd</sup> edition: © Springer-Verlag London 2008, 2016

3<sup>rd</sup> edition: © Springer Nature Switzerland AG 2022

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, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

# Preface

---

This book provides a modern, self-contained introduction to digital image processing. We designed this book to be used both by learners desiring a firm foundation on which to build as well as practitioners in search of detailed analysis and transparent implementations of the most important techniques. This is the third English edition of the original German-language book, which has been widely used by:

- Scientists and engineers who use image processing as a tool and wish to develop a deeper understanding and create custom solutions to imaging problems in their field.
- IT professionals in search of a self-study course featuring easily adaptable code and completely worked out examples, enabling them to be productive right away.
- Faculty and students desiring an example-rich introductory textbook suitable for an advanced undergraduate or graduate level course that features exercises, projects, and examples that have been honed during quite some years of experience teaching this material.

While we concentrate on practical applications and concrete implementations, we do so without glossing over the important formal details and mathematics necessary for a deeper understanding of the algorithms. In preparing this text, we started from the premise that simply creating a recipe book of imaging solutions would not provide the deeper understanding needed to apply these techniques to novel problems, so instead our solutions are developed stepwise from three different perspectives: in mathematical form, as abstract pseudocode algorithms, and as complete Java programs. We use a common notation to intertwine all three perspectives—providing multiple, but intimately linked, views of problems and their solution.

## Prerequisites

Instead of presenting digital image processing as a mathematical discipline, or strictly as a signal processing topic, we present it from a practitioner's and programmer's perspective and with a view toward replacing many of the formalisms commonly used in other texts with constructs more readily understandable by our audience. To take full advantage of the *programming* components of this book, a knowledge of basic data structures and object-oriented programming, ideally in Java, is beneficial. We selected Java for a number of reasons: it is the first programming language learned by students in a wide variety of engineering curricula, and professionals with knowledge of a

related language, especially C# or C++, will find the programming examples easy to follow and extend.

The software in this book is designed to work with ImageJ, a widely used, programmer-extensible, imaging system developed, maintained, and distributed by the National Institutes of Health (NIH).<sup>1</sup> ImageJ is implemented completely in Java, and therefore runs on all major platforms. It is popular because its “plugin”-based architecture enables it to be easily extended. While all examples run in ImageJ, they have been specifically designed to be easily ported to other environments and programming languages.

### Use in Research and Development

This book has been especially designed for use as a textbook and as such features exercises and carefully constructed examples that supplement our detailed presentation of the fundamental concepts and techniques. As both practitioners and developers, we know that the details required to successfully understand, apply, and extend classical techniques are often difficult to find, and for this reason we have been very careful to provide the missing details, many gleaned over years of practical application. While this should make the text particularly valuable to those in research and development, it is not designed as a comprehensive, fully-cited scientific research text. On the contrary, we have carefully vetted our citations so that they can be obtained from easily accessible sources. While we have only briefly discussed the fundamentals of, or entirely omitted, topics such as hierarchical methods, wavelets, or eigenimages because of space limitations, other topics have been left out deliberately, including advanced issues such as object recognition, image understanding, and three-dimensional (3D) computer vision. So, while most techniques described in this book could be called “blind and dumb”, it is our experience that straightforward, technically clean implementations of these founding methods are essential to the success of any higher-level, perhaps really “intelligent” techniques built on top of them.

If you are only in search of a programming handbook for ImageJ or Java, there are certainly better sources. While the book includes many code examples, programming in and of itself is not our main focus. Instead Java serves as just one important element for describing each technique in a precise and immediately testable way.

### Classroom Use

Whether it is called signal processing, image processing, or media computation, the manipulation of digital images has been an integral part of most computer science and engineering curricula for many years. Today, with the omnipresence of all-digital work flows, it has become an integral part of the required skill set for professionals in many diverse disciplines.

Today the topic has migrated into the early stages of many curricula, where it is often a key foundation course. This trend uncovered a

<sup>1</sup> <https://rsb.info.nih.gov/ij/>

problem in that many of the texts relied on as standards in the older graduate-level courses were not appropriate for beginners. The texts were usually too formal for novices, and at the same time did not provide detailed coverage of many of the most popular methods used in actual practice. The result was that educators had a difficult time selecting a single textbook or even finding a compact collection of literature to recommend to their students. Faced with this dilemma ourselves, we wrote this book in the sincere hope of filling this gap.

The contents of the following chapters can be presented in either a one- or two-semester sequence. Where feasible, we have added supporting material in order to make each chapter as independent as possible, providing instructors with maximum flexibility when designing the course. Chapters 18–20 offer a compact introduction to the use of spectral techniques in image processing and are essentially independent of the other material in the text. Depending on the goals of the instructor and the curriculum, they can be covered in as much detail as required or completely omitted. The following road map shows a possible partitioning of topics for a two-semester syllabus.

<b>Road Map for a 2-Semester Syllabus</b>	Sem.	1	2
1. Digital Images .....	■	■	□
2. Histograms and Image Statistics .....	■	■	□
3. Point Operations .....	■	■	□
4. Filters .....	■	■	□
5. Edges and Contours .....	■	■	□
6. Corner Detection .....	■	■	□
7. Morphological Filters .....	■	■	□
8. Regions in Binary Images .....	■	■	□
9. Automatic Thresholding .....	□	□	■
10. Fitting Straight Lines .....	□	□	■
11. Fitting Circles and Ellipses .....	□	□	■
12. Detecting Geometric Primitives .....	□	□	■
13. Color Images .....	■	■	□
14. Colorimetric Color Spaces .....	□	□	■
15. Filters for Color Images .....	□	□	■
16. Edge Detection in Color Images .....	□	□	■
17. Edge-Preserving Smoothing Filters .....	□	□	■
18. Introduction to Spectral Techniques .....	□	□	■
19. The Discrete Fourier Transform in 2D .....	□	□	■
20. The Discrete Cosine Transform (DCT) .....	□	□	■
21. Geometric Operations .....	■	■	□
22. Pixel Interpolation .....	■	■	□
23. Image Matching and Registration .....	■	■	□
24. Non-Rigid Image Matching .....	□	□	■
25. Scale-Invariant Local Features (SIFT) .....	□	□	■
26. Maximally Stable Extremal Regions (MSER) .....	□	□	■

### Addendum To The 3rd Edition

This third (and presumably final) edition offers carefully revised contents and adds some new topics that round off the original material. In particular, there are new chapters on circle and ellipse fitting,

geometric primitive detection and local feature extraction. The appendix has been greatly expanded and contains numerous additions that should be helpful for concrete implementations as well as for reference. This includes small numerical examples, which are often helpful for the understanding of a problem and allow a quick check of one's own calculations.

Special attention was paid to the concise description of all methods in the form of easily understandable algorithms, not least to make the descriptions as independent as possible from concrete programming languages. In fact, we think that the consistent algorithmic description, which was significantly improved again in this edition, is really a unique feature of this book. On the other hand, we have relegated the concrete program examples somewhat to the background without sacrificing the most essential elements, especially since virtually all examples are also available online. In fact, most algorithms are implemented one-to-one in the associated `imagingbook` software library, so readers can always check in case of any ambiguities in the mathematical notation. All newly added concepts and associated demo programs have been implemented in this library, which will be continuously maintained and extended in the future.

Unfortunately, this third edition had to be prepared without the helping hand of the second author, therefore the blame for any errors that were either newly added or overlooked in the existing text lies solely with me (i.e., the first author). Readers will hopefully be merciful in view of the fact that there was no native speaker at work this time.

### Online Resources and Contact

Visit the website for this book

[www.imagingbook.com](http://www.imagingbook.com)

to download supplementary materials, including the complete (and free)<sup>2</sup> Java source code for all examples and the underlying software library, full-size test images, useful references, and other supplements. Comments, questions, and corrections are welcome and should be addressed to

[imagingbook@gmail.com](mailto:imagingbook@gmail.com)

### Exercises and Solutions

Almost every chapter of this book contains a set of sample exercises, mainly for supporting instructors to prepare their own assignments. Most of these tasks are easy to solve after studying the corresponding chapter, while some others may require more elaborated reasoning or experimental work. We assume that scholars know best how to select and adapt individual assignments in order to fit the level and interest of their students. This is the main reason why we have abstained from publishing explicit solutions in the past. However, we are happy to

<sup>2</sup> The `imagingbook` software suite is published under the *2-Clause BSD License* (see <https://opensource.org/licenses/BSD-2-Clause>).

answer any personal request if an exercise is unclear or seems to elude a simple solution.

### **Thank You!**

This book would not have been possible without the understanding and support of our families. Our thanks go to Wayne Rasband at NIH for developing ImageJ and for his truly outstanding support of the community and to all our readers of the previous editions who provided valuable input, suggestions for improvement, and encouragement. The use of open source software for such a project always carries an element of risk, since the long-term acceptance and continuity is difficult to assess. Retrospectively, choosing ImageJ as the software basis for this work was a good decision, and we would consider ourselves happy if our books have perhaps contributed to the success of the ImageJ project itself. Finally, we owe a debt of gratitude to the professionals at Springer, particularly to Wayne Wheeler and his team who again patiently accompanied the production of this edition.

W.B. and M.J.B.  
Spring 2022

# Contents

---

---

## Part I Images and Pixels

---

<b>1</b>	<b>Digital Images</b> . . . . .	<b>3</b>
1.1	Programming with Images . . . . .	4
1.2	Image Analysis and Computer Vision . . . . .	4
1.3	Types of Digital Images . . . . .	6
1.4	Image Acquisition . . . . .	6
1.4.1	The Pinhole Camera Model . . . . .	6
1.4.2	The “Thin” Lens . . . . .	8
1.4.3	Going Digital . . . . .	9
1.4.4	Image Size and Resolution . . . . .	10
1.4.5	Image Coordinate System . . . . .	11
1.4.6	Pixel Values . . . . .	11
1.5	Image File Formats . . . . .	13
1.5.1	Raster Versus Vector Data . . . . .	14
1.5.2	Tagged Image File Format (TIFF) . . . . .	14
1.5.3	Graphics Interchange Format (GIF) . . . . .	15
1.5.4	Portable Network Graphics (PNG) . . . . .	16
1.5.5	JPEG . . . . .	16
1.5.6	Legacy File Formats . . . . .	20
1.5.7	Bits and Bytes . . . . .	21
1.6	Software for Digital Imaging . . . . .	23
1.7	ImageJ . . . . .	23
1.7.1	Key Features . . . . .	24
1.7.2	Interactive Tools . . . . .	25
1.7.3	Working With ImageJ and Java . . . . .	25
1.8	Exercises . . . . .	27
<b>2</b>	<b>Histograms and Image Statistics</b> . . . . .	<b>29</b>
2.1	What is a Histogram? . . . . .	30
2.2	Interpreting Histograms . . . . .	31
2.2.1	Image Acquisition . . . . .	31
2.2.2	Image Defects . . . . .	33
2.3	Calculating Histograms . . . . .	36
2.4	Histograms of Images With More Than 8 Bits . . . . .	37
2.4.1	Binning . . . . .	37
2.4.2	Example . . . . .	37
2.4.3	Implementation . . . . .	38
2.5	Histograms of Color Images . . . . .	38
2.5.1	Intensity Histograms . . . . .	39
2.5.2	Individual Color Channel Histograms . . . . .	39
2.5.3	Combined Color Histograms . . . . .	40



2.6	The Cumulative Histogram .....	41
2.7	Statistical Information from the Histogram .....	41
	2.7.1 Mean and Variance .....	42
	2.7.2 Median .....	43
2.8	Block Statistics .....	43
	2.8.1 Integral Images .....	43
	2.8.2 Mean Intensity .....	45
	2.8.3 Variance .....	45
	2.8.4 Practical Calculation of Integral Images ....	45
2.9	Exercises .....	46
<b>3</b>	<b>Point Operations</b> .....	<b>49</b>
3.1	Modifying Image Intensity .....	50
	3.1.1 Contrast and Brightness .....	50
	3.1.2 Limiting Values by Clamping .....	50
	3.1.3 Inverting Images .....	51
	3.1.4 Thresholding Operation .....	51
3.2	Point Operations and Histograms .....	51
3.3	Automatic Contrast Adjustment .....	52
3.4	Modified Auto-Contrast Operation .....	54
3.5	Histogram Equalization .....	55
3.6	Histogram Specification .....	58
	3.6.1 Frequencies and Probabilities .....	58
	3.6.2 Principle of Histogram Specification .....	59
	3.6.3 Adjusting to a Piecewise Linear Distribution	60
	3.6.4 Adjusting to a Given Histogram (Histogram Matching) .....	61
	3.6.5 Examples .....	63
3.7	Gamma Correction .....	67
	3.7.1 Why “Gamma”? .....	68
	3.7.2 Mathematical Definition .....	68
	3.7.3 Real Gamma Values .....	69
	3.7.4 Applications of Gamma Correction .....	70
	3.7.5 Implementation .....	71
	3.7.6 Modified Gamma Correction .....	71
3.8	Point Operations in ImageJ .....	74
	3.8.1 Point Operations with Lookup Tables .....	74
	3.8.2 Arithmetic Operations .....	75
	3.8.3 Point Operations Involving Multiple Images .	75
	3.8.4 Methods for Point Operations on Two Images	76
	3.8.5 ImageJ Plugins Involving Multiple Images ..	76
3.9	Exercises .....	77

---

**Part II Filters, Edges and Corners**


---

<b>4</b>	<b>Filters</b> .....	<b>85</b>
4.1	What is a Filter? .....	85
4.2	Linear Filters .....	87
	4.2.1 The Filter Kernel .....	87
	4.2.2 Applying the Filter .....	87
	4.2.3 Implementing Filter Operations .....	88
	4.2.4 Filter Plugin Examples .....	89

4.2.5	Integer Coefficients . . . . .	91
4.2.6	Filters of Arbitrary Size . . . . .	92
4.2.7	Types of Linear Filters . . . . .	92
4.3	Formal Properties of Linear Filters . . . . .	95
4.3.1	Linear Convolution . . . . .	96
4.3.2	Formal Properties of Linear Convolution . . . . .	96
4.3.3	Separability of Linear Filters . . . . .	98
4.3.4	Impulse Response of a Filter . . . . .	99
4.4	Nonlinear Filters . . . . .	100
4.4.1	Minimum and Maximum Filters . . . . .	101
4.4.2	Median Filter . . . . .	103
4.4.3	Weighted Median Filter . . . . .	105
4.4.4	Other Nonlinear Filters . . . . .	107
4.5	Implementing Filters . . . . .	108
4.5.1	Efficiency of Filter Programs . . . . .	108
4.5.2	Handling Image Borders . . . . .	109
4.5.3	Debugging Filter Programs . . . . .	110
4.6	Filter Operations in ImageJ . . . . .	111
4.6.1	Linear Filters . . . . .	111
4.6.2	Gaussian Filters . . . . .	112
4.6.3	Nonlinear Filters . . . . .	112
4.7	Exercises . . . . .	113
<b>5</b>	<b>Edges and Contours . . . . .</b>	<b>117</b>
5.1	What Makes an Edge? . . . . .	117
5.2	Gradient-Based Edge Detection . . . . .	118
5.2.1	Partial Derivatives and the Gradient . . . . .	119
5.2.2	Derivative Filters . . . . .	119
5.3	Simple Edge Operators . . . . .	120
5.3.1	Prewitt and Sobel Edge Operators . . . . .	121
5.3.2	Roberts Edge Operator . . . . .	124
5.3.3	Compass Operators . . . . .	124
5.3.4	Edge Operators in ImageJ . . . . .	126
5.4	Other Edge Operators . . . . .	126
5.4.1	Edge Detection Based on Second Derivatives . . . . .	126
5.4.2	Edges at Different Scales . . . . .	126
5.4.3	From Edges to Contours . . . . .	127
5.5	Canny Edge Operator . . . . .	128
5.5.1	Preprocessing . . . . .	130
5.5.2	Edge Localization . . . . .	130
5.5.3	Edge Tracing and Hysteresis Thresholding . . . . .	131
5.5.4	Additional Information . . . . .	133
5.5.5	Implementation . . . . .	134
5.6	Edge Sharpening . . . . .	134
5.6.1	Edge Sharpening with the Laplacian Filter . . . . .	135
5.6.2	Unsharp Masking . . . . .	137
5.7	Exercises . . . . .	142
<b>6</b>	<b>Corner Detection . . . . .</b>	<b>145</b>
6.1	Points of Interest . . . . .	145
6.2	Harris Corner Detector . . . . .	146
6.2.1	The Local Structure Matrix . . . . .	146

6.2.2	Significance of the Local Structure Matrix ..	147
6.2.3	Corner Response Function (CRF) .....	150
6.2.4	Selecting Corner Points .....	150
6.2.5	Examples .....	152
6.3	Alternative Formulations .....	152
6.3.1	Shi-Tomasi Corner Score .....	152
6.3.2	MOPS Corner Score .....	154
6.4	Basic Implementation .....	154
6.4.1	Summary .....	160
6.5	Sub-Pixel Corner Positions .....	161
6.5.1	Position Interpolation by Second-Order Taylor Expansion .....	161
6.5.2	Sub-Pixel Positioning Example .....	162
6.6	Exercises .....	163

---

**Part III Binary Images**


---

<b>7</b>	<b>Morphological Filters .....</b>	<b>167</b>
7.1	Shrink and Let Grow .....	168
7.1.1	Pixel Neighborhoods .....	169
7.2	Basic Morphological Operations .....	170
7.2.1	The Structuring Element .....	170
7.2.2	Point Sets .....	170
7.2.3	Dilation .....	171
7.2.4	Erosion .....	171
7.2.5	Formal Properties of Dilation and Erosion ..	172
7.2.6	Designing Morphological Filters .....	173
7.2.7	Application Example: Outline .....	176
7.3	Composite Morphological Operations .....	177
7.3.1	Opening .....	177
7.3.2	Closing .....	179
7.3.3	Properties of Opening and Closing .....	179
7.4	Thinning (Skeletonization) .....	180
7.4.1	Basic Algorithm .....	180
7.4.2	Fast Thinning Algorithm .....	181
7.4.3	Java Implementation .....	185
7.4.4	Built-in Morphological Operations in ImageJ	186
7.5	Grayscale Morphology .....	187
7.5.1	Structuring Elements .....	187
7.5.2	Dilation and Erosion .....	188
7.5.3	Grayscale Opening and Closing .....	188
7.6	Exercises .....	189
<b>8</b>	<b>Regions in Binary Images .....</b>	<b>195</b>
8.1	Finding Connected Image Regions .....	196
8.1.1	Region Labeling by Flood Filling .....	196
8.1.2	Sequential Region Segmentation .....	201
8.1.3	Region Labeling – Summary .....	205
8.2	Region Contours .....	206
8.2.1	Outer and Inner Contours .....	206
8.2.2	Combining Region Labeling and Contour Detection .....	207

8.2.3	Java Implementation . . . . .	210
8.3	Representing Image Regions . . . . .	212
8.3.1	Matrix Representation . . . . .	212
8.3.2	Run Length Encoding . . . . .	213
8.3.3	Chain Codes . . . . .	213
8.4	Properties of Binary Regions . . . . .	216
8.4.1	Shape Features . . . . .	217
8.4.2	Geometric Features . . . . .	217
8.5	Statistical Shape Properties . . . . .	220
8.5.1	Centroid . . . . .	220
8.5.2	Moments . . . . .	221
8.5.3	Central Moments . . . . .	221
8.5.4	Normalized Central Moments . . . . .	222
8.5.5	Java Implementation . . . . .	222
8.6	Moment-Based Geometric Properties . . . . .	223
8.6.1	Orientation . . . . .	223
8.6.2	Region Eccentricity . . . . .	224
8.6.3	Equivalent Ellipse . . . . .	226
8.6.4	Bounding Box Aligned to the Major Axis . . . . .	226
8.6.5	Invariant Region Moments . . . . .	227
8.7	Projections . . . . .	232
8.8	Topological Region Properties . . . . .	233
8.9	Java Implementation . . . . .	234
8.10	Exercises . . . . .	234
<b>9</b>	<b>Automatic Thresholding . . . . .</b>	<b>241</b>
9.1	Global Histogram-Based Thresholding . . . . .	241
9.1.1	Image Statistics from the Histogram . . . . .	243
9.1.2	Simple Threshold Selection . . . . .	244
9.1.3	Iterative Threshold Selection (Isodata Algorithm) . . . . .	246
9.1.4	Otsu's Method . . . . .	248
9.1.5	Maximum Entropy Thresholding . . . . .	251
9.1.6	Minimum Error Thresholding . . . . .	254
9.2	Local Adaptive Thresholding . . . . .	261
9.2.1	Bernsen's Method . . . . .	262
9.2.2	Niblack's Method . . . . .	264
9.3	Java Implementation . . . . .	273
9.3.1	Global Thresholding Methods . . . . .	274
9.3.2	Adaptive Thresholding . . . . .	275
9.4	Summary and Further Reading . . . . .	275
9.5	Exercises . . . . .	276

---

**Part IV Geometric Primitives**


---

<b>10</b>	<b>Fitting Straight Lines . . . . .</b>	<b>281</b>
10.1	Straight Line Equations . . . . .	281
10.1.1	Slope-Intercept Form . . . . .	281
10.1.2	Parametric (Point-Vector) Form . . . . .	282
10.1.3	Algebraic Form . . . . .	282
10.1.4	Hessian Normal Form . . . . .	284
10.2	Fitting Lines to Points Sets . . . . .	284

10.2.1	Linear Regression . . . . .	285
10.2.2	Orthogonal Regression . . . . .	286
10.3	Example: Contour Segmentation . . . . .	290
10.4	Java Implementation . . . . .	292
10.5	Exercises . . . . .	295
<b>11</b>	<b>Fitting Circles and Ellipses . . . . .</b>	<b>297</b>
11.1	Fitting Circles . . . . .	297
11.1.1	Circle Equations . . . . .	297
11.1.2	Algebraic Circle Fits . . . . .	298
11.1.3	Geometric Circle Fitting . . . . .	305
11.2	Fitting Ellipses . . . . .	313
11.2.1	Algebraic Ellipse Fitting . . . . .	314
11.2.2	Geometric Ellipse Fitting . . . . .	323
11.2.3	Orthogonal Distance Approximations . . . . .	331
11.3	Java Implementation . . . . .	337
11.3.1	Circle Fitting . . . . .	338
11.3.2	Ellipse Fitting . . . . .	338
<b>12</b>	<b>Detecting Geometric Primitives . . . . .</b>	<b>341</b>
12.1	Random Sample Consensus (RANSAC) . . . . .	343
12.1.1	How Many Random Draws Are Needed? . . . . .	344
12.1.2	RANSAC Line Detection Algorithm . . . . .	345
12.1.3	Detecting Multiple Lines . . . . .	346
12.1.4	RANSAC Circle Detection . . . . .	348
12.1.5	RANSAC Ellipse Detection . . . . .	352
12.1.6	RANSAC Extensions and Applications . . . . .	355
12.2	The Hough Transform . . . . .	355
12.2.1	Parameter Space . . . . .	356
12.2.2	Accumulator Map . . . . .	357
12.2.3	A Better Line Representation . . . . .	357
12.2.4	Hough Algorithm . . . . .	359
12.2.5	Hough Transform Extensions . . . . .	363
12.2.6	Hough Transform for Circles and Arcs . . . . .	366
12.2.7	Hough Transform for Ellipses . . . . .	369
12.3	Java Implementation . . . . .	369
12.4	Exercises . . . . .	370

---

**Part V Color**


---

<b>13</b>	<b>Color Images . . . . .</b>	<b>375</b>
13.1	RGB Color Images . . . . .	375
13.1.1	Structure of Color Images . . . . .	376
13.1.2	Color Images in ImageJ . . . . .	379
13.2	Color Spaces and Color Conversion . . . . .	387
13.2.1	Conversion to Grayscale . . . . .	387
13.2.2	Desaturating RGB Color Images . . . . .	389
13.2.3	HSV/HSB and HLS Color Spaces . . . . .	389
13.2.4	TV Component Color Spaces: YUV, YIQ, and $Y C_b C_r$ . . . . .	400
13.2.5	Color Spaces for Printing: CMY and CMYK . . . . .	403
13.3	Statistics of Color Images . . . . .	406

13.3.1	How Many Different Colors Are in an Image?	406
13.3.2	Color Histograms	407
13.4	Color Quantization	408
13.4.1	Scalar Color Quantization	410
13.4.2	Vector Quantization	412
13.4.3	Java Implementation	419
13.5	Exercises	419
<b>14</b>	<b>Colorimetric Color Spaces</b>	<b>425</b>
14.1	CIE Color Spaces	425
14.1.1	CIE XYZ Color Space	426
14.1.2	CIE $x, y$ Chromaticity	426
14.1.3	Standard Illuminants	428
14.1.4	Gamut	429
14.1.5	Variants of the CIE Color Space	429
14.2	CIELAB Color Space	430
14.2.1	CIEXYZ $\rightarrow$ CIELAB Conversion	430
14.2.2	CIELAB $\rightarrow$ CIEXYZ Conversion	431
14.3	CIELUV Color Space	432
14.3.1	CIEXYZ $\rightarrow$ CIELUV Conversion	432
14.3.2	CIELUV $\rightarrow$ CIEXYZ Conversion	433
14.3.3	Measuring Color Differences	434
14.4	Standard RGB (sRGB)	434
14.4.1	Linear vs. Nonlinear Color Components	435
14.4.2	CIEXYZ $\rightarrow$ sRGB Conversion	436
14.4.3	sRGB $\rightarrow$ CIEXYZ Conversion	436
14.4.4	Calculations with Nonlinear sRGB Values	437
14.5	Adobe RGB	438
14.6	Chromatic Adaptation	439
14.6.1	XYZ Scaling	439
14.6.2	Bradford Color Adaptation	440
14.7	Colorimetric Support in Java	442
14.7.1	Profile Connection Space (PCS)	442
14.7.2	Color-Related Java Classes	444
14.7.3	Implementation of the CIELAB Color Space (Example)	445
14.7.4	ICC Profiles	446
14.8	Exercises	448
<b>15</b>	<b>Filters for Color Images</b>	<b>451</b>
15.1	Linear Filters	451
15.1.1	Monochromatic Application of Linear Filters	452
15.1.2	Color Space Matters	454
15.1.3	Linear Filtering with Circular Values	458
15.2	Nonlinear Color Filters	462
15.2.1	Scalar Median Filter	462
15.2.2	Vector Median Filter	462
15.2.3	Sharpening Vector Median Filter	466
15.3	Java Implementation	469
15.4	Further Reading	471
15.5	Exercises	472

<b>16</b>	<b>Edge Detection in Color Images</b> .....	475
16.1	Monochromatic Techniques .....	476
16.2	Edges in Vector-Valued Images .....	478
16.2.1	Multi-Dimensional Gradients .....	480
16.2.2	The Jacobian Matrix .....	481
16.2.3	Squared Local Contrast .....	482
16.2.4	Color Edge Magnitude .....	483
16.2.5	Color Edge Orientation .....	485
16.2.6	Grayscale Gradients Revisited .....	486
16.3	Canny Edge Detector for Color Images .....	489
16.4	Other Color Edge Operators .....	493
16.5	Java Implementation .....	494
16.6	Exercises .....	494
<b>17</b>	<b>Edge-Preserving Smoothing Filters</b> .....	497
17.1	Kuwahara-Type Filters .....	498
17.1.1	Application to Color Images .....	500
17.2	Bilateral Filter .....	504
17.2.1	Domain Filter .....	504
17.2.2	Range Filter .....	505
17.2.3	Bilateral Filter: General Idea .....	505
17.2.4	Bilateral Filter with Gaussian Kernels .....	507
17.2.5	Application to Color Images .....	508
17.2.6	Efficient Implementation by $x/y$ Separation ..	512
17.2.7	Further Reading .....	514
17.3	Anisotropic Diffusion Filters .....	517
17.3.1	Homogeneous Diffusion and the Heat Equation .....	518
17.3.2	The Perona-Malik Filter .....	520
17.3.3	Perona-Malik Filter for Color Images .....	522
17.3.4	Geometry Preserving Anisotropic Diffusion ..	526
17.3.5	Tschumperlé-Deriche Algorithm .....	528
17.4	Java Implementation .....	532
17.5	Exercises .....	535

---

**Part VI Spectral Techniques**


---

<b>18</b>	<b>Introduction to Spectral Methods</b> .....	539
18.1	The Fourier Transform .....	540
18.1.1	Sine and Cosine Functions .....	540
18.1.2	Fourier Series Representation of Periodic Functions .....	543
18.1.3	Fourier Integral .....	543
18.1.4	Fourier Spectrum and Transformation .....	544
18.1.5	Fourier Transform Pairs .....	545
18.1.6	Important Properties of the Fourier Transform	546
18.2	Working with Discrete Signals .....	550
18.2.1	Sampling .....	550
18.2.2	Discrete and Periodic Functions .....	555
18.3	The Discrete Fourier Transform (DFT) .....	555
18.3.1	Definition of the DFT .....	557
18.3.2	Discrete Basis Functions .....	558

18.3.3	Aliasing Again! . . . . .	559
18.3.4	Units in Signal and Frequency Space . . . . .	562
18.3.5	Power Spectrum . . . . .	563
18.4	Implementing the DFT . . . . .	563
18.4.1	Direct Implementation . . . . .	563
18.4.2	Fast Fourier Transform (FFT) . . . . .	565
18.5	Exercises . . . . .	565
<b>19</b>	<b>The Discrete Fourier Transform in 2D</b> . . . . .	<b>567</b>
19.1	Definition of the 2D DFT . . . . .	567
19.1.1	2D Basis Functions . . . . .	567
19.1.2	Implementing the 2D DFT . . . . .	568
19.2	Visualizing the 2D Fourier Transform . . . . .	570
19.2.1	Range of Spectral Values . . . . .	571
19.2.2	Centered Representation of the DFT Spectrum . . . . .	571
19.3	Frequencies and Orientation in 2D . . . . .	572
19.3.1	Effective Frequency . . . . .	572
19.3.2	Frequency Limits and Aliasing in 2D . . . . .	573
19.3.3	Orientation . . . . .	574
19.3.4	Normalizing the Geometry of the 2D Spectrum . . . . .	574
19.3.5	Effects of Periodicity . . . . .	575
19.3.6	Windowing . . . . .	575
19.3.7	Common Windowing Functions . . . . .	577
19.4	2D Fourier Transform Examples . . . . .	581
19.5	Applications of the DFT . . . . .	581
19.5.1	Linear Filter Operations in Frequency Space . . . . .	582
19.5.2	Linear Convolution and Correlation . . . . .	585
19.5.3	Inverse Filters . . . . .	585
19.6	Exercises . . . . .	586
<b>20</b>	<b>The Discrete Cosine Transform (DCT)</b> . . . . .	<b>589</b>
20.1	One-Dimensional DCT . . . . .	590
20.1.1	DCT Basis Functions . . . . .	590
20.1.2	Implementing the 1D DCT . . . . .	591
20.2	Two-Dimensional DCT . . . . .	592
20.2.1	Examples . . . . .	593
20.2.2	Separability . . . . .	593
20.3	Java Implementation . . . . .	595
20.4	Other Spectral Transforms . . . . .	596
20.5	Exercises . . . . .	597

---

## Part VII Image Transformations

---

<b>21</b>	<b>Geometric Operations</b> . . . . .	<b>601</b>
21.1	Coordinate Transformations in 2D . . . . .	602
21.1.1	Linear Coordinate Transformations . . . . .	603
21.1.2	Homogeneous Coordinates . . . . .	603
21.1.3	Affine (Three-Point) Mapping . . . . .	605
21.1.4	Projective (Four-Point) Mapping . . . . .	608
21.1.5	Bilinear Mapping . . . . .	614



21.1.6	Log-Polar Mapping	615
21.1.7	Other Nonlinear Transformations	620
21.1.8	Piecewise Image Transformations	623
21.2	Resampling the Image	624
21.2.1	Source-to-Target Mapping	625
21.2.2	Target-to-Source Mapping	626
21.3	Java Implementation	627
21.3.1	Geometric Transformations	627
21.3.2	Image Transformations	629
21.3.3	Examples	630
21.4	Exercises	632
<b>22</b>	<b>Pixel Interpolation</b>	<b>639</b>
22.1	Interpolation in 1D: Simple Methods	639
22.1.1	Nearest-Neighbor Interpolation	639
22.1.2	Linear Interpolation	640
22.1.3	Ideal Low-Pass Filter	640
22.2	Interpolation by Convolution	643
22.3	Cubic Interpolation	644
22.4	Spline Interpolation	645
22.4.1	Catmull-Rom Interpolation	646
22.4.2	Cubic B-spline Approximation	647
22.4.3	Mitchell-Netravali Approximation	647
22.4.4	Lanczos Interpolation	648
22.5	Interpolation in 2D	649
22.5.1	Nearest-Neighbor Interpolation in 2D	651
22.5.2	Bilinear Interpolation	651
22.5.3	Bicubic and Spline Interpolation in 2D	652
22.5.4	Lanczos Interpolation in 2D	654
22.5.5	Examples and Discussion	655
22.6	Aliasing	657
22.6.1	Sampling the Interpolated Image	657
22.6.2	Space-Variant Low-Pass Filtering	659
22.7	Java Implementation	660
22.8	Exercises	661

---

**Part VIII Image Matching**


---

<b>23</b>	<b>Image Matching and Registration</b>	<b>667</b>
23.1	Template Matching in Intensity Images	668
23.1.1	Distance between Image Patterns	668
23.1.2	Matching Under Rotation and Scaling	675
23.1.3	Java Implementation	676
23.2	Matching Binary Images	676
23.2.1	Direct Comparison of Binary Images	677
23.2.2	The Distance Transform	678
23.2.3	Chamfer Matching	682
23.2.4	Java Implementation	684
23.3	Exercises	684
<b>24</b>	<b>Non-Rigid Image Matching</b>	<b>687</b>
24.1	The Lucas-Kanade Technique	687

24.1.1	Registration in 1D . . . . .	687
24.1.2	Extension to Multi-Dimensional Functions . .	689
24.2	The Lucas-Kanade Algorithm . . . . .	690
24.2.1	Summary of the Algorithm . . . . .	693
24.3	Inverse Compositional Algorithm . . . . .	694
24.4	Linear Transformation Parameters . . . . .	698
24.4.1	Pure Translation . . . . .	698
24.4.2	Affine Transformation . . . . .	699
24.4.3	Projective Transformation . . . . .	701
24.4.4	Concatenating Linear Transformations . . . .	701
24.4.5	Coordinate Frames . . . . .	702
24.5	Example . . . . .	702
24.6	Java Implementation . . . . .	704
24.7	Exercises . . . . .	704

---

**Part IX Local Features**


---

<b>25</b>	<b>Scale-Invariant Feature Transform (SIFT)</b> . . . . .	709
25.1	Interest Points at Multiple Scales . . . . .	710
25.1.1	The LoG Filter . . . . .	710
25.1.2	Gaussian Scale Space . . . . .	715
25.1.3	LoG/DoG Scale Space . . . . .	718
25.1.4	Hierarchical Scale Space . . . . .	720
25.1.5	Scale Space Structure in SIFT . . . . .	724
25.2	Key Point Selection and Refinement . . . . .	730
25.2.1	Local Extrema Detection . . . . .	730
25.2.2	Position Refinement . . . . .	732
25.2.3	Suppressing Responses to Edge-Like Structures . . . . .	734
25.3	Creating Local Descriptors . . . . .	736
25.3.1	Finding Dominant Orientations . . . . .	737
25.3.2	SIFT Descriptor Construction . . . . .	740
25.4	SIFT Algorithm Summary . . . . .	747
25.5	Matching SIFT Features . . . . .	748
25.5.1	Feature Distance and Match Quality . . . . .	748
25.5.2	Examples . . . . .	753
25.6	Efficient Feature Matching . . . . .	756
25.7	Java Implementation . . . . .	759
25.7.1	SIFT Feature Extraction . . . . .	760
25.7.2	SIFT Feature Matching . . . . .	762
25.8	Exercises . . . . .	763
<b>26</b>	<b>Maximally Stable Extremal Regions (MSER)</b> . . . . .	765
26.1	Threshold Sets and Extremal Regions . . . . .	766
26.2	Building the Component Tree . . . . .	768
26.2.1	Component Tree Algorithms . . . . .	768
26.2.2	Component Tree Algorithm 1: <i>Global Immersion</i> . . . . .	770
26.2.3	Component Tree Algorithm 2: <i>Local Flooding</i> . . . .	774
26.2.4	Component Tree Examples . . . . .	777
26.3	Extracting MSERs from the Component Tree . . . .	778
26.3.1	Component Size Variation (Growth Rate) . . . .	779

26.3.2	Maximally Stable Components . . . . .	781
26.3.3	Constraints on Component Size and <i>Diversity</i> . . . . .	781
26.3.4	MSER Feature Statistics and Equivalent Ellipse . . . . .	784
26.3.5	Additional Constraints . . . . .	785
26.3.6	Detecting Dark <i>And</i> Bright Blobs . . . . .	787
26.3.7	MSER Examples . . . . .	788
26.4	Matching MSERs . . . . .	789
26.5	Local Affine Frames . . . . .	792
26.6	Summary . . . . .	794

---

**Appendix**


---

<b>A</b>	<b>Mathematical Symbols and Notation . . . . .</b>	<b>799</b>
A.1	Symbols . . . . .	799
A.2	Sets . . . . .	803
A.2.1	Basic Set Symbols and Operators . . . . .	803
A.2.2	Destructive Set Operators . . . . .	804
A.2.3	Relations, Mappings and Functions . . . . .	804
A.3	Sequences . . . . .	805
A.3.1	Adding and Removing Elements . . . . .	805
A.3.2	“Stack”-Type Sequences . . . . .	805
A.3.3	“Queue”-Type Sequences . . . . .	806
A.3.4	Sorting Sequences . . . . .	806
A.4	Tuples and Objects . . . . .	806
A.4.1	Type Definition and Instantiation . . . . .	806
A.4.2	Accessing Object Components . . . . .	807
A.4.3	Duplication . . . . .	807
A.5	Complex Numbers . . . . .	807
<b>B</b>	<b>Linear Algebra . . . . .</b>	<b>809</b>
B.1	Vectors and Matrices . . . . .	809
B.1.1	Column and Row Vectors . . . . .	810
B.1.2	Extracting Submatrices and Vectors . . . . .	811
B.1.3	Length (Norm) of a Vector . . . . .	811
B.2	Matrix Multiplication . . . . .	811
B.2.1	Scalar Multiplication . . . . .	811
B.2.2	Product of Two Matrices . . . . .	812
B.2.3	Matrix-Vector Products . . . . .	812
B.3	Vector Products . . . . .	813
B.3.1	Dot (Scalar) Product . . . . .	813
B.3.2	Outer Product . . . . .	814
B.3.3	Cross Product . . . . .	814
B.4	Trace and Determinant of a Square Matrix . . . . .	814
B.5	Eigenvalues and Eigenvectors . . . . .	815
B.5.1	Calculating Eigenvalues . . . . .	816
B.5.2	Generalized Symmetric Eigenproblems . . . . .	818
B.6	Homogeneous Coordinates . . . . .	819
B.7	Basic Matrix-Vector Operations with the <i>Apache</i> <i>Commons Math</i> Library . . . . .	820
B.7.1	Vectors and Matrices . . . . .	821
B.7.2	Matrix-Vector Multiplication . . . . .	821

B.7.3	Vector Products .....	821
B.7.4	Inverse of a Square Matrix .....	822
B.7.5	Eigenvalues and Eigenvectors .....	822
B.8	Solving Systems of Linear Equations .....	822
B.8.1	Exact Solutions .....	823
B.8.2	Over-Determined System (Least-Squares Solutions) .....	824
B.8.3	Solving <i>Homogeneous</i> Linear Systems .....	825
<b>C</b>	<b>Nonlinear Least Squares</b> .....	827
C.1	Nonlinear Least-Squares Fitting .....	827
C.2	Solution Methods .....	828
C.2.1	Implementation With <i>Apache Commons Math</i> .....	829
C.2.2	Example 1: One-Dimensional Curve Fitting .	829
C.3	Multi-Dimensional NLS Problems .....	832
C.3.1	Example 2: Geometric Circle Fitting .....	832
C.3.2	Numerical Estimation of Partial Derivatives .	836
<b>D</b>	<b>Elements from Calculus</b> .....	837
D.1	Scalar and Vector Fields .....	837
D.1.1	The Jacobian Matrix .....	837
D.1.2	Gradients .....	838
D.1.3	Maximum Gradient Direction .....	839
D.1.4	Divergence of a Vector Field .....	839
D.1.5	The Laplacian Operator .....	839
D.1.6	The Hessian Matrix .....	840
D.2	Taylor Series Expansion .....	841
D.2.1	Single-Variable Functions .....	841
D.2.2	Multi-Variable Functions .....	842
D.2.3	Finding Function Extrema by 2nd-Order Taylor Expansion .....	844
D.3	Estimating Derivatives of Discrete Functions .....	847
D.3.1	First-order Derivatives .....	847
D.3.2	Second-Order Derivatives .....	848
D.3.3	Alternative Formulations .....	848
<b>E</b>	<b>Sub-Pixel Maximum Finding</b> .....	849
E.1	Second-Order Interpolation in 1D .....	849
E.2	Subpixel Interpolation in 2D .....	851
E.2.1	Quadratic Functions in 2D .....	852
E.2.2	Method A: Second-Order Taylor Interpolation	853
E.2.3	Method B: Least-Squares Quadratic Interpolation .....	855
E.2.4	Quartic Interpolation .....	857
<b>F</b>	<b>Geometry</b> .....	861
F.1	Straight Lines .....	861
F.1.1	Conversions Between Different Line Equations	861
F.1.2	Intersections of Algebraic Lines .....	863
F.1.3	Intersections of Lines in Hessian Normal Form	863
F.1.4	Numeric Line Fitting Examples .....	863
F.2	Circles .....	865

	F.2.1	Circle Equations and Conversions . . . . .	865
	F.2.2	Circle From 3 Points . . . . .	866
F.3		Ellipses . . . . .	866
	F.3.1	Ellipse Equations . . . . .	866
	F.3.2	Converting Between Algebraic and Geometric Parameters . . . . .	867
	F.3.3	Ellipse From 5 Points . . . . .	869
<b>G</b>		<b>Statistical Prerequisites . . . . .</b>	<b>871</b>
	G.1	Mean, Variance, and Covariance . . . . .	871
		G.1.1 Mean . . . . .	871
		G.1.2 Variance and Covariance . . . . .	871
		G.1.3 Biased vs. Unbiased Variance . . . . .	872
	G.2	The Covariance Matrix . . . . .	872
		G.2.1 Example . . . . .	873
		G.2.2 Practical Calculation . . . . .	874
	G.3	Mahalanobis Distance . . . . .	874
		G.3.1 Definition . . . . .	874
		G.3.2 Relation to the Euclidean Distance . . . . .	875
		G.3.3 Numerical Considerations . . . . .	875
		G.3.4 Pre-Mapping Data For Efficient Mahalanobis Matching . . . . .	876
	G.4	The Gaussian Distribution . . . . .	878
		G.4.1 Maximum Likelihood Estimation . . . . .	878
		G.4.2 Gaussian Mixtures . . . . .	880
		G.4.3 Creating Gaussian Noise . . . . .	880
<b>H</b>		<b>Gaussian Filters . . . . .</b>	<b>883</b>
	H.1	Cascading Gaussian Filters . . . . .	883
	H.2	Gaussian Filters and Scale Space . . . . .	883
	H.3	Effects of Gaussian Filtering in the Frequency Domain . . . . .	884
	H.4	LoG-Approximation by Difference of Gaussians (DoG) . . . . .	885
<b>I</b>		<b>Writing ImageJ Plugins . . . . .</b>	<b>887</b>
	I.1	ImageJ Plugins . . . . .	887
		I.1.1 Program Structure . . . . .	887
		I.1.2 A First Example: Inverting an Image . . . . .	888
		I.1.3 Plugin My_Inverter_A (PlugInFilter) . . . . .	888
		I.1.4 Plugin My_Inverter_B (PlugIn) . . . . .	890
		I.1.5 When To Use PlugIn Or PlugInFilter? . . . . .	891
		I.1.6 Executing ImageJ “Commands” . . . . .	892
		I.1.7 ImageJ’s Command Recorder . . . . .	893
<b>J</b>		<b>Java Notes . . . . .</b>	<b>895</b>
	J.1	Arithmetic . . . . .	895
		J.1.1 Integer Division . . . . .	895
		J.1.2 Modulus Operator . . . . .	896
		J.1.3 Mathematical Functions in Class <code>Math</code> . . . . .	897
		J.1.4 Numerical Rounding . . . . .	898
		J.1.5 Inverse Tangent Function . . . . .	899

J.1.6	Unsigned Byte Data .....	899
J.1.7	Classes <code>Float</code> and <code>Double</code> .....	900
J.1.8	Testing Floating-Point Values Against Zero .	901
J.2	Arrays.....	901
J.2.1	Creating Arrays .....	901
J.2.2	Array Size .....	902
J.2.3	Accessing Array Elements.....	902
J.2.4	2D Arrays .....	903
J.2.5	Arrays of Objects .....	905
J.2.6	Searching for Minimum and Maximum Values	906
J.2.7	Sorting Arrays .....	906
<hr/>		
<b>References</b>	.....	909
<b>Index</b>	.....	925