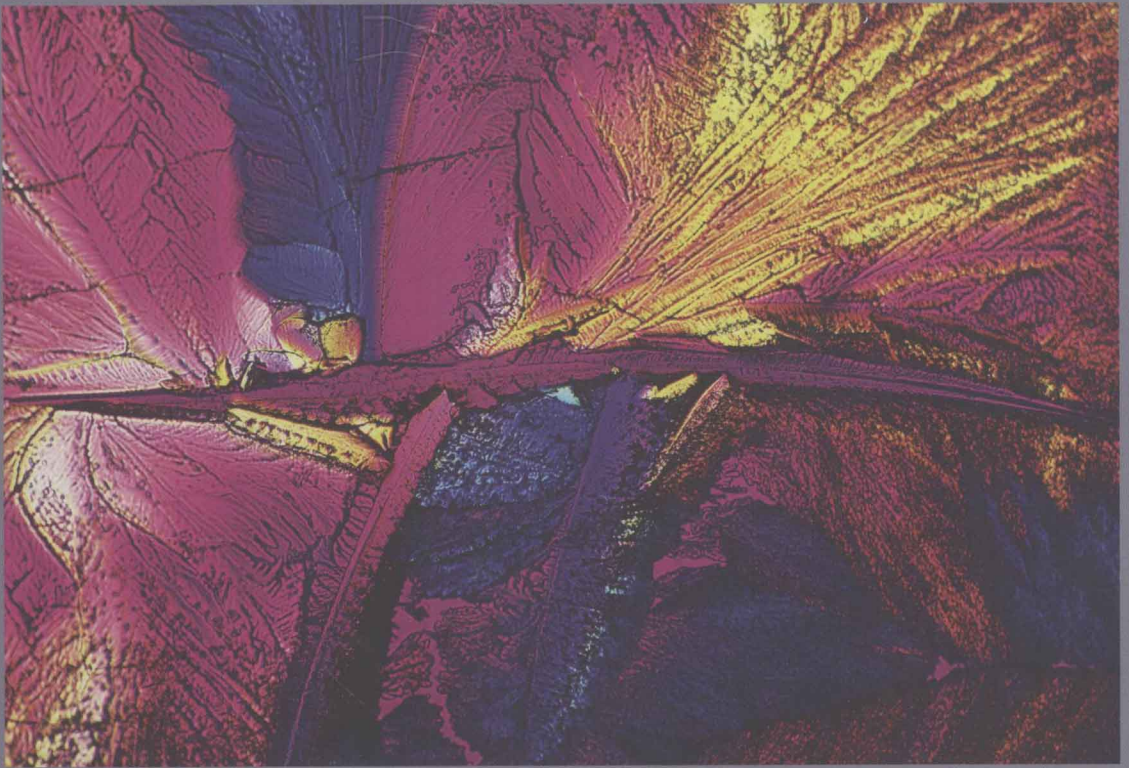


Digital Image Processing

An Algorithmic Introduction Using Java



Wilhelm Burger
Mark J. Burge

Wilhelm Burger Mark James Burge

Digital Image Processing

An Algorithmic Introduction using Java

First Edition

 Springer

Wilhelm Burger, PhD
Upper Austria University of Applied Sciences
Digital Media Department
11 Softwarepark
Hagenberg 4232
Austria
wilbur@ieee.org

Mark James Burge, PhD
National Science Foundation
4201 Wilson Blvd.
Arlington, VA 22230
USA
mburge@acm.org

ISBN 978-1-84628-379-6

e-ISBN 978-3-540-30941-3

Library of Congress Control Number: 2007938446

© 2008 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

Digital Image Processing

First Edition

Preface

This book provides a modern, self-contained introduction to digital image processing. We designed the book to be used both by learners desiring a firm foundation on which to build and practitioners in search of critical analysis and modern implementations of the most important techniques. This is the first 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.
- Information technology (IT) professionals wanting 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 our 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 linked, views of the problem and its solution.

Prerequisites

Instead of presenting digital image processing as a mathematical discipline, or strictly as signal processing, 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 required. We

selected Java for a number of reasons, one of which is that it is the first programming language learned by students in a wide variety of engineering curricula. Practitioners 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 Wayne Rasband of the National Institutes of Health (NIH).¹ ImageJ is implemented completely in Java, and therefore runs on all major platforms, and is widely used 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 the detailed synthesis of the fundamental concepts and techniques. As both practitioners and developers, we know that the details required for successful understanding and application of classical techniques are often difficult to find, and for this reason we have been very careful to provide the details, many gleaned over years of practical application, necessary to successfully apply these techniques. 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 omitted deliberately, including advanced issues such as object recognition, image understanding, or three-dimensional 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 simpler methods are essential to the success of any further domain-specific, or even “intelligent” approaches.

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

¹ <http://rsb.info.nih.gov/ij/>.

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 diverse disciplines. Previous to the explosion of digital media, it was often the case that a computing curriculum would offer only a single course, called “Digital Signal Processing” in engineering or “Digital Image Processing” in computing, and likely only as a graduate elective.

Today the topic has migrated into the early stages of many curricula, where it now serves as a key foundation course. This migration uncovered a 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 beginners, 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 a need.

The contents of the following chapters can be presented in either a one- or two-semester sequence. Where it was feasible, we have added supporting material in order to make each chapter as independent as possible and provide the instructor with as much flexibility as possible when designing the course. Chapters 13–15 offer a complete introduction to the fundamental spectral techniques used 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 road map (on page VIII) provides a sequence of topics for a one- or two-semester syllabus.

One Semester: A one-semester course can be organized around either of two major themes: image *processing* or image *analysis*. While either theme integrates easily into the early semesters of a modern computer science or IT curriculum, image analysis is especially appropriate as an early foundation course in medical informatics.

Two Semesters: When the content can be presented over two semesters, it has been designed so that it can be coherently divided (as described below) into two courses (*fundamentals* and *advanced*) where the themes are grouped according to difficulty.

Supplement to the English edition

This book was translated by the authors from the second German edition (published in 2006) [17], incorporating many enhancements throughout

Road Map for One- and Two-Semester Courses		Image Processing	Image Analysis	Fundamentals	Advanced
1. Crunching Pixels		☒	☒	☒	☐
2. Digital Images		☒	☒	☒	☐
3. ImageJ		☒	☒	☒	☐
4. Histograms		☒	☐	☒	☐
5. Point Operations		☒	☐	☒	☐
6. Filters		☒	☒	☒	☐
7. Edges and Contours		☒	☒	☒	☐
8. Finding Points of Interest		☐	☒	☐	☒
9. Detecting Simple Curves		☐	☒	☐	☒
10. Morphological Filters		☒	☐	☒	☐
11. Regions in Binary Images		☐	☒	☒	☐
12. Color Images		☒	☐	☐	☒
13. Introduction to Spectral Techniques		☐	☒	☐	☒
14. The Discrete Fourier Transform in 2D		☐	☒	☐	☒
15. The Discrete Cosine Transform		☐	☐	☐	☒
16. Geometrical Image Operations		☒	☐	☐	☒
17. Image Comparison		☐	☒	☐	☒
		1 Sem.		2 Sem.	

the text. In addition to the numerous small corrections and improvements that have been made, the presentation of histogram matching in Ch. 5, geometric region properties based on moments in Ch. 11, morphological filters in Ch. 10, and interpolation methods in Ch. 16 have been completely revised. Also, a number of example programs, such as the single-pass region labeling and contour finding algorithm (Sec. 11.2.2), have been rewritten for improved clarity and to take advantage of the new language features in Java 5.

Online resources

Visit the Website for this text

www.imagingbook.com

to download supplementary materials, including the complete Java source code for the examples, the test images used in the examples, and corrections. Additional materials are available for educators, including a complete set of formulas and figures used in the text, in a format suitable for inclusion in presentations. Comments, questions, and corrections are welcome and should be addressed to

imagingbook@gmail.com

Thank you

PREFACE

This book would not have been possible without the understanding and support of our families. Our thanks go to Wayne Rasband (NIH) for developing ImageJ and for his truly outstanding support of the community, to our colleagues Prof. Axel Pinz (TU Graz) and Prof. Vaclav Hlavac (TU Prague) for their comments, and to all the readers of the first two editions who provided valuable input, suggestions for improvement, and encouragement as we translated this edition. The authors greatly appreciate the help of their brave Sony and Apple notebooks that performed an estimated 1.6 quadrillion (10^{15}) CPU cycles to prepare this manuscript, thereby consuming about 560 kWh of electric energy and producing 196 kg of carbon dioxide. Finally, we owe a debt of gratitude to the professionals at Springer-Verlag, Ursula Zimpfer and Jutta Maria Fleschutz, who led the German edition team, and Wayne Wheeler, Catherine Brett, and Jeffrey Taub who were responsible for the English edition.

Hagenberg, Austria / Washington DC, USA
July 2007

Contents

Preface	V
1 Crunching Pixels	1
1.1 Programming with Images	2
1.2 Image Analysis and Computer Vision	3
2 Digital Images	5
2.1 Types of Digital Images	5
2.2 Image Acquisition	5
2.2.1 The Pinhole Camera Model	7
2.2.2 The “Thin” Lens	8
2.2.3 Going Digital	8
2.2.4 Image Size and Resolution	10
2.2.5 Image Coordinate System	11
2.2.6 Pixel Values	11
2.3 Image File Formats	13
2.3.1 Raster versus Vector Data	14
2.3.2 Tagged Image File Format (TIFF)	15
2.3.3 Graphics Interchange Format (GIF)	15
2.3.4 Portable Network Graphics (PNG)	16
2.3.5 JPEG	17
2.3.6 Windows Bitmap (BMP)	20
2.3.7 Portable Bitmap Format (PBM)	21
2.3.8 Additional File Formats	21
2.3.9 Bits and Bytes	22
2.4 Exercises	24
3 ImageJ	27
3.1 Image Manipulation and Processing	28
3.2 ImageJ Overview	28
3.2.1 Key Features	29
3.2.2 Interactive Tools	30
3.2.3 ImageJ Plugins	31
3.2.4 A First Example: Inverting an Image	32
3.3 Additional Information on ImageJ and Java	35
3.3.1 Resources for ImageJ	35
3.3.2 Programming with Java	35

3.4	Exercises	36
4	Histograms	37
4.1	What Is a Histogram?	38
4.2	Interpreting Histograms	39
4.2.1	Image Acquisition	40
4.2.2	Image Defects	42
4.3	Computing Histograms	44
4.4	Histograms of Images with More than 8 Bits	46
4.4.1	Binning	46
4.4.2	Example	46
4.4.3	Implementation	47
4.5	Color Image Histograms	47
4.5.1	Intensity Histograms	48
4.5.2	Individual Color Channel Histograms	48
4.5.3	Combined Color Histogram	49
4.6	Cumulative Histogram	50
4.7	Exercises	51
5	Point Operations	53
5.1	Modifying Image Intensity	54
5.1.1	Contrast and Brightness	54
5.1.2	Limiting the Results by Clamping	54
5.1.3	Inverting Images	55
5.1.4	Threshold Operation	55
5.2	Point Operations and Histograms	55
5.3	Automatic Contrast Adjustment	57
5.4	Modified Auto-Contrast	58
5.5	Histogram Equalization	59
5.6	Histogram Specification	62
5.6.1	Frequencies and Probabilities	63
5.6.2	Principle of Histogram Specification	65
5.6.3	Adjusting to a Piecewise Linear Distribution	65
5.6.4	Adjusting to a Given Histogram (Histogram Matching)	67
5.6.5	Examples	68
5.7	Gamma Correction	72
5.7.1	Why Gamma?	73
5.7.2	The Gamma Function	74
5.7.3	Real Gamma Values	74
5.7.4	Applications of Gamma Correction	75
5.7.5	Implementation	76
5.7.6	Modified Gamma Correction	76
5.8	Point Operations in ImageJ	80
5.8.1	Point Operations with Lookup Tables	80
5.8.2	Arithmetic Operations	81
5.8.3	Point Operations Involving Multiple Images	81

5.8.4	Methods for Point Operations on Two Images	82
5.8.5	ImageJ Plugins for Multiple Images	82
5.9	Exercises	83
6	Filters	87
6.1	What Is a Filter?	87
6.2	Linear Filters	89
6.2.1	The Filter Matrix	89
6.2.2	Applying the Filter	90
6.2.3	Computing the Filter Operation	91
6.2.4	Filter Plugin Examples	92
6.2.5	Integer Coefficients	93
6.2.6	Filters of Arbitrary Size	94
6.2.7	Types of Linear Filters	95
6.3	Formal Properties of Linear Filters	98
6.3.1	Linear Convolution	99
6.3.2	Properties of Linear Convolution	100
6.3.3	Separability of Linear Filters	101
6.3.4	Impulse Response of a Filter	103
6.4	Nonlinear Filters	104
6.4.1	Minimum and Maximum Filters	105
6.4.2	Median Filter	106
6.4.3	Weighted Median Filter	107
6.4.4	Other Nonlinear Filters	110
6.5	Implementing Filters	111
6.5.1	Efficiency of Filter Programs	111
6.5.2	Handling Image Borders	111
6.5.3	Debugging Filter Programs	112
6.6	Filter Operations in ImageJ	113
6.6.1	Linear Filters	113
6.6.2	Gaussian Filters	114
6.6.3	Nonlinear Filters	115
6.7	Exercises	115
7	Edges and Contours	117
7.1	What Makes an Edge?	117
7.2	Gradient-Based Edge Detection	118
7.2.1	Partial Derivatives and the Gradient	119
7.2.2	Derivative Filters	119
7.3	Edge Operators	120
7.3.1	Prewitt and Sobel Operators	120
7.3.2	Roberts Operator	123
7.3.3	Compass Operators	123
7.3.4	Edge Operators in ImageJ	125
7.4	Other Edge Operators	125
7.4.1	Edge Detection Based on Second Derivatives	126
7.4.2	Edges at Different Scales	126

7.4.3	Canny Operator	127
7.5	From Edges to Contours	127
7.5.1	Contour Following	127
7.5.2	Edge Maps	129
7.6	Edge Sharpening	130
7.6.1	Edge Sharpening with the Laplace Filter	130
7.6.2	Unsharp Masking	133
7.7	Exercises	137
8	Corner Detection	139
8.1	Points of Interest	139
8.2	Harris Corner Detector	140
8.2.1	Local Structure Matrix	140
8.2.2	Corner Response Function (CRF)	141
8.2.3	Determining Corner Points	142
8.2.4	Example	142
8.3	Implementation	142
8.3.1	Step 1: Computing the Corner Response Function	144
8.3.2	Step 2: Selecting “Good” Corner Points	148
8.3.3	Displaying the Corner Points	152
8.3.4	Summary	152
8.4	Exercises	153
9	Detecting Simple Curves	155
9.1	Salient Structures	155
9.2	Hough Transform	156
9.2.1	Parameter Space	157
9.2.2	Accumulator Array	159
9.2.3	A Better Line Representation	159
9.3	Implementing the Hough Transform	160
9.3.1	Filling the Accumulator Array	161
9.3.2	Analyzing the Accumulator Array	163
9.3.3	Hough Transform Extensions	165
9.4	Hough Transform for Circles and Ellipses	167
9.4.1	Circles and Arcs	167
9.4.2	Ellipses	170
9.5	Exercises	170
10	Morphological Filters	173
10.1	Shrink and Let Grow	174
10.1.1	Neighborhood of Pixels	175
10.2	Basic Morphological Operations	175
10.2.1	The Structuring Element	175
10.2.2	Point Sets	176
10.2.3	Dilation	177
10.2.4	Erosion	178
10.2.5	Properties of Dilation and Erosion	178

10.2.6	Designing Morphological Filters	180
10.2.7	Application Example: Outline	181
10.3	Composite Operations	183
10.3.1	Opening.....	185
10.3.2	Closing	185
10.3.3	Properties of Opening and Closing.....	186
10.4	Grayscale Morphology	187
10.4.1	Structuring Elements	187
10.4.2	Dilation and Erosion.....	187
10.4.3	Grayscale Opening and Closing	188
10.5	Implementing Morphological Filters.....	189
10.5.1	Binary Images in ImageJ	189
10.5.2	Dilation and Erosion.....	191
10.5.3	Opening and Closing	193
10.5.4	Outline	194
10.5.5	Morphological Operations in ImageJ	194
10.6	Exercises	196
11	Regions in Binary Images.....	199
11.1	Finding Image Regions	200
11.1.1	Region Labeling with Flood Filling	200
11.1.2	Sequential Region Labeling	204
11.1.3	Region Labeling—Summary	209
11.2	Region Contours	209
11.2.1	External and Internal Contours	209
11.2.2	Combining Region Labeling and Contour Finding	212
11.2.3	Implementation	213
11.2.4	Example	216
11.3	Representing Image Regions	216
11.3.1	Matrix Representation	216
11.3.2	Run Length Encoding.....	218
11.3.3	Chain Codes.....	219
11.4	Properties of Binary Regions.....	222
11.4.1	Shape Features.....	222
11.4.2	Geometric Features	223
11.4.3	Statistical Shape Properties.....	226
11.4.4	Moment-Based Geometrical Properties	228
11.4.5	Projections	233
11.4.6	Topological Properties	234
11.5	Exercises	235
12	Color Images	239
12.1	RGB Color Images	239
12.1.1	Organization of Color Images	241
12.1.2	Color Images in ImageJ	244
12.2	Color Spaces and Color Conversion	253
12.2.1	Conversion to Grayscale.....	256

12.2.2	Desaturating Color Images	257
12.2.3	HSV/HSB and HLS Color Space	258
12.2.4	TV Color Spaces—YUV, YIQ, and YC_bC_r	267
12.2.5	Color Spaces for Printing—CMY and CMYK	271
12.3	Colorimetric Color Spaces	275
12.3.1	CIE Color Spaces	276
12.3.2	CIE $L^*a^*b^*$	281
12.3.3	sRGB	283
12.3.4	Adobe RGB	288
12.3.5	Chromatic Adaptation	288
12.3.6	Colorimetric Support in Java	292
12.4	Statistics of Color Images	299
12.4.1	How Many Colors Are in an Image?	299
12.4.2	Color Histograms	299
12.5	Color Quantization	301
12.5.1	Scalar Color Quantization	303
12.5.2	Vector Quantization	305
12.6	Exercises	311
13	Introduction to Spectral Techniques	313
13.1	The Fourier Transform	314
13.1.1	Sine and Cosine Functions	314
13.1.2	Fourier Series of Periodic Functions	317
13.1.3	Fourier Integral	318
13.1.4	Fourier Spectrum and Transformation	319
13.1.5	Fourier Transform Pairs	320
13.1.6	Important Properties of the Fourier Transform	321
13.2	Working with Discrete Signals	325
13.2.1	Sampling	325
13.2.2	Discrete and Periodic Functions	330
13.3	The Discrete Fourier Transform (DFT)	332
13.3.1	Definition of the DFT	332
13.3.2	Discrete Basis Functions	334
13.3.3	Aliasing Again!	334
13.3.4	Units in Signal and Frequency Space	338
13.3.5	Power Spectrum	339
13.4	Implementing the DFT	340
13.4.1	Direct Implementation	340
13.4.2	Fast Fourier Transform (FFT)	342
13.5	Exercises	342
14	The Discrete Fourier Transform in 2D	343
14.1	Definition of the 2D DFT	343
14.1.1	2D Basis Functions	344
14.1.2	Implementing the Two-Dimensional DFT	344
14.2	Visualizing the 2D Fourier Transform	345
14.2.1	Range of Spectral Values	348

14.2.2	Centered Representation	348
14.3	Frequencies and Orientation in 2D	348
14.3.1	Effective Frequency	349
14.3.2	Frequency Limits and Aliasing in 2D	350
14.3.3	Orientation	350
14.3.4	Correcting the Geometry of a 2D Spectrum	351
14.3.5	Effects of Periodicity	352
14.3.6	Windowing	352
14.3.7	Windowing Functions	354
14.4	2D Fourier Transform Examples	359
14.5	Applications of the DFT	359
14.5.1	Linear Filter Operations in Frequency Space	363
14.5.2	Linear Convolution versus Correlation	364
14.5.3	Inverse Filters	364
14.6	Exercises	366
15	The Discrete Cosine Transform (DCT)	367
15.1	One-Dimensional DCT	367
15.1.1	DCT Basis Functions	368
15.1.2	Implementing the One-Dimensional DCT	368
15.2	Two-Dimensional DCT	370
15.2.1	Separability	371
15.2.2	Examples	371
15.3	Other Spectral Transforms	371
15.4	Exercises	373
16	Geometric Operations	375
16.1	2D Mapping Function	376
16.1.1	Simple Mappings	377
16.1.2	Homogeneous Coordinates	377
16.1.3	Affine (Three-Point) Mapping	378
16.1.4	Projective (Four-Point) Mapping	380
16.1.5	Bilinear Mapping	385
16.1.6	Other Nonlinear Image Transformations	386
16.1.7	Local Image Transformations	389
16.2	Resampling the Image	390
16.2.1	Source-to-Target Mapping	390
16.2.2	Target-to-Source Mapping	391
16.3	Interpolation	392
16.3.1	Simple Interpolation Methods	392
16.3.2	Ideal Interpolation	393
16.3.3	Interpolation by Convolution	397
16.3.4	Cubic Interpolation	397
16.3.5	Spline Interpolation	399
16.3.6	Lanczos Interpolation	402
16.3.7	Interpolation in 2D	404
16.3.8	Aliasing	410

16.4	Java Implementation	413
16.4.1	Geometric Transformations	413
16.4.2	Pixel Interpolation	423
16.4.3	Sample Applications	426
16.5	Exercises	427
17	Comparing Images	429
17.1	Template Matching in Intensity Images	430
17.1.1	Distance between Image Patterns	431
17.1.2	Implementation	438
17.1.3	Matching under Rotation and Scaling	439
17.2	Matching Binary Images	441
17.2.1	Direct Comparison	441
17.2.2	The Distance Transform	442
17.2.3	Chamfer Matching	446
17.3	Exercises	447
A	Mathematical Notation	451
A.1	Symbols	451
A.2	Set Operators	453
A.3	Complex Numbers	453
A.4	Algorithmic Complexity and \mathcal{O} Notation	454
B	Java Notes	457
B.1	Arithmetic	457
B.1.1	Integer Division	457
B.1.2	Modulus Operator	459
B.1.3	Unsigned Bytes	459
B.1.4	Mathematical Functions (Class <code>Math</code>)	460
B.1.5	Rounding	461
B.1.6	Inverse Tangent Function	462
B.1.7	<code>Float</code> and <code>Double</code> (Classes)	462
B.2	Arrays and Collections	462
B.2.1	Creating Arrays	462
B.2.2	Array Size	463
B.2.3	Accessing Array Elements	463
B.2.4	Two-Dimensional Arrays	464
B.2.5	Cloning Arrays	465
B.2.6	Arrays of Objects, Sorting	466
B.2.7	Collections	467
C	ImageJ Short Reference	469
C.1	Installation and Setup	469
C.2	ImageJ API	471
C.2.1	Images and Processors	471
C.2.2	Images (Package <code>ij</code>)	471
C.2.3	Image Processors (Package <code>ij.process</code>)	472