

Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

Wilhelm Burger · Mark J. Burge

Principles of Digital Image Processing

Fundamental Techniques



Springer

Wilhelm Burger
University of Applied Sciences
Hagenberg, Austria
wilbur@ieee.org

Mark J. Burge
noblis.org
Washington, D.C.
mburge@acm.org

Series editor

Ian Mackie, École Polytechnique, France and University of Sussex, UK

Advisory board

Samson Abramsky, University of Oxford, UK
Chris Hankin, Imperial College London, UK
Dexter Kozen, Cornell University, USA
Andrew Pitts, University of Cambridge, UK
Hanne Riis Nielson, Technical University of Denmark, Denmark
Steven Skiena, Stony Brook University, USA
Iain Stewart, University of Durham, UK
David Zhang, The Hong Kong Polytechnic University, Hong Kong

Undergraduate Topics in Computer Science ISSN 1863-7310
ISBN 978-1-84800-190-9 e-ISBN 978-1-84800-191-6
DOI 10.1007/978-1-84800-191-6

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2008942779

© Springer-Verlag London Limited 2009

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers. The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

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

Printed on acid-free paper

Springer Science+Business Media
springer.com

Preface

This book provides a modern, algorithmic introduction to digital image processing, designed 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 updated and enhanced paperback edition of our comprehensive textbook *Digital Image Processing: An Algorithmic Approach Using Java* packages the original material into a series of compact volumes, thereby supporting a flexible sequence of courses in digital image processing. Tailoring the contents to the scope of individual semester courses is also an attempt to provide affordable (and “backpack-compatible”) textbooks without compromising the quality and depth of content.

One approach to learning a new language is to become conversant in the core vocabulary and to start using it right away. At first, you may only know how to ask for directions, order coffee, and so on, but once you become confident with the core, you will start engaging others in “conversations” and rapidly learn how to get things done. This step-by-step approach works equally well in many areas of science and engineering.

In this first volume, ostentatiously titled *Fundamental Techniques*, we have attempted to compile the core “vocabulary” of digital image processing, starting from the basic concepts and elementary properties of digital images through simple statistics and point operations, fundamental filtering techniques, localization of edges and contours, and basic operations on color images. Mastering these most commonly used techniques and algorithms will enable you to start being productive right away.

The second volume of this series (*Core Algorithms*) extends the presented material, being devoted to slightly more advanced techniques and algorithms that are, nevertheless, part of the standard image processing toolbox. A forthcoming third volume (*Advanced Techniques*) will extend this series and add

important material beyond the elementary level for an advanced undergraduate or even graduate course.

Math, Algorithms, and “Real” Code

While we always concentrate on practical applications and working 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. Instead, our solutions typically develop stepwise along three different perspectives: (a) in mathematical form, (b) as abstract, pseudocode algorithms, and (c) as complete implementations in a real programming language. We use a common and consistent notation throughout to intertwine all three perspectives, thus providing multiple but linked views of the problem and its solution.

Software

The implementations in this series of texts are all based on Java and 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. It is widely used because its “plugin”-based architecture enables it to be easily extended. Although all examples run in ImageJ, they have been specifically designed to be easily ported to other environments and programming languages.

We chose Java as an implementation language because it is elegant, portable, familiar to many computing students, and more efficient than commonly thought. Although it may not be the fastest environment for numerical processing of raster images, we think that Java has great advantages when it comes to dynamic data structures and compile-time debugging. Note, however, that we use Java purely as an instructional vehicle because precise semantics are needed and, thus, everything presented here could be easily implemented in almost any other modern programming language. Although we stress the clarity and readability of our software, this is certainly not a book series on Java programming nor does it serve as a reference manual for ImageJ.

¹ <http://rsb.info.nih.gov/ij/>.

Online Resources

The authors maintain a Website for this text that provides supplementary materials, including the complete Java source code for the examples, the test images used in the figures, and corrections. Visit this site at

www.imagingbook.com

Additional materials are available for educators, including a complete set of figures, tables, and mathematical elements shown in the text, in a format suitable for easy inclusion in presentations and course notes. Comments, questions, and corrections are welcome and should be addressed to

imagingbook@gmail.com

Acknowledgements

As with its predecessors, this book would not have been possible without the understanding and steady support of our families. Thanks go to Wayne Rasband at NIH for developing and refining ImageJ and for his truly outstanding support of the growing user community. We appreciate the contribution from many careful readers who have contacted us to suggest new topics, recommend alternative solutions, or suggested corrections. Finally, we are grateful to Wayne Wheeler for initiating this book series and Catherine Brett and her colleagues at Springer's UK and New York offices for their professional support.

Hagenberg, Austria / Washington DC, USA
July 2008

Contents

Preface	v
1. Digital Images	1
1.1 Programming with Images	2
1.2 Image Acquisition	3
1.2.1 The Pinhole Camera Model	3
1.2.2 The “Thin” Lens Model	6
1.2.3 Going Digital	6
1.2.4 Image Size and Resolution	8
1.2.5 Image Coordinate System	9
1.2.6 Pixel Values	10
1.3 Image File Formats	12
1.3.1 Raster versus Vector Data	13
1.3.2 Tagged Image File Format (TIFF)	13
1.3.3 Graphics Interchange Format (GIF)	15
1.3.4 Portable Network Graphics (PNG)	15
1.3.5 JPEG	16
1.3.6 Windows Bitmap (BMP)	20
1.3.7 Portable Bitmap Format (PBM)	20
1.3.8 Additional File Formats	21
1.3.9 Bits and Bytes	21
1.4 Exercises	23
2. ImageJ	25
2.1 Image Manipulation and Processing	26
2.2 ImageJ Overview	27

2.2.1	Key Features	27
2.2.2	Interactive Tools	28
2.2.3	ImageJ Plugins	29
2.2.4	A First Example: Inverting an Image	31
2.3	Additional Information on ImageJ and Java	34
2.3.1	Resources for ImageJ	34
2.3.2	Programming with Java	35
2.4	Exercises	35
3.	Histograms	37
3.1	What Is a Histogram?	37
3.2	Interpreting Histograms	39
3.2.1	Image Acquisition	40
3.2.2	Image Defects	42
3.3	Computing Histograms	44
3.4	Histograms of Images with More than 8 Bits	47
3.4.1	Binning	47
3.4.2	Example	48
3.4.3	Implementation	48
3.5	Color Image Histograms	49
3.5.1	Intensity Histograms	49
3.5.2	Individual Color Channel Histograms	50
3.5.3	Combined Color Histograms	50
3.6	Cumulative Histogram	52
3.7	Exercises	52
4.	Point Operations	55
4.1	Modifying Image Intensity	56
4.1.1	Contrast and Brightness	56
4.1.2	Limiting the Results by Clamping	56
4.1.3	Inverting Images	57
4.1.4	Threshold Operation	57
4.2	Point Operations and Histograms	59
4.3	Automatic Contrast Adjustment	60
4.4	Modified Auto-Contrast	60
4.5	Histogram Equalization	63
4.6	Histogram Specification	66
4.6.1	Frequencies and Probabilities	67
4.6.2	Principle of Histogram Specification	68
4.6.3	Adjusting to a Piecewise Linear Distribution	69
4.6.4	Adjusting to a Given Histogram (Histogram Matching) ..	71
4.6.5	Examples	73

4.7	Gamma Correction	77
4.7.1	Why Gamma?	79
4.7.2	Power Function	79
4.7.3	Real Gamma Values	80
4.7.4	Applications of Gamma Correction	81
4.7.5	Implementation	82
4.7.6	Modified Gamma Correction	82
4.8	Point Operations in ImageJ	86
4.8.1	Point Operations with Lookup Tables	87
4.8.2	Arithmetic Operations	87
4.8.3	Point Operations Involving Multiple Images	88
4.8.4	Methods for Point Operations on Two Images	88
4.8.5	ImageJ Plugins Involving Multiple Images	90
4.9	Exercises	94
5.	Filters	97
5.1	What Is a Filter?	97
5.2	Linear Filters	99
5.2.1	The Filter Matrix	99
5.2.2	Applying the Filter	100
5.2.3	Computing the Filter Operation	101
5.2.4	Filter Plugin Examples	102
5.2.5	Integer Coefficients	104
5.2.6	Filters of Arbitrary Size	106
5.2.7	Types of Linear Filters	106
5.3	Formal Properties of Linear Filters	110
5.3.1	Linear Convolution	110
5.3.2	Properties of Linear Convolution	112
5.3.3	Separability of Linear Filters	113
5.3.4	Impulse Response of a Filter	115
5.4	Nonlinear Filters	116
5.4.1	Minimum and Maximum Filters	117
5.4.2	Median Filter	118
5.4.3	Weighted Median Filter	121
5.4.4	Other Nonlinear Filters	124
5.5	Implementing Filters	124
5.5.1	Efficiency of Filter Programs	124
5.5.2	Handling Image Borders	125
5.5.3	Debugging Filter Programs	126
5.6	Filter Operations in ImageJ	126
5.6.1	Linear Filters	127

5.6.2	Gaussian Filters	128
5.6.3	Nonlinear Filters	128
5.7	Exercises	129
6.	Edges and Contours	131
6.1	What Makes an Edge?	131
6.2	Gradient-Based Edge Detection	132
6.2.1	Partial Derivatives and the Gradient	133
6.2.2	Derivative Filters	134
6.3	Edge Operators	134
6.3.1	Prewitt and Sobel Operators	135
6.3.2	Roberts Operator	139
6.3.3	Compass Operators	139
6.3.4	Edge Operators in ImageJ	142
6.4	Other Edge Operators	142
6.4.1	Edge Detection Based on Second Derivatives	142
6.4.2	Edges at Different Scales	142
6.4.3	Canny Operator	144
6.5	From Edges to Contours	144
6.5.1	Contour Following	144
6.5.2	Edge Maps	145
6.6	Edge Sharpening	147
6.6.1	Edge Sharpening with the Laplace Filter	147
6.6.2	Unsharp Masking	150
6.7	Exercises	155
7.	Morphological Filters	157
7.1	Shrink and Let Grow	158
7.1.1	Neighborhood of Pixels	159
7.2	Basic Morphological Operations	160
7.2.1	The Structuring Element	160
7.2.2	Point Sets	161
7.2.3	Dilation	162
7.2.4	Erosion	162
7.2.5	Properties of Dilation and Erosion	163
7.2.6	Designing Morphological Filters	165
7.2.7	Application Example: Outline	167
7.3	Composite Operations	168
7.3.1	Opening	170
7.3.2	Closing	171
7.3.3	Properties of Opening and Closing	171
7.4	Grayscale Morphology	172

7.4.1	Structuring Elements	174
7.4.2	Dilation and Erosion	174
7.4.3	Grayscale Opening and Closing	174
7.5	Implementing Morphological Filters	176
7.5.1	Binary Images in ImageJ	176
7.5.2	Dilation and Erosion	180
7.5.3	Opening and Closing	181
7.5.4	Outline	181
7.5.5	Morphological Operations in ImageJ	182
7.6	Exercises	184
8.	Color Images	185
8.1	RGB Color Images	185
8.1.1	Organization of Color Images	188
8.1.2	Color Images in ImageJ	190
8.2	Color Spaces and Color Conversion	200
8.2.1	Conversion to Grayscale	202
8.2.2	Desaturating Color Images	205
8.2.3	HSV/HSB and HLS Color Space	205
8.2.4	TV Color Spaces—YUV, YIQ, and YC_bC_r	217
8.2.5	Color Spaces for Printing—CMY and CMYK	223
8.3	Statistics of Color Images	226
8.3.1	How Many Colors Are in an Image?	226
8.3.2	Color Histograms	227
8.4	Exercises	228
A.	Mathematical Notation	233
A.1	Symbols	233
A.2	Set Operators	235
A.3	Algorithmic Complexity and \mathcal{O} Notation	235
B.	Java Notes	237
B.1	Arithmetic	237
B.1.1	Integer Division	237
B.1.2	Modulus Operator	239
B.1.3	Unsigned Bytes	239
B.1.4	Mathematical Functions (Class <code>Math</code>)	240
B.1.5	Rounding	241
B.1.6	Inverse Tangent Function	242
B.1.7	<code>Float</code> and <code>Double</code> (Classes)	242
B.2	Arrays and Collections	242
B.2.1	Creating Arrays	242

B.2.2	Array Size	243
B.2.3	Accessing Array Elements	243
B.2.4	Two-Dimensional Arrays	244
B.2.5	Cloning Arrays	246
B.2.6	Arrays of Objects, Sorting	247
B.2.7	Collections	248
Bibliography		249
Index		253