

Undergraduate Topics in Computer Science

For further volumes:
<http://www.springer.com/series/7592>

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

Core Algorithms

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 978-1-84800-194-7 e-ISBN 978-1-84800-195-4
DOI 10.1007/978-1-84800-195-4

British Library Cataloguing in Publication Data

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

Library of Congress Control Number: 2008942518

© 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 is the second volume of a book series that provides a modern, algorithmic introduction to digital image processing. It is 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.

This second volume, titled *Core Algorithms*, extends the introductory material presented in the first volume (*Fundamental Techniques*) with additional techniques 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, suitable for an advanced undergraduate or even graduate course.

Math, Algorithms, and “Real” Code

It has been our experience in teaching in this field that mastering the core takes more than just reading about the techniques—it requires active construction and experimentation with the *algorithms* to acquire a feeling for how to use these methods in the real world. Internet search engines have made finding *someone’s* code for almost any imaging problem as simple as coming up with a succinct enough set of keywords. However, the problem is not to find *a* solution, but developing one’s own and understanding how it works—or why it

eventually does not. Whereas we feel that the real value of this series is not in its code, but rather in the critical selection of algorithms, illustrated explanations, and concise mathematical derivations, we continue to augment our algorithms with complete implementations, as even the best description of a method often omits some essential element necessary for the actual implementation, which only the unambiguous semantics of a real programming language can provide.

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 examples, 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 (NIH) for developing and refining ImageJ and for his truly outstanding support of the community. We appreciate the contribution from many careful readers who have contacted us to suggest new topics, recommend alternative solutions, or to suggest 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
June 2008

Contents

Preface	v
1. Introduction	1
1.1 Programming with Images	2
1.2 Image Analysis	3
2. Regions in Binary Images	5
2.1 Finding Image Regions	6
2.1.1 Region Labeling with Flood Filling	6
2.1.2 Sequential Region Labeling	11
2.1.3 Region Labeling—Summary	17
2.2 Region Contours	17
2.2.1 External and Internal Contours	18
2.2.2 Combining Region Labeling and Contour Finding	20
2.2.3 Implementation	22
2.2.4 Example	25
2.3 Representing Image Regions	26
2.3.1 Matrix Representation	26
2.3.2 Run Length Encoding	27
2.3.3 Chain Codes	28
2.4 Properties of Binary Regions	32
2.4.1 Shape Features	32
2.4.2 Geometric Features	33
2.4.3 Statistical Shape Properties	36
2.4.4 Moment-Based Geometrical Properties	38
2.4.5 Projections	44

2.4.6	Topological Properties	45
2.5	Exercises	46
3.	Detecting Simple Curves	49
3.1	Salient Structures	49
3.2	Hough Transform	50
3.2.1	Parameter Space	51
3.2.2	Accumulator Array	54
3.2.3	A Better Line Representation	54
3.3	Implementing the Hough Transform	55
3.3.1	Filling the Accumulator Array	56
3.3.2	Analyzing the Accumulator Array	56
3.3.3	Hough Transform Extensions	60
3.4	Hough Transform for Circles and Ellipses	63
3.4.1	Circles and Arcs	64
3.4.2	Ellipses	66
3.5	Exercises	67
4.	Corner Detection	69
4.1	Points of Interest	69
4.2	Harris Corner Detector	70
4.2.1	Local Structure Matrix	70
4.2.2	Corner Response Function (CRF)	71
4.2.3	Determining Corner Points	72
4.2.4	Example	72
4.3	Implementation	72
4.3.1	Step 1: Computing the Corner Response Function	76
4.3.2	Step 2: Selecting “Good” Corner Points	79
4.3.3	Displaying the Corner Points	83
4.3.4	Summary	83
4.4	Exercises	84
5.	Color Quantization	85
5.1	Scalar Color Quantization	86
5.2	Vector Quantization	88
5.2.1	Popularity algorithm	88
5.2.2	Median-cut algorithm	88
5.2.3	Octree algorithm	89
5.2.4	Other methods for vector quantization	94
5.3	Exercises	95

6. Colorimetric Color Spaces	97
6.1 CIE Color Spaces	98
6.1.1 CIE XYZ color space	98
6.1.2 CIE x, y chromaticity	99
6.1.3 Standard illuminants	101
6.1.4 Gamut	102
6.1.5 Variants of the CIE color space	103
6.2 CIE $L^*a^*b^*$	104
6.2.1 Transformation CIE XYZ \rightarrow $L^*a^*b^*$	104
6.2.2 Transformation $L^*a^*b^* \rightarrow$ CIE XYZ	105
6.2.3 Measuring color differences	105
6.3 sRGB	106
6.3.1 Linear vs. nonlinear color components	107
6.3.2 Transformation CIE XYZ \rightarrow sRGB	108
6.3.3 Transformation sRGB \rightarrow CIE XYZ	108
6.3.4 Calculating with sRGB values	109
6.4 Adobe RGB	111
6.5 Chromatic Adaptation	111
6.5.1 XYZ scaling	112
6.5.2 Bradford adaptation	113
6.6 Colorimetric Support in Java	114
6.6.1 sRGB colors in Java	114
6.6.2 Profile connection space (PCS)	115
6.6.3 Color-related Java classes	118
6.6.4 A $L^*a^*b^*$ color space implementation	120
6.6.5 ICC profiles	121
6.7 Exercises	124
7. Introduction to Spectral Techniques	125
7.1 The Fourier Transform	126
7.1.1 Sine and Cosine Functions	126
7.1.2 Fourier Series of Periodic Functions	130
7.1.3 Fourier Integral	130
7.1.4 Fourier Spectrum and Transformation	131
7.1.5 Fourier Transform Pairs	132
7.1.6 Important Properties of the Fourier Transform	136
7.2 Working with Discrete Signals	137
7.2.1 Sampling	137
7.2.2 Discrete and Periodic Functions	144
7.3 The Discrete Fourier Transform (DFT)	144
7.3.1 Definition of the DFT	144

7.3.2	Discrete Basis Functions	147
7.3.3	Aliasing Again!	148
7.3.4	Units in Signal and Frequency Space	152
7.3.5	Power Spectrum	153
7.4	Implementing the DFT	154
7.4.1	Direct Implementation	154
7.4.2	Fast Fourier Transform (FFT)	155
7.5	Exercises	156
8.	The Discrete Fourier Transform in 2D	157
8.1	Definition of the 2D DFT	157
8.1.1	2D Basis Functions	158
8.1.2	Implementing the Two-Dimensional DFT	158
8.2	Visualizing the 2D Fourier Transform	162
8.2.1	Range of Spectral Values	162
8.2.2	Centered Representation	162
8.3	Frequencies and Orientation in 2D	164
8.3.1	Effective Frequency	164
8.3.2	Frequency Limits and Aliasing in 2D	164
8.3.3	Orientation	165
8.3.4	Normalizing the 2D Spectrum	166
8.3.5	Effects of Periodicity	167
8.3.6	Windowing	169
8.3.7	Windowing Functions	169
8.4	2D Fourier Transform Examples	171
8.5	Applications of the DFT	175
8.5.1	Linear Filter Operations in Frequency Space	175
8.5.2	Linear Convolution versus Correlation	177
8.5.3	Inverse Filters	178
8.6	Exercises	180
9.	The Discrete Cosine Transform (DCT)	183
9.1	One-Dimensional DCT	183
9.1.1	DCT Basis Functions	184
9.1.2	Implementing the One-Dimensional DCT	186
9.2	Two-Dimensional DCT	187
9.2.1	Separability	187
9.2.2	Examples	188
9.3	Other Spectral Transforms	188
9.4	Exercises	190

10. Geometric Operations	191
10.1 2D Mapping Function	193
10.1.1 Simple Mappings	193
10.1.2 Homogeneous Coordinates	194
10.1.3 Affine (Three-Point) Mapping	195
10.1.4 Projective (Four-Point) Mapping	197
10.1.5 Bilinear Mapping	203
10.1.6 Other Nonlinear Image Transformations	204
10.1.7 Local Image Transformations	207
10.2 Resampling the Image	209
10.2.1 Source-to-Target Mapping	209
10.2.2 Target-to-Source Mapping	210
10.3 Interpolation	210
10.3.1 Simple Interpolation Methods	211
10.3.2 Ideal Interpolation	213
10.3.3 Interpolation by Convolution	217
10.3.4 Cubic Interpolation	217
10.3.5 Spline Interpolation	219
10.3.6 Lanczos Interpolation	223
10.3.7 Interpolation in 2D	225
10.3.8 Aliasing	234
10.4 Java Implementation	238
10.4.1 Geometric Transformations	238
10.4.2 Pixel Interpolation	248
10.4.3 Sample Applications	251
10.5 Exercises	253
11. Comparing Images	255
11.1 Template Matching in Intensity Images	257
11.1.1 Distance between Image Patterns	258
11.1.2 Implementation	266
11.1.3 Matching under Rotation and Scaling	267
11.2 Matching Binary Images	269
11.2.1 Direct Comparison	269
11.2.2 The Distance Transform	270
11.2.3 Chamfer Matching	274
11.3 Exercises	278
A. Mathematical Notation	279
A.1 Symbols	279
A.2 Set Operators	281
A.3 Complex Numbers	282

B. Source Code	283
B.1 Combined Region Labeling and Contour Tracing	283
B.1.1 <code>Contour_Tracing_Plugin</code> (Class)	283
B.1.2 <code>Contour</code> (Class)	285
B.1.3 <code>BinaryRegion</code> (Class)	286
B.1.4 <code>ContourTracer</code> (Class)	287
B.1.5 <code>ContourOverlay</code> (Class)	292
B.2 Harris Corner Detector	294
B.2.1 <code>Harris_Corner_Plugin</code> (Class)	294
B.2.2 <code>File_Corner</code> (Class)	295
B.2.3 <code>File_HarrisCornerDetector</code> (Class)	296
B.3 Median-Cut Color Quantization	301
B.3.1 <code>ColorQuantizer</code> (Interface)	301
B.3.2 <code>MedianCutQuantizer</code> (Class)	301
B.3.3 <code>ColorHistogram</code> (Class)	309
B.3.4 <code>Median_Cut_Quantization</code> (Class)	310
Bibliography	313
Index	321