

PRACTICAL IMAGE AND VIDEO PROCESSING USING MATLAB®

PRACTICAL IMAGE AND VIDEO PROCESSING USING MATLAB®

OGE MARQUES

Florida Atlantic University



A JOHN WILEY & SONS, INC., PUBLICATION

About the Cover (by Roger Dalal)

The elegant Nautilus, with its progressive chambers and near-perfect logarithmic spiral, demonstrates the beauty of mathematics and the power of digital image processing. Created exclusively for *Practical Image and Video Processing Using MATLAB®*, this composition features multiple layers and processing techniques. The primary image is doubly sharpened with an 8 pixel radius, and enhanced with posterizing and edge detection algorithms. The outer, secondary image is indexed to 20 colors, pixelized at two percent resolution of the center image, and partially hidden by a fading, offset radial mask.

Copyright © 2011 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department with the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data

Marques, Oge.

Practical image and video processing using MATLAB® / Oge Marques.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-470-04815-3 (hardback)

1. Image processing—Mathematics. 2. Digital video—Mathematics. 3. Image processing—Digital techniques. 4. MATLAB®. I. Title.

TA1637.M3375 2011

502.85'66—dc22

2011008249

oBook ISBN: 978111093467

ePDF ISBN: 9781118093481

ePub ISBN: 9781118093474

Printed in Singapore.

10 9 8 7 6 5 4 3 2 1

*To my Son Nicholas, whose
Precious Existence has
Provided the Greatest
Motivation to Pursue this
Project.*

*And in Loving Memory of my
Father, Ogé Aby Marques.*

CONTENTS

| | |
|------------------------|--------------|
| LIST OF FIGURES | xxi |
| LIST OF TABLES | xxxix |
| FOREWORD | xli |
| PREFACE | xlili |
| ACKNOWLEDGMENTS | xlix |

PART I IMAGE PROCESSING

| | |
|--|-----------|
| 1 INTRODUCTION AND OVERVIEW | 3 |
| 1.1 Motivation / 3 | |
| 1.2 Basic Concepts and Terminology / 5 | |
| 1.3 Examples of Typical Image Processing Operations / 6 | |
| 1.4 Components of a Digital Image Processing System / 10 | |
| 1.5 Machine Vision Systems / 12 | |
| 1.6 Resources / 14 | |
| 1.7 Problems / 18 | |
| 2 IMAGE PROCESSING BASICS | 21 |
| 2.1 Digital Image Representation / 21 | |
| 2.1.1 Binary (1-Bit) Images / 23 | |

- 2.1.2 Gray-Level (8-Bit) Images / 24
- 2.1.3 Color Images / 25
- 2.1.4 Compression / 26
- 2.2 Image File Formats / 27
- 2.3 Basic Terminology / 28
- 2.4 Overview of Image Processing Operations / 30
 - 2.4.1 Global (Point) Operations / 31
 - 2.4.2 Neighborhood-Oriented Operations / 31
 - 2.4.3 Operations Combining Multiple Images / 32
 - 2.4.4 Operations in a Transform Domain / 32

3 MATLAB BASICS

35

- 3.1 Introduction to MATLAB / 35
- 3.2 Basic Elements of MATLAB / 36
 - 3.2.1 Working Environment / 36
 - 3.2.2 Data Types / 37
 - 3.2.3 Array and Matrix Indexing in MATLAB / 37
 - 3.2.4 Standard Arrays / 37
 - 3.2.5 Command-Line Operations / 38
- 3.3 Programming Tools: Scripts and Functions / 38
 - 3.3.1 M-Files / 39
 - 3.3.2 Operators / 40
 - 3.3.3 Important Variables and Constants / 42
 - 3.3.4 Number Representation / 42
 - 3.3.5 Flow Control / 43
 - 3.3.6 Code Optimization / 43
 - 3.3.7 Input and Output / 43
- 3.4 Graphics and Visualization / 43
- 3.5 Tutorial 3.1: MATLAB—a Guided Tour / 44
- 3.6 Tutorial 3.2: MATLAB Data Structures / 46
- 3.7 Tutorial 3.3: Programming in MATLAB / 53
- 3.8 Problems / 59

4 THE IMAGE PROCESSING TOOLBOX AT A GLANCE

61

- 4.1 The Image Processing Toolbox: an Overview / 61
- 4.2 Essential Functions and Features / 62
 - 4.2.1 Displaying Information About an Image File / 62
 - 4.2.2 Reading an Image File / 64

- 4.2.3 Data Classes and Data Conversions / 65
- 4.2.4 Displaying the Contents of an Image / 68
- 4.2.5 Exploring the Contents of an Image / 69
- 4.2.6 Writing the Resulting Image onto a File / 70
- 4.3 Tutorial 4.1: MATLAB Image Processing Toolbox—a Guided Tour / 72
- 4.4 Tutorial 4.2: Basic Image Manipulation / 74
- 4.5 Problems / 80

5 IMAGE SENSING AND ACQUISITION

83

- 5.1 Introduction / 83
- 5.2 Light, Color, and Electromagnetic Spectrum / 84
 - 5.2.1 Light and Electromagnetic Spectrum / 84
 - 5.2.2 Types of Images / 85
 - 5.2.3 Light and Color Perception / 86
 - 5.2.4 Color Encoding and Representation / 87
- 5.3 Image Acquisition / 89
 - 5.3.1 Image Sensors / 89
 - 5.3.2 Camera Optics / 92
- 5.4 Image Digitization / 93
 - 5.4.1 Sampling / 95
 - 5.4.2 Quantization / 96
 - 5.4.3 Spatial and Gray-Level Resolution / 97
- 5.5 Problems / 101

6 ARITHMETIC AND LOGIC OPERATIONS

103

- 6.1 Arithmetic Operations: Fundamentals and Applications / 103
 - 6.1.1 Addition / 104
 - 6.1.2 Subtraction / 106
 - 6.1.3 Multiplication and Division / 109
 - 6.1.4 Combining Several Arithmetic Operations / 110
- 6.2 Logic Operations: Fundamentals and Applications / 111
- 6.3 Tutorial 6.1: Arithmetic Operations / 113
- 6.4 Tutorial 6.2: Logic Operations and Region of Interest Processing / 118
- 6.5 Problems / 122

7 GEOMETRIC OPERATIONS 125

- 7.1 Introduction / 125
- 7.2 Mapping and Affine Transformations / 127
- 7.3 Interpolation Methods / 130
 - 7.3.1 The Need for Interpolation / 130
 - 7.3.2 A Simple Approach to Interpolation / 131
 - 7.3.3 Zero-Order (Nearest-Neighbor) Interpolation / 132
 - 7.3.4 First-Order (Bilinear) Interpolation / 132
 - 7.3.5 Higher Order Interpolations / 132
- 7.4 Geometric Operations Using MATLAB / 132
 - 7.4.1 Zooming, Shrinking, and Resizing / 133
 - 7.4.2 Translation / 134
 - 7.4.3 Rotation / 134
 - 7.4.4 Cropping / 134
 - 7.4.5 Flipping / 134
- 7.5 Other Geometric Operations and Applications / 134
 - 7.5.1 Warping / 134
 - 7.5.2 Nonlinear Image Transformations / 135
 - 7.5.3 Morphing / 137
 - 7.5.4 Seam Carving / 137
 - 7.5.5 Image Registration / 137
- 7.6 Tutorial 7.1: Image Cropping, Resizing, Flipping, and Rotation / 138
- 7.7 Tutorial 7.2: Spatial Transformations and Image Registration / 142
- 7.8 Problems / 149

8 GRAY-LEVEL TRANSFORMATIONS 151

- 8.1 Introduction / 151
- 8.2 Overview of Gray-level (Point) Transformations / 152
- 8.3 Examples of Point Transformations / 155
 - 8.3.1 Contrast Manipulation / 155
 - 8.3.2 Negative / 157
 - 8.3.3 Power Law (Gamma) Transformations / 157
 - 8.3.4 Log Transformations / 159
 - 8.3.5 Piecewise Linear Transformations / 160
- 8.4 Specifying the Transformation Function / 161
- 8.5 Tutorial 8.1: Gray-level Transformations / 163
- 8.6 Problems / 169

9 HISTOGRAM PROCESSING 171

- 9.1 Image Histogram: Definition and Example / 171
- 9.2 Computing Image Histograms / 173
- 9.3 Interpreting Image Histograms / 174
- 9.4 Histogram Equalization / 176
- 9.5 Direct Histogram Specification / 181
- 9.6 Other Histogram Modification Techniques / 184
 - 9.6.1 Histogram Sliding / 185
 - 9.6.2 Histogram Stretching / 185
 - 9.6.3 Histogram Shrinking / 186
- 9.7 Tutorial 9.1: Image Histograms / 188
- 9.8 Tutorial 9.2: Histogram Equalization and Specification / 191
- 9.9 Tutorial 9.3: Other Histogram Modification Techniques / 195
- 9.10 Problems / 200

10 NEIGHBORHOOD PROCESSING 203

- 10.1 Neighborhood Processing / 203
- 10.2 Convolution and Correlation / 204
 - 10.2.1 Convolution in the One-Dimensional Domain / 204
 - 10.2.2 Convolution in the Two-Dimensional Domain / 206
 - 10.2.3 Correlation / 208
 - 10.2.4 Dealing with Image Borders / 210
- 10.3 Image Smoothing (Low-pass Filters) / 211
 - 10.3.1 Mean Filter / 213
 - 10.3.2 Variations / 213
 - 10.3.3 Gaussian Blur Filter / 215
 - 10.3.4 Median and Other Nonlinear Filters / 216
- 10.4 Image Sharpening (High-pass Filters) / 218
 - 10.4.1 The Laplacian / 219
 - 10.4.2 Composite Laplacian Mask / 220
 - 10.4.3 Directional Difference Filters / 220
 - 10.4.4 Unsharp Masking / 221
 - 10.4.5 High-Boost Filtering / 221
- 10.5 Region of Interest Processing / 222
- 10.6 Combining Spatial Enhancement Methods / 223
- 10.7 Tutorial 10.1: Convolution and Correlation / 223
- 10.8 Tutorial 10.2: Smoothing Filters in the Spatial Domain / 225

- 10.9 Tutorial 10.3: Sharpening Filters in the Spatial Domain / 228
- 10.10 Problems / 234

11 FREQUENCY-DOMAIN FILTERING

235

- 11.1 Introduction / 235
- 11.2 Fourier Transform: the Mathematical Foundation / 237
 - 11.2.1 Basic Concepts / 237
 - 11.2.2 The 2D Discrete Fourier Transform: Mathematical Formulation / 239
 - 11.2.3 Summary of Properties of the Fourier Transform / 241
 - 11.2.4 Other Mathematical Transforms / 242
- 11.3 Low-pass Filtering / 243
 - 11.3.1 Ideal LPF / 244
 - 11.3.2 Gaussian LPF / 246
 - 11.3.3 Butterworth LPF / 246
- 11.4 High-pass Filtering / 248
 - 11.4.1 Ideal HPF / 248
 - 11.4.2 Gaussian HPF / 250
 - 11.4.3 Butterworth HPF / 250
 - 11.4.4 High-Frequency Emphasis / 251
- 11.5 Tutorial 11.1: 2D Fourier Transform / 252
- 11.6 Tutorial 11.2: Low-pass Filters in the Frequency Domain / 254
- 11.7 Tutorial 11.3: High-pass Filters in the Frequency Domain / 258
- 11.8 Problems / 264

12 IMAGE RESTORATION

265

- 12.1 Modeling of the Image Degradation and Restoration Problem / 265
- 12.2 Noise and Noise Models / 266
 - 12.2.1 Selected Noise Probability Density Functions / 267
 - 12.2.2 Noise Estimation / 269
- 12.3 Noise Reduction Using Spatial-domain Techniques / 269
 - 12.3.1 Mean Filters / 273
 - 12.3.2 Order Statistic Filters / 275
 - 12.3.3 Adaptive Filters / 278
- 12.4 Noise Reduction Using Frequency-domain Techniques / 278
 - 12.4.1 Periodic Noise / 279
 - 12.4.2 Bandreject Filter / 280
 - 12.4.3 Bandpass Filter / 281

- 12.4.4 Notch Filter / 282
- 12.5 Image Deblurring Techniques / 283
 - 12.5.1 Wiener Filtering / 286
- 12.6 Tutorial 12.1: Noise Reduction Using Spatial-domain Techniques / 289
- 12.7 Problems / 296

13 MORPHOLOGICAL IMAGE PROCESSING

299

- 13.1 Introduction / 299
- 13.2 Fundamental Concepts and Operations / 300
 - 13.2.1 The Structuring Element / 301
- 13.3 Dilation and Erosion / 304
 - 13.3.1 Dilation / 305
 - 13.3.2 Erosion / 307
- 13.4 Compound Operations / 310
 - 13.4.1 Opening / 310
 - 13.4.2 Closing / 311
 - 13.4.3 Hit-or-Miss Transform / 313
- 13.5 Morphological Filtering / 314
- 13.6 Basic Morphological Algorithms / 315
 - 13.6.1 Boundary Extraction / 317
 - 13.6.2 Region Filling / 319
 - 13.6.3 Extraction and Labeling of Connected Components / 321
- 13.7 Grayscale Morphology / 322
 - 13.7.1 Dilation and Erosion / 323
 - 13.7.2 Opening and Closing / 323
 - 13.7.3 Top-Hat and Bottom-Hat Transformations / 325
- 13.8 Tutorial 13.1: Binary Morphological Image Processing / 325
- 13.9 Tutorial 13.2: Basic Morphological Algorithms / 330
- 13.10 Problems / 334

14 EDGE DETECTION

335

- 14.1 Formulation of the Problem / 335
- 14.2 Basic Concepts / 336
- 14.3 First-order Derivative Edge Detection / 338
- 14.4 Second-order Derivative Edge Detection / 343
 - 14.4.1 Laplacian of Gaussian / 345
- 14.5 The Canny Edge Detector / 347

- 14.6 Edge Linking and Boundary Detection / 348
 - 14.6.1 The Hough Transform / 349
- 14.7 Tutorial 14.1: Edge Detection / 354
- 14.8 Problems / 363

15 IMAGE SEGMENTATION

365

- 15.1 Introduction / 365
- 15.2 Intensity-based Segmentation / 367
 - 15.2.1 Image Thresholding / 368
 - 15.2.2 Global Thresholding / 369
 - 15.2.3 The Impact of Illumination and Noise on Thresholding / 370
 - 15.2.4 Local Thresholding / 371
- 15.3 Region-based Segmentation / 373
 - 15.3.1 Region Growing / 374
 - 15.3.2 Region Splitting and Merging / 377
- 15.4 Watershed Segmentation / 377
 - 15.4.1 The Distance Transform / 378
- 15.5 Tutorial 15.1: Image Thresholding / 379
- 15.6 Problems / 386

16 COLOR IMAGE PROCESSING

387

- 16.1 The Psychophysics of Color / 387
 - 16.1.1 Basic Concepts / 388
 - 16.1.2 The CIE XYZ Chromaticity Diagram / 390
 - 16.1.3 Perceptually Uniform Color Spaces / 393
 - 16.1.4 ICC Profiles / 395
- 16.2 Color Models / 396
 - 16.2.1 The RGB Color Model / 396
 - 16.2.2 The CMY and CMYK Color Models / 398
 - 16.2.3 The HSV Color Model / 398
 - 16.2.4 The YIQ (NTSC) Color Model / 401
 - 16.2.5 The YCbCr Color Model / 401
- 16.3 Representation of Color Images in MATLAB / 401
 - 16.3.1 RGB Images / 402
 - 16.3.2 Indexed Images / 403
- 16.4 Pseudocolor Image Processing / 406
 - 16.4.1 Intensity Slicing / 406

- 16.4.2 Gray Level to Color Transformations / 407
- 16.4.3 Pseudocoloring in the Frequency Domain / 408
- 16.5 Full-color Image Processing / 409
 - 16.5.1 Color Transformations / 410
 - 16.5.2 Histogram Processing / 412
 - 16.5.3 Color Image Smoothing and Sharpening / 412
 - 16.5.4 Color Noise Reduction / 414
 - 16.5.5 Color-Based Image Segmentation / 414
 - 16.5.6 Color Edge Detection / 417
- 16.6 Tutorial 16.1: Pseudocolor Image Processing / 419
- 16.7 Tutorial 16.2: Full-color Image Processing / 420
- 16.8 Problems / 425

17 IMAGE COMPRESSION AND CODING

427

- 17.1 Introduction / 427
- 17.2 Basic Concepts / 428
 - 17.2.1 Redundancy / 428
 - 17.2.2 Image Encoding and Decoding Model / 431
- 17.3 Lossless and Lossy Compression Techniques / 432
 - 17.3.1 Lossless Compression Techniques / 432
 - 17.3.2 Lossy Compression Techniques / 433
- 17.4 Image Compression Standards / 435
 - 17.4.1 Binary Image Compression Standards / 435
 - 17.4.2 Continuous Tone Still Image Compression Standards / 435
 - 17.4.3 JPEG / 436
 - 17.4.4 JPEG 2000 / 437
 - 17.4.5 JPEG-LS / 437
- 17.5 Image Quality Measures / 438
 - 17.5.1 Subjective Quality Measurement / 438
 - 17.5.2 Objective Quality Measurement / 439
- 17.6 Tutorial 17.1: Image Compression / 440

18 FEATURE EXTRACTION AND REPRESENTATION

447

- 18.1 Introduction / 447
- 18.2 Feature Vectors and Vector Spaces / 448
 - 18.2.1 Invariance and Robustness / 449
- 18.3 Binary Object Features / 450

| | |
|---------|--|
| 18.3.1 | Area / 450 |
| 18.3.2 | Centroid / 450 |
| 18.3.3 | Axis of Least Second Moment / 451 |
| 18.3.4 | Projections / 451 |
| 18.3.5 | Euler Number / 452 |
| 18.3.6 | Perimeter / 453 |
| 18.3.7 | Thinness Ratio / 453 |
| 18.3.8 | Eccentricity / 454 |
| 18.3.9 | Aspect Ratio / 454 |
| 18.3.10 | Moments / 455 |
| 18.4 | Boundary Descriptors / 456 |
| 18.4.1 | Chain Code, Freeman Code, and Shape Number / 459 |
| 18.4.2 | Signatures / 461 |
| 18.4.3 | Fourier Descriptors / 462 |
| 18.5 | Histogram-based (Statistical) Features / 464 |
| 18.6 | Texture Features / 466 |
| 18.7 | Tutorial 18.1: Feature Extraction and Representation / 470 |
| 18.8 | Problems / 474 |

19 VISUAL PATTERN RECOGNITION

475

| | |
|--------|--|
| 19.1 | Introduction / 475 |
| 19.2 | Fundamentals / 476 |
| 19.2.1 | Design and Implementation of a Visual Pattern Classifier / 476 |
| 19.2.2 | Patterns and Pattern Classes / 478 |
| 19.2.3 | Data Preprocessing / 479 |
| 19.2.4 | Training and Test Sets / 480 |
| 19.2.5 | Confusion Matrix / 480 |
| 19.2.6 | System Errors / 481 |
| 19.2.7 | Hit Rates, False Alarm Rates, and ROC Curves / 481 |
| 19.2.8 | Precision and Recall / 482 |
| 19.2.9 | Distance and Similarity Measures / 485 |
| 19.3 | Statistical Pattern Classification Techniques / 487 |
| 19.3.1 | Minimum Distance Classifier / 488 |
| 19.3.2 | k -Nearest Neighbors Classifier / 490 |
| 19.3.3 | Bayesian Classifier / 490 |
| 19.4 | Tutorial 19.1: Pattern Classification / 491 |
| 19.5 | Problems / 497 |

PART II VIDEO PROCESSING**20 VIDEO FUNDAMENTALS 501**

- 20.1 Basic Concepts and Terminology / 501
- 20.2 Monochrome Analog Video / 507
 - 20.2.1 Analog Video Raster / 507
 - 20.2.2 Blanking Intervals / 508
 - 20.2.3 Synchronization Signals / 509
 - 20.2.4 Spectral Content of Composite Monochrome Analog Video / 509
- 20.3 Color in Video / 510
- 20.4 Analog Video Standards / 512
 - 20.4.1 NTSC / 513
 - 20.4.2 PAL / 513
 - 20.4.3 SECAM / 514
 - 20.4.4 HDTV / 514
- 20.5 Digital Video Basics / 514
 - 20.5.1 Advantages of Digital Video / 515
 - 20.5.2 Parameters of a Digital Video Sequence / 516
 - 20.5.3 The Audio Component / 517
- 20.6 Analog-to-Digital Conversion / 517
- 20.7 Color Representation and Chroma Subsampling / 520
- 20.8 Digital Video Formats and Standards / 521
 - 20.8.1 The Rec. 601 Digital Video Format / 522
 - 20.8.2 The Common Intermediate Format / 523
 - 20.8.3 The Source Intermediate Format / 524
- 20.9 Video Compression Techniques and Standards / 524
 - 20.9.1 Video Compression Standards, Codecs, and Containers / 525
- 20.10 Video Processing in MATLAB / 526
 - 20.10.1 Reading Video Files / 527
 - 20.10.2 Processing Video Files / 527
 - 20.10.3 Playing Video Files / 527
 - 20.10.4 Writing Video Files / 528
- 20.11 Tutorial 20.1: Basic Digital Video Manipulation in MATLAB / 528
- 20.12 Tutorial 20.2: Working with YUV Video Data / 534
- 20.13 Problems / 539

21 VIDEO SAMPLING RATE AND STANDARDS CONVERSION 541

- 21.1 Video Sampling / 541
- 21.2 Sampling Rate Conversion / 542
- 21.3 Standards Conversion / 543
 - 21.3.1 Deinterlacing / 543
 - 21.3.2 Conversion between PAL and NTSC Signals / 545
 - 21.3.3 Color Space Conversion / 545
 - 21.3.4 Aspect Ratio Conversion / 546
 - 21.3.5 3:2 Pull-Down / 547
- 21.4 Tutorial 21.1: Line Down-Conversion / 548
- 21.5 Tutorial 21.2: Deinterlacing / 550
- 21.6 Tutorial 21.3: NTSC to PAL Conversion / 556
- 21.7 Tutorial 21.4: 3:2 Pull-Down / 557
- 21.8 Problems / 559

22 DIGITAL VIDEO PROCESSING TECHNIQUES AND APPLICATIONS 561

- 22.1 Fundamentals of Motion Estimation and Motion Compensation / 561
- 22.2 General Methodologies in Motion Estimation / 564
 - 22.2.1 Motion Representation / 566
 - 22.2.2 Motion Estimation Criteria / 567
 - 22.2.3 Optimization Methods / 567
- 22.3 Motion Estimation Algorithms / 568
 - 22.3.1 Exhaustive Search Block Matching Algorithm / 568
 - 22.3.2 Fast Algorithms / 570
 - 22.3.3 Hierarchical Block Matching Algorithm / 571
 - 22.3.4 Phase Correlation Method / 573
- 22.4 Video Enhancement and Noise Reduction / 573
 - 22.4.1 Noise Reduction in Video / 574
 - 22.4.2 Interframe Filtering Techniques / 575
- 22.5 Case Study: Object Segmentation and Tracking in the Presence of Complex Background / 576
- 22.6 Tutorial 22.1: Block-based Motion Estimation / 579
- 22.7 Tutorial 22.2: Intraframe and Interframe Filtering Techniques / 585
- 22.8 Problems / 589

Appendix A: HUMAN VISUAL PERCEPTION **591**

- A.1 Introduction / 591
- A.2 The Human Eye / 592
- A.3 Characteristics of Human Vision / 596
 - A.3.1 Resolution, Viewing Distance, and Viewing Angle / 596
 - A.3.2 Detail and Sharpness Perception / 598
 - A.3.3 Optical Transfer Function and Modulation Transfer Function / 599
 - A.3.4 Brightness Perception / 600
 - A.3.5 Contrast Ratio and Contrast Sensitivity Function / 603
 - A.3.6 Perception of Motion / 605
 - A.3.7 Spatiotemporal Resolution and Frequency Response / 606
 - A.3.8 Masking / 608
- A.4 Implications and Applications of Knowledge about the Human Visual System / 609

Appendix B: GUI DEVELOPMENT **611**

- B.1 Introduction / 611
- B.2 GUI File Structure / 611
- B.3 Passing System Control / 613
- B.4 The UserData Object / 615
- B.5 A Working GUI Demo / 616
- B.6 Concluding Remarks / 618

REFERENCES **619****INDEX** **627**

LIST OF FIGURES

| | | |
|------|--|----|
| 1.1 | Image sharpening: (a) original image; (b) after sharpening. | 7 |
| 1.2 | Noise removal: (a) original (noisy) image; (b) after removing noise. | 7 |
| 1.3 | Deblurring: (a) original (blurry) image; (b) after removing the (motion) blur. Original image: courtesy of MathWorks. | 8 |
| 1.4 | Edge extraction: (a) original image; (b) after extracting its most relevant edges. Original image: courtesy of MathWorks. | 8 |
| 1.5 | Binarization: (a) original grayscale image; (b) after conversion to a black-and-white version. Original image: courtesy of MathWorks. | 9 |
| 1.6 | Blurring: (a) original image; (b) after blurring to remove unnecessary details. Original image: courtesy of MathWorks. | 9 |
| 1.7 | Contrast enhancement: (a) original image; (b) after histogram equalization to improve contrast. | 9 |
| 1.8 | Object segmentation and labeling: (a) original image; (b) after segmenting and labeling individual objects. Original image: courtesy of MathWorks. | 10 |
| 1.9 | Components of a digital image processing system. Adapted and redrawn from [Umb05]. | 11 |
| 1.10 | Diagram of a machine vision system. Adapted and redrawn from [GW08]. | 13 |
| 1.11 | Test image for the design of a machine vision system to read the label of the main integrated circuit on a printed circuit board. | 18 |

| | | |
|------|--|----|
| 1.12 | (a) Test image for distance estimation: parallel lines with up to 5% difference in length. (b) Test image for area estimation: circles with up to 10% difference in radius. Both images are adapted and redrawn from [Jah05]. | 19 |
| 1.13 | (a) Test image for texture-based object segmentation. (b) Test image for object segmentation based on “interpolation” of object boundaries. Both images are adapted and redrawn from [Jah05]. | 19 |
| 2.1 | A monochrome image and the convention used to represent rows (x) and columns (y) adopted in this book. | 22 |
| 2.2 | A binary image and the pixel values in a 6×6 neighborhood. Original image: courtesy of MathWorks. | 23 |
| 2.3 | A grayscale image and the pixel values in a 6×6 neighborhood. | 24 |
| 2.4 | Color image (a) and its R (b), G (c), and B (d) components. | 25 |
| 2.5 | An indexed color image and the indices in a 4×4 neighborhood. Original image: courtesy of MathWorks. | 26 |
| 2.6 | Pixels within a neighborhood. | 28 |
| 2.7 | Concept of neighborhood of pixel p (from an image topology perspective): (a) 4-neighborhood; (b) diagonal neighborhood; (c) 8-neighborhood. | 28 |
| 2.8 | Connected components: (a) original (binary) image; (b) results for 8-connectivity; (c) results for 4-connectivity. | 29 |
| 2.9 | Example of intensity reduction using a transformation function: (a) original image; (b) output image. | 31 |
| 2.10 | A 3×3 convolution mask, whose generic weights are W_1, \dots, W_9 . | 32 |
| 2.11 | Pixel-by-pixel arithmetic and logic operations. | 33 |
| 2.12 | Operations in a transform domain. | 33 |
| 3.1 | MATLAB environment. | 45 |
| 4.1 | Displaying an image: (a) without scaling; (b) scaling for display purposes; (c) selecting only pixels within a specified range. Original image: courtesy of MathWorks. | 69 |
| 4.2 | Displaying an image and exploring its contents with the <i>Pixel Region</i> tool. Original image: courtesy of MathWorks. | 69 |
| 4.3 | The <i>Image Information</i> tool. | 70 |
| 4.4 | The <i>Adjust Contrast</i> tool. Original image: courtesy of MathWorks. | 70 |
| 4.5 | The <i>Distance</i> tool. Original image: courtesy of MathWorks. | 71 |
| 4.6 | Reading and writing images: (a) Original image (PNG); (b) compressed image (JPG, $q = 75$, file size = 24 kB); (c) compressed image (JPG, $q = 5$, file size = 8 kB); (d) compressed image (JPG, $q = 95$, file size = 60 kB). Original image: courtesy of MathWorks. | 72 |

| | | |
|------|---|-----|
| 4.7 | Division of a figure using subplot. | 77 |
| 5.1 | Image acquisition, formation, and digitization. Adapted and redrawn from [GW08]. | 84 |
| 5.2 | Electromagnetic spectrum. | 85 |
| 5.3 | Recording the various types of interaction of radiation with objects and surfaces. Redrawn from [Bov00a]. | 86 |
| 5.4 | Newton's prism: many "colors" in the sunlight. | 87 |
| 5.5 | Spectral power distributions of common physical light sources. Redrawn from [Pra07]. | 88 |
| 5.6 | The Bayer pattern for single-CCD cameras. | 90 |
| 5.7 | The beam splitter for three-CCD color cameras. | 91 |
| 5.8 | X3 color sensor. | 91 |
| 5.9 | Image formation using a lens. | 92 |
| 5.10 | Examples of lens aberrations: (a) pincushion distortion; (b) barrel distortion. | 93 |
| 5.11 | The main components of MATLAB Image Acquisition Toolbox. | 94 |
| 5.12 | Digitization = sampling + quantization. Redrawn from [Poy03]. | 95 |
| 5.13 | Pixel arrays of several imaging standards. Redrawn from [Poy03]. | 95 |
| 5.14 | 1D aliasing explanation. Redrawn from [Wat00]. | 96 |
| 5.15 | A mapping function for uniform quantization ($N = 4$). | 97 |
| 5.16 | Effects of sampling resolution on image quality: (a) A 1944×2592 image, 256 gray levels, at a 1250 dpi resolution. The same image resampled at (b) 300 dpi; (c) 150 dpi; (d) 72 dpi. | 98 |
| 5.17 | (a) A 480×640 image, 256 gray levels; (b–h) image requantized to 128, 64, 32, 16, 8, 4, and 2 gray levels. | 99 |
| 6.1 | Adding two images: (a) first image (X); (b) second image (Y); (c) result ($Z = X + Y$). | 104 |
| 6.2 | Additive image offset: (a) original image (X); (b) brighter version ($Z = X + 75$). | 104 |
| 6.3 | Adding noise to an image: (a) original image (X); (b) zero-mean Gaussian white noise (variance = 0.01) (N); (c) result ($Z = X + N$). | 105 |
| 6.4 | Subtractive image offset: (a) original image (X); (b) darker version ($Z = X - 75$). | 107 |
| 6.5 | Example of an image negative: (a) original image; (b) negative image. | 109 |
| 6.6 | Multiplication and division by a constant: (a) original image (X); (b) multiplication result ($X \times 0.7$); (c) division result ($X/0.7$). | 109 |
| 6.7 | Logic operations on binary images. | 112 |

| | | |
|------|---|-----|
| 6.8 | The AND operation applied to monochrome images: (a) X ; (b) Y ; (c) $X \text{ AND } Y$. | 112 |
| 6.9 | The OR operation applied to monochrome images: (a) X ; (b) Y ; (c) $X \text{ OR } Y$. | 112 |
| 6.10 | The XOR operation applied to monochrome images: (a) X ; (b) Y ; (c) $X \text{ XOR } Y$. | 113 |
| 6.11 | The NOT operation applied to a monochrome image: (a) X ; (b) NOT X . | 113 |
| 7.1 | Examples of typical geometric operations: (a) original image; (b) translation (shifting); (c) scaling (resizing); (d) rotation. | 126 |
| 7.2 | Mapping one triangle onto another by an affine transformation. | 128 |
| 7.3 | Forward mapping: for each pixel position in the input image, the corresponding (continuous) target position—resulting from applying a geometric transformation T —is found in the output image. In general, the target position (x', y') does not coincide with any discrete raster point, and the value of the pixel in the input image is copied to one of the adjacent target pixels. Redrawn from [BB08]. | 130 |
| 7.4 | Backward mapping: for each discrete pixel position in the output image, the corresponding continuous position in the input image (x, y) is found by applying the inverse mapping function T^{-1} . The new pixel value is found by interpolation among the neighbors of (x, y) in the input image. Redrawn from [BB08]. | 131 |
| 7.5 | Effects of different interpolation techniques on rotated images: (a) original image; zoomed-in versions of rotated (35°) image using (b) zero-order (nearest-neighbor) interpolation; (c) first-order (bilinear) interpolation; (d) third-order (bicubic) interpolation. | 133 |
| 7.6 | Image deformation effects using <i>Photo Booth</i> . | 136 |
| 7.7 | Using seam carving for content-aware resizing: (a) original image (334×500 pixels); (b) cropped image (256×256 pixels). Original image from Flickr. Seam carving results were obtained using the publicly available implementation by Mathias Lux: http://code.google.com/p/java-imageseams/ . | 138 |
| 7.8 | Image registration using MATLAB and the IPT. | 145 |
| 7.9 | Interactive image registration: (a) base image; (b) unregistered image. | 145 |
| 7.10 | The <i>Control Point Selection</i> tool. | 146 |
| 7.11 | Selected points. | 146 |
| 8.1 | The image enhancement process. Adapted and redrawn from [Umb05]. | 152 |
| 8.2 | Basic gray-level transformation functions. | 153 |

| | | |
|------|--|-----|
| 8.3 | Linear point transformations example: input image. | 154 |
| 8.4 | Linear point transformations and their impact on the overall brightness and contrast of an image: brightening (left), darkening (middle), and contrast reduction (right). | 155 |
| 8.5 | Examples of gray-level transformations for contrast enhancement. Redrawn from [GW08]. | 155 |
| 8.6 | Autocontrast operation. Redrawn from [BB08]. | 156 |
| 8.7 | (a) Example of an image whose original gray-level range was [90, 162]; (b) the result of applying the autocontrast transformation (equation (8.4)). | 157 |
| 8.8 | Examples of power law transformations for different values of γ . | 158 |
| 8.9 | Examples of gamma correction for two different values of γ : 0.5 (left) and 2.2 (right). | 158 |
| 8.10 | Example of using log transformation: (a) Fourier spectrum (amplitude only) of the rice image (available in MATLAB); (b) result of applying equation (8.6) with $c = 1$ followed by autocontrast. | 159 |
| 8.11 | Piecewise linear transformation using <code>glsdemo</code> . | 160 |
| 8.12 | Gray-level slicing using <code>glsdemo</code> . Original image: courtesy of MathWorks. | 160 |
| 8.13 | Example of using a lookup table: (a) input image; (b) transformation function specified by equation (8.7); (c) output image. | 162 |
| 9.1 | Example of histogram for an image with eight gray levels. | 173 |
| 9.2 | Examples of images and corresponding histograms. Original image in part (b): courtesy of MathWorks. | 175 |
| 9.3 | Transformation function used for histogram equalization. | 177 |
| 9.4 | Equalized histogram—graph. | 178 |
| 9.5 | Use of histogram equalization to improve image contrast. | 179 |
| 9.6 | Global versus local histogram equalization. Original image: courtesy of MathWorks. | 180 |
| 9.7 | Histogram matching: (a) desired (specified) histogram; (b) resulting histogram. | 182 |
| 9.8 | Histogram matching: (a) original image; (b) resulting image; (c) original histogram; (d) desired histogram; (e) resulting histogram. | 184 |
| 9.9 | Histogram sliding: (a) original image; (b) result of sliding to the right by 50; (c) result of sliding to the left by 50; (d–f) histograms corresponding to images in (a)–(c). | 185 |
| 9.10 | Example of using histogram stretching to improve contrast: (a) original image ($r_{\min} = 129$, $r_{\max} = 204$); (b) result of | |

| | | |
|-------|---|-----|
| | stretching using equation (9.12); (c and d) histograms corresponding to images in (a) and (b). | 186 |
| 9.11 | Example of using histogram shrinking to reduce contrast: (a) original image; (b) result of shrinking using equation (9.13) with $r_{\min} = 4$, $r_{\max} = 254$, $s_{\min} = 49$, and $s_{\max} = 140$; (c and d) histograms corresponding to images in (a) and (b). | 187 |
| 9.12 | Gamma transformations for different values of gamma. Redrawn from [GWE04]. | 196 |
| 10.1 | Neighborhood processing for the case of linear filtering. | 205 |
| 10.2 | Two-dimensional convolution example. | 208 |
| 10.3 | Applying different convolution masks to the same input image: (a) original image; (b–d) result of 2D convolution using the masks in Table 10.1. | 209 |
| 10.4 | Border geometry. Redrawn from [BB08]. | 211 |
| 10.5 | Examples of applying the averaging filter with different mask sizes: (a) input image (899×675 pixels); (b–d) output images corresponding to averaging masks of size 7×7 , 15×15 , and 31×31 . | 214 |
| 10.6 | A 2D Gaussian function (with $\sigma = 3$). | 216 |
| 10.7 | Example of using Gaussian blur filters. | 217 |
| 10.8 | Median filter. Redrawn from [BB08]. | 217 |
| 10.9 | (a) Original image; (b) image with salt and pepper noise; (c) result of 3×3 median filtering; (d) result of 3×3 neighborhood averaging. | 218 |
| 10.10 | Example of using Laplacian masks to enhance an image. | 221 |
| 10.11 | Example of region of interest processing: (a) original image; (b) result of applying a Gaussian blur to a selected ROI; (c) result of applying a HPF to a selected ROI; (d) result of applying a Laplacian mask to a selected ROI. | 222 |
| 10.12 | A 3×3 image region. | 224 |
| 10.13 | A 3×3 mask. | 225 |
| 10.14 | Uniform and nonuniform averaging masks. | 227 |
| 10.15 | Laplacian masks that account for corner pixels (standard and composite). | 230 |
| 10.16 | Unsharp masking process including histogram adjustment. | 230 |
| 10.17 | Unsharp masking process with sharpening image. | 231 |
| 10.18 | Unsharp masking process using convolution mask. | 232 |
| 10.19 | High-boost masks with and without regard to corner pixels. | 232 |
| 11.1 | Frequency-domain operations. | 236 |

| | | |
|-------|--|-----|
| 11.2 | Two examples of response functions for frequency-domain filters: (a) low-pass filter equivalent to a 3×3 average filter in the spatial domain; (b) high-pass filter equivalent to a 3×3 composite Laplacian sharpening filter in the spatial domain. | 237 |
| 11.3 | Operations in a transform domain. | 238 |
| 11.4 | (a) Original image (256×256 pixels); (b) Fourier spectrum of the image in (a). | 240 |
| 11.5 | Original image (a) and its 2D FT spectrum (b); rotated image (c) and its 2D FT spectrum (d). | 242 |
| 11.6 | Example of using LPF to smooth false contours: (a) original image; (b) result of applying a LPF. | 243 |
| 11.7 | Example of using LPF for noise reduction: (a) original image; (b) result of applying a LPF. | 243 |
| 11.8 | Frequency response plot for an ideal LPF: (a) 3D view; (b) 2D view from the top. | 244 |
| 11.9 | (a) Original image (256×256 pixels); (b) Fourier spectrum of the image in (a). The rings represent cutoff frequencies for the low-pass filter examples described later. | 244 |
| 11.10 | (a) Original image (256×256 pixels); (b–f) ideal LPF results for filters with cutoff frequency corresponding to the radii in Figure 11.9b, namely, 8, 16, 32, 64, and 128 pixels. | 245 |
| 11.11 | Frequency response plot for a Gaussian LPF: (a) 3D view; (b) 2D view from the top. | 246 |
| 11.12 | (a) Original image (256×256 pixels); (b–f) Gaussian LPF results for filters with different values for σ : 5, 10, 20, 30, and 75. | 247 |
| 11.13 | Frequency response plot for a Butterworth LPF of order $n = 4$: (a) 3D view; (b) 2D view from the top. | 248 |
| 11.14 | (a) Original image (512×512 pixels); (b–f) fourth-order Butterworth LPF results for filters with cutoff frequency corresponding to the radii in Figure 11.9b, namely, 8, 16, 32, 64, and 128 pixels. | 249 |
| 11.15 | Frequency response plot for an ideal HPF: (a) 3D view; (b) 2D view from the top. | 250 |
| 11.16 | Frequency response plot for a Gaussian HPF: (a) 3D view; (b) 2D view from the top. | 250 |
| 11.17 | Frequency response plot for a Butterworth HPF of order $n = 4$: (a) 3D view; (b) 2D view from the top. | 251 |
| 11.18 | High-frequency emphasis: (a) input image; (b) result of applying a second-order Butterworth HPF (with $D_0 = 30$) to the input image; (c) result of high-frequency emphasis with $a = 0.5$ and $b = 1$. | 251 |

| | | |
|-------|--|-----|
| 12.1 | Image degradation and restoration. | 266 |
| 12.2 | Histograms of representative noise types: (a) Gaussian, (b) impulse (salt and pepper), (c) uniform, (d) Rayleigh, (e) gamma (Erlang), and (e) exponential. Redrawn from [Pra07]. | 270 |
| 12.3 | Test images and corresponding histograms for different types of noise: (a and b) Gaussian; (c and d) exponential; (e and f) salt and pepper. | 271 |
| 12.4 | Test images and corresponding histograms for different types of noise: (a and b) Rayleigh; (c and d) Gamma; (e and f) uniform. | 272 |
| 12.5 | Estimating noise type from a homogeneous patch within an image: (a) original image; (b) noisy image (where the rectangle indicates a manually selected patch); (c) histogram of the original image; (d) histogram of the noisy image; (e) histogram of selected patch showing clearly that the noise is of Gaussian type in this case. | 273 |
| 12.6 | (a) Original image; (b) image with Gaussian noise; (c) result of 3×3 arithmetic mean filtering; (d) result of 5×5 arithmetic mean filtering; (e) result of 3×3 geometric mean filtering; (f) result of 3×3 harmonic mean filtering. | 275 |
| 12.7 | (a) Image with salt and pepper noise; (b) result of 3×3 arithmetic mean filtering; (c) result of 3×3 geometric mean filtering; (d) result of 3×3 harmonic mean filtering; (e) result of 3×3 contraharmonic mean filtering with $R = 0.5$; (f) result of 3×3 contraharmonic mean filtering with $R = -0.5$. | 276 |
| 12.8 | (a) Image with salt and pepper noise; (b) result of 3×3 arithmetic mean filtering (for comparison); (c) result of 3×3 median filtering; (d) result of 3×3 midpoint filtering. | 279 |
| 12.9 | Example of an image corrupted by periodic noise: (a) noisy image; (b) periodic noise component; (c) the Fourier spectrum of the noise component (bright dots were enlarged for viewing purposes). | 280 |
| 12.10 | Example of using a bandreject filter to reduce periodic noise: (a) noisy image; (b) noisy image spectrum (the eight spots corresponding to the noise have been made brighter and bigger for visualization purposes); (c) the Fourier spectrum of the image after applying the bandreject filter; (d) resulting image. | 281 |
| 12.11 | Example of image restoration using inverse filtering: (a) input (blurry) image; (b) result of naive inverse filtering; (c) applying a 10th-order Butterworth low-pass filter with cutoff frequency of 20 to the division; (d) same as (c), but with cutoff frequency of 50; (e) results of using constrained division, with threshold $T = 0.01$; (f) same as (e), but with threshold $T = 0.001$. | 285 |

| | | |
|-------|---|-----|
| 12.12 | Example of motion deblurring using inverse filtering: (a) input image; (b) result of applying inverse filtering with constrained division and threshold $T = 0.05$: the motion blurred has been removed at the expense of the appearance of vertical artifacts. | 286 |
| 12.13 | Example of image restoration using Wiener filtering: (a) input image (blurry and noisy); (b) result of inverse filtering, applying a 10th-order Butterworth low-pass filter with cutoff frequency of 50 to the division; (c) results of Wiener filter, with $K = 10^{-3}$; (d) same as (c), but with $K = 0.1$. | 287 |
| 12.14 | Example of image restoration using Wiener filtering: (a) input (blurry) image; (b) result of inverse filtering, applying a 10th-order Butterworth low-pass filter with cutoff frequency of 50 to the division; (c) results of Wiener filter, with $K = 10^{-5}$; (d) same as (c), but with $K = 0.1$. | 288 |
| 13.1 | Basic set operations: (a) set A ; (b) translation of A by $x = (x_1, x_2)$; (c) set B ; (d) reflection of B ; (e) set A and its complement A^c ; (f) set difference $(A - B)$. | 301 |
| 13.2 | Logical equivalents of set theory operations: (a) Binary image (A); (b) Binary image (B); (c) Complement (A^c); (d) Union ($A \cup B$); (e) Intersection ($A \cap B$); (f) Set difference ($A - B$). | 302 |
| 13.3 | Examples of structuring elements: (a) square; (b) cross. | 302 |
| 13.4 | Example of dilation using three different rectangular structuring elements. | 305 |
| 13.5 | Example of erosion using three different rectangular structuring elements. | 308 |
| 13.6 | Example of morphological opening. | 311 |
| 13.7 | Geometric interpretation of the morphological opening operation. | 311 |
| 13.8 | Example of morphological closing. | 312 |
| 13.9 | Geometric interpretation of the morphological closing operation. Adapted and redrawn from [GW08]. | 313 |
| 13.10 | Example of HoM transform. | 314 |
| 13.11 | Morphological filtering. (a) input (noisy) image; (b) partial result (after opening) with SE of radius = 2 pixels; (c) final result with SE of radius = 2 pixels; (d) final result with SE of radius = 4 pixels. | 316 |
| 13.12 | Morphological algorithms. (a) input image; (b) skeleton of (a); (c) pruning spurious pixels from (b); (d) removing interior pixels from (a); (e) thickening the image in (d); (f) thinning the image in (e). Original image: courtesy of MathWorks. | 318 |
| 13.13 | Boundary extraction. | 319 |
| 13.14 | Region filling: (a) input image; (b) complement of (a); (c) partial results (numbered according to the iteration in the algorithm described by equation (13.27); (d) final result; (e) structuring element. | 320 |

| | | |
|-------|---|-----|
| 13.15 | Extraction of connected components: (a) input image; (b) first iteration; (c) second iteration; (d) final result, showing the contribution of each iteration (indicated by the numbers inside the squares); (e) structuring element. | 322 |
| 13.16 | Grayscale erosion and dilation with a nonflat ball-shaped structuring element with radius 5: (a) input image; (b) result of dilation; (c) result of erosion. | 323 |
| 13.17 | Grayscale opening and closing with a flat disk-shaped structuring element with radius 3: (a) input image (Gaussian noise); (b) result of opening image (a); (c) result of closing image (b); (d) input image (salt and pepper noise); (e) result of opening image (d); (f) result of closing image (e). | 324 |
| 13.18 | Example of using top-hat and bottom-hat filtering for contrast improvement: (a) input image; (b) output image. | 326 |
| 13.19 | Combining two structuring elements into one for the HoM transformation. | 330 |
| 14.1 | Ideal and ramp edges: (a) ideal edge on a digital image and corresponding profile along a horizontal line; (b) ramp edge and corresponding profile. | 337 |
| 14.2 | Grayscale image containing two regions separated by a ramp edge: intensity profile and, first and second derivative results. | 338 |
| 14.3 | First- and second-order edge detectors with and without noise: (a) original image; (b) first derivative; (c) second derivative; (d–f) horizontal profiles for images (a)–(c); (g–i) noisy versions of images (a)–(c); (j–l) horizontal profiles for images (g)–(i). | 339 |
| 14.4 | Edge detection example: (a) original image; (b) result of Prewitt horizontal kernel; (c) result of Prewitt vertical kernel; (d) combination of (b) and (c). | 341 |
| 14.5 | Edge detection using Sobel operator: (a) original image; (b) result of Sobel horizontal kernel; (c) result of Sobel vertical kernel; (d) combination of (b) and (c). | 342 |
| 14.6 | Kirsch compass masks. | 343 |
| 14.7 | Robinson compass masks. | 343 |
| 14.8 | Edge detection using Sobel operator and thresholding (the original image is the same as Figure 14.5a): (a) threshold of 0; (b) threshold of 0.05; (c) threshold of 0.1138 (the best value); (d) threshold of 0.2. | 344 |
| 14.9 | Edge detection using the zero-cross edge detector: (a) input image (without noise); (b) results using default values; (c) results using threshold zero; (d) noisy input image; (e) results using | |

| | | |
|-------|--|-----|
| | default values; (f) results using threshold zero. Edge results have been inverted for clarity. | 345 |
| 14.10 | Laplacian of Gaussian: (a) 3D plot; (b) 2D intensity plot; (c) cross section of (a). | 346 |
| 14.11 | Edge detection using the LoG edge detector: (a) input image; (b) results using default values; (c) results using $\sigma = 1$; (d) results using $\sigma = 3$. Edge results have been inverted for clarity. | 347 |
| 14.12 | Edge detection using the Canny edge detector: (a) default values ($\sigma = 1$, $T_{\text{low}} = 0.0625$, $T_{\text{high}} = 0.1563$); (b) $\sigma = 0.5$; (c) $\sigma = 2$; (d) $\sigma = 1$, $T_{\text{low}} = 0.01$, $T_{\text{high}} = 0.1$. | 349 |
| 14.13 | The Hough transform maps a point into a line. | 350 |
| 14.14 | The Hough transform: intersections in the transform domain correspond to aligned points in the image. | 350 |
| 14.15 | The Hough transform: a line and its parameters in the polar coordinate system. | 351 |
| 14.16 | Hough transform example: (a) input image; (b) results of Hough transform, highlighting the intersections corresponding to the predominant lines in the input image. | 352 |
| 14.17 | Hough transform example: (a) results of Hough transform highlighting the two highest peaks; (b) (negative of) edge detection results; (c) lines corresponding to the longest peaks overlaid on top of original image. | 353 |
| 14.18 | Kirsch masks stored in a $3 \times 3 \times 8$ matrix. | 359 |
| 15.1 | Test images for segmentation algorithms: (a) a <i>hard</i> test image and (b) its grayscale equivalent; (c) an easier test image (courtesy of MathWorks) and (d) the result of morphological preprocessing and thresholding. | 366 |
| 15.2 | The histogram for the image in Figure 15.1c: an example of histogram suitable for partitioning using a single threshold. | 368 |
| 15.3 | Image thresholding results for the image in Figure 15.1c using iterative threshold selection algorithm (a) and manually selected threshold (b). | 370 |
| 15.4 | An example of uneven illumination pattern used to generate the image in Figure 15.5a. | 371 |
| 15.5 | Effect of illumination (left) and noise (right) on thresholding. See text for details. | 372 |
| 15.6 | Local thresholding. Using a single threshold for the entire image (a) and dividing it up into six slices and choosing a different threshold for each vertical slice (b). | 373 |
| 15.7 | Region growing: (a) seed pixels; (b) first iteration; (c) final iteration. | 375 |
| 15.8 | Region growing results for two test images. See text for details. | 376 |

| | | |
|-------|--|-----|
| 15.9 | The quadtree data structure used in the split and merge segmentation algorithm (a) and the corresponding regions in the image (b). | 377 |
| 15.10 | Segmentation using the morphological watershed transform: (a) complement of the image shown in Figure 15.3; (b) distance transform; (c) watershed ridge lines; (d) result of segmentation. | 380 |
| 15.11 | Histogram plot with data cursor selection. | 381 |
| 16.1 | Spectral absorption curves of the short (S), medium (M), and long (L) wavelength pigments in human cone and rod (R) cells. Courtesy of Wikimedia Commons. | 389 |
| 16.2 | Additive (a) and subtractive (b) color mixtures. | 389 |
| 16.3 | RGB color matching function (CIE 1931). Courtesy of Wikimedia Commons. | 390 |
| 16.4 | XYZ color matching function (CIE 1931). Courtesy of Wikimedia Commons. | 391 |
| 16.5 | CIE XYZ color model. | 393 |
| 16.6 | Color gamut for three different devices: (a) CRT monitor; (b) printer; (c) film. The RGB triangle is the same in all figures to serve as a reference for comparison. | 393 |
| 16.7 | MacAdam ellipses overlapped on the CIE 1931 chromaticity diagram. Courtesy of Wikimedia Commons. | 394 |
| 16.8 | RGB color model. | 396 |
| 16.9 | RGB color cube. | 397 |
| 16.10 | The HSV color model as a hexagonal cone. | 399 |
| 16.11 | The HSV color model as a cylinder. | 400 |
| 16.12 | The HSV color model as a cone. | 400 |
| 16.13 | RGB color image representation. | 402 |
| 16.14 | RGB image and its three color components (or <i>channels</i>). Original image: courtesy of MathWorks. | 403 |
| 16.15 | Indexed color image representation. | 404 |
| 16.16 | A built-in indexed image. Original image: courtesy of MathWorks. | 405 |
| 16.17 | Pseudocoloring with intensity slicing. | 407 |
| 16.18 | An alternative representation of the intensity slicing technique for an image with L gray levels pseudocolored using four colors. | 407 |
| 16.19 | Pseudocoloring using intensity slicing: original image (a) and results of pseudocoloring using different color maps (b–d). Original image: courtesy of MathWorks. | 408 |
| 16.20 | Block diagram for pseudocoloring using color transformation functions. | 409 |

| | | |
|-------|---|-----|
| 16.21 | (a) Block diagram for pseudocoloring in the frequency domain; (b) frequency response of the filters. Redrawn from [Umb05]. | 409 |
| 16.22 | RGB processing. | 411 |
| 16.23 | Intensity processing using RGB to YIQ color space conversions. | 411 |
| 16.24 | Example of color histogram equalization. (a) Original image and its Y channel histogram; (b) output image and its equalized Y channel histogram. Original image: courtesy of MathWorks. | 413 |
| 16.25 | Spatial convolution masks for grayscale and RGB color images. | 413 |
| 16.26 | Thresholding in RGB space. | 415 |
| 16.27 | Defining spherical (ellipsoidal) regions in RGB space. | 416 |
| 16.28 | Example of color segmentation using requantization. | 416 |
| 16.29 | Another example of color segmentation using requantization: (a) original image; (b) requantized image with two color levels; (c) requantized image with five color levels. | 417 |
| 16.30 | Color edge detection example: (a) original image; (b) grayscale equivalent; (c) edge detection on (b); (d) edge detection on individual RGB components; (e) edge detection on Y component only; (f) edge detection on V component only. | 418 |
| 17.1 | Two ways to represent the same information using different amounts of data. See text for details. | 429 |
| 17.2 | A general image encoding and decoding model. | 431 |
| 17.3 | Source encoder. | 431 |
| 17.4 | Lossless predictive encoder. | 433 |
| 17.5 | Transform coding diagram. | 434 |
| 17.6 | JPEG encoder and decoder. | 436 |
| 17.7 | Measuring objective image quality after compression: (a) original; (b) compressed version of (a) (using quality factor = 90), $e_{\text{rms}} = 2.1647$, PSNR = 41.4230 dB; (c) compressed version of (a) (using quality factor = 5), $e_{\text{rms}} = 7.6188$, PSNR = 30.4931 dB. | 441 |
| 17.8 | The problem of poor correlation between objective and subjective measures of image quality: (a) original; (b) blurred version of (a) (using a 5×5 average filter), $e_{\text{rms}} = 0.0689$, PSNR = 71.3623 dB; (c) partially blurred version of (a) (after applying a severe blurring filter only to a small part of the image), $e_{\text{rms}} = 0.0629$, PSNR = 72.1583 dB. | 442 |
| 18.1 | Test image (a) and resulting 2D feature vectors (b). | 449 |
| 18.2 | Axis of least second moment. | 451 |
| 18.3 | Horizontal and vertical projections. | 452 |

| | | |
|-------|--|-----|
| 18.4 | Examples of two regions with Euler numbers equal to 0 and -1 , respectively. | 453 |
| 18.5 | Examples of a compact (a) and a noncompact (b) regions. | 454 |
| 18.6 | Eccentricity (A/B) of a region. | 454 |
| 18.7 | Elongatedness (a/b) of a region. | 455 |
| 18.8 | Tracing boundaries of objects. | 459 |
| 18.9 | Tracing boundaries of objects and holes. | 459 |
| 18.10 | Chain code and Freeman code for a contour: (a) original contour; (b) subsampled version of the contour; (c) chain code representation; (d) Freeman code representation. | 460 |
| 18.11 | Chain code, first differences, and shape number. | 461 |
| 18.12 | Distance \times angle signatures for two different objects. Redrawn from [GW08]. | 462 |
| 18.13 | Effect of noise on signatures for two different objects. Redrawn from [GW08]. | 462 |
| 18.14 | Fourier descriptor of a boundary. | 463 |
| 18.15 | Example of boundary reconstruction using Fourier descriptors: (a) original image; (b–f) reconstructed image using 100%, 50%, 25%, 2.5%, and 1% of the total number of points, respectively. | 464 |
| 18.16 | Example of images with smooth (a), coarse (b), and regular (c) texture. Images from the Brodatz textures data set. Courtesy of http://tinyurl.com/brodatz . | 466 |
| 18.17 | Histograms of images in Figure 18.16. | 467 |
| 18.18 | An image (a) and its cooccurrence matrix for $\mathbf{d} = (0, 1)$ (b). | 468 |
| 18.19 | An image (a) and its cooccurrence matrix for $\mathbf{d} = (1, 0)$ (b). | 468 |
| 18.20 | Test images for this tutorial: (a) steps 1–6; (b) step 7; (c) step 11. | 470 |
| 19.1 | Diagram of a statistical pattern classifier. Redrawn from [SS01]. | 476 |
| 19.2 | The interplay between feature extraction, feature selection, and pattern classification as a function of the application at hand. Adapted and redrawn from [Umb05]. | 477 |
| 19.3 | Example of two classes (<i>sumo wrestlers</i> —red circles—and <i>table tennis players</i> —blue diamonds) described by two measurements (<i>weight</i> and <i>height</i>). | 479 |
| 19.4 | Example of 4×4 confusion matrix. | 481 |
| 19.5 | Example of ROC curve. | 482 |
| 19.6 | Example of precision–recall (PR) graph. | 484 |
| 19.7 | Precision–recall graph for Example 19.4. | 485 |
| 19.8 | Discrimination functions for a three-class classifier in a 2D feature space. | 487 |

| | | |
|-------|--|-----|
| 19.9 | Example of two classes and their mean vectors. | 489 |
| 19.10 | Example of three classes with relatively complex structure. | 489 |
| 19.11 | (a) Example of a KNN classifier ($k = 1$) for a five-class classifier in a 2D feature space (obtained using the STPRTool toolbox). (b) Minimum distance classifier results for the same data set. | 490 |
| 19.12 | Feature space for training set. Obtained using the <i>Statistical Pattern Recognition Toolbox (STPRtool)</i> , available at http://cmp.felk.cvut.cz/cmp/software/stprtool/ . | 493 |
| 19.13 | Confusion matrix with results of KNN classifier for the selected features. Obtained using the <i>Statistical Pattern Recognition Toolbox (STPRtool)</i> , available at http://cmp.felk.cvut.cz/cmp/software/stprtool/ . | 495 |
| 19.14 | Number of Confusion matrix with results of KNN classifier for the case where the images' gray values are used as "features." Obtained using the <i>Statistical Pattern Recognition Toolbox (STPRtool)</i> , available at http://cmp.felk.cvut.cz/cmp/software/stprtool/ . | 496 |
| 19.15 | Confusion matrix for Problem 19.1. | 497 |
| 20.1 | Scanning raster. Redrawn from [LI99]. | 502 |
| 20.2 | Scan and retrace: (a) progressive scan (dashed lines indicate horizontal retrace); (b) interlaced scan (solid and dashed lines represent even and odd fields, respectively). Adapted and redrawn from [WOZ02]. | 503 |
| 20.3 | Aspect ratios of SDTV, HDTV, and film. Redrawn from [Poy03]. | 505 |
| 20.4 | Gamma correction in video and TV systems: (a) composite video; (b) component video. | 506 |
| 20.5 | Typical interlaced video raster. Redrawn from [WOZ02]. | 508 |
| 20.6 | Fine-grained frequency spectrum of a monochrome analog video signal. Redrawn from [LI99]. | 509 |
| 20.7 | NTSC spectrum, showing how luminance (Y) and chrominance (I and Q) signals are interleaved. Redrawn from [LD04]. | 512 |
| 20.8 | Sampling in the horizontal, vertical, and temporal dimensions. Redrawn from [Poy03]. | 515 |
| 20.9 | Analog-to-digital converters for composite (top) and component (bottom) video. Redrawn from [LI99]. | 517 |
| 20.10 | Location of sampling points in component video signals. Redrawn from [LI99]. | 518 |
| 20.11 | Assignment of quantization levels for component and composite video. Redrawn from [LI99]. | 519 |
| 20.12 | The most common chroma subsampling patterns: 4:4:4, 4:2:2, and 4:2:0. | 520 |

| | | |
|-------|--|-----|
| 20.13 | Rec. 601 format: screen dimensions and active area for the 525/59.94/2:1 (a) and 625/50/2:1 (b) variants. Redrawn from [WOZ02]. | 523 |
| 20.14 | Visual representation of a YUV file. | 534 |
| 21.1 | The deinterlacing process. Fields t and $t + 1$ form one interlaced frame. | 544 |
| 21.2 | A practical method for converting PAL to NTSC formats. | 545 |
| 21.3 | 3:2 pull-down. Redrawn from [Ack01]. | 547 |
| 21.4 | The problem of judder in telecine using 3:2 pull-down. Redrawn from [Wat94b]. | 548 |
| 22.1 | Two frames at different time instants (t_1 and $t_1 + \Delta t$) and the resulting optical flow. Redrawn from [SHB08]. | 562 |
| 22.2 | The aperture problem. | 563 |
| 22.3 | Motion compensation: interpolation axes are aligned with each moving object. Redrawn from [Wat94b]. | 565 |
| 22.4 | Anchor and target frames in forward and backward motion estimation. | 565 |
| 22.5 | Motion estimation methods: (a) global; (b) pixel-based; (c) block-based; (d) object-based. | 566 |
| 22.6 | Exhaustive search block matching algorithm (EBMA). | 568 |
| 22.7 | Block matching with half-pixel accuracy. The MV in this case is (1, 1.5). Redrawn from [WOZ02]. | 569 |
| 22.8 | EBMA example: (a) target frame; (b) anchor frame; (c) motion field overlapped on anchor frame; (d) reconstructed frame. | 570 |
| 22.9 | 2D log search. In this case, the final MV is $(-6, 2)$. Redrawn from [MPG85]. | 571 |
| 22.10 | Three-step search. In this case, the final MV is $(-6, 2)$. Redrawn from [MPG85]. | 572 |
| 22.11 | Hierarchical block matching algorithm (HBMA) using three levels. Redrawn from [WOZ02]. | 572 |
| 22.12 | Object detection and tracking system. | 577 |
| 22.13 | Keeping track of existing and candidate objects. | 578 |
| 22.14 | Updating the coordinates of an existing object. | 578 |
| 22.15 | Screenshot of the object detection and tracking system, showing the bounding box of an object being tracked, its trajectory since it started being tracked, and its properties (on a separate window). | 579 |
| A.1 | Simplified view of the connection from the eye to the brain via the optic nerve. Adapted and redrawn from [Umb05]. | 592 |
| A.2 | The eye: a cross-sectional view. 1, sclera; 2, ciliary body; 3, iris; 4, pupil and anterior chamber filled with aqueous humor; | |

| | | |
|------|--|-----|
| | 5, optical axis; 6, line of sight; 7, cornea; 8, crystalline lens; 9, choroid; 10, optic nerve; 11, optic disk; 12, fovea; 13, retina; 14, vitreous humor. | 593 |
| A.3 | The eye-camera analogy. Adapted and redrawn from [Pal99]. | 593 |
| A.4 | Dark adaptation. Adapted and redrawn from [Pal99]. | 594 |
| A.5 | Distribution of rods and cones in the human retina for the right eye (seen from the bottom). Adapted and redrawn from [Ost35]. | 595 |
| A.6 | EIA 1956 standard test pattern. Courtesy of http://www.bealecorner.com/trv900/respat/ . | 596 |
| A.7 | Angular frequency concept. | 597 |
| A.8 | Sinusoidal gratings commonly used for measures of resolution—based on MATLAB code by Alex Petrov: http://alexpetrov.com/softw/utills/ . | 597 |
| A.9 | Viewing distance for SDTV and HDTV displays. Adapted and redrawn from [Poy03]. | 599 |
| A.10 | Picture (viewing) angles for SDTV and HDTV displays. Adapted and redrawn from [Poy03]. | 600 |
| A.11 | (a) The definition of contrast index; (b) A test image with constant CI results in an output image with falling CI; (c) modulation transfer function: the ratio of output and input CIs. <i>Note:</i> When LF response is unity, CI and MTF are interchangeable. Redrawn from [Wat00]. | 601 |
| A.12 | Range of subjective brightness sensations showing a particular adaptation level. Redrawn from [GW08]. | 602 |
| A.13 | Simultaneous contrast: the center square is perceived as progressively darker as the background becomes brighter (from (a) to (d)) even though it is identical in all four cases. | 602 |
| A.14 | Mach bands. | 603 |
| A.15 | Contrast sensitivity test pattern. | 603 |
| A.16 | Contrast sensitivity function for various retinal illuminance values (expressed in Td). Redrawn from [VNB67]. | 604 |
| A.17 | Temporal frequency response of the HVS. Redrawn from [Kel61]. | 606 |
| A.18 | Spatiotemporal frequency response of the HVS: (a) spatial frequency responses for different temporal frequencies (in cpd); (b) temporal frequency responses for different spatial (angular) frequencies (in Hz). Redrawn from [Rob66]. | 607 |
| A.19 | Temporal frequency as a function of eye movements. Redrawn from [Wat00]. | 608 |
| B.1 | System control diagram. | 615 |
| B.2 | Variable stack. | 616 |

LIST OF TABLES

| | | |
|------|--|-----|
| 3.1 | MATLAB Data Classes | 37 |
| 3.2 | MATLAB Array and Matrix Arithmetic Operators | 40 |
| 3.3 | Examples of MATLAB Specialized Matrix Operations | 41 |
| 3.4 | Specialized Arithmetic Functions Supported by the IPT | 41 |
| 3.5 | Relational Operators | 41 |
| 3.6 | Logical Operators | 42 |
| 3.7 | Logical Functions | 42 |
| 3.8 | Selected Built-In Variables and Constants | 42 |
| 4.1 | IPT Functions to Perform Image Data Class Conversion | 65 |
| 4.2 | IPT Functions to Perform Image Data Class Conversion | 68 |
| 7.1 | Summary of Transformation Coefficients for Selected Affine Transformations | 128 |
| 8.1 | Examples of Linear Point Transformations (Images and Curves in Figure 8.4) | 154 |
| 9.1 | Example of a Histogram | 172 |
| 9.2 | Equalized Histogram: Values | 178 |
| 9.3 | Desired Histogram | 182 |
| 9.4 | Direct Histogram Specification: Summary | 183 |
| 9.5 | Resulting Histogram | 184 |
| 10.1 | Examples of Convolution Masks | 209 |

| | | |
|------|---|-----|
| 13.1 | Operations Supported by <i>bwmorph</i> | 317 |
| 16.1 | IPT Functions for CIE <i>XYZ</i> and CIELAB Color Spaces | 395 |
| 16.2 | IPT Functions for <i>ICC</i> Profile Manipulation | 395 |
| 16.3 | <i>R</i> , <i>G</i> , and <i>B</i> Values for Eight Representative Colors Corresponding to the Vertices of the <i>RGB</i> Cube | 397 |
| 16.4 | Color Maps in MATLAB | 404 |
| 17.1 | Objective Quality Measures for Three JPEG Images with Different Quality Factors | 440 |
| 17.2 | Compression Ratio for Five JPEG Images with Different Quality Factors | 444 |
| 17.3 | Objective Quality Measures for Five JPEG Images with Different Quality Factors | 444 |
| 18.1 | RST-Invariant Moments | 456 |
| 18.2 | Properties of Labeled Regions | 457 |
| 18.3 | Statistical Texture Descriptors for the Three Images in Figure 18.16 | 467 |
| 18.4 | Statistical Texture Descriptors for the Three Images in Figure 18.16 | 470 |
| 18.5 | Table for Feature Extraction Results | 471 |
| 20.1 | Parameters of Analog Color TV Systems | 514 |
| 20.2 | Representative Digital Video Formats | 522 |
| 20.3 | ITU-R Recommendation BT.601-5 Parameters | 523 |
| 20.4 | Intermediate Formats | 524 |

FOREWORD

In packing for an office move earlier this year, I was struck by how many of my books (in many cases books sent to me free by their publishers for evaluation and potential use) were knockoffs: books that repackaged material that had been thoroughly consolidated in good textbooks years or even decades ago. Oge Marques' textbook is not a knockoff, even though much its subject matter has been around for years. It is a thoughtful and original compilation of material that is highly relevant today for students of imaging science, imaging technology, image understanding, and, foremost, image processing.

Imaging is my principal field of expertise. My interest in this book is great because imaging and image processing have grown together in recent years. Some forms of imaging—computational imaging is the buzzword that comes first to mind—presume that there will be (must be, in fact) postdetection processing of the raw information collected by the sensor system. Indeed, the sensor may output something that makes little or no sense to the observer in the absence of critical postprocessing operations.

Ultimately, in commercial mass-produced systems, image processing is implemented by specialized hardware. In the research and development stages of an imaging system, however, the processing is almost certain to be implemented using MATLAB[®]. Marques' book addresses this fact by linking directly to MATLAB[®] the many processing operations that are described.

There are of course numerous texts that describe digital image processing operations and algorithms. None, however, emphasizes as this one does the human vision system and the interaction and intercomparison between that system and machine vision systems.

The book contains a wealth of practical material and an invaluable up-to-date list of references, including journals, periodicals, and web sites. I would hope that the

book sees subsequent editions, with correspondingly updated lists. Also invaluable to the teacher is the inclusion, now characteristic of many contemporary textbooks, of concise chapter summaries that address the question “What have we learned?” Tutorials punctuate the text, taking the students through important material in an active-learning process.

I write this foreword not just because I think this book addresses its intended image processing audience well but also because I see it influencing the thinking of my own students, students interested in imaging systems from a physics and technological standpoint but who must understand the relationship between imaging systems and image processing systems.

WILLIAM T. RHODES

William T. Rhodes is Emeritus Professor of Electrical & Computer Engineering at Georgia Institute of Technology. In 2005 he joined the Electrical Engineering faculty at Florida Atlantic University and became Associate Director of that university's Imaging Technology Center. A Fellow of the Optical Society of America and of the SPIE, he is editor-in-chief of the Springer Series in Optical Sciences and editor-in-chief of the online journal *SPIE Reviews*.

PREFACE

The prospect of using computers to emulate some of the attributes of the human visual system has attracted the interest of scientists, engineers, and mathematicians for more than 30 years, making the field of image processing one of the fastest-growing branches of applied computer science research. During the past 15 years, the fields of image and video processing have experienced tremendous growth and become more popular and accessible. This growth has been driven by several factors: widely available and relatively inexpensive hardware; a variety of software tools for image and video editing, manipulation, and processing; popularization of the Web and its strong emphasis on visual information, a true revolution in photography that has rendered film-based cameras all but obsolete; advances in the movie industry; and groundbreaking changes on the way we watch, record, and share TV programs and video clips.

APPROACH

This book provides a practical introduction to the most important topics in image and video processing, using MATLAB[®] (and its Image Processing Toolbox) as a tool to demonstrate relevant techniques and algorithms. The word *Practical* in its title is not meant to suggest a coverage of all the latest consumer electronics products in these fields; this knowledge would be superficial at best and would be obsolete shortly after (or even before!) the publication of the book. The word *Practical* should rather be interpreted in the sense of “enabling the reader/student to develop practical projects, that is, working prototypes, using the knowledge gained from the book.” It also has other implications, such as the adoption of a “just enough math” philosophy,

which favors the computational, algorithmic, and conceptual aspects of the techniques described along the book, over excessive mathematical formalism.

As a result, the book should appeal not only to its original target audience when used as a textbook (namely, upper-level undergraduate and early graduate students in Computer Science, Computer Engineering, Electrical Engineering, and related courses) but also to researchers and practitioners who have access to MATLAB[®], solid computing/programming skills, and want to teach themselves the fundamentals of image and video processing.

KEY FEATURES

- This is the first book to combine image processing, video processing, and a practical, MATLAB[®]-oriented approach to experimenting with image and video algorithms and techniques.
- Complete, up-to-date, technically accurate, and practical coverage of essential topics in image and video processing techniques.
- 37 MATLAB[®] tutorials, which can be used either as step-by-step guides to exploring image and video processing techniques using MATLAB[®] on your own or as lab assignments by instructors adopting the textbook.
- More than 330 figures and 30 tables illustrating and summarizing the main techniques and concepts described in the text.
- This book adopts a “just enough math” philosophy. Many students are intimidated by image and video processing books with heavy emphasis on the mathematical aspects of the techniques. This book addresses this issue by offering the minimal mathematical treatment necessary to fully understand a technique without sacrificing the integrity of its explanation.
- The book emphasizes and encourages practical experimentation. After presenting a topic, it invites the readers to play on their own, reinforcing and expanding what they have just learned and venturing into new avenues along the same theme.
- The book has been designed to answer the most basic questions a student/reader is likely to have when first presented with a topic. It builds on my experience teaching image and video processing courses for 20 years, and the insights acquired along the way.
- The book includes many extra features to reinforce the understanding of its topics and allow the reader to learn more about them, such as exercises and programming projects, useful Web sites, and an extensive list of bibliographical references at the end of the chapters.

A TOUR OF THE BOOK

This book has been organized into two parts: *Image Processing* and *Video Processing*.

Part I (Image Processing) starts with an introduction and overview of the field (Chapter 1) that should motivate students to devote time and effort to the material in the remaining chapters. Chapter 2 introduces the fundamental concepts, notation, and terminology associated with image representation and basic image processing operations. Chapters 3 and 4 are devoted to MATLAB[®] and its Image Processing Toolbox, respectively, and establish the beginning of a series of chapters with hands-on activities, presented in the form of step-by-step tutorials at the end of each chapter from this point onward (except Chapter 5). Chapter 5 discusses the factors involved in image acquisition and digitization. Chapter 6 presents arithmetic and logic operations and introduces region of interest (ROI) processing. Chapter 7 covers geometric operations, such as resizing, rotation, cropping, and warping. Chapters 8–10 are devoted to point-based (Chapter 8), histogram-based (Chapter 9), and neighborhood-based (Chapter 10) image enhancement techniques. Chapter 11 extends the reach of image processing operations to the frequency domain and presents the Fourier transform and relevant frequency-domain image filtering techniques. Solutions to the problem of image restoration—particularly in cases of noise and blurring—are discussed in Chapter 12. Chapter 13 presents a detailed coverage of mathematical morphology and its use in image processing. Chapter 14 is devoted to edge detection techniques. Chapter 15 covers image segmentation. Chapter 16 transitions from grayscale to color images and presents representative color image processing techniques and algorithms. Image compression and coding, including the most recent and relevant standards, are the subject of Chapter 17. Chapter 18 looks at the problem of feature extraction and representation and leads naturally to Chapter 19 where the resulting feature vectors could be used for classification and recognition purposes.

Part II (Video Processing) starts by presenting the main concepts and terminology associated with analog video signals and systems and digital video formats and standards (Chapter 20). It then proceeds to describe the technically involved problem of standards conversion (Chapter 21). Chapter 22 discusses motion estimation and compensation techniques, shows how video sequences can be filtered, and concludes with an example of a simple solution to the problem of object detection and tracking in video sequences using MATLAB[®].

The book contains two appendices. Appendix A presents selected aspects of the human visual system that bear implications in the design of image and video processing systems. Appendix B provides a tutorial on how to develop graphical user interfaces (GUIs) in MATLAB[®].

NOTES TO INSTRUCTORS

This book can be used for upper-level undergraduate or introductory graduate courses in image and video processing, for one or two semesters. Most of the material included in this book has been extensively tested in many such courses during the past 20 years. The following is a summary of recommendations for instructors adopting this textbook.

Part I is organized around a typical machine vision system, from image acquisition to pattern classification. All chapters (except Chapters 16 and 17) in Part I follow a natural logic sequence, which covers all the steps involved in acquiring images, preprocessing them to remove imperfections or improve their properties, segmenting them into objects of interest, extracting objects' features, and classifying the objects into categories. The goal of Chapter 1 is to provide breadth, perspective, early examples of what can be achieved with image processing algorithms, and a systemic view of what constitutes a machine vision system. Some instructors may want to combine this information with the material from Chapter 2 as they introduce the topic early in their courses.

The material from Chapters 3 and 4 has been carefully selected to make the book self-contained, providing students all the MATLAB® and Image Processing Toolbox information they might need for the corresponding tutorials. Readers will likely keep these two chapters for future reference should they ever require MATLAB®-related help later in the course. Instructors with limited lecture time may choose to cover both chapters briefly, assign the corresponding tutorials, and monitor students' progress as they work on the tutorials and answer the associated questions.

Chapter 5 briefly introduces the topic of image sensing and acquisition. Its main goal is to equip the reader with information on the steps needed to convert a three-dimensional (3D) real-world scene into a two-dimensional (2D) digitized version of it. Instructors teaching courses with a strong emphasis on image capture and acquisition hardware may want to supplement this material with detailed references, for example, on sensors that operate outside the visible spectrum, stereo-mounted cameras, camera calibration, and many other topics.

Chapters 6–10 are straightforward and cover essential topics in any image processing class. They also provide room for many interesting discussions, lab assignments, and small projects.

Chapter 11 may be a bit challenging to some students, due to the mathematical formalism associated with the Fourier transform. Instructors may find the interactive MATLAB® frequency-domain demo (`fd_demo`) introduced in that chapter a valuable tool to develop students' confidence on their understanding of the basic concepts of frequency-domain filtering techniques. Chapter 12 builds on the knowledge from Chapters 10 and 11, with focus on noise reduction and deblurring techniques. Some instructors may prefer to tone down the discussion of noise models (Sections 12.1 and 12.2) and present the techniques described in Sections 12.3–12.5 earlier on as applications of spatial-domain and frequency-domain filtering techniques.

Chapter 13 is self-contained, which gives instructors the flexibility to adjust their level of coverage—from skipping it altogether, to covering it in detail—without major impact on the other topics in their courses.

Chapters 14 and 15 provide introductory coverage of two essential topics in any image processing course. Instructors who want to present some of these contents earlier in the course or in a different sequence should be able to easily extract the associated sections and move them to a different point in time.

Chapter 16 comprises information on color image processing and is somehow related to earlier chapters (particularly, those on enhancement, segmentation, and edge

extraction). We have made a conscious decision of keeping color in a separate chapter rather than spreading color image processing throughout the text. We believe that by the time readers reach Chapter 16, they will be able to easily navigate through its contents focusing on the differences between what they learned earlier for grayscale images and later for their color equivalent. Instructors who do not agree with this decision can easily bring sections of Chapter 16 to an earlier point in their courses.

Chapter 17 deals with image compression and coding, very extensive and technically complex topics upon which entire books have been written. Since the focus of the book is on building *practical* image processing and machine vision solutions using MATLAB®, we decided to approach the topic of image coding and compression from a broad perspective (standards in use today, categories of compression techniques and their chief characteristics, etc.) instead of attempting to embed a deeper discussion of these topics that could be potentially distracting and would most likely add little value. From a pragmatic viewpoint, since the reader's goal is to process images using MATLAB® and its rich capabilities for reading and writing images from/to a wide variety of formats (most of which use some type of compression), we focused on how to use these capabilities in a meaningful way. Instructors may want to proceed in different ways, depending on their goals, ranging from expanding the material in Chapter 17 with additional references (if image coding and compression is an important part of their course syllabus) to skipping the chapter altogether (if the course's main goal is to build a machine vision solution to a practical problem, which probably would not require that type of knowledge).

Chapters 18 and 19 are tightly interrelated. They provide the information needed to design and implement two of the most critical stages of image processing and machine vision solutions: feature extraction and pattern classification. Chapter 18 offers a wide array of choices for feature extraction and representation techniques, depending on the type of image and the specific needs of the solution being designed. Instructors may appreciate the fact that Chapter 19 provides all the basic concepts that students may need from the associated fields of pattern recognition, data mining, and information retrieval, without requiring additional references. This is particularly important if the course does not enforce prerequisites in any of these areas. The tutorial at the end of Chapter 19 was created to put the selection, design, and fine-tuning of the algorithms presented in Chapters 18 and 19 under perspective. It is my hope that at this point in the book, students will not only be fluent in MATLAB® and image processing but will also have acquired the ability to look back and reflect critically on what works, what does not, and why.

Part II is organized in three chapters, which can be used in the later part of a one- or two-semester course that combines image and video processing or at the early stages of course devoted exclusively to video processing. In the latter case, the instructor may want to supplement the material in Part II with additional references (e.g., scholarly papers in video processing and related topics in the case of graduate-level courses).

Chapter 20 covers a very broad range of topics, from basic analog video concepts to digital video standards and codecs. It offers room for expansion in multiple directions, from a deeper study of TV broadcasting systems to a more detailed analysis of contemporary video compression schemes and standards. Chapter 21 covers

the topic of standards conversion and discusses the most popular techniques used to accomplish it. Chapter 22 expands the discussion to include motion estimation and compensation, as well as (interframe and intraframe) video filtering techniques. It concludes with a practical project implemented in MATLAB[®] by one of my former students: an object detection and tracking system in video sequences with fixed camera and moving, complex, background. The goal of including this case study is to conclude the discussion of Part II (and the book) reminding the reader that at this point they should be knowledgeable enough to attempt similar projects (which instructors may assign as end-of-course projects).

The material in Appendix A is very relevant to image and video processing systems because it explains the relationship between properties of the human visual system and their impact on design decisions involved in building such systems. Some instructors may choose to present (part of) it earlier in their courses.

Appendix B is a practical guide to the development of GUIs for MATLAB[®] applications. It should empower students to develop visually attractive, interactive, and functional interfaces to their MATLAB[®] projects.

A note about MATLAB[®] and the tutorials at the end of chapters. Having used MATLAB[®] (and its Image Processing Toolbox) for more than a decade, I wholeheartedly agree with Rudra Pratap [Pra02] who wrote, “MATLAB[®]’s ease of use is its main feature.” MATLAB[®] has a shallow learning curve, which allows the user to engage in an interactive learning style that accommodates the right degree of challenge needed to raise the user’s skills by a certain amount, and so on, in a staircase-like progression. The MATLAB[®] tutorials included in this book have been conceived under this philosophy.

Web site

The book’s companion web site (<http://www.ogemarques.com>) contains many supplementary materials for students and instructors: MATLAB[®] code for all tutorials in the book, MATLAB[®] code for selected figures, test images and video sequences, supplementary problems, tutorials, and projects (that could not make it to the printed version), and an ever-growing and frequently maintained list of useful web sites—including (links to) image processing conferences, software, hardware, research groups, test image databases, and much more.

OGE MARQUES

Boca Raton, FL

ACKNOWLEDGMENTS

I am deeply grateful to many people who have collaborated—directly or indirectly—on this project. This book would not have been possible without their help.

I want to thank many inspiring professors, supervisors, and colleagues who have guided my steps in the fields of image and video processing and related areas, particularly Maria G. Te Vaarwerk, Wim Hoeks, Bart de Greef, Eric Persoon, John Bernsen, Borko Furht, and Bob Cooper.

A very special thank you to my friend and colleague Hugo Vieira Neto, who has been a great supporter of this project from its early planning stages to its completion.

I am deeply indebted to Gustavo Benvenuti Borba for his excellent work in the creation of most of the figures in this book, his insightful reviews and comments, and continuous encouragement, investing many hours of his time in return for not much more than these few lines of thanks.

Many thanks to Liam M. Mayron for his encouragement, support, and expert help throughout all the steps of this project.

This book could not have been produced without the invaluable contributions of Jeremy Jacob, who wrote, revised, and documented most of the MATLAB® code associated with the book and its tutorials and also contributed the contents for Appendix B.

Special thanks to the MathWorks Book Program (Courtney Esposito, Naomi Fernandes, and Meg Vulliez) for their support over the years.

Several friends and colleagues reviewed draft versions of selected portions of the text: Liam M. Mayron, Hugo Vieira Neto, Mathias Lux, Gustavo Benvenuti Borba, Pierre Baillargeon, Humberto Remigio Gamba, Vladimir Nedovic, Pavani Chilamakuri, and Joel Gibson. I would like to thank them for their careful reviews and insightful comments and suggestions. I have done my best to correct the mistakes

they pointed out and improve the contents of the book according to their suggestions. If any error remains, it is entirely my responsibility, not theirs. If you should find any errors, please e-mail me at omarques@ieee.org, and I will correct them in future printings of this book.

My biggest thanks to my publisher George J. Telecki and his wonderful staff at John Wiley & Sons, Inc. who have patiently worked with me throughout the lifetime of this project: Lucy Hitz, Rachel Witmer, and Melissa Valentine. Their kindness and professionalism have made the creation of this book a very enjoyable process.

Thanks to Amy Hendrickson (TeXnology Inc.) for her expert help with L^AT_EX issues.

I am also indebted to Roger Dalal who designed the inspiring cover art for the book.

Last but certainly not least, I want to thank my family for their unfailing love, patience, and understanding.

Oge Marques