

CHRIS SOLOMON | TOBY BRECKON

FUNDAMENTALS OF DIGITAL IMAGE PROCESSING

A PRACTICAL APPROACH WITH
EXAMPLES IN MATLAB



WILEY-BLACKWELL

Fundamentals of Digital Image Processing

A Practical Approach
with Examples in Matlab

Chris Solomon

School of Physical Sciences, University of Kent, Canterbury, UK

Toby Breckon

School of Engineering, Cranfield University, Bedfordshire, UK



 **WILEY-BLACKWELL**

A John Wiley & Sons, Ltd., Publication

This edition first published 2011, © 2011 by John Wiley & Sons, Ltd

Wiley-Blackwell is an imprint of John Wiley & Sons, formed by the merger of Wiley's global Scientific, Technical and Medical business with Blackwell Publishing.

Registered office: John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Offices:

9600 Garsington Road, Oxford, OX4 2DQ, UK

111 River Street, Hoboken, NJ 07030-5774, USA

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com/wiley-blackwell

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. 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 or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

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

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book. This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

Library of Congress Cataloguing-in-Publication Data

Solomon, Chris and Breckon, Toby

Fundamentals of digital image processing : a practical approach with examples in Matlab / Chris Solomon and Toby Breckon

p. cm.

Includes index.

Summary: "Fundamentals of Digital Image Processing is an introductory text on the science of image processing and employs the Matlab programming language to illustrate some of the elementary, key concepts in modern image processing and pattern recognition drawing on specific examples from within science, medicine and electronics"— Provided by publisher.

ISBN 978-0-470-84472-4 (hardback) – ISBN 978-0-470-84473-1 (pbk.)

1. Image processing—Digital techniques. 2. Matlab. I. Breckon, Toby. II. Title.

TA1637.S65154 2010

621.36'7—dc22

2010025730

This book is published in the following electronic formats: eBook 9780470689783; Wiley Online Library 9780470689776

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

Set in 10/12.5 pt Minion by Thomson Digital, Noida, India

Printed and bound in Singapore by Markono Print Media Pte Ltd

Fundamentals of Digital Image Processing

Preface

Scope of this book

This is an introductory text on the science (*and art*) of image processing. The book also employs the Matlab programming language and toolboxes to illuminate and consolidate some of the elementary but key concepts in modern image processing and pattern recognition.

The authors are firm believers in the old adage, “**Hear** and forget. . . , **See** and remember. . . , **Do** and know”. For most of us, it is through good examples and gently guided experimentation that we really learn. Accordingly, the book has a large number of carefully chosen examples, graded exercises and computer experiments designed to help the reader get a real grasp of the material. All the program code (.m files) used in the book, corresponding to the examples and exercises, are made available to the reader/course instructor and may be downloaded from the book’s dedicated web site – www.fundipbook.com.

Who is this book for?

For undergraduate and graduate students in the technical disciplines, for technical professionals seeking a direct introduction to the field of image processing and for instructors looking to provide a hands-on, structured course. This book intentionally starts with simple material but we also hope that relative experts will nonetheless find some interesting and useful material in the latter parts.

Aims

What then are the specific aims of this book ? Two of the principal aims are –

- To introduce the reader to some of the key concepts and techniques of modern image processing.
- To provide a framework within which these concepts and techniques can be understood by a series of examples, exercises and computer experiments.

These are, perhaps, aims which one might reasonably expect from *any* book on a technical subject. However, we have one further aim namely to provide the reader with the fastest, most direct route to acquiring a real hands-on understanding of image processing. We hope this book will give you a real fast-start in the field.

Assumptions

We make no assumptions about the reader's mathematical background beyond that expected at the undergraduate level in the technical sciences – ie reasonable competence in calculus, matrix algebra and basic statistics.

Why write this book?

There are already a number of excellent and comprehensive texts on image processing and pattern recognition and we refer the interested reader to a number in the appendices of this book. There are also some exhaustive and well-written books on the Matlab language. What the authors felt was lacking was *an image processing book which combines a simple exposition of principles with a means to quickly test, verify and experiment with them in an instructive and interactive way.*

In our experience, formed over a number of years, Matlab and the associated image processing toolbox are extremely well-suited to help achieve this aim. It is simple but powerful and its key feature in this context is that it enables one to *concentrate on the image processing concepts and techniques* (i.e. the real business at hand) while keeping concerns about programming syntax and data management to a minimum.

What is Matlab?

Matlab is a programming language with an associated set of specialist software toolboxes. It is an industry standard in scientific computing and used worldwide in the scientific, technical, industrial and educational sectors. Matlab is a commercial product and information on licences and their cost can be obtained direct by enquiry at the web-site www.mathworks.com. Many Universities all over the world provide site licenses for their students.

What knowledge of Matlab is required for this book?

Matlab is very much part of this book and we use it extensively to demonstrate how certain processing tasks and approaches can be quickly implemented and tried out in practice. Throughout the book, we offer comments on the Matlab language and the best way to achieve certain image processing tasks in that language. Thus the learning of concepts in image processing and their implementation within Matlab go hand-in-hand in this text.

Is the book any use then if I don't know Matlab?

Yes. This is fundamentally a book about image processing which aims to make the subject accessible and practical. It is not a book about the Matlab programming language. Although some prior knowledge of Matlab is an advantage and will make the practical implementation easier, we have endeavoured to maintain a self-contained discussion of the concepts which will stand up apart from the computer-based material.

If you have not encountered Matlab before and you wish to get the maximum from this book, please refer to the Matlab and Image Processing primer on the book website (<http://www.fundipbook.com>). This aims to give you the essentials on Matlab with a strong emphasis on the basic properties and manipulation of images.

Thus, you do not have to be knowledgeable in Matlab to profit from this book.

Practical issues

To carry out the vast majority of the examples and exercises in the book, the reader will need access to a current licence for *Matlab* and the *Image Processing Toolbox* only.

Features of this book and future support

This book is accompanied by a dedicated website (<http://www.fundipbook.com>). The site is intended to act as a point of contact with the authors, as a repository for the code examples (Matlab .m files) used in the book and to host additional supporting materials for the reader and instructor.

About the authors

Chris Solomon gained a B.Sc in theoretical physics from Durham University and a Ph.D in Medical imaging from the Royal Marsden Hospital, University of London. Since 1994, he has been on the Faculty at the School of Physical Sciences where he is currently a Reader in Forensic Imaging. He has broad research interests focussing on evolutionary and genetic algorithms, image processing and statistical learning methods with a special interest in the human face. Chris is also Technical Director of Visionmetric Ltd, a company he founded in 1999 and which is now the UK's leading provider of facial composite software and training in facial identification to police forces. He has received a number of UK and European awards for technology innovation and commercialisation of academic research.

Toby Breckon holds a Ph.D in Informatics and B.Sc in Artificial Intelligence and Computer Science from the University of Edinburgh. Since 2006 he has been a lecturer in image processing and computer vision in the School of Engineering at Cranfield University. His key research interests in this domain relate to 3D sensing, real-time vision, sensor fusion, visual surveillance and robotic deployment. He is additionally a visiting member of faculty at *Ecole Supérieure des Technologies Industrielles Avancées* (France) and has held visiting faculty positions in China and Japan. In 2008 he led the development of

image-based automatic threat detection for the winning Stellar Team system in the UK MoD Grand Challenge. He is a Chartered Engineer (CEng) and an Accredited Imaging Scientist (AIS) as an Associate of the Royal Photographic Society (ARPS).

Thanks

The authors would like to thank the following people and organisations for their various support and assistance in the production of this book: the authors families and friends for their support and (frequent) understanding, Professor Chris Dainty (National University of Ireland), Dr. Stuart Gibson (University of Kent), Dr. Timothy Lukins (University of Edinburgh), The University of Kent, Cranfield University, VisionMetric Ltd and Wiley-Blackwell Publishers.

For further examples and exercises see <http://www.fundipbook.com>

Using the book website

There is an associated website which forms a vital supplement to this text. It is:

www.fundipbook.com

The material on the site is mostly organised by chapter number and this contains –

EXERCISES: intended to consolidate and highlight concepts discussed in the text. Some of these exercises are numerical/conceptual, others are based on Matlab.

SUPPLEMENTARY MATERIAL: Proofs, derivations and other supplementary material referred to in the text are available from this section and are intended to consolidate, highlight and extend concepts discussed in the text.

Matlab CODE: The Matlab code to all the examples in the book as well as the code used to create many of the figures are available in the Matlab code section.

IMAGE DATABASE: The Matlab software allows direct access and use to a number of images as an integral part of the software. Many of these are used in the examples presented in the text.

We also offer a modest repository of images captured and compiled by the authors which the reader may freely download and work with. Please note that some of the example Matlab code contained on the website and presented in the text makes use of these images. **You will therefore need to download these images to run some of the Matlab code shown.**

We strongly encourage you to make use of the website and the materials on it. It is a vital link to making your exploration of the subject both practical and more in-depth. Used properly, it will help you to get much more from this book.

Contents

Preface	xi
Using the book website	xv
1 Representation	1
1.1 What is an image?	1
1.1.1 Image layout	1
1.1.2 Image colour	2
1.2 Resolution and quantization	3
1.2.1 Bit-plane splicing	4
1.3 Image formats	5
1.3.1 Image data types	6
1.3.2 Image compression	7
1.4 Colour spaces	9
1.4.1 RGB	10
1.4.1.1 RGB to grey-scale image conversion	11
1.4.2 Perceptual colour space	12
1.5 Images in Matlab	14
1.5.1 Reading, writing and querying images	14
1.5.2 Basic display of images	15
1.5.3 Accessing pixel values	16
1.5.4 Converting image types	17
Exercises	18
2 Formation	21
2.1 How is an image formed?	21
2.2 The mathematics of image formation	22
2.2.1 Introduction	22
2.2.2 Linear imaging systems	23
2.2.3 Linear superposition integral	24
2.2.4 The Dirac delta or impulse function	25
2.2.5 The point-spread function	28

2.2.6	Linear shift-invariant systems and the convolution integral	29
2.2.7	Convolution: its importance and meaning	30
2.2.8	Multiple convolution: N imaging elements in a linear shift-invariant system	34
2.2.9	Digital convolution	34
2.3	The engineering of image formation	37
2.3.1	The camera	38
2.3.2	The digitization process	40
2.3.2.1	Quantization	40
2.3.2.2	Digitization hardware	42
2.3.2.3	Resolution versus performance	43
2.3.3	Noise	44
	Exercises	46
3	Pixels	49
3.1	What is a pixel?	49
3.2	Operations upon pixels	50
3.2.1	Arithmetic operations on images	51
3.2.1.1	Image addition and subtraction	51
3.2.1.2	Multiplication and division	53
3.2.2	Logical operations on images	54
3.2.3	Thresholding	55
3.3	Point-based operations on images	57
3.3.1	Logarithmic transform	57
3.3.2	Exponential transform	59
3.3.3	Power-law (gamma) transform	61
3.3.3.1	Application: gamma correction	62
3.4	Pixel distributions: histograms	63
3.4.1	Histograms for threshold selection	65
3.4.2	Adaptive thresholding	66
3.4.3	Contrast stretching	67
3.4.4	Histogram equalization	69
3.4.4.1	Histogram equalization theory	69
3.4.4.2	Histogram equalization theory: discrete case	70
3.4.4.3	Histogram equalization in practice	71
3.4.5	Histogram matching	73
3.4.5.1	Histogram-matching theory	73
3.4.5.2	Histogram-matching theory: discrete case	74
3.4.5.3	Histogram matching in practice	75
3.4.6	Adaptive histogram equalization	76
3.4.7	Histogram operations on colour images	79
	Exercises	81

4	Enhancement	85
4.1	Why perform enhancement?	85
4.1.1	Enhancement via image filtering	85
4.2	Pixel neighbourhoods	86
4.3	Filter kernels and the mechanics of linear filtering	87
4.3.1	Nonlinear spatial filtering	90
4.4	Filtering for noise removal	90
4.4.1	Mean filtering	91
4.4.2	Median filtering	92
4.4.3	Rank filtering	94
4.4.4	Gaussian filtering	95
4.5	Filtering for edge detection	97
4.5.1	Derivative filters for discontinuities	97
4.5.2	First-order edge detection	99
4.5.2.1	Linearly separable filtering	101
4.5.3	Second-order edge detection	102
4.5.3.1	Laplacian edge detection	102
4.5.3.2	Laplacian of Gaussian	103
4.5.3.3	Zero-crossing detector	104
4.6	Edge enhancement	105
4.6.1	Laplacian edge sharpening	105
4.6.2	The unsharp mask filter	107
	Exercises	109
5	Fourier transforms and frequency-domain processing	113
5.1	Frequency space: a friendly introduction	113
5.2	Frequency space: the fundamental idea	114
5.2.1	The Fourier series	115
5.3	Calculation of the Fourier spectrum	118
5.4	Complex Fourier series	118
5.5	The 1-D Fourier transform	119
5.6	The inverse Fourier transform and reciprocity	121
5.7	The 2-D Fourier transform	123
5.8	Understanding the Fourier transform: frequency-space filtering	126
5.9	Linear systems and Fourier transforms	129
5.10	The convolution theorem	129
5.11	The optical transfer function	131
5.12	Digital Fourier transforms: the discrete fast Fourier transform	134
5.13	Sampled data: the discrete Fourier transform	135
5.14	The centred discrete Fourier transform	136
6	Image restoration	141
6.1	Imaging models	141
6.2	Nature of the point-spread function and noise	142

6.3	Restoration by the inverse Fourier filter	143
6.4	The Wiener–Helstrom Filter	146
6.5	Origin of the Wiener–Helstrom filter	147
6.6	Acceptable solutions to the imaging equation	151
6.7	Constrained deconvolution	151
6.8	Estimating an unknown point-spread function or optical transfer function	154
6.9	Blind deconvolution	156
6.10	Iterative deconvolution and the Lucy–Richardson algorithm	158
6.11	Matrix formulation of image restoration	161
6.12	The standard least-squares solution	162
6.13	Constrained least-squares restoration	163
6.14	Stochastic input distributions and Bayesian estimators	165
6.15	The generalized Gauss–Markov estimator	165
7	Geometry	169
7.1	The description of shape	169
7.2	Shape-preserving transformations	170
7.3	Shape transformation and homogeneous coordinates	171
7.4	The general 2-D affine transformation	173
7.5	Affine transformation in homogeneous coordinates	174
7.6	The Procrustes transformation	175
7.7	Procrustes alignment	176
7.8	The projective transform	180
7.9	Nonlinear transformations	184
7.10	Warping: the spatial transformation of an image	186
7.11	Overdetermined spatial transformations	189
7.12	The piecewise warp	191
7.13	The piecewise affine warp	191
7.14	Warping: forward and reverse mapping	194
8	Morphological processing	197
8.1	Introduction	197
8.2	Binary images: foreground, background and connectedness	197
8.3	Structuring elements and neighbourhoods	198
8.4	Dilation and erosion	200
8.5	Dilation, erosion and structuring elements within Matlab	201
8.6	Structuring element decomposition and Matlab	202
8.7	Effects and uses of erosion and dilation	204
8.7.1	Application of erosion to particle sizing	207
8.8	Morphological opening and closing	209
8.8.1	The rolling-ball analogy	210
8.9	Boundary extraction	212
8.10	Extracting connected components	213

8.11	Region filling	215
8.12	The hit-or-miss transformation	216
8.12.1	Generalization of hit-or-miss	219
8.13	Relaxing constraints in hit-or-miss: 'don't care' pixels	220
8.13.1	Morphological thinning	222
8.14	Skeletonization	222
8.15	Opening by reconstruction	224
8.16	Grey-scale erosion and dilation	227
8.17	Grey-scale structuring elements: general case	227
8.18	Grey-scale erosion and dilation with flat structuring elements	228
8.19	Grey-scale opening and closing	229
8.20	The top-hat transformation	230
8.21	Summary	231
	Exercises	233
9	Features	235
9.1	Landmarks and shape vectors	235
9.2	Single-parameter shape descriptors	237
9.3	Signatures and the radial Fourier expansion	239
9.4	Statistical moments as region descriptors	243
9.5	Texture features based on statistical measures	246
9.6	Principal component analysis	247
9.7	Principal component analysis: an illustrative example	247
9.8	Theory of principal component analysis: version 1	250
9.9	Theory of principal component analysis: version 2	251
9.10	Principal axes and principal components	253
9.11	Summary of properties of principal component analysis	253
9.12	Dimensionality reduction: the purpose of principal component analysis	256
9.13	Principal components analysis on an ensemble of digital images	257
9.14	Representation of out-of-sample examples using principal component analysis	257
9.15	Key example: eigenfaces and the human face	259
10	Image Segmentation	263
10.1	Image segmentation	263
10.2	Use of image properties and features in segmentation	263
10.3	Intensity thresholding	265
10.3.1	Problems with global thresholding	266
10.4	Region growing and region splitting	267
10.5	Split-and-merge algorithm	267
10.6	The challenge of edge detection	270
10.7	The Laplacian of Gaussian and difference of Gaussians filters	270
10.8	The Canny edge detector	271

10.9	Interest operators	274
10.10	Watershed segmentation	279
10.11	Segmentation functions	280
10.12	Image segmentation with Markov random fields	286
10.12.1	Parameter estimation	288
10.12.2	Neighbourhood weighting parameter θ_n	289
10.12.3	Minimizing $U(x y)$: the iterated conditional modes algorithm	290
11	Classification	291
11.1	The purpose of automated classification	291
11.2	Supervised and unsupervised classification	292
11.3	Classification: a simple example	292
11.4	Design of classification systems	294
11.5	Simple classifiers: prototypes and minimum distance criteria	296
11.6	Linear discriminant functions	297
11.7	Linear discriminant functions in N dimensions	301
11.8	Extension of the minimum distance classifier and the Mahalanobis distance	302
11.9	Bayesian classification: definitions	303
11.10	The Bayes decision rule	304
11.11	The multivariate normal density	306
11.12	Bayesian classifiers for multivariate normal distributions	307
11.12.1	The Fisher linear discriminant	310
11.12.2	Risk and cost functions	311
11.13	Ensemble classifiers	312
11.13.1	Combining weak classifiers: the AdaBoost method	313
11.14	Unsupervised learning: k-means clustering	313
	Further reading	317
	Index	319

1

Representation

In this chapter we discuss the representation of images, covering basic notation and information about images together with a discussion of standard image types and image formats. We end with a practical section, introducing Matlab's facilities for reading, writing, querying, converting and displaying images of different image types and formats.

1.1 What is an image?

A digital image can be considered as a discrete representation of data possessing both spatial (layout) and intensity (colour) information. As we shall see in Chapter 5, we can also consider treating an image as a multidimensional signal.

1.1.1 Image layout

The two-dimensional (2-D) discrete, digital image $I(m, n)$ represents the response of some sensor (or simply a value of some interest) at a series of fixed positions ($m = 1, 2, \dots, M$; $n = 1, 2, \dots, N$) in 2-D Cartesian coordinates and is derived from the 2-D continuous spatial signal $I(x, y)$ through a sampling process frequently referred to as discretization. Discretization occurs naturally with certain types of imaging sensor (such as CCD cameras) and basically effects a local averaging of the continuous signal over some small (typically square) region in the receiving domain.

The indices m and n respectively designate the rows and columns of the image. The individual picture elements or pixels of the image are thus referred to by their 2-D (m, n) index. Following the Matlab[®] convention, $I(m, n)$ denotes the response of the pixel located at the m th row and n th column starting from a top-left image origin (see Figure 1.1). In other imaging systems, a column–row convention may be used and the image origin in use may also vary.

Although the images we consider in this book will be discrete, it is often theoretically convenient to treat an image as a continuous spatial signal: $I(x, y)$. In particular, this sometimes allows us to make more natural use of the powerful techniques of integral and differential calculus to understand properties of images and to effectively manipulate and

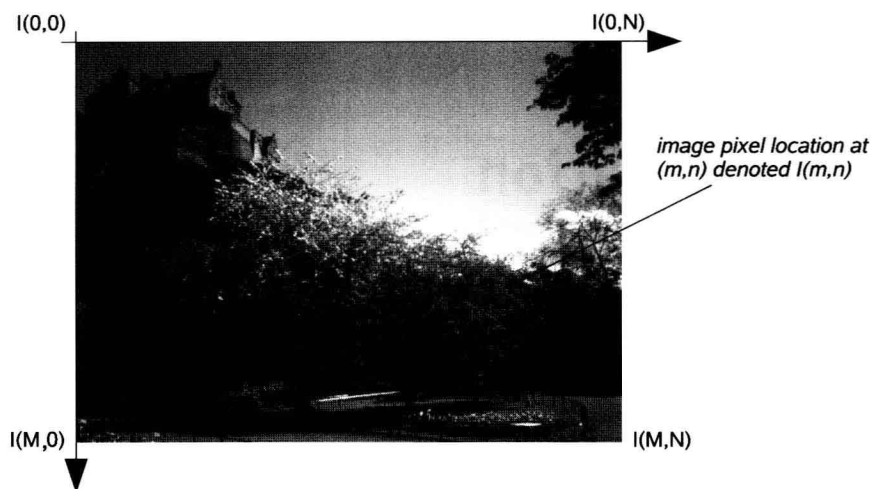


Figure 1.1 The 2-D Cartesian coordinate space of an $M \times N$ digital image

process them. Mathematical analysis of discrete images generally leads to a linear algebraic formulation which is better in some instances.

The individual pixel values in most images do actually correspond to some physical response in real 2-D space (e.g. the optical intensity received at the image plane of a camera or the ultrasound intensity at a transceiver). However, we are also free to consider images in abstract spaces where the coordinates correspond to something other than physical space and we may also extend the notion of an image to three or more dimensions. For example, medical imaging applications sometimes consider full three-dimensional (3-D) reconstruction of internal organs and a time sequence of such images (such as a beating heart) can be treated (if we wish) as a single four-dimensional (4-D) image in which three coordinates are spatial and the other corresponds to time. When we consider 3-D imaging we are often discussing spatial volumes represented by the image. In this instance, such 3-D pixels are denoted as voxels (volumetric pixels) representing the smallest spatial location in the 3-D volume as opposed to the conventional 2-D image.

Throughout this book we will usually consider 2-D digital images, but much of our discussion will be relevant to images in higher dimensions.

1.1.2 Image colour

An image contains one or more colour channels that define the intensity or colour at a particular pixel location $I(m, n)$.

In the simplest case, each pixel location only contains a single numerical value representing the signal level at that point in the image. The conversion from this set of numbers to an actual (displayed) image is achieved through a colour map. A colour map assigns a specific shade of colour to each numerical level in the image to give a visual representation of the data. The most common colour map is the greyscale, which assigns all shades of grey from black (zero) to white (maximum) according to the signal level. The