Wilhelm Burger · Mark J. Burge

Digital Image Processing

An algorithmic introduction using Java

Second Edition

2016

Springer
Berlin Heidelberg New York
Hong Kong London
Milano Paris Tokyo

Preface

This book provides a modern, self-contained introduction to digital image processing. We designed the book to be used both by learners desiring a firm foundation on which to build as well as practitioners in search of detailed analysis and transparent implementations of the most important techniques. This is the second English edition of the original German-language book, which has been widely used by:

- Scientists and engineers who use image processing as a tool and wish to develop a deeper understanding and create custom solutions to imaging problems in their field.
- IT professionals wanting a self-study course featuring easily adaptable code and completely worked out examples, enabling them to be productive right away.
- Faculty and students desiring an example-rich introductory textbook suitable for an advanced undergraduate or graduate level course that features exercises, projects, and examples that have been honed during our years of experience teaching this material.

While we concentrate on practical applications and concrete implementations, we do so without glossing over the important formal details and mathematics necessary for a deeper understanding of the algorithms. In preparing this text, we started from the premise that simply creating a recipe book of imaging solutions would not provide the deeper understanding needed to apply these techniques to novel problems, so instead our solutions are developed stepwise from three different perspectives: in mathematical form, as abstract pseudocode algorithms, and as complete Java programs. We use a common notation to intertwine all three perspectives—providing multiple, but linked, views of the problem and its solution.

Prerequisites

Instead of presenting digital image processing as a mathematical discipline, or strictly as signal processing, we present it from a practitioner's and programmer's perspective and with a view toward replacing many of the formalisms commonly used in other texts with constructs more readily understandable by our audience. To take full advantage of the *programming* components of this book, a knowledge of basic data structures and object-oriented programming, ideally in Java, is required. We selected Java for a number of reasons: it is the first programming language learned by students in a wide variety of engineering curricula, and professionals with knowledge of a related language, especially C# or C++, will find the programming examples easy to follow and extend.

The software in this book is designed to work with ImageJ, a widely used, programmer-extensible, imaging system developed, maintained, and distributed by the National Institutes of Health (NIH).[1] ImageJ is implemented completely in Java, and therefore runs on all major platforms, and is widely used because its "plugin"-based architecture enables it to be easily extended. While all examples run in ImageJ, they have been specifically designed to be easily ported to other environments and programming languages.

Use in research and development

This book has been especially designed for use as a textbook and as such features exercises and carefully constructed examples that supplement our detailed presentation of the fundamental concepts and techniques. As both practitioners and developers, we know that the details required to successfully understand, apply, and extend classical techniques are often difficult to find, and for this reason we have been very careful to provide the missing details, many gleaned over years of practical application. While this should make the text particularly valuable to those in research and development, it is not designed as a comprehensive, fully-cited scientific research text. On the contrary, we have carefully vetted our citations so that they can be obtained from easily accessible sources. While we have only briefly discussed the fundamentals of, or entirely omitted, topics such as hierarchical methods, wavelets, or eigenimages because of space limitations, other topics have been left out deliberately, including advanced issues such as object recognition, image understanding, and three-dimensional (3D) computer vision. So, while most techniques described in this book could be called "blind and dumb", it is our experience that straightforward, technically clean implementations of these simpler methods are essential to the success of any further domain-specific, or even "intelligent", approaches.

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

Classroom use

Whether it is called signal processing, image processing, or media computation, the manipulation of digital images has been an integral part of most computer science and engineering curricula for many years. Today, with the omnipresence of all-digital work flows, it has become an integral part of the required skill set for professionals in many diverse disciplines.

Today the topic has migrated into the early stages of many curricula, where it is often a key foundation course. This migration uncovered a problem in that many of the texts relied on as standards

[1] http://rsb.info.nih.gov/ij/.

in the older graduate-level courses were not appropriate for beginners. The texts were usually too formal for novices, and at the same time did not provide detailed coverage of many of the most popular methods used in actual practice. The result was that educators had a difficult time selecting a single textbook or even finding a compact collection of literature to recommend to their students. Faced with this dilemma ourselves, we wrote this book in the sincere hope of filling this gap.

The contents of the following chapters can be presented in either a one- or two-semester sequence. Where feasible, we have added supporting material in order to make each chapter as independent as possible, providing the instructor with maximum flexibility when designing the course. Chapters 18–20 offer a complete introduction to the fundamental spectral techniques used in image processing and are essentially independent of the other material in the text. Depending on the goals of the instructor and the curriculum, they can be covered in as much detail as required or completely omitted. The following road map shows a possible partitioning of topics for a two-semester syllabus.

Road Map for a 1/2-Semester Syllabus	Sem. 1	2
1. Digital Images	■	□
2. ImageJ	■	□
3. Histograms and Image Statistics	■	□
4. Point Operations	■	□
5. Filters	■	□
6. Edges and Contours	■	□
7. Corner Detection	□	■
8. The Hough Transform: Finding Simple Curves	□	■
9. Morphological Filters	■	□
10. Regions in Binary Images	■	□
11. Automatic Thresholding	□	■
12. Color Images	■	□
13. Color Quantization	□	■
14. Colorimetric Color Spaces	□	■
15. Filters for Color Images	□	■
16. Edge Detection in Color Images	□	■
17. Edge-Preserving Smoothing Filters	□	■
18. Introduction to Spectral Techniques	□	■
19. The Discrete Fourier Transform in 2D	□	■
20. The Discrete Cosine Transform (DCT)	□	■
21. Geometric Operations	■	□
22. Pixel Interpolation	■	□
23. Image Matching and Registration	■	□
24. Non-Rigid Image Matching	□	■
25. Scale-Invariant Local Features (SIFT)	□	■
26. Fourier Shape Descriptors	□	■

Addendum to the second edition

This second edition is based on our completely revised German third edition and contains both additional material and several new chap-

ters including: *automatic thresholding* (Ch. 11), *filters and edge detection for color images* (Chs. 15 and 16), *edge-preserving smoothing filters* (Ch. 17), *non-rigid image matching* (Ch. 24), and *Fourier shape descriptors* (Ch. 26). Much of this new material is presented for the first time at the level of detail necessary to completely understand and create a working implementation.

The two final chapters on SIFT and Fourier shape descriptors are particularly detailed to demonstrate the actual level of granularity and the special cases which must be considered when actually implementing complex techniques. Some other chapters have been rearranged or split into multiple parts for more clarity and easier use in teaching. The mathematical notation and programming examples were completely revised and almost all illustrations were adapted or created anew for this full-color edition.

For this edition, the *ImageJ Short Reference* and ancillary source code have been relocated from the Appendix and the most recently versions are freely available in electronic form from the book's website. The complete source code, consisting of the common `imagingbook` library, sample ImageJ plugins for each book chapter, and extended documentation are available from the book's SourceForge site.[2]

Online resources

Visit the website for this book

www.imagingbook.com

to download supplementary materials, including the complete Java source code for all examples and the underlying software library, full-size test images, useful references, and other supplements. Comments, questions, and corrections are welcome and may be addressed to

imagingbook@gmail.com

Exercises and solutions

Each chapter of this book contains a set of sample exercises, mainly for supporting instructors to prepare their own assignments. Most of these tasks are easy to solve after studying the corresponding chapter, while some others may require more elaborated reasoning or experimental work. We assume that scholars know best how to select and adapt individual assignments in order to fit the level and interest of their students. This is the main reason why we have abstained from publishing explicit solutions in the past. However, we are happy to answer any personal request if an exercise is unclear or seems to elude a simple solution.

Thank you!

This book would not have been possible without the understanding and support of our families. Our thanks go to Wayne Rasband at NIH for developing ImageJ and for his truly outstanding support of

[2] http://sourceforge.net/projects/imagingbook/.

the community and to all our readers of the previous editions who provided valuable input, suggestions for improvement, and encouragement. The use of open source software for such a project always carries an element of risk, since the long-term acceptance and continuity is difficult to assess. Retrospectively, choosing ImageJ as the software basis for this work was a good decision, and we would consider ourselves happy if our book has indirectly contributed to the success of the ImageJ project itself. Finally, we owe a debt of gratitude to the professionals at Springer, particularly to Wayne Wheeler and Simon Reeves who were responsible for the English edition.

Hagenberg / Washington D.C.
Fall 2015

Contents