

学习OpenCV(影印版)



Learning

OpenCV

*Computer Vision with
the OpenCV Library*

O'REILLY®

東南大學出版社

Gary Bradski & Adrian Kaehler 著

学习 OpenCV (影印版)

Learning OpenCV

江苏工业学院图书馆
藏书章

Gary Bradski & Adrian Kaehler

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

学习 OpenCV: 英文 / (美) 布拉德斯基 (Bradski, G.), (美) 克勒 (Kaehler, A.) 著. —影印本. —南京: 东南大学出版社, 2009.4

书名原文: Learning OpenCV

ISBN 978-7-5641-1629-3

I .学… II .①布… ②克… III .①计算机视觉—英文
②图像处理—C语言—语言设计—英文 IV .TP312 TP391.41

中国版本图书馆 CIP 数据核字 (2009) 第 048352 号

江苏省版权局著作权合同登记

图字: 10-2009-141 号

©2008 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2009. Authorized reprint of the original English edition, 2008 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2008。

英文影印版由东南大学出版社出版 2009。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

学习 OpenCV (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出版人: 江汉

网 址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 36 印张

字 数: 605 千字

版 次: 2009 年 4 月第 1 版

印 次: 2009 年 4 月第 1 次印刷

书 号: ISBN 978-7-5641-1629-3

印 数: 1~2000 册

定 价: 74.00 元 (册)

About the Authors

Dr. Gary Rost Bradski is a consulting professor in the CS department at the Stanford University AI Lab, where he mentors robotics, machine learning, and computer vision research. He is also senior scientist at Willow Garage (<http://www.willowgarage.com>), a recently founded robotics research institute/incubator. He holds a B.S. in EECS from UC Berkeley and a Ph.D. from Boston University. He has 20 years of industrial experience applying machine learning and computer vision, spanning option-trading operations at First Union National Bank, to computer vision at Intel Research, to machine learning in Intel Manufacturing, and several startup companies in between.

Gary started the Open Source Computer Vision Library (OpenCV, <http://sourceforge.net/projects/opencvlibrary>), which is used around the world in research, in government, and commercially; the statistical Machine Learning Library (which comes with OpenCV); and the Probabilistic Network Library (PNL). The vision libraries helped develop a notable part of the commercial Intel Performance Primitives Library (IPP, <http://tinyurl.com/36ua5s>). Gary also organized the vision team for Stanley, the Stanford robot that won the DARPA Grand Challenge autonomous race across the desert for a \$2M team prize, and he helped found the Stanford AI Robotics project at Stanford (<http://www.cs.stanford.edu/group/stair>) working with Professor Andrew Ng. Gary has more than 50 publications and 13 issued patents with 18 pending. He lives in Palo Alto, CA, with his wife and three daughters and bikes road or mountain as much as he can.

Dr. Adrian Kaehler is a senior scientist at Applied Minds Corporation. His current research includes topics in machine learning, statistical modeling, computer vision, and robotics. Adrian received his Ph.D. in Theoretical Physics from Columbia University in 1998. He has since held positions at Intel Corporation and the Stanford University AI Lab and was a member of the winning Stanley race team in the DARPA Grand Challenge. He has a variety of published papers and patents in physics, electrical engineering, computer science, and robotics.

Colophon

The image on the cover of *Learning OpenCV* is a giant, or great, peacock moth (*Saturnia pyri*). Native to Europe, the moth's range includes southern France and Italy, the Iberian Peninsula, and parts of Siberia and northern Africa. It inhabits open landscapes with scattered trees and shrubs and can often be found in parklands, orchards, and vineyards, where it rests under shade trees during the day.

The largest of the European moths, giant peacock moths have a wingspan of up to six inches; their size and nocturnal nature can lead some observers to mistake them for bats. Their wings are gray and grayish-brown with accents of white and yellow. In the center of each wing, giant peacock moths have a large eyespot, a distinctive pattern most commonly associated with the birds they are named for.

The cover image is from Cassell's *Natural History*, Volume 5. The cover font is Adobe-ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSansMonoCondensed.

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。

Preface

This book provides a working guide to the Open Source Computer Vision Library (OpenCV) and also provides a general background to the field of computer vision sufficient to use OpenCV effectively.

Purpose

Computer vision is a rapidly growing field, partly as a result of both cheaper and more capable cameras, partly because of affordable processing power, and partly because vision algorithms are starting to mature. OpenCV itself has played a role in the growth of computer vision by enabling thousands of people to do more productive work in vision. With its focus on real-time vision, OpenCV helps students and professionals efficiently implement projects and jump-start research by providing them with a computer vision and machine learning infrastructure that was previously available only in a few mature research labs. The purpose of this text is to:

- Better document OpenCV—detail what function calling conventions really mean and how to use them correctly.
- Rapidly give the reader an intuitive understanding of how the vision algorithms work.
- Give the reader some sense of what algorithm to use and when to use it.
- Give the reader a boost in implementing computer vision and machine learning algorithms by providing many working coded examples to start from.
- Provide intuitions about how to fix some of the more advanced routines when something goes wrong.

Simply put, this is the text the authors wished we had in school and the coding reference book we wished we had at work.

This book documents a tool kit, OpenCV, that allows the reader to do interesting and fun things rapidly in computer vision. It gives an intuitive understanding as to how the algorithms work, which serves to guide the reader in designing and debugging vision

applications and also to make the formal descriptions of computer vision and machine learning algorithms in other texts easier to comprehend and remember.

After all, it is easier to understand complex algorithms and their associated math when you start with an intuitive grasp of how those algorithms work.

Who This Book Is For

This book contains descriptions, working coded examples, and explanations of the computer vision tools contained in the OpenCV library. As such, it should be helpful to many different kinds of users.

Professionals

For those practicing professionals who need to rapidly implement computer vision systems, the sample code provides a quick framework with which to start. Our descriptions of the intuitions behind the algorithms can quickly teach or remind the reader how they work.

Students

As we said, this is the text we wish had back in school. The intuitive explanations, detailed documentation, and sample code will allow you to boot up faster in computer vision, work on more interesting class projects, and ultimately contribute new research to the field.

Teachers

Computer vision is a fast-moving field. We've found it effective to have the students rapidly cover an accessible text while the instructor fills in formal exposition where needed and supplements with current papers or guest lecturers from experts. The students can meanwhile start class projects earlier and attempt more ambitious tasks.

Hobbyists

Computer vision is fun, here's how to hack it.

We have a strong focus on giving readers enough intuition, documentation, and working code to enable rapid implementation of real-time vision applications.

What This Book Is Not

This book is not a formal text. We do go into mathematical detail at various points,* but it is all in the service of developing deeper intuitions behind the algorithms or to make clear the implications of any assumptions built into those algorithms. We have not attempted a formal mathematical exposition here and might even incur some wrath along the way from those who do write formal expositions.

This book is not for theoreticians because it has more of an “applied” nature. The book will certainly be of general help, but is not aimed at any of the specialized niches in computer vision (e.g., medical imaging or remote sensing analysis).

* Always with a warning to more casual users that they may skip such sections.

That said, it is the belief of the authors that having read the explanations here first, a student will not only learn the theory better but remember it longer. Therefore, this book would make a good adjunct text to a theoretical course and would be a great text for an introductory or project-centric course.

About the Programs in This Book

All the program examples in this book are based on OpenCV version 2.0. The code should definitely work under Linux or Windows and probably under OS-X, too. Source code for the examples in the book can be fetched from this book's website (<http://www.oreilly.com/catalog/9780596516130>). OpenCV can be loaded from its source forge site (<http://sourceforge.net/projects/opencvlibrary>).

OpenCV is under ongoing development, with official releases occurring once or twice a year. As a rule of thumb, you should obtain your code updates from the source forge CVS server (http://sourceforge.net/cvs/?group_id=22870).

Prerequisites

For the most part, readers need only know how to program in C and perhaps some C++. Many of the math sections are optional and are labeled as such. The mathematics involves simple algebra and basic matrix algebra, and it assumes some familiarity with solution methods to least-squares optimization problems as well as some basic knowledge of Gaussian distributions, Bayes' law, and derivatives of simple functions.

The math is in support of developing intuition for the algorithms. The reader may skip the *math* and the algorithm descriptions, using only the function definitions and code examples to get vision applications up and running.

How This Book Is Best Used

This text need not be read in order. It can serve as a kind of user manual: look up the function when you need it; read the function's description if you want the gist of how it works "under the hood". The intent of this book is more tutorial, however. It gives you a basic understanding of computer vision along with details of how and when to use selected algorithms.

This book was written to allow its use as an adjunct or as a primary textbook for an undergraduate or graduate course in computer vision. The basic strategy with this method is for students to read the book for a rapid overview and then supplement that reading with more formal sections in other textbooks and with papers in the field. There are exercises at the end of each chapter to help test the student's knowledge and to develop further intuitions.

You could approach this text in any of the following ways.

Grab Bag

Go through Chapters 1–3 in the first sitting, then just hit the appropriate chapters or sections as you need them. This book does not have to be read in sequence, except for Chapters 11 and 12 (Calibration and Stereo).

Good Progress

Read just two chapters a week until you’ve covered Chapters 1–12 in six weeks (Chapter 13 is a special case, as discussed shortly). Start on projects and start in detail on selected areas in the field, using additional texts and papers as appropriate.

The Sprint

Just cruise through the book as fast as your comprehension allows, covering Chapters 1–12. Then get started on projects and go into detail on selected areas in the field using additional texts and papers. This is probably the choice for professionals, but it might also suit a more advanced computer vision course.

Chapter 13 is a long chapter that gives a general background to machine learning in addition to details behind the machine learning algorithms implemented in OpenCV and how to use them. Of course, machine learning is integral to object recognition and a big part of computer vision, but it’s a field worthy of its own book. Professionals should find this text a suitable launching point for further explorations of the literature—or for just getting down to business with the code in that part of the library. This chapter should probably be considered optional for a typical computer vision class.

This is how the authors like to teach computer vision: Sprint through the course content at a level where the students get the gist of how things work; then get students started on meaningful class projects while the instructor supplies depth and formal rigor in selected areas by drawing from other texts or papers in the field. This same method works for quarter, semester, or two-term classes. Students can get quickly up and running with a general understanding of their vision task and working code to match. As they begin more challenging and time-consuming projects, the instructor helps them develop and debug complex systems. For longer courses, the projects themselves can become instructional in terms of project management. Build up working systems first; refine them with more knowledge, detail, and research later. The goal in such courses is for each project to aim at being worthy of a conference publication and with a few project papers being published subsequent to further (postcourse) work.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, path names, directories, and Unix utilities.

Constant width

Indicates commands, options, switches, variables, attributes, keys, functions, types, classes, namespaces, methods, modules, properties, parameters, values, objects,

events, event handlers, XMLtags, HTMLtags, the contents of files, or the output from commands.

Constant width bold

Shows commands or other text that should be typed literally by the user. Also used for emphasis in code samples.

Constant width italic

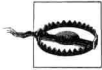
Shows text that should be replaced with user-supplied values.

[...]

Indicates a reference to the bibliography.



Shows text that should be replaced with user-supplied values. This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

OpenCV is free for commercial or research use, and we have the same policy on the code examples in the book. Use them at will for homework, for research, or for commercial products. We would very much appreciate referencing this book when you do, but it is not required. Other than how it helped with your homework projects (which is best kept a secret), we would like to hear how you are using computer vision for academic research, teaching courses, and in commercial products when you do use OpenCV to help you. Again, not required, but you are always invited to drop us a line.

Safari® Books Online



When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

We'd Like to Hear from You

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list examples and any plans for future editions. You can access this information at:

<http://www.oreilly.com/catalog/9780596516130/>

You can also send messages electronically. To be put on the mailing list or request a catalog, send an email to:

info@oreilly.com

To comment on the book, send an email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://www.oreilly.com>

Acknowledgments

A long-term open source effort sees many people come and go, each contributing in different ways. The list of contributors to this library is far too long to list here, but see the [.../opencv/docs/HTML/Contributors/doc_contributors.html](http://opencv/docs/HTML/Contributors/doc_contributors.html) file that ships with OpenCV.

Thanks for Help on OpenCV

Intel is where the library was born and deserves great thanks for supporting this project the whole way through. Open source needs a champion and enough development support in the beginning to achieve critical mass. Intel gave it both. There are not many other companies where one could have started and maintained such a project through good times and bad. Along the way, OpenCV helped give rise to—and now takes (optional) advantage of—Intel's Integrated Performance Primitives, which are hand-tuned assembly language routines in vision, signal processing, speech, linear algebra, and more. Thus the lives of a great commercial product and an open source product are intertwined.

Mark Holler, a research manager at Intel, allowed OpenCV to get started by knowingly turning a blind eye to the inordinate amount of time being spent on an unofficial project back in the library's earliest days. As divine reward, he now grows wine up in Napa's Mt. Vieder area. Stuart Taylor in the Performance Libraries group at Intel enabled OpenCV by letting us "borrow" part of his Russian software team. Richard Wirt was key to its continued growth and survival. As the first author took on management responsibility at Intel, lab director Bob Liang let OpenCV thrive; when Justin Rattner became CTO, we were able to put OpenCV on a more firm foundation under Software Technology Lab—supported by software guru Shinn-Horng Lee and indirectly under his manager, Paul Wiley. Omid Moghadam helped advertise OpenCV in the early days. Mohammad Haghghat and Bill Butera were great as technical sounding boards. Nuriel Amir, Denver

Dash, John Mark Agosta, and Marzia Polito were of key assistance in launching the machine learning library. Rainer Lienhart, Jean-Yves Bouguet, Radek Grzeszczuk, and Ara Nefian were able technical contributors to OpenCV and great colleagues along the way; the first is now a professor, the second is now making use of OpenCV in some well-known Google projects, and the others are staffing research labs and start-ups. There were many other technical contributors too numerous to name.

On the software side, some individuals stand out for special mention, especially on the Russian software team. Chief among these is the Russian lead programmer Vadim Pisarevsky, who developed large parts of the library and also managed and nurtured the library through the lean times when boom had turned to bust; he, if anyone, is the true hero of the library. His technical insights have also been of great help during the writing of this book. Giving him managerial support and protection in the lean years was Valery Kuriakin, a man of great talent and intellect. Victor Eruhimov was there in the beginning and stayed through most of it. We thank Boris Chudinovich for all of the contour components.

Finally, very special thanks go to Willow Garage [WG], not only for its steady financial backing to OpenCV's future development but also for supporting one author (and providing the other with snacks and beverages) during the final period of writing this book.

Thanks for Help on the Book

While preparing this book, we had several key people contributing advice, reviews, and suggestions. Thanks to John Markoff, Technology Reporter at the *New York Times* for encouragement, key contacts, and general writing advice born of years in the trenches. To our reviewers, a special thanks go to Evgeniy Bart, physics postdoc at CalTech, who made many helpful comments on every chapter; Kjerstin Williams at Applied Minds, who did detailed proofs and verification until the end; John Hsu at Willow Garage, who went through all the example code; and Vadim Pisarevsky, who read each chapter in detail, proofed the function calls and the code, and also provided several coding examples. There were many other partial reviewers. Jean-Yves Bouguet at Google was of great help in discussions on the calibration and stereo chapters. Professor Andrew Ng at Stanford University provided useful early critiques of the machine learning chapter. There were numerous other reviewers for various chapters—our thanks to all of them. Of course, any errors result from our own ignorance or misunderstanding, not from the advice we received.

Finally, many thanks go to our editor, Michael Loukides, for his early support, numerous edits, and continued enthusiasm over the long haul.

Gary Adds . . .

With three young kids at home, my wife Sonya put in more work to enable this book than I did. Deep thanks and love—even OpenCV gives her recognition, as you can see in the face detection section example image. Further back, my technical beginnings started with the physics department at the University of Oregon followed by undergraduate years at

UC Berkeley. For graduate school, I'd like to thank my advisor Steve Grossberg and Gail Carpenter at the Center for Adaptive Systems, Boston University, where I first cut my academic teeth. Though they focus on mathematical modeling of the brain and I have ended up firmly on the engineering side of AI, I think the perspectives I developed there have made all the difference. Some of my former colleagues in graduate school are still close friends and gave advice, support, and even some editing of the book: thanks to Frank Guenther, Andrew Worth, Steve Lehar, Dan Cruthirds, Allen Gove, and Krishna Govindarajan.

I specially thank Stanford University, where I'm currently a consulting professor in the AI and Robotics lab. Having close contact with the best minds in the world definitely rubs off, and working with Sebastian Thrun and Mike Montemerlo to apply OpenCV on Stanley (the robot that won the \$2M DARPA Grand Challenge) and with Andrew Ng on STAIR (one of the most advanced personal robots) was more technological fun than a person has a right to have. It's a department that is currently hitting on all cylinders and simply a great environment to be in. In addition to Sebastian Thrun and Andrew Ng there, I thank Daphne Koller for setting high scientific standards, and also for letting me hire away some key interns and students, as well as Kunle Olukotun and Christos Kozyrakis for many discussions and joint work. I also thank Oussama Khatib, whose work on control and manipulation has inspired my current interests in visually guided robotic manipulation. Horst Haussecker at Intel Research was a great colleague to have, and his own experience in writing a book helped inspire my effort.

Finally, thanks once again to Willow Garage for allowing me to pursue my lifelong robotic dreams in a great environment featuring world-class talent while also supporting my time on this book and supporting OpenCV itself.

Adrian Adds . . .

Coming from a background in theoretical physics, the arc that brought me through supercomputer design and numerical computing on to machine learning and computer vision has been a long one. Along the way, many individuals stand out as key contributors. I have had many wonderful teachers, some formal instructors and others informal guides. I should single out Professor David Dorfan of UC Santa Cruz and Hartmut Sadrozinski of SLAC for their encouragement in the beginning, and Norman Christ for teaching me the fine art of computing with the simple edict that "if you can not make the computer do it, you don't know what you are talking about". Special thanks go to James Guzzo, who let me spend time on this sort of thing at Intel—even though it was miles from what I was supposed to be doing—and who encouraged my participation in the Grand Challenge during those years. Finally, I want to thank Danny Hillis for creating the kind of place where all of this technology can make the leap to wizardry and for encouraging my work on the book while at Applied Minds.

I also would like to thank Stanford University for the extraordinary amount of support I have received from them over the years. From my work on the Grand Challenge team with Sebastian Thrun to the STAIR Robot with Andrew Ng, the Stanford AI Lab was always

generous with office space, financial support, and most importantly ideas, enlightening conversation, and (when needed) simple instruction on so many aspects of vision, robotics, and machine learning. I have a deep gratitude to these people, who have contributed so significantly to my own growth and learning.

No acknowledgment or thanks would be meaningful without a special thanks to my lady Lyssa, who never once faltered in her encouragement of this project or in her willingness to accompany me on trips up and down the state to work with Gary on this book. My thanks and my love go to her.

Contents

Preface	ix
1. Overview	1
What Is OpenCV?	1
Who Uses OpenCV?	1
What Is Computer Vision?	2
The Origin of OpenCV	6
Downloading and Installing OpenCV	8
Getting the Latest OpenCV via CVS	10
More OpenCV Documentation	11
OpenCV Structure and Content	13
Portability	14
Exercises	15
2. Introduction to OpenCV	16
Getting Started	16
First Program—Display a Picture	16
Second Program—AVI Video	18
Moving Around	19
A Simple Transformation	22
A Not-So-Simple Transformation	24
Input from a Camera	26
Writing to an AVI File	27
Onward	29
Exercises	29

3. Getting to Know OpenCV	31
OpenCV Primitive Data Types	31
CvMat Matrix Structure	33
IplImage Data Structure	42
Matrix and Image Operators	47
Drawing Things	77
Data Persistence	82
Integrated Performance Primitives	86
Summary	87
Exercises	87
4. HighGUI	90
A Portable Graphics Toolkit	90
Creating a Window	91
Loading an Image	92
Displaying Images	93
Working with Video	102
ConvertImage	106
Exercises	107
5. Image Processing	109
Overview	109
Smoothing	109
Image Morphology	115
Flood Fill	124
Resize	129
Image Pyramids	130
Threshold	135
Exercises	141
6. Image Transforms	144
Overview	144
Convolution	144
Gradients and Sobel Derivatives	148
Laplace	150
Canny	151

Hough Transforms	153
Remap	162
Stretch, Shrink, Warp, and Rotate	163
CartToPolar and PolarToCart	172
LogPolar	174
Discrete Fourier Transform (DFT)	177
Discrete Cosine Transform (DCT)	182
Integral Images	182
Distance Transform	185
Histogram Equalization	186
Exercises	190
7. Histograms and Matching	193
Basic Histogram Data Structure	195
Accessing Histograms	198
Basic Manipulations with Histograms	199
Some More Complicated Stuff	206
Exercises	219
8. Contours	222
Memory Storage	222
Sequences	223
Contour Finding	234
Another Contour Example	243
More to Do with Contours	244
Matching Contours	251
Exercises	262
9. Image Parts and Segmentation	265
Parts and Segments	265
Background Subtraction	265
Watershed Algorithm	295
Image Repair by Inpainting	297
Mean-Shift Segmentation	298
Delaunay Triangulation, Voronoi Tessellation	300
Exercises	313