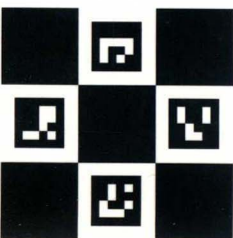


O'REILLY®

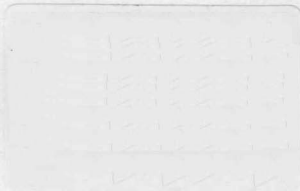


Learning OpenCV 3

COMPUTER VISION IN C++ WITH THE OPENCV LIBRARY



Adrian Kaehler & Gary Bradski



Learning OpenCV 3

Computer Vision in C++ with the OpenCV Library

Adrian Kaehler and Gary Bradski



Beijing • Boston • Farnham • Sebastopol • Tokyo

Learning OpenCV 3

by Adrian Kaehler and Gary Bradski

Copyright © 2017 Adrian Kaehler, Gary Bradski. All rights reserved.

Printed in the United States of America.

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

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://www.oreilly.com/safari>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editor: Dawn Schanafelt

Indexer: Ellen Troutman

Production Editor: Kristen Brown

Interior Designer: David Futato

Copyeditor: Rachel Monaghan

Cover Designer: Karen Montgomery

Proofreader: James Fraleigh

Illustrator: Rebecca Demarest

December 2016: First Edition

Revision History for the First Edition

2016-12-09: First Release

2017-01-20: Second Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781491937990> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Learning OpenCV 3*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-93799-0

[M]

Preface

This book provides a working guide to the C++ Open Source Computer Vision Library (OpenCV) version 3.x and gives a general background on the field of computer vision sufficient to help readers use OpenCV effectively.

Purpose of This Book

Computer vision is a rapidly growing field largely because of four trends:

- The advent of mobile phones put millions of cameras in people's hands.
- The Internet and search engines aggregated the resulting giant flows of image and video data into huge databases.
- Computer processing power became a cheap commodity.
- Vision algorithms themselves became more mature (now with the advent of deep neural networks, which OpenCV is increasingly supporting; see *dnn* at `opencv_contrib` (https://github.com/opencv/opencv_contrib) [`opencv_contrib`]).

OpenCV has played a role in the growth of computer vision by enabling hundreds of thousands of people to do more productive work in vision. OpenCV 3.x now allows students, researchers, professionals, and entrepreneurs to efficiently implement projects and jump-start research by providing them with a coherent C++ computer vision architecture that is optimized over many platforms.

The purpose of this book is to:

- Comprehensively document OpenCV by detailing what function calling conventions really mean and how to use them correctly
- 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 code examples to start from
- Suggest ways to fix some of the more advanced routines when something goes wrong

This book documents OpenCV in a way that allows the reader to rapidly do interesting and fun things in computer vision. It gives an intuitive understanding of how the algorithms work, which serves to guide the reader in designing and debugging vision applications and also makes the formal descriptions of computer vision and machine learning algorithms in other texts easier to comprehend and remember.

Who This Book Is For

This book contains descriptions, working code examples, and explanations of the C++ computer vision tools contained in the OpenCV 3.x library. Thus, it should be helpful to many different kinds of users:

Professionals and entrepreneurs

For practicing professionals who need to rapidly prototype or professionally implement computer vision systems, the sample code provides a quick framework with which to start. Our descriptions of the algorithms can quickly teach or remind the reader how they work. OpenCV 3.x sits on top of a *hardware acceleration layer* (HAL) so that implemented algorithms can run efficiently, seamlessly taking advantage of a variety of hardware platforms.

Students

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 students rapidly cover an accessible text while the instructor fills in formal exposition where needed and supplements that with current papers or guest lectures 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,¹ but it is all in the service of developing deeper intuitions behind the algorithms or to clarify 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 has more of an “applied” nature. It 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).

That said, we believe that by reading the explanations here first, a student will not only learn the theory better, but remember it longer as well. 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 3.x. The code should work under Linux, Windows, and OS X. Using references online, OpenCV 3.x has full support to run on Android and iOS. Source code for the examples in the book can be fetched from this book’s website (<http://bit.ly/learningOpenCV3>); source code for OpenCV is available on GitHub (<https://github.com/opencv/opencv>); and prebuilt versions of OpenCV can be loaded from its SourceForge site (<http://sourceforge.net/projects/opencvlibrary>).

OpenCV is under ongoing development, with official releases occurring quarterly. To stay completely current, you should obtain your code updates from the aforementioned GitHub site. OpenCV maintains a website at <http://opencv.org>; for developers, there is a wiki at <https://github.com/opencv/opencv/wiki>.

Prerequisites

For the most part, readers need only know how to program in C++. Many of the math sections in this book are optional and are labeled as such. The mathematics involve simple algebra and basic matrix algebra, and assume 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.

¹ Always with a warning to more casual users that they may skip such sections.

The math in this book 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, and read the function's description if you want the gist of how it works "under the hood." However, the intent of this book is tutorial. It gives you a basic understanding of computer vision along with details of how and when to use selected algorithms.

This book is written to allow its use as an adjunct or 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 1–5 in the first sitting, and 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 18 and 19 (which cover camera calibration and stereo imaging) and Chapters 20, 21, and 22 (which cover machine learning). Entrepreneurs and students doing project-based courses might go this way.

Good progress

Read just two chapters a week until you've covered Chapters 1–22 in 11 weeks (Chapter 23 will go by in an instant). Start on projects and dive into details on selected areas in the field, using additional texts and papers as appropriate.

The sprint

Cruise through the book as fast as your comprehension allows, covering Chapters 1–23. 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 20 is a brief chapter that gives general background on machine learning, which is followed by Chapters 21 and 22, which give more details on 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. The machine learning interface has been substantially simplified and unified in OpenCV 3.x.

This is how we 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 supplying 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 be worthy of a conference publication and with a few project papers being published subsequent to further (post-course) work. In OpenCV 3.x, the C++ code framework, Buildbots, GitHub use, pull request reviews, unit and regression tests, and documentation are together a good example of the kind of professional software infrastructure a startup or other business should put together.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, pathnames, 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.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at https://github.com/oreillymedia/Learning-OpenCV-3_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, research, or for commercial products! We would very much appreciate you referencing this book when you do so, but it is not required. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Learning OpenCV 3* by Adrian Kaehler and Gary Bradski (O’Reilly). Copyright 2017 Adrian Kaehler, Gary Bradski, 978-1-491-93799-0.”

Other than hearing how it helped with your homework projects (which is best kept a secret), we would love 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, it’s not required, but you are always invited to drop us a line.

O’Reilly Safari



Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O’Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit <http://oreilly.com/safari>.

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://bit.ly/learningOpenCV3>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

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` 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 as it started and grew. From time to time, Intel still funds contests and contributes work to OpenCV. Intel also donated the built-in performance primitives code, which provides for seamless speedup on Intel architectures. Thank you for that.

Google has been a steady funder of development for OpenCV by sponsoring interns for OpenCV under its Google Summer of Code project; much great work has been done through this funding. Willow Garage provided several years of funding that enabled OpenCV to go from version 2.x through to version 3.0. During this time, the

computer vision R&D company Itseez (recently bought by Intel Corporation) has provided extensive engineering support and web services hosting over the years. Intel has indicated verbal agreement to continue this support (thanks!).

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 is the largest single contributor to the library. Vadim also managed and nurtured the library through the lean times when boom had turned to bust and then bust to boom; 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 has been Victor Eruhimov, a cofounder of Itseez [Itseez] and now CEO of Itseez3D.

Several people consistently help out with managing the library during weekly meetings: Grace Vesom, Vincent Rabaud, Stefano Fabri, and of course, Vadim Pisarevsky. The developer notes for these meetings can be seen at https://github.com/opencv/opencv/wiki/Meeting_notes.

Many people have contributed to OpenCV over time; a list of more recent ones is: Dinar Ahmaturov, Pablo Alcantarilla, Alexander Alekhin, Daniel Angelov, Dmitriy Anisimov, Anatoly Baksheev, Cristian Balint, Alexandre Benoit, Laurent Berger, Leonid Beynenson, Alexander Bokov, Alexander Bovyryn, Hilton Bristow, Vladimir Bystritsky, Antonella Cascitelli, Manuela Chessa, Eric Christiansen, Frederic Devernay, Maria Dimashova, Roman Donchenko, Vladimir Dudnik, Victor Eruhimov, Georgios Evangelidis, Stefano Fabri, Sergio Garrido, Harris Gasparakis, Yuri Gitman, Lluís Gomez, Yury Gorbachev, Elena Gvozdeva, Philipp Hasper, Fernando J. Iglesias Garcia, Alexander Kalistratov, Andrey Kamaev, Alexander Karsakov, Rahul Kavi, Pat O'Keefe, Siddharth Kherada, Eugene Khvedchenya, Anna Kogan, Marina Kolpakova, Kirill Korniyakov, Ivan Korolev, Maxim Kostin, Evgeniy Kozhinov, Ilya Krylov, Laksono Kurnianggoro, Baisheng Lai, Ilya Lavrenov, Alex Leontiev, Gil Levi, Bo Li, Ilya Lysenkov, Vitaliy Lyudvichenko, Bence Magyar, Nikita Manovich, Juan Manuel Perez Rua, Konstantin Matskevich, Patrick Mihelich, Alexander Mordvintsev, Fedor Morozov, Gregory Morse, Marius Muja, Mircea Paul Muresan, Sergei Nosov, Daniil Osokin, Seon-Wook Park, Andrey Pavlenko, Alexander Petrikov, Philip aka Dikay900, Prasanna, Francesco Puja, Steven Puttemans, Vincent Rabaud, Edgar Riba, Cody Rigney, Pavel Rojtberg, Ethan Rublee, Alfonso Sanchez-Beato, Andrew Senin, Maksim Shabunin, Vlad Shakhuro, Adi Shavit, Alexander Shishkov, Sergey Sivolgin, Marvin Smith, Alexander Smorkalov, Fabio Solari, Adrian Stratulat, Evgeny Talanin, Manuele Tamburrano, Ozan Tonkal, Vladimir Tyan, Yannick Verdie, Pierre-Emmanuel Viel, Vladislav Vinogradov, Pavel Vlasov, Philipp Wagner, Yida Wang, Jiaolong Xu, Marian Zajko, Zoran Zivkovic.

Other contributors show up over time at <https://github.com/opencv/opencv/wiki/ChangeLog>. Finally, Arraiy [Arraiy] is now also helping maintain OpenCV.org (the free and open codebase).

Thanks for Help on This Book

While preparing this book and the previous version of this book, we'd like to thank John Markoff, science reporter at the *New York Times*, for encouragement, key contacts, and general writing advice born of years in the trenches. We also thank our many editors at O'Reilly, especially Dawn Schanafelt, who had the patience to continue on as slips became the norm while the errant authors were off trying to found a startup. This book has been a long project that slipped from OpenCV 2.x to the current OpenCV 3.x release. Many thanks to O'Reilly for sticking with us through all that.

Adrian Adds...

In the first edition (*Learning OpenCV*) I singled out some of the great teachers who helped me reach the point where a work like this would be possible. In the intervening years, the value of the guidance received from each of them has only grown more clear. My many thanks go out to each of them. I would like to add to this list of extraordinary mentors Tom Tombrello, to whom I owe a great debt, and in whose memory I would like to dedicate my contribution to this book. He was a man of exceptional intelligence and deep wisdom, and I am honored to have been given the opportunity to follow in his footsteps. Finally, deep thanks are due the OpenCV community, for welcoming the first edition of this book and for your patience through the many exciting, but perhaps distracting, endeavors that have transpired while this edition was being written.

This edition of the book has been a long time coming. During those intervening years, I have had the fortune to work with dozens of different companies advising, consulting, and helping them build their technology. As a board member, advisory board member, technical fellow, consultant, technical contributor, and founder, I have had the fortune to see and love every dimension of the technology development process. Many of those years were spent with Applied Minds, Inc., building and running our robotics division there, or at Applied Invention corporation, a spinout of Applied Minds, as a Fellow there. I was constantly pleased to find OpenCV at the heart of outstanding projects along the way, ranging from health care and agriculture to aviation, defense, and national security. I have been equally pleased to find the first edition of this book on people's desks in almost every institution along the way. The technology that Gary and I used to build Stanley has become integral to countless projects since, not the least of which are the many self-driving car projects now under way—any one of which, or perhaps all of which, stand ready to change and improve daily life for countless people. What a joy it is to be part of all of this! The number of

incredible minds that I have encountered over the years—who have told me what benefit the first edition was to them in the classes they took, the classes they taught, the careers they built, and the great accomplishments that they completed—has been a continuous source of happiness and wonder. I am hopeful that this new edition of the book will continue to serve you all, as well as to inspire and enable a new generation of scientists, engineers, and inventors.

As the last chapter of this book closes, we start new chapters in our lives working in robotics, AI, vision, and beyond. Personally, I am deeply grateful for all of the people who have contributed the many works that have enabled this next step in my own life: teachers, mentors, and writers of books. I hope that this new edition of our book will enable others to make the next important step in their own lives, and I hope to see you there!

Gary Adds...

I founded OpenCV in 1999 with the goal to accelerate computer vision and artificial intelligence and give everyone the infrastructure to work with that I saw at only the top labs at the time. So few goals actually work out as intended in life, and I'm thankful this goal did work out 17 (!) years later. Much of the credit for accomplishing that goal was due to the help, over the years, of many friends and contributors too numerous to mention.² But I will single out the original Russian group I started working with at Intel, who ran a successful computer vision company (Itseez.com) that was eventually bought back into Intel; we started out as coworkers but have since become deep friends.

With three teenagers at home, my wife, Sonya Bradski, put in more work to enable this book than I did. Many thanks and love to her. The teenagers I love, but I can't say they accelerated the book. :)

This version of the book was started back at the former startup I helped found, Industrial Perception Inc., which sold to Google in 2013. Work continued in fits and starts on random weekends and late nights ever since. Somehow it's now 2016—time flies when you are overwhelmed! Some of the speculation that I do toward the end of Chapter 23 was inspired by the nature of robot minds that I experienced with the PR2, a two-armed robot built by Willow Garage, and with the Stanley project at Stanford—the robot that won the \$2 million DARPA Grand Challenge.

² We now have many contributors, as you can see by scrolling past the updates in the change logs at <https://github.com/opencv/opencv/wiki/ChangeLog>. We get so many new algorithms and apps that we now store the best in self-maintaining and self-contained modules in *opencv_contrib* (https://github.com/opencv/opencv_contrib).

As we close the writing of this book, we hope to see you in startups, research labs, academic sites, conferences, workshops, VC offices, and cool company projects down the road. Feel free to say hello and chat about cool new stuff that you're doing. I started OpenCV to support and accelerate computer vision and AI for the common good; what's left is your part. We live in a creative universe where someone can create a pot, the next person turns that pot into a drum, and so on. Create! Use OpenCV to create something uncommonly good for us all!

What Is OpenCV?

OpenCV (OpenCL) is an open source (<http://opencv.org>) computer vision library available from <http://openi.org>. In 1999 Gary Bradski (Bradski, Intel Corporation), launched OpenCV with the hopes of accelerating computer vision and artificial intelligence by providing a solid infrastructure for everyone working in the field. The library is written in C and C++ and runs under Linux, Windows, and Mac OS X. There is active development on interfaces for Python, Java, Matlab, and other languages, including porting the library to Android and iOS for mobile applications. OpenCV has received much of its support over the years from Intel and OpenCL, but especially from Intel's (Intel's) OpenCL community, which did the bulk of the early development work. Figure 3-10 (Intel's) OpenCL community to maintain the always open and free OpenCV.org (OpenCV).

OpenCV was designed for cross-platform (Apple, Windows, and Linux) support in real-time applications. It is written in optimized C++ and can take advantage of multi-core processors. If you desire further performance optimization (e.g., Intel architecture [Intel]), you can use Intel's integrated performance framework (IPP) libraries (IPP), which consist of low-level optimized routines in many different algorithmic areas. OpenCV automatically uses the appropriate one library at runtime if that library is installed. Starting with OpenCV 3.0, Intel created the OpenCL wrapper and OpenCV community a free-of-charge subset of IPP (code-named IPPICV), which is built into and included with OpenCV by default.

One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build highly sophisticated vision applications quickly. The OpenCV library contains over 300 functions that span many areas of work, including factory product inspection, medical imaging, creature user interfaces, camera calibration,

Table of Contents

Preface	xv
1. Overview	1
What Is OpenCV?	1
Who Uses OpenCV?	2
What Is Computer Vision?	3
The Origin of OpenCV	6
OpenCV Block Diagram	8
Speeding Up OpenCV with IPP	9
Who Owns OpenCV?	10
Downloading and Installing OpenCV	10
Installation	10
Getting the Latest OpenCV via Git	13
More OpenCV Documentation	13
Supplied Documentation	14
Online Documentation and the Wiki	14
OpenCV Contribution Repository	17
Downloading and Building Contributed Modules	17
Portability	18
Summary	19
Exercises	19
2. Introduction to OpenCV	21
Include Files	21
Resources	22
First Program—Display a Picture	23
Second Program—Video	25
Moving Around	27

A Simple Transformation	31
A Not-So-Simple Transformation	32
Input from a Camera	35
Writing to an AVI File	36
Summary	38
Exercises	38
3. Getting to Know OpenCV Data Types.....	41
The Basics	41
OpenCV Data Types	41
Overview of the Basic Types	42
Basic Types: Getting Down to Details	44
Helper Objects	52
Utility Functions	60
The Template Structures	67
Summary	68
Exercises	69
4. Images and Large Array Types.....	71
Dynamic and Variable Storage	71
The cv::Mat Class: N-Dimensional Dense Arrays	72
Creating an Array	73
Accessing Array Elements Individually	78
The N-ary Array Iterator: NArrayMatIterator	81
Accessing Array Elements by Block	84
Matrix Expressions: Algebra and cv::Mat	85
Saturation Casting	87
More Things an Array Can Do	88
The cv::SparseMat Class: Sparse Arrays	89
Accessing Sparse Array Elements	90
Functions Unique to Sparse Arrays	92
Template Structures for Large Array Types	94
Summary	97
Exercises	97
5. Array Operations.....	99
More Things You Can Do with Arrays	99
cv::abs()	102
cv::absdiff()	103
cv::add()	103
cv::addWeighted()	104
cv::bitwise_and()	106

cv::bitwise_not()	107
cv::bitwise_or()	107
cv::bitwise_xor()	108
cv::calcCovarMatrix()	108
cv::cartToPolar()	110
cv::checkRange()	111
cv::compare()	111
cv::completeSymm()	112
cv::convertScaleAbs()	112
cv::countNonZero()	113
cv::cvarrToMat()	113
cv::dct()	114
cv::dft()	115
cv::cvtColor()	117
cv::determinant()	119
cv::divide()	120
cv::eigen()	120
cv::exp()	121
cv::extractImageCOI()	121
cv::flip()	122
cv::gemm()	122
cv::getConvertElem() and cv::getConvertScaleElem()	123
cv::idct()	124
cv::idft()	124
cv::inRange()	124
cv::insertImageCOI()	125
cv::invert()	126
cv::log()	126
cv::LUT()	127
cv::magnitude()	127
cv::Mahalanobis()	128
cv::max()	129
cv::mean()	130
cv::meanStdDev()	130
cv::merge()	131
cv::min()	131
cv::minMaxIdx()	132
cv::minMaxLoc()	133
cv::mixChannels()	134
cv::mulSpectrums()	136
cv::multiply()	136
cv::mulTransposed()	136