



# ОГЛАВЛЕНИЕ

<b>Об авторах .....</b>	<b>8</b>
<b>О рецензентах.....</b>	<b>11</b>
<b>Предисловие .....</b>	<b>13</b>
Структура книги .....	13
Что необходимо для чтения этой книги.....	14
Предполагаемая аудитория .....	15
Обозначения и графические выделения.....	15
Отзывы.....	16
Поддержка клиентов .....	16
Загрузка кода примеров .....	16
Загрузка цветных иллюстраций .....	17
Опечатки.....	17
Нарушение авторских прав .....	17
Вопросы.....	18
<b>Глава 1. Работа с файлами изображений и видео ...</b>	<b>19</b>
Введение в OpenCV .....	19
Загрузка и установка OpenCV .....	21
Получение компилятора и настройка CMake .....	22
Настройка OpenCV с помощью CMake .....	23
Компиляция и установка библиотеки.....	26
Структура каталогов OpenCV.....	27
Создание проекта, включающего OpenCV .....	28
Общие замечания об использовании библиотеки .....	29
Средства для разработки новых проектов.....	30
Создание приложения OpenCV на C++ в Qt Creator.....	31
Чтение и запись файлов изображений.....	33
Основные элементы API .....	33
Поддерживаемые форматы графических файлов .....	36
Пример программы .....	36

Чтение и запись видеофайлов .....	41
Пример программы .....	41
Средства взаимодействия с пользователем .....	43
Полосы прокрутки .....	46
Управление с помощью мыши .....	47
Кнопки .....	48
Рисование и отображение текста .....	49
Резюме .....	51
<b>Глава 2. Инструменты обработки изображений .....</b>	<b>52</b>
Основные типы данных .....	52
Доступ к пикселям .....	55
Хронометраж .....	56
Типичные операции над изображениями .....	56
Арифметические операции .....	58
Сохранение данных .....	61
Гистограммы .....	63
Пример программы .....	65
Пример программы .....	69
Резюме .....	73
<b>Глава 3. Коррекция и улучшение изображений .....</b>	<b>74</b>
Фильтрация изображений .....	74
Сглаживание .....	75
Повышение резкости .....	79
Работа с пирамидами изображений .....	82
Пирамиды Лапласа .....	83
Морфологические операции .....	84
Пример программы .....	87
LUT-фильтры .....	88
Пример программы .....	89
Геометрические преобразования .....	90
Аффинное преобразование .....	91
Ретуширование .....	101
Пример программы .....	103
Очистка от шумов .....	107
Пример программы .....	108
Резюме .....	110
<b>Глава 4. Работа с цветом .....</b>	<b>111</b>
Цветовые пространства .....	111

Преобразования цветовых пространств (cvtColor) .....	112
Сегментация на основе цветового пространства.....	132
HSV-сегментация.....	133
YCrCb-сегментация.....	134
Цветоперенос .....	136
Пример программы .....	136
Резюме .....	138
<b>Глава 5. Обработка видео.....</b>	<b>139</b>
Стабилизация видео .....	139
Сверхвысокое разрешение .....	146
Сшивка изображений .....	155
Резюме .....	167
<b>Глава 6. Вычислительная фотография.....</b>	<b>169</b>
Изображения с широким динамическим диапазоном.....	169
Создание HDR-изображений .....	172
Тональная компрессия .....	176
Совмещение .....	177
Экспозиционное объединение.....	178
Бесшовное клонирование .....	179
Обесцвечивание .....	181
Нефотореалистичный рендеринг .....	183
Резюме .....	186
<b>Глава 7. Ускорение обработки изображений.....</b>	<b>187</b>
Установка OpenCV с поддержкой OpenCL.....	189
Краткое описание установки OpenCV с поддержкой OpenCL.....	194
Проверка использования GPU.....	194
Ускорение собственных функций .....	196
Проверка поддержки OpenCL.....	196
Ваша первая программа для GPU .....	198
А теперь в реальном времени .....	200
Резюме .....	205
<b>Предметный указатель .....</b>	<b>206</b>



## ОБ АВТОРАХ

**Глория Буэно Гарсия** имеет степень доктора по машинному зрению, присвоенную университетом Ковентри, Великобритания. Она работала в должности старшего научного сотрудника в нескольких исследовательских центрах, в частности в подразделении UMR 7005 Национального научно-исследовательского центра при университете Луи Пастера в Страсбурге (Франция), в компании Gilbert Gilkes & Gordon Technology (Великобритания) и в центре научно-технических исследований CEIT San Sebastian (Испания). Она автор двух патентов, одного зарегистрированного типа программ и более 100 рецензированных публикаций. Ее научные интересы лежат в области двумерной и трехмерной мультимодальной обработки изображений и искусственного интеллекта. Возглавляет исследовательскую группу VISILAB в университете Кастилия–Ла-Манча. Является соавтором книги по программированию для мобильных устройств с применением OpenCV «OpenCV Essentials», вышедшей в издательстве Packt Publishing.

---

*Посвящается нашим сыновьям в качестве компенсации за то время, когда мы не могли поиграть с ними, и нашим родителям за постоянную поддержку безо всяких условий. Спасибо от Глории и Оскара.*

---

Научные интересы **Оскара Дениза Суареса** сосредоточены в основном в области машинного зрения и распознавания образов. Он автор более 50 рецензированных работ в журналах и на конференциях. Занял второе место на конкурсе работ докторов наук по машинному зрению и распознаванию образов, проводимом AERFAI (Испанская ассоциация по распознаванию образов и анализу изображений), а также получил награду от компании Innocentive Inc. за лучшую программу по обработке и переформатированию файлов изображений. Его работы используются такими передовыми компаниями, как Existor, Gliif, Tarpmedia, E-Twenty и другими. Он внес вклад в разра-

ботку OpenCV. В настоящее время работает доцентом в университете Кастилия–Ла-Манча и сотрудничает с VISILAB. Является старшим членом IEEE и участвует в организациях AAAI, SIANI, CEA-IFAC, AEPiA и AERFAI-IAPR. Исполняет функции академического редактора журнала PLoS ONE. По приглашению занимался исследовательской работой в университете Карнеги-Меллон, Имперском колледже Лондона и в компании Leica Biosystems. Соавтор двух книг по OpenCV.

**Хосе Луис Эспиноса Аранда** имеет степень доктора информатики, присвоенную университетом Кастилия–Ла-Манча. Он вышел в финал испанского конкурса Certamen Universitario Arquímedes de Introducción a la Investigación científica 2009 года со своим дипломным проектом. В сферу его научных интересов входят машинное зрение, эвристические алгоритмы и исследование операций. В настоящее время работает в группе VISILAB младшим научным сотрудником, занимаясь разработками в области машинного зрения.

---

*Посвящаю своим родителям и братьям.*

---

**Хесус Салидо Терсеро** получил степень доктора электротехники в 1996 году в Мадридском политехническом университете (Испания). Затем два года работал приглашенным научным сотрудником в Институте робототехники (университет Карнеги-Меллон, Питтсбург, США), занимаясь кооперативными системами из нескольких роботов. После возвращения в испанский университет Кастилия–Ла-Манча ведет курсы по робототехнике и промышленной информатике, а также занимается исследованиями в области интеллектуальных систем машинного зрения. Последние три года направлял основные усилия на разработку приложений машинного зрения для мобильных устройств. Соавтор книги по программированию мобильных устройств с применением OpenCV.

---

*Посвящаю тем, кому обязан всем в своей жизни: родителям, Сагарио и Марии.*

---

**Исмаэль Серрано Грасиа** получил ученую степень по информатике в 2012 году в университете Кастилия–Ла-Манча. Удостоен высшей

оценки за дипломный проект, посвященный распознаванию людей. В этом приложении используются камеры глубины, программируемые с помощью библиотеки OpenCV. В настоящее время работает над докторской диссертацией, получив грант от испанского Министерства науки и исследований. По совместительству работает младшим научным сотрудником в группе VISILAB, занимаясь различными вопросами машинного зрения.

---

*Посвящаю родителям, которые дали мне возможность получить образование и поддерживают на протяжении всей жизни. А также своему научному руководителю, доктору Оскару Денизу, моему другу, наставнику и помощнику. И еще друзьям и своей девушке, которая всегда помогала мне и верила, что я смогу это сделать.*

---

**Ноэлия Валлез Энано** с детства обожала компьютеры, но свою первую машину получила только в пятнадцать лет. В 2009 году она с высшими баллами окончила курс информатики в университете Кастилия–Ла-Манча. Начав работать в группе VISILAB, занималась проектом в области САD-систем для маммографии и обработкой медицинских электронных данных. Затем получила степень магистра физико-математических наук и приступила к работе над докторской диссертацией, посвященной методам обработки изображений и распознавания образов. Также любит преподавать и заниматься различными вопросами искусственного интеллекта.



## О РЕЦЕНЗЕНТАХ

**Вальтер Лучетти**, известный в Интернете под ником *Muzhar*, – инженер-компьютерщик из Италии, специализируется в робототехнике и робототехническом восприятии. Получил степень лауреата в 2005 года за написанную в Пизанском исследовательском центре «E.Piaggio» диссертацию по 3D-маппингу с помощью двумерного лазера, управляемого серводвигателем, когда на трехмерную поверхность проецируются RGB-данные. В ходе работы над диссертацией он впервые познакомился с *OpenCV* – это было в начале 2004 года, когда библиотека еще находилась в зачаточном состоянии. После защиты занимался разработкой низкоуровневых встраиваемых систем и высокоуровневых систем для настольных компьютеров. Знания в области машинного зрения и машинного обучения он серьезно обогатил, работая инженером-исследователем в Центре передовой робототехники Густаво Стефанини в Ла Специи (Италия), филиале лаборатории PERCRO Пизанской Высшей школы *Sant’Anna*.

В настоящее время работает в программной индустрии, пишет прошивки для встраиваемых ARM-систем, программы для настольных компьютеров на базе библиотеки *Qt* и разрабатывает интеллектуальные алгоритмы для систем видеонаблюдения на основе *OpenCV* и *CUDA*. Также работает над личным робототехническим проектом: *MuzharBot*. Это гусеничный наземный подвижный робот, который пользуется машинным зрением для распознавания препятствий, анализа и исследования окружающей среды. Управление роботом осуществляется с помощью алгоритмов, основанных на *ROS*, *CUDA* и *OpenCV*. Познакомиться с проектом можно на сайте <http://muzharbot.robot-home.it>.

**Андре де Суза Морейра** получил степень магистра информатики со специализацией в компьютерной графике от Папского католического университета Рио де Жанейро (Бразилия).

Закончил Федеральный университет штата Мараньян (Бразилия) со степенью бакалавра информатики. Во время работы над дипломом входил в состав исследовательской группы *Labmint* и занимался об-

работкой медицинских изображений, а конкретно распознаванием и диагностикой рака молочной железы.

В настоящее время работает исследователем и системным аналитиком в Институте Текграф (Instituto Tecgraf), одном из главных научно-исследовательских центров компьютерной графики в Бразилии. С 2007 года активно использует PHP, HTML и CSS, а сейчас применяет для разработки C++11/C++14, SQLite, Qt, Boost, и OpenGL. Дополнительные сведения можно найти на его персональном сайте по адресу [www.andredsm.com](http://www.andredsm.com).

**Марвин Смит** в настоящее время работает инженером-программистом в оборонной промышленности и специализируется в области фотограмметрии и дистанционного зондирования. Степень бакалавра информатики получил в университете Невады в Рино. В область его технических интересов входят высокопроизводительные вычисления, распределенная обработка изображений и многоспектральные формирователи изображений. До перехода в оборонную отрасль Марвин проходил практику в группе интеллектуальной робототехники в Исследовательском центре НАСА в Амесе и в Центре ходовых испытаний в Неваде.





# ПРЕДИСЛОВИЕ

OpenCV является, наверное, самой широко распространенной библиотекой компьютерного зрения. Она включает сотни готовых функций обработки изображений и используется как в академических учреждениях, так и в промышленности. Камеры становятся все дешевле, а спрос на обработку изображений растет, поэтому спектр приложений OpenCV как на настольных, так и на мобильных платформах постоянно расширяется.

В этой книге на примерах демонстрируются основные алгоритмы обработки изображений, реализованные в OpenCV. Есть книги, где объясняется теория, лежащая в основе OpenCV. Есть и такие, где приводятся примеры больших, почти законченных приложений. Мы же ориентируемся на читателей, которым нужны – и притом быстро – простые для понимания работающие примеры, на которые можно навешивать дополнительные функции.

В первой, вступительной, части объясняется, как установить библиотеку, описывается ее общая структура и приводятся простые примеры чтения и записи изображений и видео. Далее рассматриваются следующие темы: работа с изображениями и видео, основные инструменты обработки изображений, коррекция и улучшение изображений, цвет, обработка видео, вычислительная фотография. И наконец обсуждаются более сложные темы, в частности, повышение быстродействия за счет использования графических процессоров. На протяжении всей книги описываются новые функции и методы, появившиеся в последней основной версии, OpenCV 3.

## Структура книги

В главе 1 «Работа с файлами изображений и видео» описывается, как читать такие файлы. Демонстрируются также основные средства взаимодействия с пользователем, которые позволяют изменять значения параметров, выбирать интересующие области и т. д.

В главе 2 «Инструменты обработки изображений» описываются основные структуры данных и процедуры, которые понадобятся в последующих главах.

В главе 3 «Коррекция и улучшение изображений» речь пойдет о преобразованиях, которые обычно применяются для исправления дефектов изображения. Здесь рассматриваются фильтры, точечные преобразования с использованием справочных таблиц, геометрические преобразования и алгоритмы ретуширования и очистки от шумов.

Глава 4 «Работа с цветом» посвящена месту цвета в обработке изображений. Здесь объясняется, как устроены различные цветовые пространства и как осуществляется цветоперенос с одного изображения на другое.

В главе 5 «Обработка видео» рассматриваются методы работы с последовательностями изображений, т. е. видео. Здесь будут описаны реализации алгоритмов стабилизации видео, сверхвысокого разрешения и сшивки.

В главе 6 «Вычислительная фотография» объясняется, как читать HDR-изображения и выполнять для них тональную компрессию.

Глава 7 «Ускорение обработки изображений» посвящена очень важному для обработки изображений вопросу: быстрдействию. Современные графические процессоры (GPU) – наилучшая на сегодняшний день технология ускорения длительных операций над изображениями.

## **Что необходимо для чтения этой книги**

Цель этой книги – научить применению OpenCV для обработки изображений на ряде практических примеров. Используется последняя версия – OpenCV 3.0.

Каждая глава содержит несколько готовых к работе примеров, иллюстрирующих излагаемый материал. Таким образом, мы стремимся как можно быстрее предоставить код, над которым можно надстраивать дополнительный функционал.

Для проработки примеров нужно только бесплатное ПО. Все примеры разрабатывались и тестировались в бесплатной интегрированной среде Qt Creator с использованием компилятора GNU/GCC. Для конфигурирования процесса сборки библиотеки OpenCV используется платформенно-независимая программа CMake. Для примеров

применения ускорителей в главе 7 понадобится еще бесплатный комплект средств разработки OpenCL SDK.

## Предполагаемая аудитория

Эта книга рассчитана на читателей, уже знакомых с языком C++ и желающих научиться обработке изображений с помощью библиотеки OpenCV. Предполагается, что у читателя имеются минимальные теоретические познания в области обработки изображений. Мы не рассматриваем вопросы, в большей степени относящиеся к компьютерному зрению, например: выделение признаков, обнаружение объектов, трассировка или машинное обучение.

## Обозначения и графические выделения

В этой книге для выделения семантически различной информации применяются различные стили. Ниже приведены примеры стилей с пояснениями.

Фрагменты кода в тексте, имена папок и файлов, имена системных переменных, URL-адреса и данные, вводимые пользователем, выглядят следующим образом: «С каждым модулем ассоциирован файл-заголовок (например, `core.hpp`)».

Отдельно стоящие фрагменты кода набраны так:

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace std;
using namespace cv;

int main(int argc, char *argv[])
{
    Mat frame; // контейнер фрейма
```

Чтобы привлечь внимание к части кода, строки или отдельные слова выделяются полужирным шрифтом:

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <iostream>

using namespace std;
```

```
using namespace cv;

int main(int argc, char *argv[])
{
```

Текст, который вводится на консоли или выводится на консоль, напечатан следующим образом:

```
C:\opencv-buildQt\install
```

**Новые термины и важные слова** набраны полужирным шрифтом. Также выделяются элементы интерфейса: «Для перехода к следующему экрану нажмите кнопку **Next**».



Предупреждения и важные замечания оформлены так.



Советы и рекомендации выглядят так.

---

## Отзывы

Мы всегда рады отзывам читателей. Расскажите нам, что вы думаете об этой книге – что вам понравилось или, быть может, не понравилось. Читательские отзывы важны для нас, так как помогают выпускать книги, из которых вы черпаете максимум полезного для себя.

Чтобы отправить обычный отзыв, просто пошлите письмо на адрес [feedback@packtpub.com](mailto:feedback@packtpub.com), указав название книги в качестве темы.

Если вы являетесь специалистом в некоторой области и хотели бы стать автором или соавтором книги, познакомьтесь с инструкциями для авторов по адресу [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Поддержка клиентов

Счастливым обладателям книг Packt мы можем предложить ряд услуг, которые позволят извлечь из приобретения максимум пользы.

### **Загрузка кода примеров**

Вы можете скачать код примеров ко всем книгам издательства Packt, купленным на сайте <http://www.PacktPub.com>. Если книга была

куплена в другом месте, зайдите на страницу <http://www.PacktPub.com/support>, зарегистрируйтесь, и мы отправим файлы по электронной почте.

## **Загрузка цветных иллюстраций**

Мы также предлагаем PDF-файл с цветными изображениями снимков экрана и диаграмм, встречающихся в книге. Имея цветные изображения, вы сможете полнее оценить, как меняется вывод. Загрузить файл можно со страницы [https://www.packtpub.com/sites/default/files/downloads/ImageProcessingwithOpenCV\\_Graphics.pdf](https://www.packtpub.com/sites/default/files/downloads/ImageProcessingwithOpenCV_Graphics.pdf).

## **Опечатки**

Мы проверяли содержимое книги со всем тщанием, но какие-то ошибки все же могли проскользнуть. Если вы найдете в нашей книге ошибку, в тексте или в коде, пожалуйста, сообщите нам о ней. Так вы избавите других читателей от разочарования и поможете нам сделать следующие издания книги лучше. При обнаружении опечатки просьба зайти на страницу <http://www.packtpub.com/submit-errata>, выбрать книгу, щелкнуть по ссылке **Errata Submission Form** и ввести информацию об опечатке. Проверив ваше сообщение, мы поместим информацию об опечатке на нашем сайте или добавим ее в список замеченных опечаток в разделе Errata для данной книги. Список подтвержденных опечаток можно просмотреть, введя название книги в поле поиска на странице <http://www.packtpub.com/books/content/support>. Информация появится в разделе **Errata**.

## **Нарушение авторских прав**

Незаконное размещение защищенного авторским правом материала в Интернете – проблема для всех носителей информации. В издательстве Packt мы относимся к защите прав интеллектуальной собственности и лицензированию очень серьезно. Если вы обнаружите незаконные копии наших изданий в любой форме в Интернете, пожалуйста, незамедлительно сообщите нам адрес или название веб-сайта, чтобы мы могли предпринять соответствующие меры.

Просим отправить ссылку на вызывающий подозрение в пиратстве материал по адресу [copyright@packtpub.com](mailto:copyright@packtpub.com).

Мы будем признательны за помощь в защите прав наших авторов и содействие в наших стараниях предоставлять читателям полезные сведения.

## Вопросы

Если вас смущает что-то в этой книге, вы можете связаться с нами по адресу [questions@packtpub.com](mailto:questions@packtpub.com), и мы сделаем все возможное для решения проблемы.



# ГЛАВА 1.

## Работа с файлами изображений и видео

В этой главе мы познакомимся с установкой OpenCV и напишем первые, очень простые программы. Будут рассмотрены следующие вопросы:

- краткое введение в OpenCV для начинающих, сопровождаемое пошаговой инструкцией по установке библиотеки;
- краткий обзор структуры каталогов OpenCV после ее установки на локальный диск;
- рекомендации по созданию проектов в некоторых распространенных средах программирования;
- как пользоваться функциями чтения и записи изображений и видео;
- описание библиотечных функций для реализации интерфейса приложения, в том числе взаимодействия с помощью мыши, рисования примитивов и поддержки библиотеки Qt.

## Введение в OpenCV

Библиотека **OpenCV (Open Source Computer Vision)**, первоначально разработанная компанией Intel, теперь является бесплатной, кросс-платформенной библиотекой обработки изображений в реальном времени и стала стандартом де факто для всего, что связано с компьютерным зрением. Ее первая версия была выпущена в 2000 году на условиях **лицензии BSD**, но с тех пор функциональность существенно обогатилась благодаря усилиям научного сообщества. В 2012 году некоммерческий фонд OpenCV.org взял на себя задачу ведения сайта для поддержки разработчиков и пользователей.



На момент написания этой книги вышла новая основная версия OpenCV 3.0, но пока она находится на стадии бета-тестирования. В этой книге мы расскажем о самых существенных изменениях в этой версии.

OpenCV реализована для большинства популярных операционных систем, в т. ч. GNU/Linux, OS X, Windows, Android, iOS и некоторых других. Первая реализация была написана на языке C, но популярность библиотеки значительно возросла с выходом версии 2.0, переписанной на C++. Новые функции пишутся только на C++. Впрочем, в настоящее время имеются полнофункциональные интерфейсы между OpenCV и другими языками программирования, в частности Java, Python и MATLAB/Octave. Разработаны также обертки для таких языков, как C#, Ruby и Perl, чтобы популяризировать библиотеку среди программистов.

Стремясь добиться максимальной производительности при решении счетных (потребляющих много процессорного времени) задач машинного зрения, авторы OpenCV включили поддержку следующих возможностей:

- многопоточность при работе на многоядерных компьютерах за счет использования шаблонной библиотеки **Threading Building Blocks (TBB)**, разработанной Intel;
- подмножество библиотеки **Integrated Performance Primitives (IPP)** на процессорах Intel для резкого повышения производительности. Благодаря любезности Intel эти примитивы бесплатно включены в бета-версию 3.0;
- интерфейсы для работы с **графическими процессорами (GPU)** с применением архитектуры **Compute Unified Device Architecture (CUDA)** и языка **Open Computing Language (OpenCL)**.

OpenCV применяется в таких областях, как сегментация, выделение признаков в двумерных и трехмерных изображениях, идентификация объектов, распознавание лиц, трассировка движения, распознавание жестов, шивка изображений, обработка изображений с широким динамическим диапазоном (**high dynamic range, HDR**), дополненная реальность и т. д. Кроме того, для поддержки некоторых из вышеперечисленных задач включен модуль, содержащий функции статистического машинного обучения.



# Загрузка и установка OpenCV

OpenCV можно бесплатно скачать с сайта <http://opencv.org>, где имеется как последняя версия (в настоящее время 3.0 beta), так и предыдущие.



Скачивая нестабильную версию, например 3.0 beta, всегда нужно помнить о возможности ошибок в программе.

---

На странице <http://opencv.org/downloads.html> имеются версии OpenCV для всех поддерживаемых платформ. Код библиотеки и информацию о нем можно получить из различных репозиторийев (в зависимости от поставленной цели).

- Основной репозиторий (<http://sourceforge.net/projects/opencvlibrary>), ориентированный на конечных пользователей. Содержит двоичные версии библиотеки и готовые к компиляции исходные коды для конечной платформы.
- Репозиторий тестовых данных ([https://github.com/itseez/opencv\\_extra](https://github.com/itseez/opencv_extra)), содержащий наборы данных для тестирования некоторых библиотечных модулей.
- Репозиторий дополнений ([http://github.com/itseez/opencv\\_contrib](http://github.com/itseez/opencv_contrib)), содержащий исходный код дополнений и самых передовых возможностей, добавленный сторонними разработчиками. Он не так хорошо протестирован, как код в стволовой ветви, и может содержать ошибки.



В последней версии OpenCV 3.0 дополнительные модули в основной пакет не включены. Их следует скачивать отдельно и явно включать в процесс компиляции с помощью соответствующих флагов. Будьте внимательны, потому что некоторые модули зависят от сторонних программ, не входящих в состав OpenCV.

---

- Сайт документации (<http://docs.opencv.org/master/>) по каждому модулю, включая дополнительные.
- Репозиторий разработчиков (<https://github.com/Itseez/opencv>), содержащий текущую разрабатываемую версию би-

библиотеки. Предназначен для разработчиков основной функциональности библиотеки и «нетерпеливых» пользователей, которым хочется попробовать последние обновления до их официального выпуска.

В отличие от платформ GNU/Linux и OS X, на которых OpenCV распространяется только в исходном виде, для Windows существуют откомпилированные (с помощью компиляторов Microsoft Visual C++ v10, v11 и v12) версии библиотеки. Откомпилированные версии совместимы с компиляторами Microsoft. Но если требуется разработать проект в другой среде компиляции (например, GNU GCC), то библиотеку придется перекомпилировать для нее.



Самый быстрый способ начать работу с OpenCV – воспользоваться откомпилированной версией, включенной в состав дистрибутива. Но лучше бы самостоятельно собрать библиотеку с настройками, оптимальными для конкретной платформы и планируемого использования. В этой главе приведены инструкции по сборке и установке OpenCV для Windows. Дополнительные сведения о настройке библиотеки в Linux можно найти на страницах [http://docs.opencv.org/doc/tutorials/introduction/linux\\_install](http://docs.opencv.org/doc/tutorials/introduction/linux_install) и <https://help.ubuntu.com/community/OpenCV>.

## Получение компилятора и настройка CMake

Для кросс-платформенной разработки с применением OpenCV лучше всего использовать **комплект инструментов GNU** (включающий gmake, g++ и gdb). Его нетрудно получить для популярных операционных систем. Мы предпочитаем среду разработки, состоящую из комплекта инструментов GNU и кросс-платформенного **каркаса Qt**, в который входит библиотека Qt и **интегрированная среда разработки (IDE) Qt Creator**. Каркас Qt можно бесплатно скачать с сайта <http://qt-project.org/>.

После установки компилятора в Windows не забудьте добавить в переменную окружения PATH путь к исполняемому файлу компилятора, например `c:\Qt\Qt5.2.1\5.2.1\mingw48_32\bin` в случае каркаса Qt. Для Windows имеется средство **Rapid Environment Editor** (на сай-

те <http://www.rapidee.com>), которое позволяет удобно изменять PATH и другие переменные окружения.

Для управления процессом сборки OpenCV способом, независимым от компилятора, рекомендуется система CMake, бесплатная кросс-платформенная программа с открытым исходным кодом, доступная на сайте <http://www.cmake.org/>.

## Настройка OpenCV с помощью CMake

После копирования исходного кода библиотеки на локальный диск необходимо настроить make-файлы для ее компиляции. CMake – основной инструмент конфигурирования процесса установки OpenCV. Его можно использовать как из командной строки, так и из графического интерфейса.

Ниже перечислены шаги конфигурирования OpenCV с помощью CMake:

1. Задать начальный (ниже он обозначается `OPENCV_SRC`) и конечный (`OPENCV_BUILD`) каталог. В конечный каталог будут помещены откомпилированные двоичные файлы.
2. Отметить флажки **Grouped** и **Advanced** и нажать кнопку **Configure**.
3. Выбрать компилятор (например, предлагаемый по умолчанию компилятор GNU, MSVC и т. д.)
4. Включить нужные и отключить ненужные возможности.
5. Нажать кнопку **Configure** и повторять шаги 4 и 5, пока не перестанут выдаваться сообщения об ошибках.
6. Нажать кнопку **Generate** и выйти из CMake.

На рисунке ниже показано главное окно CMake, в котором заданы начальный и конечный каталог и отмечены флажки для группировки доступных функций.

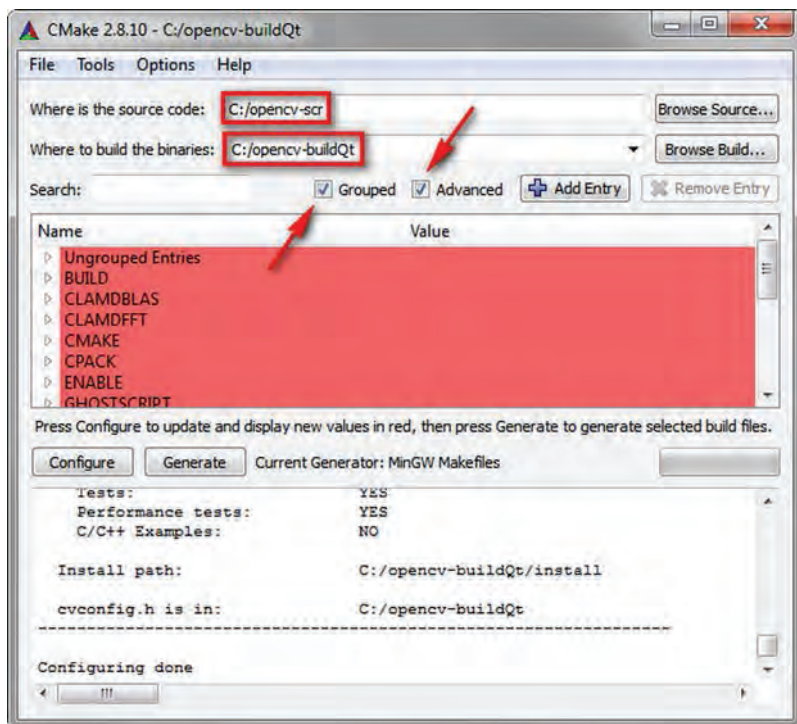


Для краткости мы обозначаем начальный и конечный каталоги OpenCV в процедуре установки `OPENCV_SRC` и `OPENCV_BUILD` соответственно. Но помните, что все каталоги должны соответствовать вашей машине.

---

До начала конфигурирования CMake находит установленные на машине компиляторы и определяет другие свойства локального окружения, существенные для сборки OpenCV. На рисунке ниже по-

казано главное окно **CMake** по завершении этапа предварительного конфигурирования, а красным цветом выделены сгруппированные параметры.



Главное окно CMake до начала конфигурирования

Можно вообще не менять параметры по умолчанию и продолжить процесс конфигурирования. Однако лучше бы для удобства установить некоторые параметры.

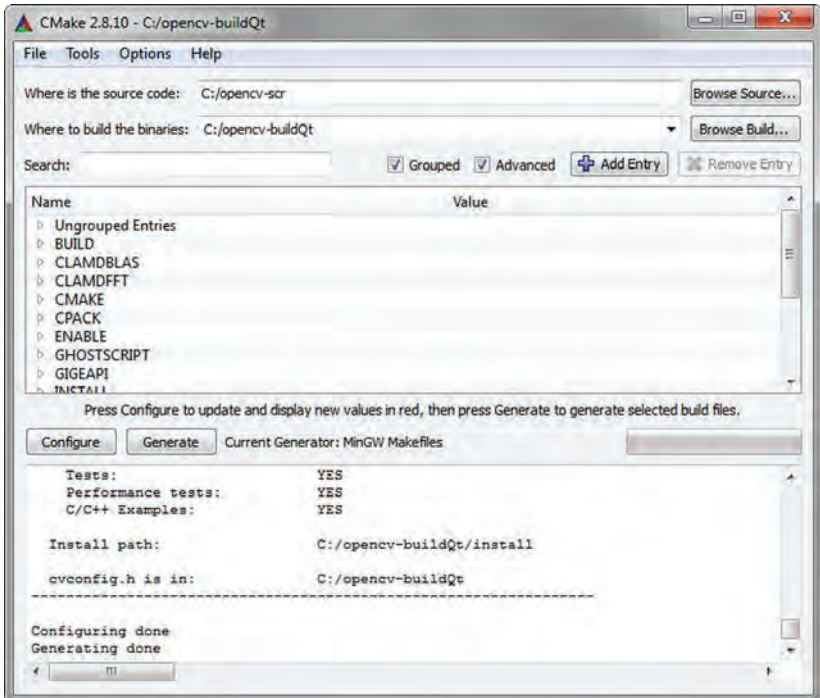
- `BUILD_EXAMPLES`: собирать примеры использования OpenCV.
- `BUILD_opencv_<module_name>`: включить в процесс сборки модуль с указанным именем.
- `OPENCV_EXTRA_MODULES_PATH`: используется, если нужен какой-то дополнительный модуль; задайте путь к исходному коду дополнительных модулей (например, `C:/opencv_contrib-master/modules`).
- `WITH_QT`: включить в библиотеку функциональность Qt.

- `WITH_IPP`: ЭТОТ параметр по умолчанию включен. В текущую версию OpenCV 3.0 включено подмножество библиотеки **Intel Integrated Performance Primitives (IPP)**, ускоряющее работу библиотеки.



При компиляции новой версии OpenCV 3.0 (beta) будьте внимательны – сообщалось о неожиданных ошибках, связанных с включением IPP (т. е. при компиляции с параметрами по умолчанию). Мы рекомендуем выключить параметр `WITH_IPP`.

Если на этапе конфигурирования (шаги 4 и 5) CMake не выдает ошибок, то можно приступать к генерации окончательных make-файлов для процесса сборки. На рисунке ниже показано главное окно CMake после завершения шага генерации без ошибок.



Главное окно CMake после завершения конфигурирования

## Компиляция и установка библиотеки

Следующий шаг после генерации make-файлов – компиляция библиотеки с помощью подходящей программы. Обычно эта программа запускается из командной строки, когда текущим является конечный каталог (заданный на этапе конфигурирования в CMake). Например, в Windows компиляция из командной строки запускается так:

```
OPENCV_BUILD>mingw32-make
```

При этом начинается процесс сборки с использованием make-файлов, сгенерированных CMake. Процесс компиляции обычно занимает несколько минут. Если компиляция завершится без ошибок, то можно переходить к установке:

```
OPENCV_BUILD>mingw32-make install
```

Эта команда копирует исполняемые файлы OpenCV в каталог `OPENCV_BUILD\install`.

Если во время компиляции что-то пойдет не так, то нужно перезапустить CMake и изменить выбранные ранее параметры, а затем заново сгенерировать make-файлы.

По завершении установки нужно добавить путь к каталогу двоичных файлов библиотеки (например, в Windows созданные DLL-файлы находятся в каталоге `OPENCV_BUILD\install\x64\mingw\bin`) в переменную окружения **PATH**. Иначе при попытке выполнить любой исполняемый файл, собранный с OpenCV, будет выдано сообщение о том, что не найдена библиотека.

Чтобы проверить, успешно ли завершился процесс установки, можно попробовать выполнить какой-нибудь пример, откомпилированный вместе с библиотекой (если в CMake был задан параметр `BUILD_EXAMPLES`). Код примеров (на C++) находится в каталоге `OPENCV_BUILD\install\x64\mingw\samples\cpp`.



Краткие инструкции по установке OpenCV относятся к Windows. Подробное описание необходимых условий для установки в Linux можно найти на странице [http://docs.opencv.org/doc/tutorials/introduction/linux\\_install/linux\\_install.html](http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html). И хотя это пособие относится к OpenCV 2.0, почти все написанное в нем справедливо и для версии 3.0.

---

# Структура каталогов OpenCV

После установки OpenCV в каталоге `OPENCV_BUILD\install` будут находиться файлы трех типов.

- **Заголовки.** Они находятся в подкаталоге `OPENCV_BUILD\install\include` и используются при разработке новых проектов.
- **Двоичные файлы библиотеки.** Это статические или динамические библиотеки (в зависимости от режима сборки, заданного в CMake), содержащие функционал отдельных модулей. Они находятся в подкаталоге `bin` (например, `x64\mingw\bin`, если использовался компилятор GNU).
- **Двоичные файлы примеров.** Это исполняемые файлы примеров использования библиотеки. Их исходный код можно найти в соответствующем каталоге (например, `OPENCV_SRC\sources\samples`).

OpenCV имеет модульную структуру, т. е. пакет включает статическую или динамическую (DLL) библиотеку для каждого модуля. Официальную документацию по модулям можно найти по адресу <http://docs.opencv.org/master/>. Перечислим основные модули.

- `core`: базовые функции, используемые всеми остальными модулями, а также фундаментальные структуры данных, включая тип многомерного массива `mat`.
- `highgui`: простые средства организации **пользовательского интерфейса**. При сборке библиотеки с поддержкой Qt (параметр `WITH_QT`) обеспечивается совместимость с этим каркасом.
- `imgproc`: функции обработки изображений, в том числе фильтрация (линейная и нелинейная), геометрические преобразования, преобразование цветового пространства, вычисление гистограмм и т. д.
- `imgcodecs`: простой интерфейс для чтения и записи изображений.



Обращайте внимание на изменения, внесенные в версии OpenCV 3.0, т. к. функциональность частично перемещена из одного модуля в другой (например, функции чтения и записи изображений теперь находятся не в `highgui`, а в `imgcodecs`).

---

- `photo`: включает средства вычислительной фотографии, в т. ч. ретуширование, очистку от шумов, обработку изображений с **широким динамическим диапазоном (HDR)** и другие.
- `stitching`: нужен для сшивки изображений.
- `videoio`: содержит простой интерфейс для захвата видео и видеокодеки.
- `video`: функциональность для анализа видео (оценка движения, выделение заднего плана, трассировка объектов).
- `features2d`: функции для обнаружения признаков (углов и плоскостных объектов), их описания, сопоставления и т. д.
- `objdetect`: функции для обнаружения объектов и некоторые предопределенные детекторы (например, лиц, глаз, улыбок, людей, автомобилей и т. д.).

Назовем еще некоторые модули: `calib3d` (калибровка камеры), `flann` (кластеризация и поиск), `ml` (машинное обучение), `shape` (вычисление расстояния между фигурами и сопоставление фигур), `superres` (сверхвысокое разрешение), `video` (анализ видео), `videostab` (стабилизация видео).



Начиная с версии 3.0 beta, новые дополнительные модули распространяются в отдельном пакете (`opencv_contrib-master.zip`), который можно скачать по адресу [https://github.com/itseez/opencv\\_contrib](https://github.com/itseez/opencv_contrib). Чтобы работать с функциями, которые находятся в этих модулях, нужно отчетливо понимать, что они делают. Краткий обзор новой функциональности, включенной в версию OpenCV 3.0, имеется в документе <http://opencv.org/opencv-3-0-beta.html>.

## Создание проекта, включающего OpenCV

В этой книге предполагается, что основным языком для программирования приложений для обработки изображений является C++, хотя существуют интерфейсы и к другим языкам (например, Python, Java, MATLAB/Octave и т. д.).

В этом разделе объясняется, как разрабатывать приложения с использованием C++ API в простой и удобной кросс-платформенной среде.