



Бесплатная электронная книга

УЧУСЬ opencv

Free unaffiliated eBook created from
Stack Overflow contributors.

#opencv

.....	1
1: opencv	2
.....	2
.....	2
OpenCV 3	2
OpenCV 2	2
Examples.....	3
OpenCV.....	3
OpenCV	4
.....	5
.....	5
.....	6
, Java.....	6
-.....	7
.....	7
C ++	7
.....	8
OpenCV 3.1 Windows.....	8
OPENCV?.....	14
2:	16
.....	16
Examples.....	16
cv :: Mat :: at ().....	16
cv :: Mat :: ptr	17
C ++.....	18
Matiterator.....	19
.....	22
3:	24
Examples.....	24
.....	24
.....	24

Live Capture.....	25
.....	25
.....	26
4: OpenCV Android.....	28
Examples.....	28
.....	28
OpenCV Manager.....	29
.....	29
5: VideoCapture OpenCV Python.....	32
Examples.....	32
.....	32
VideoCapture OpenCV Java.....	32
6: Java.....	34
.....	34
.....	34
Examples.....	35
,	35
.....	36
Mat BufferedImage.....	37
.....	38
7:	41
Examples.....	41
.....	41
.....	41
.....	41
.....	41
Java.....	42
.....	42
.....	43
C ++.....	43
8: C ++.....	46
.....	46

.....	46
.....	46
$g(i, j) = f(i, j) +$	46
Examples.....	47
c ++.....	47
9:	49
Examples.....	49
.....	49
.....	49
OpenCV - kmeans ().	50
10: BLOB	51
Examples.....	51
blob.....	51
11:	53
.....	53
.....	53
Examples.....	53
Canny	53
Canny Algorithm - C ++.....	54
Canny Thresholds.....	55
Canny Edge - Capture - Python.....	55
Canny Edge Thresholds	55
12:	57
Examples.....	57
Java.....	57
Java.....	57
.....	57
13:	59
.....	59
.....	59
.....	59

Examples.....	60
C ++.....	60
Thresholding.....	61
.....	61
14:	63
.....	63
Examples.....	63
.....	63
.....	63
Vec.....	64
15: OpenCV.....	66
Examples.....	66
.....	66
MJPEG IP-.....	66
OpenCV Java.....	67
16: (, ,) C ++.....	68
.....	68
Examples.....	68
.....	68
17: opencv 3.1.0-dev Python2 Windows CMake.....	71
.....	71
Examples.....	80
.....	80
18:	82
.....	82
Examples.....	82
OpenCV (Java).....	82
19: OpenCV.....	83
.....	83
Examples.....	83
OpenCV Ubuntu.....	83

20: Java.....86

Examples.....86

.....86

.....87

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [opencv](#)

It is an unofficial and free opencv ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official opencv.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с opencv

замечания

В этом разделе представлен обзор того, что такое opencv, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в opencv и ссылаться на связанные темы. Поскольку документация для opencv нова, вам может потребоваться создать начальные версии этих связанных тем.

Версии

OpenCV 3

Версия	Дата выхода
3,2	2016-12-23
3,1	2015-12-18
3.0	2015-06-03
3.0 RC1	2015-04-23
3,0 бета	2014-11-07
3,0 альфа	2014-08-21

OpenCV 2

Версия	Дата выхода
2.4.13	2016-05-19
2.4.12	2015-07-30
2.4.11	2015-02-25
2.4.10	2014-10-01
2.4.9	2014-04-14

Версия	Дата выхода
2.3.1	2011-08-17
2.3.0	2011-07-04
2.2.0	2010-12-05
2.1.0	2010-04-06
2.0.0	2009-10-01
1.0.0	2006-10-19

Examples

Загрузите и покажите изображение с помощью OpenCV

В этом примере мы увидим, как загрузить цветное изображение с диска и отобразить его с помощью встроенных функций OpenCV. Для этого мы можем использовать привязки C / C++, Python или Java.

В C ++:

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

#include <iostream>

using namespace cv;

int main(int argc, char** argv) {
    // We'll start by loading an image from the drive
    Mat image = imread("image.jpg", CV_LOAD_IMAGE_COLOR);

    // We check that our image has been correctly loaded
    if(image.empty()) {
        std::cout << "Error: the image has been incorrectly loaded." << std::endl;
        return 0;
    }

    // Then we create a window to display our image
    namedWindow("My first OpenCV window");

    // Finally, we display our image and ask the program to wait for a key to be pressed
    imshow("My first OpenCV window", image);
    waitKey(0);

    return 0;
}
```

В Python:

```

import sys
import cv2

# We load the image from disk
img = cv2.imread("image.jpg", cv2.CV_LOAD_IMAGE_COLOR)

# We check that our image has been correctly loaded
if img.size == 0
    sys.exit("Error: the image has not been correctly loaded.")

# We create a window to display our image
cv2.namedwindow("My first OpenCV window")

# We display our image and ask the program to wait until a key is pressed
cv2.imshow("My first OpenCV window", img)
cv2.waitKey(0)

# We close the window
cv2.destroyAllWindows()

```

В Java:

```

import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.CvType;
import org.opencv.highgui.Highgui;
public class Sample{
public static void main (String[] args) {

    //Load native opencv library
    System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

    //Read image from file first param:file location ,second param:color space
    Mat img = imread("image.jpg", CV_LOAD_IMAGE_COLOR);

    //If the image is successfully read.
    if (img.size() == 0) {
        System.exit(1);
    }
}
}

```

У HighGui нет именованных окон или эквивалентов imshow в opencv java. Используйте swing или swt для отображения изображения.

Создайте и установите OpenCV из источника

Это пошаговое руководство по установке OpenCV 3 в Linux-системе Debian из источника. Шаги должны оставаться неизменными для других дистрибутивов, просто замените соответствующие команды диспетчера пакетов при установке пакетов для сборки.

Примечание. Если вам не нравится тратить время на создание материала или не нравится терминал, вы, скорее всего, можете установить OpenCV из графического интерфейса диспетчера пакетов Synaptic. Однако эти библиотеки часто устаревают.

Подготовьтесь к сборке

Для установки необходимых пакетов выполните следующие команды в своем терминале:

```
sudo apt-get update
sudo apt-get install build-essential
sudo apt-get install cmake git libgtk2.0-dev pkg-config \
    libavcodec-dev libavformat-dev libswscale-dev
```

Следующие пакеты являются необязательными:

```
sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev \
    libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev
```

Выполните следующую команду, чтобы получить исходный код OpenCV и подготовить сборку:

```
mkdir ~/src
cd ~/src
git clone https://github.com/opencv/opencv.git
cd opencv
mkdir build && cd build
```

Создание и установка

Мы включаем примеры в сборку, но не стесняемся их оставлять. Также не стесняйтесь устанавливать другие флаги и настраивать свою сборку по своему усмотрению.

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D INSTALL_C_EXAMPLES=ON ..
```

Если CMake не сообщал о каких-либо ошибках или недостающих библиотеках, продолжайте сборку.

```
make -j$(nproc)
```

Если ошибок не было, мы можем продолжить установку OpenCV в систему:

```
sudo make install
```

Теперь OpenCV должен быть доступен вашей системе. Вы можете использовать следующие строки, чтобы узнать, где был установлен OpenCV и какие библиотеки были установлены:

```
pkg-config --cflags opencv # get the include path (-I)
pkg-config --libs opencv   # get the libraries path (-L) and the libraries (-l)
```

Тестирование

Сначала мы создаем примеры на C ++:

```
cd ~/src/opencv/samples
cmake .
make
```

Если ошибок не было, запустите любой образец, например

```
./cpp/cpp-example-edge
```

Если образец запускается, то библиотеки C ++ будут правильно установлены.

Затем проверьте привязки Python:

```
python
>> import cv2
>> print cv2.__version__
```

Если эти команды импортируют OpenCV и печатают правильную версию без жалоб, тогда привязки Python будут правильно установлены.

Поздравляем, вы только что создали и установили OpenCV. Счастливое программирование!

Для Mac см. Здесь [Установка OpenCV в Mac OS X](#)

Привет, мир в Java

Изображение OpenCv, прочитанное из файловой системы в Java

```
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.imgcodecs.Imgcodecs;

public class Giris {
    public static void main(String[] args) {
        //Load native library
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
        //image container object
        Mat imageArray;
        //Read image from file system
        imageArray=Imgcodecs.imread("C:\\Users\\mesutpiskin\\sample.jpg");
        //Get image with & height
        System.out.println(imageArray.rows());
        System.out.println(imageArray.cols());
    }
}
```

```
}  
}
```

Получить изображение с веб-камеры

Отображение живого видеопотока, взятого с веб-камеры, с использованием класса VideoCapture от OpenCV с Java, C / C ++ и Python.

Джава

```
import org.opencv.core.Core;  
import org.opencv.core.Mat;  
import org.opencv.videoio.VideoCapture;  
  
public class Camera {  
    public static void main(String[] args) {  
        // Load Native Library  
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);  
        // image container object  
        Mat imageArray = new Mat();  
        // Video device acces  
        VideoCapture videoDevice = new VideoCapture();  
        // 0:Start default video device 1,2 etc video device id  
        videoDevice.open(0);  
        // is contected  
        if (videoDevice.isOpened()) {  
            // Get frame from camera  
            videoDevice.read(imageArray);  
            // image array  
            System.out.println(imageArray.toString());  
            // Release video device  
            videoDevice.release();  
        } else {  
            System.out.println("Error.");  
        }  
    }  
}
```

C ++

```
#include "opencv2/opencv.hpp"  
#include "iostream"  
  
int main(int, char**) {  
    // open the first webcam plugged in the computer  
    cv::VideoCapture camera(0);  
    if (!camera.isOpened()) {  
        std::cerr << "ERROR: Could not open camera" << std::endl;  
        return 1;  
    }  
  
    // create a window to display the images from the webcam
```

```

cv::namedWindow("Webcam", CV_WINDOW_AUTOSIZE);

// this will contain the image from the webcam
cv::Mat frame;

// capture the next frame from the webcam
camera >> frame;

// display the frame until you press a key
while (1) {
    // show the image on the window
    cv::imshow("Webcam", frame);
    // wait (10ms) for a key to be pressed
    if (cv::waitKey(10) >= 0)
        break;
}
return 0;
}

```

ПИТОН

```

import numpy as np
import cv2

# Video source - can be camera index number given by 'ls /dev/video*'
# or can be a video file, e.g. '~/Video.avi'
cap = cv2.VideoCapture(0)

while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

```

Начало работы с OpenCV 3.1 в Windows

Мы устанавливаем OpenCV 3.1.0 в Windows и начинаем работу. Существует два способа установки OpenCV на окна. Один из них - загрузить установщик и запустить его. Другое - строить из источника.

Это самый простой способ установить OpenCV и начать работу. OpenCV дает предварительно сборку бинарные файлы для установки на Windows , [здесь](#) . После завершения загрузки загрузите его и установите по выбранному пути.

ProTip: Убедитесь, что ваш путь OpenCV не содержит пробелов. Итак, лучше, если вы просто установите его в каталог C: \ или D: \ root

Проблема с указанным выше методом заключается в том, что вы не можете использовать модули opencv_contrib. Кроме того, он не поставляется со всеми сторонними инструментами и библиотеками. Итак, если вы хотите использовать все это, просто следуйте за ним.

Я объясню наименьший минимум, чтобы установить OpenCV из источника. Для более продвинутого, см. [Здесь](#) .

- Установите [CMake](#) .
- Клонировать источник OpenCV от <https://github.com/Itseez/opencv.git> в некотором каталоге, который не имеет пробелов. Позволяет называть его «OpenCVdir».

 Itseez / opencv

<> Code

! Issues 880

🔗 Pull requests 32

📖 Wiki

⚡ Pu

Open Source Computer Vision Library <http://opencv.org>

📄 18,556 commits

🔗 3 branches


Branch: master ▾

New pull request

New file

Fi

- Теперь откройте CMake GUI и добавьте исходный каталог (OpenCVdir) в меню «Источники» и создайте каталог в меню сборки. **Совет.** Если каталог сборки отсутствует, создайте его в папке opencv.

 CMake 2.8.11.2 - C:/opencv/build

File Tools Options Help

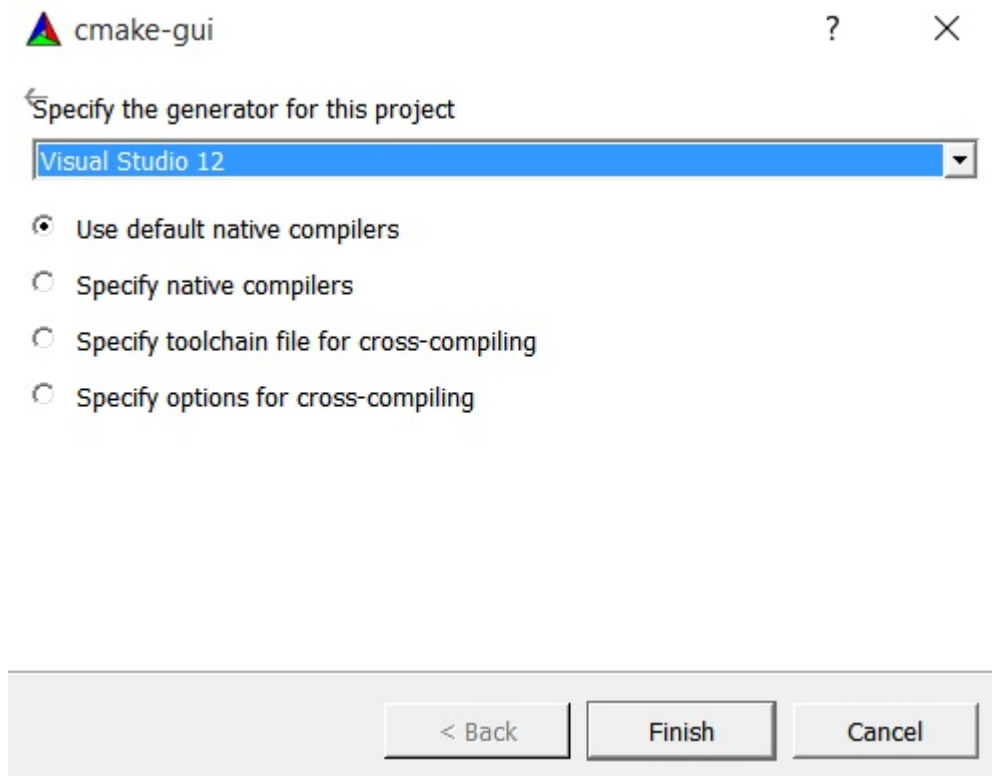
Where is the source code: C:/opencv/source

Where to build the binaries: C:/opencv/build

Search:

☒ Group

- Нажмите «Настроить» и выберите версию компилятора Visual Studio. У меня была Visual Studio 2013 Professional 32-бит, поэтому я выбрал компилятор Visual Studio 12.



Совет. Вы можете скачать Visual Studio 2013 Professional [здесь](#). Он поставляется с пробной пробкой в течение 30 дней + расширенный маршрут продолжительностью 90 дней после входа в систему.

- Нажмите Finish, и CMake автоматически загрузит все пакеты. Вы можете добавлять или удалять пакеты. Нажмите «Настроить» еще раз.
- Если вы хотите создать дополнительные модули opencv_contrib, вам необходимо загрузить их [здесь](#). Затем извлеките их и добавьте каталог opencv_contrib / modules в ваш CMake, как показано ниже.

Where is the source code: C:/opencv/source

Where to build the binaries: C:/opencv/build

Search:

☒ Group

Name	Value
+ CUDA	
+ DOXYGEN	
+ ENABLE	
+ GIGEAPI	
+ INSTALL	
+ JAVA	
+ MATLAB	
- OPENCV	
OPENCV_CONFIG_FILE_INCLUDE_DIR	C:/opencv/build11
OPENCV_EXTRA_MODULES_PATH	C:/opencv/source/opencv
OPENCV_WARNINGS_ARE_ERRORS	<input type="checkbox"/>
+ PYTHON2	

Press Configure to update and display new values in red, then press Generate to generate

Configure

Generate

Current Generator: Visual Studio 12

- Теперь нажмите Configure еще раз, а затем нажмите Generate.
- Закройте CMake. Перейдите в папку your_opencv \ build и откройте файл с именем « OpenCV.sln». - Он откроет Visual Studio. Теперь запустите его как в Debug

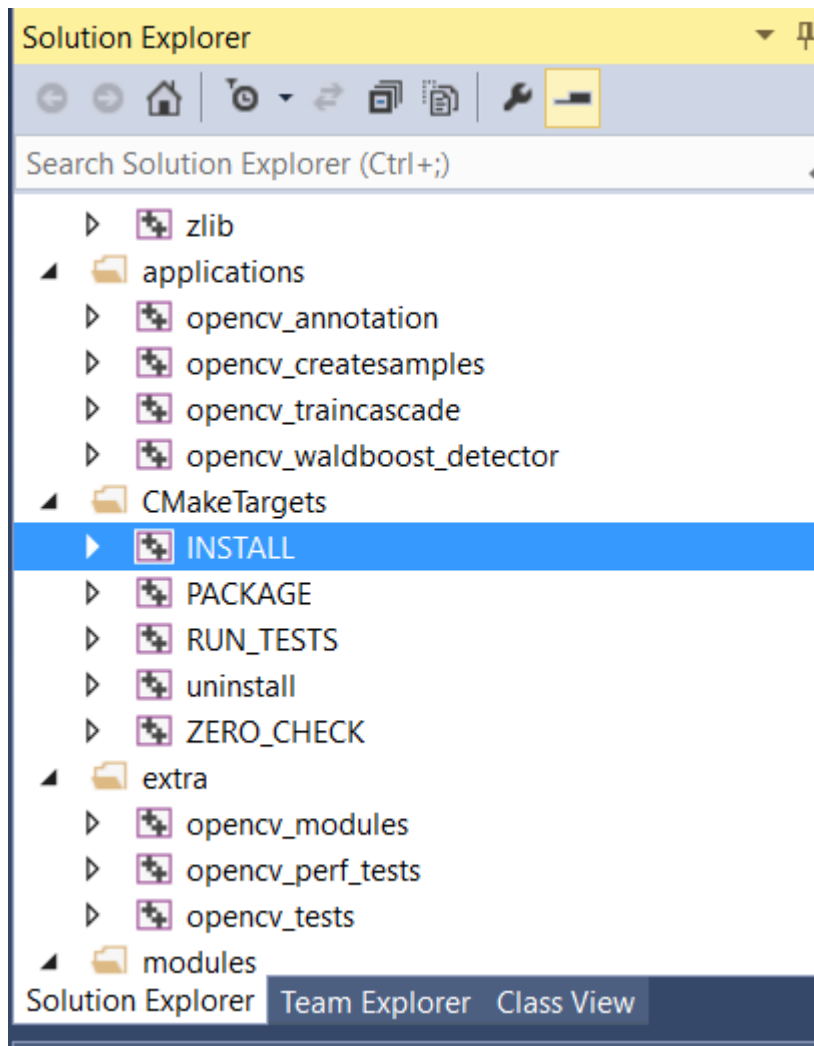


режим и выпуск



Режим.

- Теперь, в проводнике решений в правом верхнем углу вашей Visual Studio, выберите «УСТАНОВИТЬ проект» и постройте его.







Ура!! Наслаждайтесь OpenCV.

Добавление каталога OpenCV в переменную PATH переменных среды:

- Перейдите в раздел «Свойства системы» и нажмите «Дополнительные параметры системы».

Control Panel Home

-  Device Manager
-  Remote settings
-  System protection
-  [Advanced system settings](#)



View basic information about your computer

Windows edition

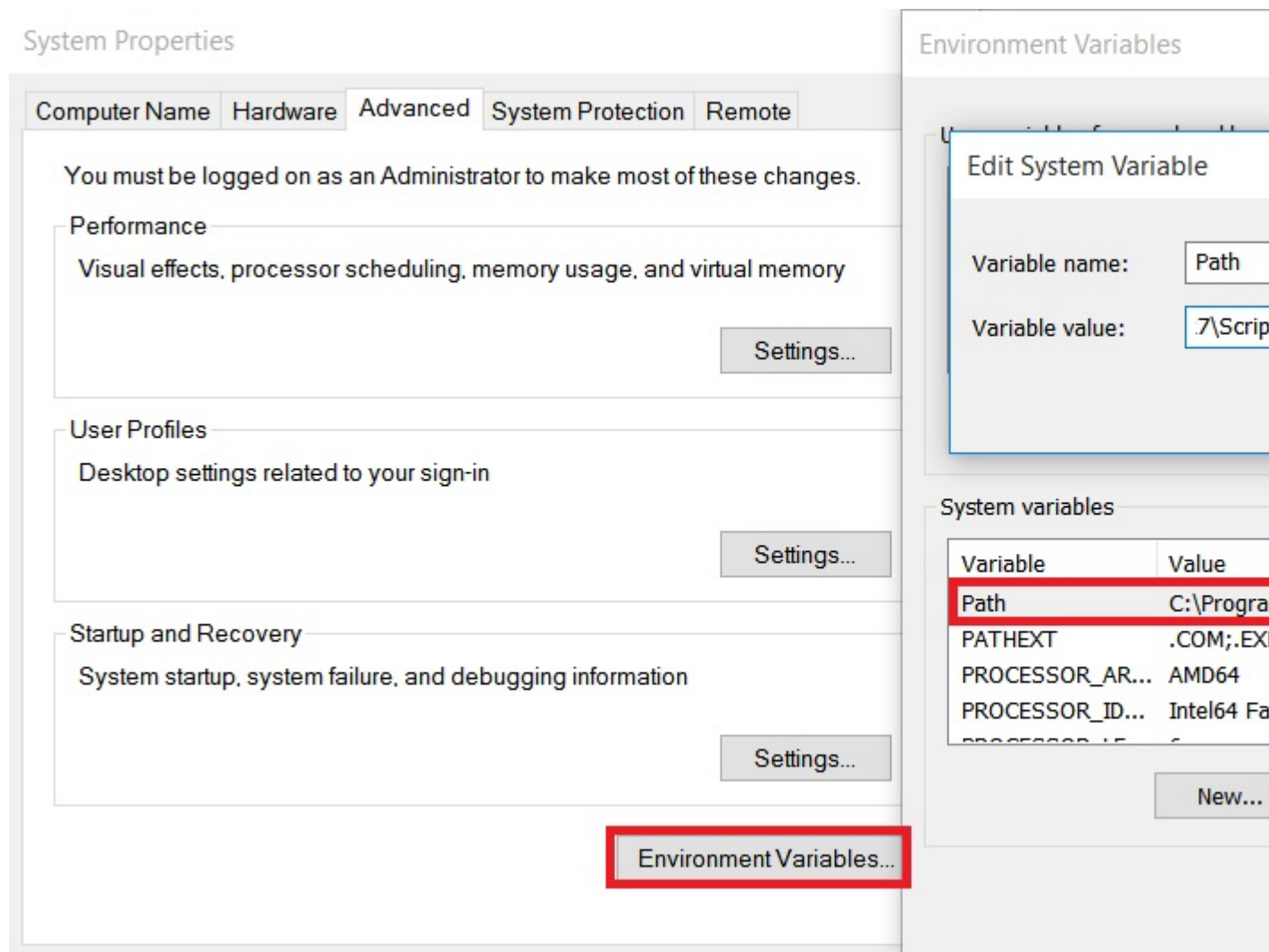
Windows 10 Home Single Language

© 2015 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i5-4210U CPU (
Installed memory (RAM):	6.00 GB (5.89 GB usable)
System type:	64-bit Operating System, x64-ba
Pen and Touch:	No Pen or Touch Input is availabl

- Теперь нажмите «Переменные среды» >> Путь >> Изменить.



- Здесь добавьте папку bin, расположенную в вашей OpenCVdir / build / install / x86 / vc ** / bin, к этой переменной. Будьте осторожны, чтобы не заменить существующие значения Path.
- После этого вам нужно перезапустить систему для изменения переменных среды, и теперь вы готовы к работе.

Что и почему OPENCV?

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку программного обеспечения для компьютерного зрения с открытым исходным кодом и компьютерного обучения. Он был построен для различных целей, таких как машинное обучение, компьютерное зрение, алгоритм, математические операции, захват видео, обработка изображений и т. Д. На протяжении многих лет он стал очень популярен среди исследователей и разработчиков как для поддержки на разных платформах (Windows, Linux, android, ios). Также он имеет обертку на разных известных языках программирования. В соответствии с лицензионным соглашением у него есть доступ к компаниям для использования и изменения кода.

Библиотека содержит более 2500 оптимизированных алгоритмов, которые имеют отличную

точность в производительности и скорости. Эти алгоритмы могут использоваться для обнаружения и распознавания лиц, идентификации объектов, классификации действий человека в видео, отслеживания движения камеры, отслеживания движущихся объектов, извлечения 3D-моделей объектов, создания 3D-облаков точек из стереокамер, сшивания изображений вместе для получения высокого разрешения изображение всей сцены, найти похожие изображения из базы данных изображений, удалять красные глаза с изображений, сделанных с помощью вспышки, следить за движениями глаз, распознавать декорации и устанавливать маркеры, чтобы накладывать их на дополненную реальность и т. д. В OpenCV есть отличные люди и сообщество, разработчиками и исследователями, их число составляет более 47 тысяч, а количество загрузок превышает 7 миллионов.

Библиотека обширна в профессиональных компаниях, исследовательских группах и других группах.

Многие известные компании, такие как Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, используют библиотеку, есть много стартапов, таких как Applied Minds, VideoSurf и Zeitera, которые широко используют OpenCV. Развернутые применения OpenCV охватывают диапазон от сшивания изображений уличного обзора вместе, обнаруживая вторжения в видео наблюдения в Израиле, контролируя оборудование шахт в Китае, помогая роботам перемещаться и собирать объекты в Willow Garage, обнаруживать несчастные случаи в бассейнах в Европе, запускать интерактивное искусство в Испания и Нью-Йорк, проверяя взлетно-посадочные полосы для мусора в Турции, проверяя этикетки на продуктах на заводах по всему миру, чтобы быстро обнаружить лицо в Японии. Он имеет интерфейсы C ++, C, Python, Java и MATLAB и поддерживает Windows, Linux, Android и Mac OS. OpenCV ориентируется в основном на приложения в режиме реального времени и использует преимущества команд MMX и SSE, когда они доступны. В настоящее время активно развиваются полнофункциональные интерфейсы CUDA и OpenCL. Существует более 500 алгоритмов и около 10-кратного числа функций, которые составляют или поддерживают эти алгоритмы. OpenCV написан на языке C ++ и имеет шаблонный интерфейс, который работает без проблем с контейнерами STL.

Информация, собранная с [официального сайта](#)

Прочитайте Начало работы с opencv онлайн: <https://riptutorial.com/ru/opencv/topic/800/начало-работы-с-opencv>

глава 2: Доступ к пикселям

замечания

Будьте осторожны, чтобы знать тип `cv::Mat` вы имеете дело. Например, если у вас есть `cv::Mat` типа `CV_8UC3`, но получите доступ к нему с помощью `image.at<uchar>(r,c)` ошибка не произойдет, но ваша программа будет иметь некоторое неожиданное поведение.

Examples

Доступ к отдельным значениям пикселей с помощью `cv::Mat::at()`

Чтобы получить доступ к значениям пикселей в объекте `cv::Mat` OpenCV, вам сначала нужно узнать *тип* вашей матрицы.

Наиболее распространенными типами являются:

- `CV_8UC1` для 8-битных одноканальных изображений в оттенках серого;
- `CV_32FC1` для 32-битных изображений с `CV_32FC1` 1-канальным полутоном;
- `CV_8UC3` для 8-битных трехканальных цветных изображений; а также
- `CV_32FC3` для 32-битных `CV_32FC3` цветных изображений с плавающей запятой.

Значение по умолчанию `cv::imread` создаст матрицу `CV_8UC3`.

Для доступа к отдельным пикселям самым безопасным способом, хотя и не самым эффективным, является использование метода `cv::Mat::at<T>(r,c)` где *r* - строка матрицы, а *c* - столбец. Аргумент шаблона зависит от типа матрицы.

Скажем, у вас есть изображение `cv::Mat image`. Согласно его типу, метод доступа и тип цвета пикселей будут разными.

- Для `CV_8UC1`: `uchar pixelGrayValue = image.at<uchar>(r,c)`.
- Для `CV_8UC3`: `cv::Vec3b pixelColor = image.at<cv::Vec3b>(r,c)`. Объект `cv::Vec3b` представляет собой триплет значений `uchar` (целые числа от 0 до 255).
- Для `CV_32FC1`: `float pixelGrayValue = image.at<float>(r,c)`.
- Для `CV_32FC3`: `cv::Vec3f pixelColor = image.at<cv::Vec3f>(r,c)`. Объект `cv::Vec3f` представляет собой триплет значений `float`.

Обратите внимание, что OpenCV представляет изображения в **строчном** порядке, например, например, Matlab или как соглашение в алгебре. Таким образом, если ваши координаты пикселей (x,y) , то вы получите доступ к пикселю, используя `image.at<...>(y,x)`.

Кроме того, `at<>` также поддерживается доступ через один аргумент `cv::Point`.

В этом случае доступ осуществляется в *столбце* :

```
image.at<..>(cv::Point(x,y));
```

Взгляните на [документацию OpenCV](#) для получения более подробной информации об этом методе.

Эффективный доступ к пикселям с использованием `cv::Mat::ptr` указатель

Если эффективность важна, быстрый способ итерации по пикселям в объекте `cv::Mat` заключается в использовании его метода `ptr<T>(int r)` для получения указателя на начало строки `r` (индекс на основе 0).

В соответствии с типом матрицы указатель будет иметь другой шаблон.

- Для `CV_8UC1` : `uchar* ptr = image.ptr<uchar>(r);`
- Для `CV_8UC3` : `cv::Vec3b* ptr = image.ptr<cv::Vec3b>(r);`
- Для `CV_32FC1` : `float* ptr = image.ptr<float>(r);`
- Для `CV_32FC3` : `cv::Vec3f* ptr = image.ptr<cv::Vec3f>(r);`

Этот объект `ptr` можно затем использовать для доступа к значению пикселя в строке `r` и столбце `c` , вызывая `ptr[c]` .

Чтобы проиллюстрировать это, вот пример, где мы загружаем изображение с диска и инвертируем его синие и красные каналы, работая пиксель за пикселем:

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

int main(int argc, char** argv) {
    cv::Mat image = cv::imread("image.jpg", CV_LOAD_IMAGE_COLOR);

    if(!image.data) {
        std::cout << "Error: the image wasn't correctly loaded." << std::endl;
        return -1;
    }

    // We iterate over all pixels of the image
    for(int r = 0; r < image.rows; r++) {
        // We obtain a pointer to the beginning of row r
        cv::Vec3b* ptr = image.ptr<cv::Vec3b>(r);

        for(int c = 0; c < image.cols; c++) {
            // We invert the blue and red values of the pixel
            ptr[c] = cv::Vec3b(ptr[c][2], ptr[c][1], ptr[c][0]);
        }
    }

    cv::imshow("Inverted Image", image);
    cv::waitKey();
}
```



```
    return 0;
}
```

Настройка и получение значений пикселей серого изображения в C ++

```
// PixelAccessTutorial.cpp : Defines the entry point for the console
// Environment: Visual studio 2015, Windows 10
// Assumptions: Opecv is installed configured in the visual studio project
// Opencv version: OpenCV 3.1

#include "stdafx.h"
#include<opencv2/core/core.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<string>
#include<iostream>

int main()
{
    cv::Mat imgOriginal;          // input image
    cv::Mat imgGrayscale;         // grayscale of input image

    std::cout << "Please enter an image filename : ";
    std::string img_addr;
    std::cin >> img_addr;

    std::cout << "Searching for " + img_addr << std::endl;

    imgOriginal = cv::imread(img_addr);          // open image

    if (imgOriginal.empty()) {                    // if unable to open
image                                     // if unable to open
        std::cout << "error: image not read from file\n\n";        // show error message
on command line
        return(0);                                // and exit program
    }

    cv::cvtColor(imgOriginal, imgGrayscale, CV_BGR2GRAY);        // convert to grayscale

    const int channels = imgGrayscale.channels();
    printf("Number of channels = %d", channels);

    cv::Mat output ;
    imgGrayscale.copyTo(output); // Just to make sure the Mat objects are of the same size.

    //Set the threshhold to your desired value
    uchar threshold = 127;

    if (channels == 1)
    {
        for (int x = 0; x<imgGrayscale.rows; x++) {
            for (int y = 0; y<imgGrayscale.cols; y++) {
                // Accesssing values of each pixel
                if (imgGrayscale.at<uchar>(x, y) >= threshold) {
                    // Setting the pixel values to 255 if it's above the threshold
                    output.at<uchar>(x, y) = 254;
                }
            }
        }
    }
}
```



```

        else if (imgGrayscale.at<uchar>(x, y) < threshold) {
            // Setting the pixel values to 255 if it's below the threshold
            output.at<uchar>(x, y) = 0;
        }
        else {
            // Just in case
            printf("The value at (%d, %d) are not right. Value: %d\n", x, y,
imgGrayscale.at<uchar>(x, y));
        }
    }
}
}
else if (channels == 3)
{
    // This is only for gray scale images
    printf("\tThe image has 3 channels. The function does not support images with 3
channels.\n");
}

//Create windows to show image
cv::namedWindow("Gray scale", CV_WINDOW_AUTOSIZE);
cv::namedWindow("Binary", CV_WINDOW_AUTOSIZE);

cv::imshow("Gray scale", imgGrayscale);
cv::imshow("Binary", output);

cv::waitKey(0); // hold windows open until user presses a key

return 0;
}

```

Альтернативный доступ к пикселям с помощью Matiterator

Это не лучший способ итерации через пиксели; однако это лучше, чем `cv :: Mat :: at <T>`.

Предположим, у вас есть цветное изображение в вашей папке, и вы хотите перебирать все пиксели этого изображения и стирать зеленые и красные каналы (обратите внимание, что это пример, вы можете сделать это более оптимизированными способами);

```

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

int main(int argc, char **argv)
{
    // Create a container
    cv::Mat im;

    //Create a vector
    cv::Vec3b *vec;

    // Create an mat iterator
    cv::MatIterator_<cv::Vec3b> it;

    // Read the image in color format
    im = cv::imread("orig1.jpg", 1);
}

```

```

// iterate through each pixel
for(it = im.begin<cv::Vec3b>(); it != im.end<cv::Vec3b>(); ++it)
{
    // Erase the green and red channels
    (*it)[1] = 0;
    (*it)[2] = 0;
}

// Create a new window
cv::namedWindow("Resulting Image");

// Show the image
cv::imshow("Resulting Image", im);

// Wait for a key
cv::waitKey(0);

return 0;
}

```

Чтобы скомпилировать это с помощью Cmake:

```

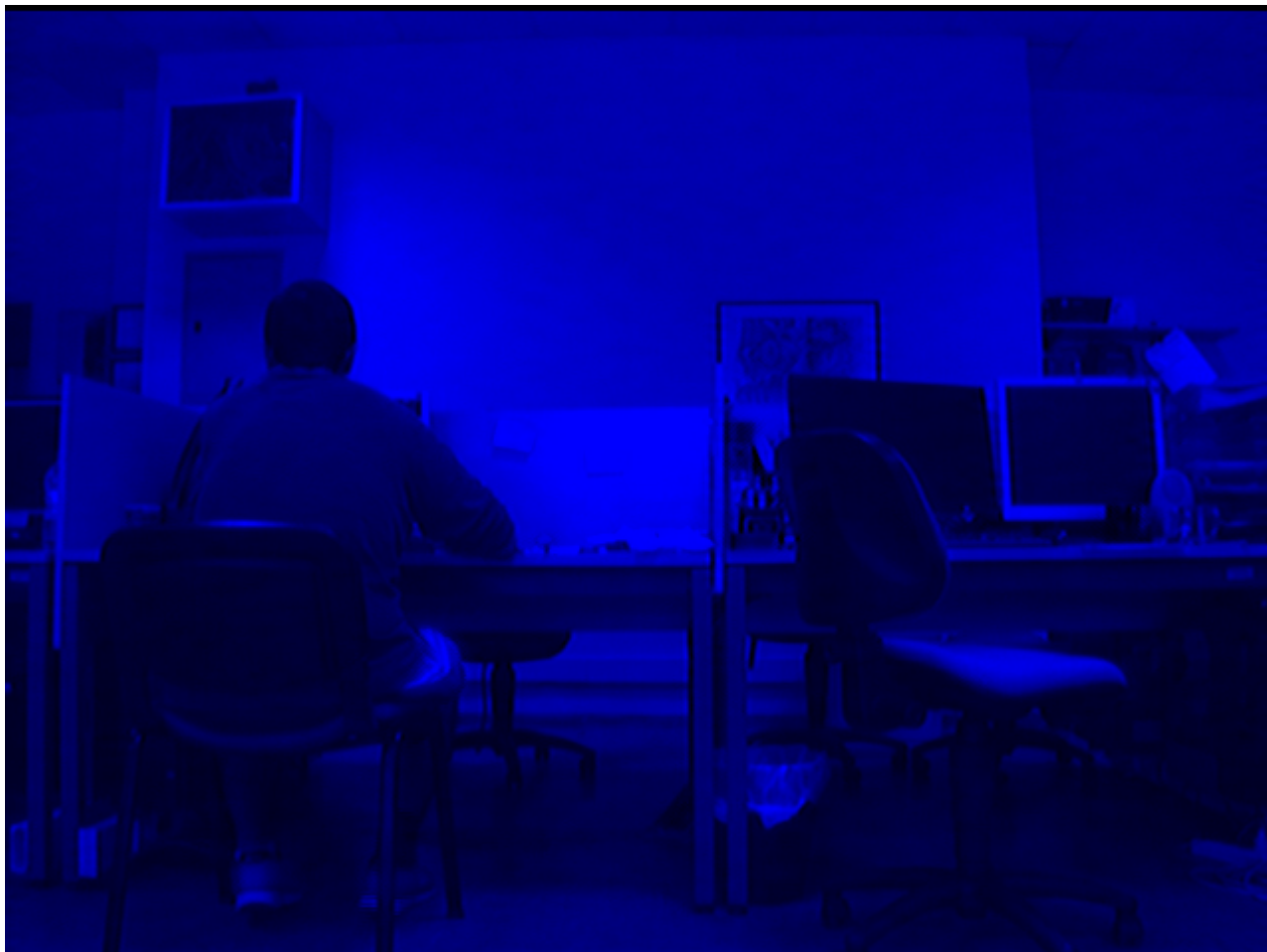
cmake_minimum_required(VERSION 2.8)
project(Main)
find_package(OpenCV REQUIRED)
add_executable(Main main.cpp)
target_link_libraries(Main ${OpenCV_LIBS})

```

Исходное изображение:



Обработанное изображение:



Обратите внимание: мы не трогаем только синий канал.

Для получения дополнительной информации: http://docs.opencv.org/2.4/opencv_tutorials.pdf

Страница: 145

Доступ к пикселям в матрице

Индивидуальный доступ к пикселям в структуре Matrix OpenCV может быть достигнут несколькими способами. Чтобы понять, как получить доступ, лучше сначала изучить типы данных.

Основные структуры объясняют основные типы данных. `CV_<bit-`

`depth>{U|S|F}C(<number_of_channels>)` , `CV_<bit-depth>{U|S|F}C(<number_of_channels>)` является базовой структурой типа. Наряду с этим важно понимать структуры `Vec` .

```
typedef Vec<type, channels> Vec< channels>< one char for the type>
```

где `type` - один из `uchar`, `short`, `int`, `float`, `double` а символы для каждого типа - `b`, `s`, `i`, `f`, `d` соответственно.

Например, `Vec2b` указывает `unsigned char vector of 2 channels` .

Рассмотрим `Mat mat(R,C,T)` где `R` является `#rows`, `C` является `#cols` и `T` является типом.

Некоторые примеры для доступа к координате $(i, j)_{mat}$:

2D:

```
If the type is CV_8U or CV_8UC1 ---- //they are alias
mat.at<uchar>(i,j) // --> This will give char value of index (i,j)
//If you want to obtain int value of it
(int)mat.at<uchar>(i,j)

If the type is CV_32F or CV_32FC1 ---- //they are alias
mat.at<float>(i,j) // --> This will give float value of index (i,j)
```

3D:

```
If the type is CV_8UC2 or CV_8UC3 or more channels
mat.at<Vec2b/Vec3b>(i,j)[k] // note that (k < #channels)
//If you want to obtain int value of it
(int)mat.at<uchar>(i,j)[k]

If the type is CV_64FC2 or CV_64FC3
mat.at<Vec2d/Vec3d>(i,j)[k] // note that k < #channels
```

Обратите внимание, что очень важно ввести правильный тип в $\langle \dots \rangle$, в противном случае вы можете иметь ошибку времени выполнения или нежелательные результаты.

Прочитайте Доступ к пикселям онлайн: <https://riptutorial.com/ru/opencv/topic/1957/доступ-к-пикселям>

глава 3: Загрузка и сохранение различных форматов мультимедиа

Examples

Загрузка изображений

```
#include <highgui.h>

//...

cv::Mat img = cv::imread("img.jpg");
```

...

Загрузка видео

Покажите, как использовать `cv::VideoCapture` . Вот пример загрузки видео из файла:

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

using namespace cv;
VideoCapture videoSource;
Mat frame;
#define VIDEO_PATH "video.avi"

int main()
{
    //Open video
    if (!videoSource.open(VIDEO_PATH))
    {
        std::cout<<"Video not found at "<<VIDEO_PATH<<std::endl;
        return 1;    // Exit if fail
    }
    videoSource.set(CV_CAP_PROP_CONVERT_RGB, 1);

    int cameraWidth = videoSource.get(CV_CAP_PROP_FRAME_WIDTH);
    int cameraHeight = videoSource.get(CV_CAP_PROP_FRAME_HEIGHT);
    float cameraAspectRatio = cameraWidth / cameraHeight;

    std::cout <<"Camera resolution: " << cameraWidth<<", "<<cameraHeight<<" aspect ratio:
"<<cameraAspectRatio<< std::endl;

    while(true)
    {
        videoSource >> frame;
        if(frame.empty())
            break;
        //Resize frame
```

```

        cv::resize(frame, frame, cv::Size(320, 320 / cameraAspectRatio));
        imshow("frame", frame);
        waitKey(20);
    }
    waitKey(0);
    return 0;
}

```

Live Capture

Покажите, как использовать `cv::VideoCapture` с помощью веб-камеры. Захват кадров с веб-камеры и отображение его. Вот пример кода:

```

#include <iostream>

#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/core/core.hpp"

using namespace cv;
VideoCapture videoSource;
Mat frame;

int main()
{
    if(!videoSource.open(0)) //if more cameras available use 1,2,...
        return 1;

    while(true)
    {
        videoSource >> frame;
        if(frame.empty())
            break;
        imshow("Webcam", frame); //or any kind of precessing
        if(waitKey(1)==27)
            break;//stop capturing is ESC pressed
    }

    return 0;
}

```

Сохранение видео

Покажите, как использовать `cv::VideoWriter`.

```

#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main(int argc, char* argv[])
{
    VideoCapture cap(0); // open the video camera no. 0

    if (!cap.isOpened()) // if not success, exit program

```

```

{
    cout << "ERROR: Cannot open the video file" << endl;
    return -1;
}

namedWindow("MyVideo", CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"

double dWidth = cap.get(CV_CAP_PROP_FRAME_WIDTH); //get the width of frames of the video
double dHeight = cap.get(CV_CAP_PROP_FRAME_HEIGHT); //get the height of frames of the
video

cout << "Frame Size = " << dWidth << "x" << dHeight << endl;

Size frameSize(static_cast<int>(dWidth), static_cast<int>(dHeight));

VideoWriter oVideoWriter ("D:/MyVideo.avi", CV_FOURCC('P','I','M','l'), 20, frameSize,
true); //initialize the VideoWriter object

if ( !oVideoWriter.isOpened() ) //if not initialize the VideoWriter successfully, exit the
program
{
    cout << "ERROR: Failed to write the video" << endl;
    return -1;
}

while (1)
{

    Mat frame;

    bool bSuccess = cap.read(frame); // read a new frame from video

    if (!bSuccess) //if not success, break loop
    {
        cout << "ERROR: Cannot read a frame from video file" << endl;
        break;
    }

    oVideoWriter.write(frame); //writer the frame into the file

    imshow("MyVideo", frame); //show the frame in "MyVideo" window

    if (waitKey(10) == 27) //wait for 'esc' key press for 30ms. If 'esc' key is pressed, break
loop
    {
        cout << "esc key is pressed by user" << endl;
        break;
    }
}

return 0;

}

```

Сохранение изображений

Фактически, пример Live Capture хорош для захвата изображений, поэтому я использую его для захвата изображений и сохранения их в папке.


```

#include <fstream>
#include <string>

#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>

int main()
{
    std::stringstream file; // to write the file name

    cv::VideoCapture cap(0); // create a capture object

    int counter = 0; // Create counter

    while(true) // infinite loop
    {
        cv::Mat frame; // Create a object

        cap.read(frame); // read the frame

        file << "/home/user/path_to_your_folder/image" << counter << ".jpg"; // file name

        cv::imwrite(file.str(), frame);

        counter++; // increment the counter
    }

    return 0;
}

```

Прочитайте [Загрузка и сохранение различных форматов мультимедиа онлайн](https://riptutorial.com/ru/opencv/topic/6658/загрузка-и-сохранение-различных-форматов-мультимедиа-онлайн):

<https://riptutorial.com/ru/opencv/topic/6658/загрузка-и-сохранение-различных-форматов-мультимедиа>

глава 4: Инициализация OpenCV в Android

Examples

Асинхронная инициализация

Использование асинхронной инициализации является рекомендуемым способом разработки приложений. Он использует [OpenCV Manager](#) для доступа к библиотекам OpenCV, установленным извне в целевой системе.

Фрагмент кода, реализующий инициализацию async:

```
public class MainActivity extends Activity implements CvCameraViewListener2 {

    private BaseLoaderCallback mLoaderCallback = new BaseLoaderCallback(this) {
        @Override
        public void onManagerConnected(int status) {
            switch(status) {
                case LoaderCallbackInterface.SUCCESS:
                    Log.i(TAG, "OpenCV Manager Connected");
                    //from now onwards, you can use OpenCV API
                    Mat m = new Mat(5, 10, CvType.CV_8UC1, new Scalar(0));
                    break;
                case LoaderCallbackInterface.INIT_FAILED:
                    Log.i(TAG, "Init Failed");
                    break;
                case LoaderCallbackInterface.INSTALL_CANCELED:
                    Log.i(TAG, "Install Cancelled");
                    break;
                case LoaderCallbackInterface.INCOMPATIBLE_MANAGER_VERSION:
                    Log.i(TAG, "Incompatible Version");
                    break;
                case LoaderCallbackInterface.MARKET_ERROR:
                    Log.i(TAG, "Market Error");
                    break;
                default:
                    Log.i(TAG, "OpenCV Manager Install");
                    super.onManagerConnected(status);
                    break;
            }
        }
    };

    @Override
    public void onResume() {
        super.onResume();
        OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_1_0, this, mLoaderCallback);
    }

    ...
}
```

В этом случае наше приложение работает с OpenCV Manager в асинхронном режиме.

`OnManagerConnected` обратный вызов будет вызываться в потоке пользовательского интерфейса, когда инициализация завершается.

Обратите внимание, что до вызова этого обратного вызова запрещается использовать вызовы OpenCV или загружать зависящие от OpenCV собственные библиотеки. Загрузите собственные библиотеки, которые зависят от OpenCV после успешной инициализации OpenCV.

По умолчанию `BaseLoaderCallback` реализует контекст приложения как `Activity` и вызывает метод `Activity.finish()` для выхода в случае отказа инициализации. Чтобы переопределить это поведение, вам необходимо переопределить метод `finish()` класса `BaseLoaderCallback` и реализовать свой собственный метод финализации.

OpenCV Manager

OpenCV Manager - это служба Android, предназначенная для управления двоичными файлами библиотеки OpenCV на устройствах конечных пользователей. Он позволяет совместно использовать динамические библиотеки OpenCV между приложениями на одном устройстве.

Менеджер предоставляет следующие преимущества:

- Меньше использования памяти (около 40 МБ). Все приложения используют одни и те же двоичные файлы из службы и не содержат внутри себя собственные библиотеки.
- Оптимизация оборудования для всех поддерживаемых платформ.
- Надежный источник библиотеки OpenCV. Все пакеты с OpenCV публикуются на рынке Google Play.
- Регулярные обновления и исправления ошибок.

Единственным недостатком является то, что пользователю предлагается загрузить и дополнительное приложение, поэтому пользовательский интерфейс немного уменьшается.

Дополнительная информация: [Android OpenCV Manager](#)

Обновлено 18/10/16:

В версии OpenCV Manager, распространяемой в [Play Store](#), есть ошибка (обновлено 21/09/15).

Это влияет только на версию OpenCV 3.1.0. Когда вы запускаете некоторые функции OpenCV, вы получаете ошибку `SIGSEGV`. Версия, распространяемая с Android SDK, отлично работает (`opencv-android-sdk/apk/OpenCV_3.1.0_Manager_3.10_{platform}.apk`). Его можно загрузить с [сайта OpenCV](#) .

Дополнительная информация: [выпуск № 6247](#) .

Статическая инициализация

Согласно этому подходу все файлы OpenCV включены в ваш пакет приложений. Он предназначен в основном для разработки и отладки. Этот подход **устарел** для производственного кода, рекомендуется инициализация асинхронного режима.

Если ваш проект приложения не имеет части JNI, просто скопируйте соответствующие родные библиотеки OpenCV из `OpenCV-3.1.0-android-sdk/sdk/native/libs` в каталог вашего проекта в папку `app/src/main/jniLibs`.

В случае проекта приложения с частью JNI вместо ручного копирования библиотек вам необходимо изменить файл `Android.mk`: добавьте следующие две строки кода после

`"include $(CLEAR_VARS)"` и до `"include path_to_OpenCV-3.1.0-android-sdk/sdk/native/jni/OpenCV.mk"`:

```
OPENCV_CAMERA_MODULES:=on
OPENCV_INSTALL_MODULES:=on
```

Результат должен выглядеть следующим образом:

```
include $(CLEAR_VARS)
# OpenCV
OPENCV_CAMERA_MODULES:=on
OPENCV_INSTALL_MODULES:=on
include ../../sdk/native/jni/OpenCV.mk
```

После этого библиотеки OpenCV будут скопированы в папку `jniLibs` вашего приложения во время сборки JNI.

Последним шагом включения OpenCV в ваше приложение является код инициализации Java перед вызовом API OpenCV. Это можно сделать, например, в статическом разделе класса `Activity`:

```
static {
    if (!OpenCVLoader.initDebug()) {
        // Handle initialization error
    }
}
```

Если приложение включает другие родственные библиотеки, зависящие от OpenCV, вы должны загрузить их после инициализации OpenCV:

```
static {
    if (!OpenCVLoader.initDebug()) {
        // Handle initialization error
    } else {
        System.loadLibrary("my_jni_lib1");
        System.loadLibrary("my_jni_lib2");
    }
}
```

Примечание. `initDebug()` устарел для производственного кода. Он предназначен только для экспериментального и локального развития. Если вы хотите опубликовать свой подход к использованию приложения с инициализацией `async`.

Прочитайте Инициализация OpenCV в Android онлайн:

<https://riptutorial.com/ru/opencv/topic/7545/инициализация-opencv-в-android>

глава 5: Использование VideoCapture с OpenCV Python

Examples

Чтение кадров из предварительно захваченного видео



```
import numpy as np
import cv2

#access a video from your disk
#to use the GIF in this example, convert to avi!
cap = cv2.VideoCapture('eg_videoRead.avi')

#we are going to read 10 frames
#we store the frames in a numpy structure
#then we'll generate a minimum projection of those frames

frameStack=[]
numFrames=10

for fr in range(numFrames):
    cap.set(cv2.CAP_PROP_POS_FRAMES,fr) #specifies which frame to read next
    frame=cap.read() #read the frame
    #gray = cv2.cvtColor(frame[1], cv2.COLOR_BGR2GRAY) #convert to gray scale
    frameStack.append(frame[1]) #add current frame to our frame Stack

minProjection=np.min(frameStack,axis=0) #find the minimum across frames
cv2.imshow("projection", minProjection) #show the result
```

Использование VideoCapture с OpenCV Java

В java нет imshow, вам нужно написать метод для этого. Этот метод является Mat2bufferedImage. Принимает объект mat в качестве параметра и возвращает изображение.

```
public static void main(String[] args) {
    Mat frame = new Mat();
    //0; default video device id
```

```
VideoCapture camera = new VideoCapture(0);
JFrame jframe = new JFrame("Title");
jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel vidpanel = new JLabel();
jframe.setContentPane(vidpanel);
jframe.setVisible(true);

while (true) {
    if (camera.read(frame)) {

        ImageIcon image = new ImageIcon(Mat2bufferedImage(frame));
        vidpanel.setIcon(image);
        vidpanel.repaint();

    }
}
```

Прочитайте Использование VideoCapture с OpenCV Python онлайн:

<https://riptutorial.com/ru/opencv/topic/6803/использование-videocapture-с-opencv-python>

глава 6: Использование каскадных классификаторов в Java

Синтаксис

- `CascadeClassifier cascade = новый CascadeClassifier ("cascade.xml");` // Создает каскадный классификатор из `cascade.xml`
- `Mat image = Imgcodecs.imread ("image.png");` // Преобразует `image.png` в объект `Mat` (`Matrix`)
- Определение `MatOfRect = новый MatOfRect ();` // Создает пустой файл `MatOfRect` (`Matrix of Rectangles`), используемый в качестве вывода для наших классов обнаружения
- `detections.toArray ();` // Возвращает массив объектов `Rect`, которые могут быть повторены
- `Imgproc.rectangle (изображение, новая точка (rect.x, rect.y), новая точка (rect.x + rect.width, rect.y + rect.height), новый Scalar (0, 255, 0));` // Рисует зеленый очерченный прямоугольник из местоположений `x` и `y` первой точки в положение `x` и `y` второй точки на объект «Изображение» объекта `Mat`. «`rect`» - это объект `Rect`, обычно предоставляемый функциями `detections.toArray ()`. Использует класс `Point` `OpenCV`.
- `Imgcodecs.imwrite ("output.png", изображение);` // Записывает измененный объект «объект изображения» `Mat` в «`output.png`»
- `CascadeClassifier.detectMultiScale (изображение, обнаружение);` // Обнаруживает любой объект в объекте «изображение» объекта `Mat` и выводит обнаруженные в объекте `MatOfRect` «обнаружения»,
- `CascadeClassifier.detectMultiScale (изображение, обнаружение, scaleFactor , minNeighbors , flags , minSize , maxSize);` // Выполняет обнаружение с дополнительными параметрами. См. Подробности ниже.
- `Imgproc.ellipse (изображение, центр, оси , 0, 0, 360, новый Scalar (255, 0, 255), толщина , lineType , 0);` // Рисуем эллипс на изображении в `center` точки. Использует класс `Point` `OpenCV`.

параметры

параметр	подробности
масштаб	Насколько размер изображения уменьшается при каждой шкале изображения. По умолчанию = 1.1
minNeighbors	Сколько соседей должен иметь прямоугольник-кандидат, прежде чем выбирать его как обнаруженный объект. По умолчанию = 4

параметр	подробности
флаги	Устаревшие флаги. В большинстве случаев это должно быть установлено равным 0. По умолчанию = 0
MinSize	Минимальным размером может быть прямоугольник кандидата. Это использует класс <code>Size</code> OpenCV. Может использоваться для уменьшения времени обнаружения и использования ЦП, а также для уменьшения ложных срабатываний.
MaxSize	Максимальный размер прямоугольника-кандидата. Это использует класс <code>Size</code> OpenCV. Может использоваться для уменьшения времени обнаружения и использования ЦП, а также для уменьшения ложных срабатываний.
оси	Использует класс размера OpenCV. Определяет ширину и высоту эллипса.
толщина	Толщина линии, в пикселях.
LineType	Имеет различные параметры. 0 - сплошная линия, 8 - для 8-связной линии, 4 - для 4-связной линии, а <code>CV_AA</code> - для <code>CV_AA</code> линии. По умолчанию = 8

Examples

Получение статического изображения, определение элементов на нем и вывод результатов.

Обратите внимание, что в этом примере используется OpenCV 3.1.

```
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;

public class Classifier {
    private CascadeClassifier diceCascade = new
        CascadeClassifier("res/newMethod/diceCascade.xml");
    private Mat image;
    private String loc = "path/to/image.png";
    private String output = "path/to/output.png";

    public void detImg() {

        Mat image = Imgcodecs.imread(loc); // Reads the image
```

```

MatOfRect diceDetections = new MatOfRect(); // Output container
diceCascade.detectMultiScale(image, diceDetections); // Performs the detection

// Draw a bounding box around each detection.
for (Rect rect : diceDetections.toArray()) {
    Imgproc.rectangle(image, new Point(rect.x, rect.y),
        new Point(rect.x + rect.width, rect.y + rect.height),
        new Scalar(0, 255, 0));
}

// Save the visualized detection.
Imgcodecs.imwrite(output, image);
}
}

```

`Rect[]` возвращаемый `diceDetections.toArray()` может быть повторен. Каждый `Rect` внутри массива будет иметь четыре основных свойства: `x`, `y`, `width` и `height`. `x` и `y` определяет верхнее левое положение прямоугольника, а `width` и `height` возвращают `int` ширины и высоты прямоугольника. Это используется при рисовании прямоугольников на изображениях. Минимальные требуемые параметры функции `Imgproc.rectangle` следующие:

```

Imgproc.rectangle(Mat image, Point start, Point end, Scalar color);

```

Обе `Point` используются для позиций верхнего левого угла и нижнего правого угла. Эти позиции являются *абсолютными* по отношению к изображению, представленному в качестве первого параметра, а не друг к другу. Таким образом, вы должны добавить как `x` и `y` позицию прямоугольника в дополнение к `width` или `height` чтобы правильно определить `end` точку.

Обратите внимание, что класс `Point` используемый в этих параметрах, **не** является классом `Point` стандартной библиотеки Java. Вы должны импортировать класс `Point` OpenCV вместо этого!

Обнаружение изображений с видеоустройства

В этом примере представлен класс `VideoCapture`, где мы используем его для получения изображения с веб-камеры и сохранения его на изображении.

```

import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;
import org.opencv.videoio.VideoCapture;

public class Classifier {
    private CascadeClassifier diceCascade = new

```

```

        CascadeClassifier("res/newMethod/diceCascade.xml");
private Mat image;
private String loc = "path/to/image.png";
private String output = "path/to/output.png";
private VideoCapture vc = new VideoCapture();

public void detImg() {
    vc.open(0); // Opens the video stream

    Mat image = new Mat(); // Creates an empty matrix
    vc.read(image); // Reads the image from the video stream and
                    // writes it to the image matrix.

    MatOfRect diceDetections = new MatOfRect(); // Output container
    diceCascade.detectMultiScale(image, diceDetections); // Performs the detection

    // Draw a bounding box around each detection.
    for (Rect rect : diceDetections.toArray()) {
        Imgproc.rectangle(image, new Point(rect.x, rect.y),
            new Point(rect.x + rect.width, rect.y + rect.height),
            new Scalar(0, 255, 0));
    }

    // Save the visualized detection.
    Imgcodecs.imwrite(output, image);

    vc.release(); // Closes the stream.
}
}

```

Преобразование объекта Mat в объект BufferedImage

Этот пример Дэниела Баджо был взят непосредственно из [этого ответа StackExchange](#) , но был пересмотрен для видимости.

Этот класс принимает объект Mat и возвращает объект BufferedImage, используемый библиотеками `javax.swing` . Это может использоваться объектом `Graphics` для рисования изображения.

```

private BufferedImage toBufferedImage(Mat m) {
    if (!m.empty()) {
        int type = BufferedImage.TYPE_BYTE_GRAY;
        if (m.channels() > 1) {
            type = BufferedImage.TYPE_3BYTE_BGR;
        }
        int bufferSize = m.channels() * m.cols() * m.rows();
        byte[] b = new byte[bufferSize];
        m.get(0, 0, b); // get all the pixels
        BufferedImage image = new BufferedImage(m.cols(), m.rows(), type);
        final byte[] targetPixels = ((DataBufferByte)
            image.getRaster().getDataBuffer()).getData();
        System.arraycopy(b, 0, targetPixels, 0, b.length);
        return image;
    }

    return null;
}

```

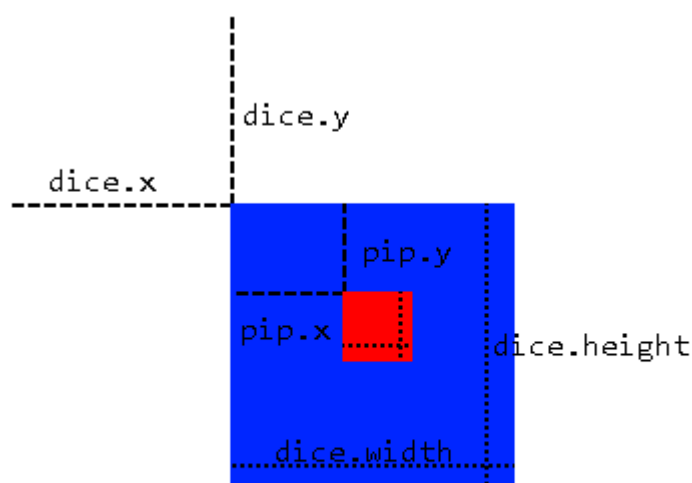
```
}
```

Обнаружения в пределах детектирования

Этот пример использует Dice и черные пятна на кости (пипсы) в качестве нашего объекта. Поскольку пример довольно длинный, сначала объяснение некоторых ключевых понятий имеет решающее значение для понимания примера.

Понимание первого примера: «Получение статического изображения, определение элементов на нем и вывод результатов». имеет решающее значение для понимания этого примера, особенно в том, как OpenCV рисует прямоугольники.

Взгляните на следующее изображение:



Мы будем использовать метод субимации, где мы используем обнаруженную область в качестве нашей базы для применения большего количества обнаружений. Это возможно только в том случае, если объект всегда будет находиться внутри другого объекта, который мы можем обнаружить, например, наши пипсы на наших игровых костях. Этот метод имеет несколько преимуществ:

- Вместо сканирования всего изображения нам нужно только сканировать область, в которой мы знаем, что объект будет находиться.
- Удаляет любую вероятность ложных срабатываний вне зоны обнаружения.

Мы делаем это, сначала применяя одно сканирование каскадного классификатора по всему изображению, чтобы дать нам объект `MatOfRect` содержащий наши большие объекты (в данном случае кости). Затем мы перебираем массив `Rect[]` заданный функцией `toArray()` из объекта `MatOfRect`. Этот объект `Rect` используется для создания временного объекта `Mat` который «обрезается» для свойств объекта `Rect (x, y, width, height)` из исходного изображения, где мы можем затем выполнять обнаружение на временном объекте `Mat`. Другими словами, мы говорим, что классификатор только выполняет обнаружение на

частях кости изображения, и мы указываем положение каждой кости, используя объекты `Rect` которые мы получили от выполнения обнаружения на всем изображении.

Однако объекты `Rect` (pips) имеют свои свойства относительно своих костей, а не самого изображения. Чтобы решить эту проблему, когда мы хотим нарисовать прямоугольники для фактического изображения, показывающего местоположения пипсов, мы добавляем оба значения `dice.x` и `dice.y` в начальную `Point`.

```
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;
import org.opencv.videoio.VideoCapture;

public class Classifier {

    private CascadeClassifier diceCascade =
        new CascadeClassifier("res/newMethod/diceCascade.xml");
    private CascadeClassifier pipCascade =
        new CascadeClassifier("res/newMethod/pipCascade6.xml");
    private VideoCapture vc = new VideoCapture();
    private Mat image;

    public void openVC(int index) {
        vc.open(index);
    }

    public void closeVC() {
        vc.close();
    }

    public Mat getNextImage() {
        image = new Mat();
        vc.read(image); // Sets the matrix to the current livestream frame.

        MatOfRect diceDetections = new MatOfRect(); // Output container

        // See syntax for explanations on addition parameters
        diceCascade.detectMultiScale(image, diceDetections, 1.1, 4, 0, new Size(20, 20),
            new Size(38, 38));

        // Iterates for every Dice ROI
        for (int i = 0; i < diceDetections.toArray().length; i++) {
            Rect diceRect = diceDetections.toArray()[i];

            // Draws rectangles around our detected ROI
            Point startingPoint = new Point(diceRect.x, diceRect.y);
            Point endingPoint = new Point(diceRect.x + diceRect.width,
                diceRect.y + diceRect.height);
            Imgproc.rectangle(image, startingPoint, endingPoint, new Scalar(255, 255, 0));

            MatOfRect pipDetections = new MatOfRect();
```

```

        pipCascade.detectMultiScale(image.submat(diceRect), pipDetections, 1.01, 4, 0,
            new Size(2, 2), new Size(10, 10));

        // Gets the number of detected pips and draws a cricle around the ROI
        for (int y = 0; y < pipDetections.toArray().length; y++) {
            // Provides the relative position of the pips to the dice ROI
            Rect pipRect = pipDetections.toArray()[y];

            // See syntax explanation
            // Draws a circle around our pips
            Point center = new Point(diceRect.x + pipRect.x + pipRect.width / 2,
                diceRect.y + pipRect.y + pipRect.height / 2);
            Imgproc.ellipse(image, center, new Size(pipRect.width / 2, pipRect.height /
2),
                0, 0, 360, new Scalar(255, 0, 255), 1, 0, 0);
        }
    }

    return image;
}
}

```

Функция `getNextImage()` возвращает объект `Mat`, который вместе с другими опубликованными примерами может быть вызван постоянно и может быть преобразован в `BufferImage`, чтобы обеспечить отображение живого изображения с обнаружением.

Прочитайте [Использование каскадных классификаторов в Java онлайн](https://riptutorial.com/ru/opencv/topic/6377/использование-каскадных-классификаторов-в-java):

<https://riptutorial.com/ru/opencv/topic/6377/использование-каскадных-классификаторов-в-java>

глава 7: Каскадные классификаторы

Examples

Использование каскадных классификаторов для обнаружения лица

ПИТОН

Код

```
import numpy as np
import cv2

#loading haarcascade classifiers for face and eye
#You can find these cascade classifiers here
#https://github.com/opencv/opencv/tree/master/data/haarcascades
#or where you download opencv inside data/haarcascades

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

#loading the image
img = cv2.imread('civil_war.jpg')

#converting the image to gray scale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#detecting face in the grayscale image
faces = face_cascade.detectMultiScale(gray, 1.3, 5)

#iterate through each detected face
for (x,y,w,h) in faces:
    cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0),2) #draw rectangle to each detected face

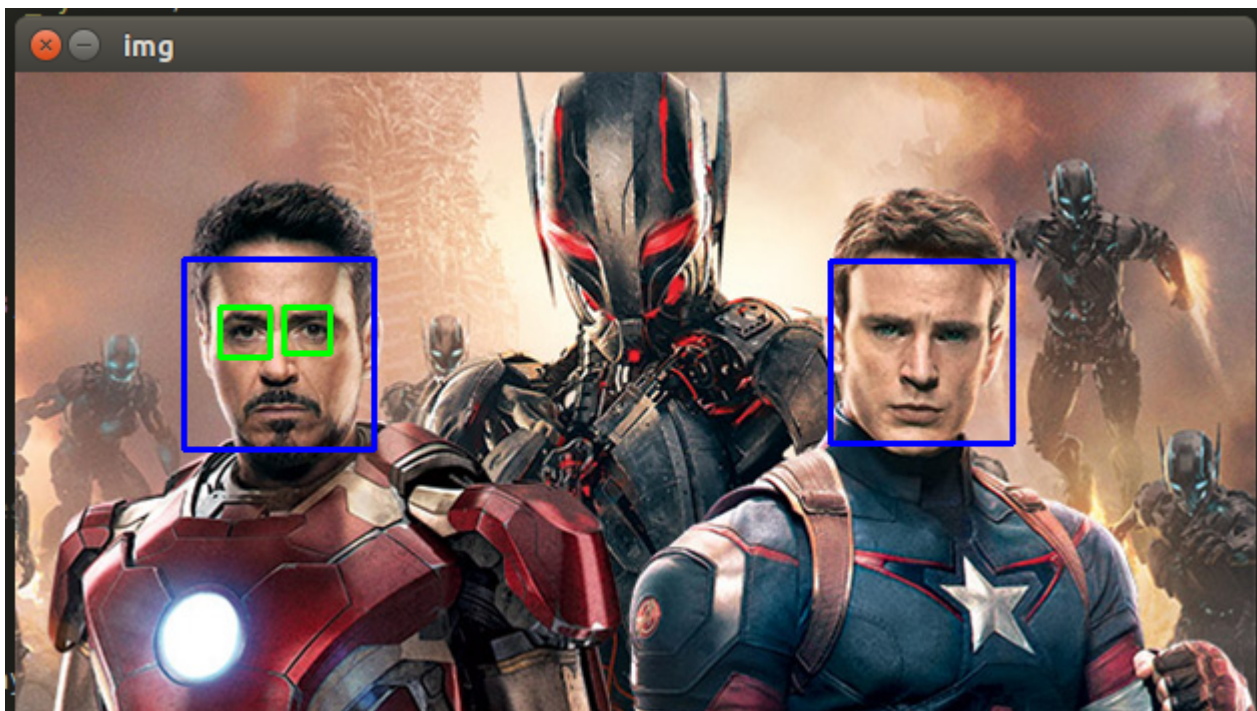
    #take the roi of the face (region of interest)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]

    #detect the eyes
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:

        #draw rectangle for each eye
        cv2.rectangle(roi_color, (ex,ey), (ex+ew,ey+eh), (0,255,0),2)

#show the image
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Результат



Классификаторы каскадов для обнаружения лица с помощью Java

Джава

Код

```
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.highgui.Highgui;
import org.opencv.highgui.VideoCapture;
import org.opencv.objdetect.CascadeClassifier;

public class FaceDetector{

    public static void main(String[] args) {

        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
        //Create object
        CascadeClassifier faceDetector = new
        CascadeClassifier(FaceDetector.class.getResource("haarcascade_frontalface_default.xml").getPath());

        //Read image
        Mat image = Highgui.imread("sourceimage.jpg");

        /*
        //Or read from webcam

        * Mat image=new Mat();
        * VideoCapture videoCapture=new VideoCapture(0);
        * videoCapture.read(image);
        */
    }
}
```



```

MatOfRect faceDetections = new MatOfRect();
//Result list
faceDetector.detectMultiScale(image, faceDetections);

for (Rect rect : faceDetections.toArray()) {
    //Draw rectangle on result

    Core.rectangle(image, new Point(rect.x, rect.y), new Point(rect.x + rect.width,
rect.y + rect.height),
        new Scalar(0, 255, 0));
}

//write result
Highgui.imwrite("result.png", image);
System.out.println("Succesfull");
}
}

```

Результат



Обнаружение лица с использованием хаскального каскадного классификатора

C ++

```

#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;

// Function Headers
void detectAndDisplay(Mat frame);

// Global variables
string face_cascade_name = "../data/haarcascade_frontalface_alt2.xml";
CascadeClassifier face_cascade;

```

```

// Function main
int main(void)
{
    // Load the cascade
    if (!face_cascade.load(face_cascade_name)){
        printf("--(!)Error on cascade loading\n");
        return (-1);
    }

    // Read the image file
    Mat frame = imread("d:/obama_01.jpg");

    // Apply the classifier to the frame
    if (!frame.empty())
        detectAndDisplay(frame);
    waitKey(0);
    return 0;
}

// Function detectAndDisplay
void detectAndDisplay(Mat frame)
{
    std::vector<Rect> faces;
    Mat frame_gray;

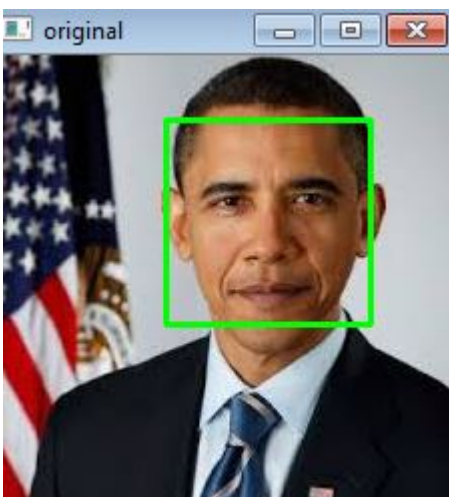
    cvtColor(frame, frame_gray, COLOR_BGR2GRAY);
    equalizeHist(frame_gray, frame_gray);

    // Detect faces
    face_cascade.detectMultiScale(frame_gray, faces, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, Size(30,
30));

    for (int ic = 0; ic < faces.size(); ic++) // Iterate through all current elements
(detected faces)
    {
        Point pt1(faces[ic].x, faces[ic].y); // Display detected faces on main window - live
stream from camera
        Point pt2((faces[ic].x + faces[ic].height), (faces[ic].y + faces[ic].width));
        rectangle(frame, pt1, pt2, Scalar(0, 255, 0), 2, 8, 0);
    }

    imshow("original", frame);
}

```



Прочитайте Каскадные классификаторы онлайн: <https://riptutorial.com/ru/opencv/topic/6562/каскадные-классификаторы>

глава 8: Контрастность и яркость в C ++

Синтаксис

- `void cv :: Mat :: convertTo (OutputArray m, int rtype, double alpha = 1, double beta = 0)`
`const`

параметры

параметр	подробности
м	выходная матрица; если он не имеет правильного размера или типа перед операцией, он перераспределяется
RTYPE	желаемый тип выходной матрицы или, точнее, глубина, так как количество каналов совпадает с количеством входных данных; если rtype отрицательный, выходная матрица будет иметь тот же тип, что и вход
альфа	дополнительный масштабный коэффициент. Это изменяет контрастность изображения. Значения ниже 1 уменьшают контраст, а выше один увеличивает контраст
бета	дополнительная величина добавлена к масштабированным значениям. Положительные значения увеличивают яркость, а отрицательные значения уменьшают яркость

замечания

Контраст :

Контрастность - это разница в яркости или цвете, которая делает объект (или его изображение на изображении или дисплее) различимым. Чем выше разница между пикселем и его соседями, тем выше контраст в этой области.

Яркость :

Другими словами, яркость - это восприятие, вызванное яркостью визуальной цели. С точки зрения пикселей, чем выше значение пикселя, тем ярче этот пиксель.

Корректировка контрастности и яркости:

$$g(i, j) = \alpha \cdot f(i, j) + \beta$$

$f(x)$ в качестве пикселей исходного изображения и $g(x)$ в качестве пикселей выходного изображения.

i и j указывает, что пиксель расположен в i -й строке и j -м столбце.

Параметры $\alpha > 0$ и β часто называются параметрами усиления и смещения; иногда эти параметры, как говорят, контролируют *контраст* и *яркость* соответственно.

У OpenCV есть функция, называемая `convertTo()`, которая может применять эти две операции.

Источники:

http://docs.opencv.org/trunk/d3/d63/classcv_1_1Mat.html#adf88c60c5b4980e05bb556080916978b
<http://opencv-srf.blogspot.ca/2013/07/change-contrast-of-image-or-video.html> <http://opencv-srf.blogspot.ca/2013/07/change-brightness.html>

Examples

Настройка яркости и контрастности изображения в C++

```
// main.cpp : Defines the entry point for the console application.
//
#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main(int argc, const char** argv)
{
    Mat img = imread("lena30.jpg", CV_LOAD_IMAGE_COLOR); //open and read the image

    if (img.empty())
    {
        cout << "Image cannot be loaded...!!" << endl;
        return -1;
    }

    Mat img_higher_contrast;
    img.convertTo(img_higher_contrast, -1, 2, 0); //increase the contrast (double)

    Mat img_lower_contrast;
    img.convertTo(img_lower_contrast, -1, 0.5, 0); //decrease the contrast (halve)

    Mat img_higher_brightness;
    img.convertTo(img_higher_brightness, -1, 1, 20); //increase the brightness by 20 for each
    pixel

    Mat img_lower_brightness;
    img.convertTo(img_lower_brightness, -1, 1, -20); //decrease the brightness by 20 for each
    pixel
```

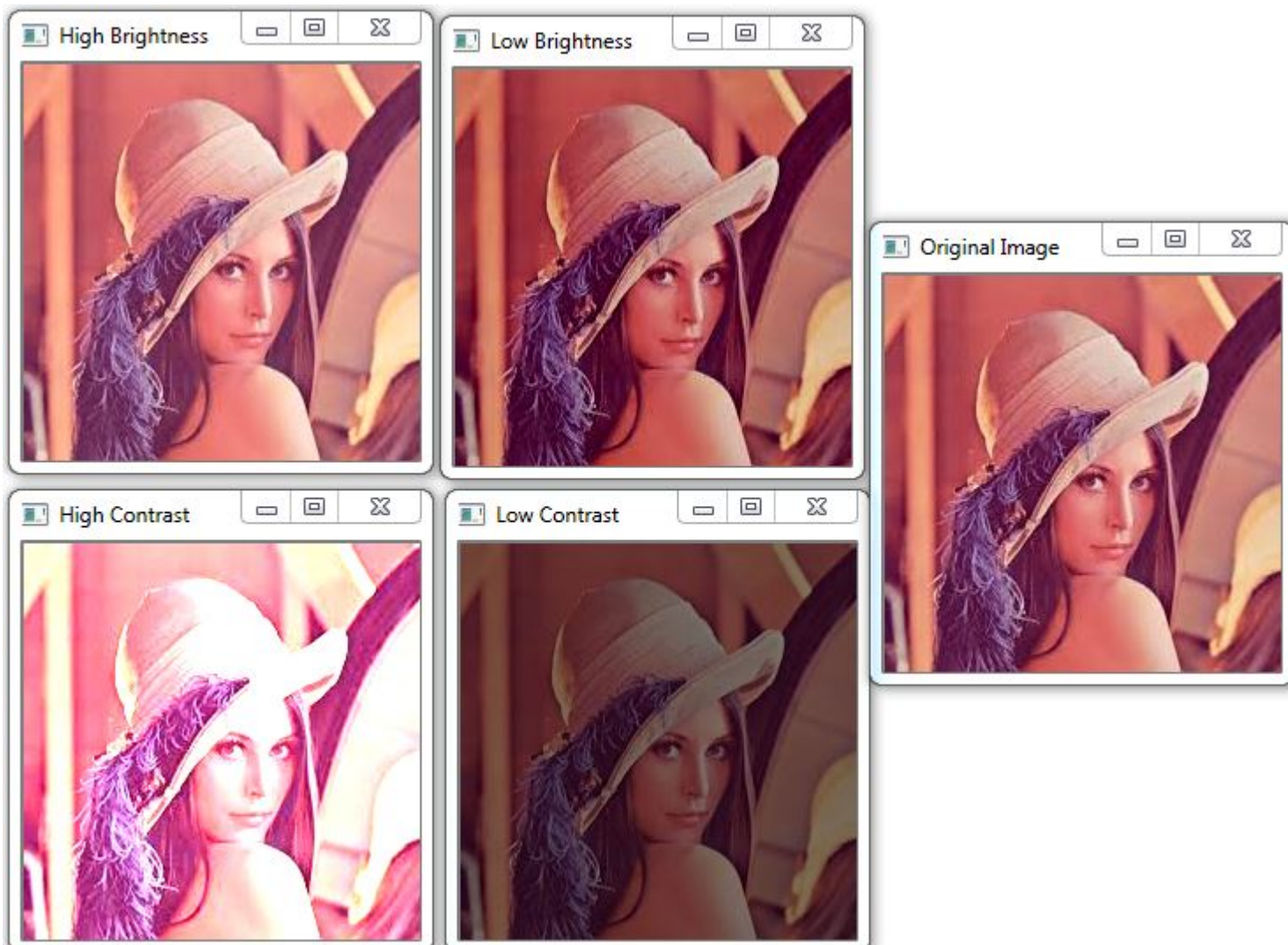
```

//create windows
namedWindow("Original Image", CV_WINDOW_AUTOSIZE);
namedWindow("High Contrast", CV_WINDOW_AUTOSIZE);
namedWindow("Low Contrast", CV_WINDOW_AUTOSIZE);
namedWindow("High Brightness", CV_WINDOW_AUTOSIZE);
namedWindow("Low Brightness", CV_WINDOW_AUTOSIZE);
//show the image
imshow("Original Image", img);
imshow("High Contrast", img_higher_contrast);
imshow("Low Contrast", img_lower_contrast);
imshow("High Brightness", img_higher_brightness);
imshow("Low Brightness", img_lower_brightness);

waitKey(0); //wait for key press
destroyAllWindows(); //destroy all open windows
return 0;
}

```

Вывод программы:



Прочитайте Контрастность и яркость в C ++ онлайн:

<https://riptutorial.com/ru/opencv/topic/6917/контрастность-и-яркость-в-c-plusplus>

глава 9: Модификация содержимого изображения

Examples

Задайте полное изображение сплошному цвету

Учитывая непустой `cv::Mat img` некоторого размера, мы можем заполнить его до сплошного цвета несколькими способами:

```
img = cv::Scalar(blueVal, greenVal, redVal);
```

или, более общая поддержка маски, `cv::Mat::setTo()` :

```
img.setTo(cv::Scalar(blueVal, greenVal, redVal));
```

Если вы используете старый API OpenCV C с `IplImage* img` :

Использование:

```
cvSet(img, CV_RGB(redVal, greenVal, blueVal));
```

Пиксельная модификация изображений

В OpenCV изображения могут быть RGB / BGR, HSV, серого, черно-белого и так далее. Крайне важно знать тип данных перед обращением к изображениям.

Типы данных изображения в основном `CV_8UC3` (Matrix of uchar с 3 каналами) и `CV_8U` (матрица канала с 1 каналом), однако возможны и преобразования в другие типы, такие как `CV_32FC3`, `CV_64F`. (см. [типы данных](#))

Рассмотрим, что изображение представляет собой изображение RGB, которое считывается функцией `imread`.

```
Mat rgb = imread('path/to/rgb/image', CV_LOAD_IMAGE_COLOR);  
//to set RED pixel value of (i,j)th to X,  
rgb.at<Vec3b>(i, j) [0] = X;
```

Аналогично, если изображение выполнено в виде серого,

```
gray.at<uchar>(i, j) = X;
```

Обратите внимание, что в OpenCV черно-белые изображения сохраняются как `CV_8U` с

значениями 0 и 255. Поэтому изменение BW-изображений аналогично серым изображениям.

Изменение цвета изображения в OpenCV - kmeans (). Чтобы сканировать все пиксели изображения и заменить значения пикселей на общие цвета.

```
#include opencv2/opencv.hpp> #include vector> using namespace std; using namespace cv; int  
main() { Mat3b img = imread("test.jpg"); z }
```

```
imshow("Original", img);  
  
// Cluster  
  
int K = 8;  
int n = img.rows * img.cols;  
Mat data = img.reshape(1, n);  
data.convertTo(data, CV_32F);  
  
Mat labels;  
Mat1f colors;  
kmeans(data, K, labels, cv::TermCriteria(), 1, cv::KMEANS_PP_CENTERS, colors);  
  
for (int i = 0; i < n; ++i)  
{  
    data.at<float>(i, 0) = colors(labels.at<int>(i), 0);  
    data.at<float>(i, 1) = colors(labels.at<int>(i), 1);  
    data.at<float>(i, 2) = colors(labels.at<int>(i), 2);  
}  
  
Mat reduced = data.reshape(3, img.rows);  
reduced.convertTo(reduced, CV_8U);  
  
imshow("Reduced", reduced);  
waitKey(0);  
  
return 0;
```

```
#include opencv2/opencv.hpp> #include vector> using namespace std; using namespace cv; int  
main() { Mat3b img = imread("test.jpg"); z }
```

Прочитайте Модификация содержимого изображения онлайн:

<https://riptutorial.com/ru/opencv/topic/6307/модификация-содержимого-изображения>

глава 10: Обнаружение BLOB

Examples

Обнаружение кругового blob

В этом примере показано, как найти круговые капли в изображении в градациях серого. Оценка округлости капли выполняется с использованием площади и периметра (длины дуги) контура. Центральная точка оценивается с использованием моментов контура.

```
#include "opencv/cv.h"
#include "opencv/highgui.h"
#include "opencv/cxcore.h"

using namespace cv;

int main(int argc, char** argv)
{
    Mat img = imread("image.jpg", CV_LOAD_IMAGE_GRAYSCALE);
    Mat resultImg;
    cvtColor(img, resultImg, CV_GRAY2BGR);

    // threshold the image with gray value of 100
    Mat binImg;
    threshold(img, binImg, 100, 255, THRESH_BINARY);

    // find the contours
    vector<vector<Point>> contours;
    vector<Vec4i> hierarchy;
    findContours(binImg, contours, hierarchy, CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE);

    if(contours.size() <= 0)
    {
        printf("no contours found");
        return 0;
    }

    // filter the contours
    vector<vector<Point>> filteredBlobs;
    Mat centers = Mat::zeros(0,2,CV_64FC1);
    for(int i = 0; i < contours.size(); i++)
    {
        // calculate circularity
        double area = contourArea(contours[i]);
        double arclength = arcLength(contours[i], true);
        double circularity = 4 * CV_PI * area / (arclength * arclength);
        if(circularity > 0.8)
        {
            filteredBlobs.push_back(contours[i]);

            //calculate center
            Moments mu = moments(contours[i], false);
            Mat centerpoint = Mat(1,2,CV_64FC1);
            centerpoint.at<double>(i,0) = mu.m10 / mu.m00; // x-coordinate
            centerpoint.at<double>(i,1) = mu.m01 / mu.m00; // y-coordinate
            centers.push_back(centerpoint);
        }
    }
}
```

```
    }  
}  
  
if(filteredBlobs.size() <= 0)  
{  
    printf("no circular blobs found");  
    return 0;  
}  
drawContours(resultImg, filteredBlobs, -1, Scalar(0,0,255), CV_FILLED, 8);  
  
imshow("Blobs", resultImg);  
waitKey(0);  
return 0;  
}
```

Прочитайте Обнаружение BLOB онлайн: <https://riptutorial.com/ru/opencv/topic/6589/обнаружение-blob>

глава 11: Обнаружение кромок

Синтаксис

- `edge = cv2. Canny (изображение, порог1, порог2 [, edge [, apertureSize [, L2gradient]]])`
- `void Canny (изображение InputArray, края OutputArray, double threshold1, double threshold2, int apertureSize = 3, bool L2gradient = false`

параметры

параметр	подробности
образ	Входное изображение
края	Выходное изображение
threshold1	Первый порог для процедуры гистерезиса
порог2	Второй порог для процедуры гистерезиса
размер диафрагмы	Размер диафрагмы для оператора Sobel
L2gradient	Флаг, указывающий, следует ли использовать более точный алгоритм градиента изображения

Examples

Canny алгоритм

Алгоритм Canny - это более поздний краевой детектор, разработанный как проблема обработки сигналов. В OpenCV он выводит двоичное изображение, обозначающее обнаруженные края.

Python:

```
import cv2
import sys

# Load the image file
image = cv2.imread('image.png')

# Check if image was loaded improperly and exit if so
if image is None:
    sys.exit('Failed to load image')
```

```
# Detect edges in the image. The parameters control the thresholds
edges = cv2.Canny(image, 100, 2500, apertureSize=5)

# Display the output in a window
cv2.imshow('output', edges)
cv2.waitKey()
```

Canny Algorithm - C ++

Ниже приводится использование canny алгоритма в с ++. Обратите внимание, что изображение сначала преобразуется в изображение в оттенках серого, затем гауссовский фильтр используется для уменьшения шума на изображении. Затем для обнаружения края используется алгоритм Canny.

```
// CannyTutorial.cpp : Defines the entry point for the console application.
// Environment: Visual studio 2015, Windows 10
// Assumptions: Opecv is installed configured in the visual studio project
// Opencv version: OpenCV 3.1

#include "stdafx.h"
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<string>
#include<iostream>

int main()
{
    //Modified from source:
    https://github.com/MicrocontrollersAndMore/OpenCV_3_Windows_10_Installation_Tutorial
    cv::Mat imgOriginal;           // input image
    cv::Mat imgGrayscale;         // grayscale of input image
    cv::Mat imgBlurred;           // intermediate blurred image
    cv::Mat imgCanny;             // Canny edge image

    std::cout << "Please enter an image filename : ";
    std::string img_addr;
    std::cin >> img_addr;

    std::cout << "Searching for " + img_addr << std::endl;

    imgOriginal = cv::imread(img_addr);           // open image

    if (imgOriginal.empty()) {                    // if unable to open image
        std::cout << "error: image not read from file\n\n";           // show error message on
command line
        return(0);                               // and exit program
    }

    cv::cvtColor(imgOriginal, imgGrayscale, CV_BGR2GRAY);           // convert to grayscale

    cv::GaussianBlur(imgGrayscale,                // input image
        imgBlurred,                               // output image
        cv::Size(5, 5),                           // smoothing window width and height in pixels
        1.5);                                     // sigma value, determines how much the image
```

will be blurred

```
cv::Canny(imgBlurred,          // input image
          imgCanny,            // output image
          100,                 // low threshold
          200);                // high threshold

// Declare windows
// Note: you can use CV_WINDOW_NORMAL which allows resizing the window
// or CV_WINDOW_AUTOSIZE for a fixed size window matching the resolution of the image
// CV_WINDOW_AUTOSIZE is the default
cv::namedWindow("imgOriginal", CV_WINDOW_AUTOSIZE);
cv::namedWindow("imgCanny", CV_WINDOW_AUTOSIZE);

//Show windows
cv::imshow("imgOriginal", imgOriginal);
cv::imshow("imgCanny", imgCanny);

cv::waitKey(0);                // hold windows open until user presses a key
return 0;
}
```

Вычисление Canny Thresholds

[Автоматический расчет низких и высоких порогов для операции Canny в opencv](#)

Canny Edge Видео с веб-камеры Capture - Python

```
import cv2

def canny_webcam():
    "Live capture frames from webcam and show the canny edge image of the captured frames."

    cap = cv2.VideoCapture(0)

    while True:
        ret, frame = cap.read() # ret gets a boolean value. True if reading is successful (I
        think). frame is an
        # uint8 numpy.ndarray

        frame = cv2.GaussianBlur(frame, (7, 7), 1.41)
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        edge = cv2.Canny(frame, 25, 75)

        cv2.imshow('Canny Edge', edge)

        if cv2.waitKey(20) == ord('q'): # Introduce 20 milisecond delay. press q to exit.
            break

    canny_webcam()
```

Canny Edge Thresholds прототипирование с использованием трекбаров

```

"""
CannyTrackbar function allows for a better understanding of
the mechanisms behind Canny Edge detection algorithm and rapid
prototyping. The example includes basic use case.

2 of the trackbars allow for tuning of the Canny function and
the other 2 help with understanding how basic filtering affects it.
"""
import cv2

def empty_function(*args):
    pass

def CannyTrackbar(img):
    win_name = "CannyTrackbars"

    cv2.namedWindow(win_name)
    cv2.resizeWindow(win_name, 500,100)

    cv2.createTrackbar("canny_th1", win_name, 0, 255, empty_function)
    cv2.createTrackbar("canny_th2", win_name, 0, 255, empty_function)
    cv2.createTrackbar("blur_size", win_name, 0, 255, empty_function)
    cv2.createTrackbar("blur_amp", win_name, 0, 255, empty_function)

    while True:
        cth1_pos = cv2.getTrackbarPos("canny_th1", win_name)
        cth2_pos = cv2.getTrackbarPos("canny_th2", win_name)
        bsize_pos = cv2.getTrackbarPos("blur_size", win_name)
        bamp_pos = cv2.getTrackbarPos("blur_amp", win_name)

        img_blurred = cv2.GaussianBlur(img.copy(), (trackbar_pos3 * 2 + 1, trackbar_pos3 * 2 +
1), bamp_pos)
        canny = cv2.Canny(img_blurred, cth1_pos, cth2_pos)
        cv2.imshow(win_name, canny)

        key = cv2.waitKey(1) & 0xFF
        if key == ord("c"):
            break

    cv2.destroyAllWindows()
    return canny

img = cv2.imread("image.jpg")
canny = CannyTrackbar(img)
cv2.imwrite("result.jpg", canny)

```

Прочитайте Обнаружение кромок онлайн: <https://riptutorial.com/ru/opencv/topic/6099/обнаружение-кромки>

глава 12: Обнаружение объекта

Examples

Совместимость шаблона с Java

Исходный код Java

```
import org.opencv.core.Core;
import org.opencv.core.Core.MinMaxLocResult;
import org.opencv.core.Mat;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;

public class TemplateMatching {

    public static void main(String[] args) {

        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
        Mat source=null;
        Mat template=null;
        String filePath="C:\\Users\\mesutpiskin\\Desktop\\Object Detection\\Template
Matching\\Sample Image\\";
        //Load image file
        source=Imgcodecs.imread(filePath+"kapadokya.jpg");
        template=Imgcodecs.imread(filePath+"balon.jpg");

        Mat outputImage=new Mat();
        int machMethod=Imgproc.TM_CCOEFF;
        //Template matching method
        Imgproc.matchTemplate(source, template, outputImage, machMethod);

        MinMaxLocResult mmr = Core.minMaxLoc(outputImage);
        Point matchLoc=mmr.maxLoc;
        //Draw rectangle on result image
        Imgproc.rectangle(source, matchLoc, new Point(matchLoc.x + template.cols(),
            matchLoc.y + template.rows()), new Scalar(255, 255, 255));

        Imgcodecs.imwrite(filePath+"sonuc.jpg", source);
        System.out.println("Complated.");
    }

}
```

РЕЗУЛЬТАТ



Resource Image



Template



Result Im

Прочитайте Обнаружение объекта онлайн: <https://riptutorial.com/ru/opencv/topic/6735/обнаружение-объекта>

глава 13: Обработка изображения

Синтаксис

1. **Gaussian Blur Синтаксис C ++:** `void GaussianBlur (InputArray src, OutputArray dst, размер ksize, double sigmaX, double sigmaY = 0, int borderType = BORDER_DEFAULT)`

параметры

Параметры гауссовского размытия	подробности
ЦСИ	Входное изображение, изображение может иметь любое количество каналов, которые обрабатываются независимо, но глубина должна быть <code>CV_8U</code> , <code>CV_16U</code> , <code>CV_16S</code> , <code>CV_32F</code> или <code>CV_64F</code> .
ДСТ	Выходное изображение того же размера и типа, что и <code>src</code>
ksize	Гауссовский размер ядра. <code>ksize.width</code> и <code>ksize.height</code> могут отличаться, но оба они должны быть положительными и нечетными . Или они могут быть нулями, а затем они вычисляются из сигмы *.
sigmaX	Гауссовское стандартное отклонение ядра в направлении X .
sigmaY	Гауссовское стандартное отклонение ядра в направлении Y . если <code>sigmaY</code> равно нулю, он устанавливается равным <code>sigmaX</code> , если обе сигмы являются нулями, они вычисляются из <code>ksize.width</code> и <code>ksize.height</code> . Чтобы полностью контролировать результат независимо от возможных будущих модификаций всей этой семантики, рекомендуется указать все <code>ksize</code> , <code>sigmaX</code> и <code>sigmaY</code> .
borderType	Метод экстраполяции пикселей.

замечания

Я не думаю, что имеет смысл поставить синтаксис и параметры, характерные для гауссовского размытия, в этом месте, поскольку тема настолько широка, что она должна включать множество других примеров

Examples

Сглаживание изображений с использованием гауссовского размытия в C++

Сглаживание, также известное как **размытие**, является одной из наиболее часто используемых операций обработки изображений.

Наиболее частое использование операции сглаживания заключается в **уменьшении шума** изображения для дальнейшей обработки.

Существует множество алгоритмов для выполнения операций сглаживания.

Мы рассмотрим один из наиболее часто используемых фильтров для смазывания изображения, **Гауссовский фильтр** с использованием функции библиотеки OpenCV `GaussianBlur()`. Этот фильтр разработан специально для снятия *высокочастотного шума* с изображений.

```
#include <opencv2/opencv.hpp>
#include <iostream>

using namespace std;
using namespace cv;

int main(int argc, char** argv){

    Mat image , blurredImage;

    // Load the image file
    image = imread(argv[1], CV_LOAD_IMAGE_COLOR);

    // Report error if image could not be loaded
    if(!image.data){
        cout<<"Error loading image" << "\n";
        return -1;
    }

    // Apply the Gaussian Blur filter.
    // The Size object determines the size of the filter (the "range" of the blur)
    GaussianBlur( image, blurredImage, Size( 9, 9 ), 1.0);

    // Show the blurred image in a named window
    imshow("Blurred Image" , blurredImage);

    // Wait indefinitely untill the user presses a key
    waitKey(0);

    return 0;
}
```

Для детального математического определения и других типов фильтров вы можете проверить [исходную документацию](#).

Thresholding

B Python:



```
import cv2
image_path= 'd:/contour.png'
img = cv2.imread(image_path)

#display image before thresholding
cv2.imshow('I am an image display window',img)
cv2.waitKey(0)

#convert image to gray scale - needed for thresholding
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#apply threshold to gray image to obtain binary image

threshold=150 #value above which pixel values will be set to max_value
max_value=255 #value to which pixels above threshold will be set
threshold_stype=cv2.THRESH_BINARY #default threshold method

ret, img_binary = cv2.threshold(img_gray, threshold, max_value, threshold_stype)

#display image after thresholding
cv2.imshow('image after applying threshold',img_binary)
cv2.waitKey(0)

#save the binary image
cv2.imwrite('d:/binary.png',img_binary)
cv2.destroyAllWindows()
```



Двусторонняя фильтрация

В приложениях обработки изображений двусторонние фильтры представляют собой особый тип **нелинейных фильтров** .

Существует компромисс между потерями структуры и удалением шума, потому что наиболее популярным методом удаления шума является гауссовское размытие, которое не осознает структуру изображения; поэтому он также удаляет края. В большинстве случаев края содержат ценную информацию о сцене, и мы не хотим ее потерять. **Двусторонний фильтр** знает о структуре сцены и имеет тенденцию действовать как классический фильтр размытия, когда он находится на области без краев; однако, когда он видит край, он меняет свое поведение; так что размытие не работает по краям, но оно работает по краям, что означает, что они являются **фильтрами, сохраняющими края**.

```
#include <opencv2/opencv.hpp>
#include <iostream>

void main(int argc, char* argv[]) {
    if(argc==1) {
        std::cout << argv[0] << " <image>" << endl;
        return;
    }

    cv::Mat image, output;
    image = cv::imread(argv[1]);
    if(image.empty()) {
        std::cout << "Unable to load the image: " << argv[1] << endl;
        return;
    }

    cv::bilateralFilter(image, output, 3, 5, 3);
}
```

Прочитайте Обработка изображения онлайн: <https://riptutorial.com/ru/opencv/topic/2032/обработка-изображения>

глава 14: Основные структуры

Вступление

В этом разделе рассматриваются основные структуры OpenCV. Структуры, которые будут обсуждаться в этом разделе, это `DataType` , `Point` , `Vec` , `Size` , `Rect` , `Scalar` , `Ptr` и `Mat` .

Examples

Тип данных

Примитивными типами в OpenCV являются `unsigned char` , `bool` , `signed char` , `unsigned short` , `signed short` , `int` , `float` , `double` . Любой тип данных в OpenCV определяется как `CV_<bit-depth>{U|S|F}C(<number_of_channels>)` где U: unsigned , S:signed и F:floating point .

Например, `CV_32FC2` представляет собой 32-битную, с плавающей точкой и 2-канальную структуру. и определение базового, одного типа каналов

```
#define CV_8U    0
#define CV_8S    1
#define CV_16U   2
#define CV_16S   3
#define CV_32S   4
#define CV_32F   5
#define CV_64F   6
#define CV_USRTYPE1 7
```

Другие типы с более высоким каналом получают из них следующим определением:

```
#define CV_MAKETYPE(depth,cn) (CV_MAT_DEPTH(depth) + (((cn)-1) << CV_CN_SHIFT))
```

Используя эти типы данных, могут быть созданы другие структуры.

Мат

`Mat` (Matrix) - это n-мерный массив, который может использоваться для хранения различных типов данных, таких как изображения RGB, HSV или оттенки серого, векторы с реальными или сложными значениями, другие матрицы и т. Д.

`Mat` содержит следующую информацию: `width` , `height` , `type` , `channels` , `data` , `flags` , `datastart` , `dataend` и так далее.

Он имеет несколько методов, некоторые из них: `create` , `copyTo` , `convertTo` , `isContinuous` и т. Д.

Существует множество способов создания переменной Mat. Рассмотрим, что я хочу создать матрицу со 100 строками, 200 столбцами, тип CV_32FC3:

```
int R = 100, C = 200;
Mat m1; m1.create(R,C,CV_32FC3); //creates empty matrix
Mat m2(cv::Size(R, C), CV_32FC3); // creates a matrix with R rows, C columns with data type T
where R and C are integers,
Mat m3(R,C,CV_32FC3); // same as m2
```

Инициализация Mat:

```
Mat m1 = Mat::zeros(R,C,CV_32FC3); // This initialized to zeros, you can use one, eye or
cv::randn etc.
Mat m2(R,C,CV_32FC3);
for (int i = 0; i < m2.rows; i++)
    for (int j = 0; j < m2.cols; j++)
        for (int k = 0; k < m2.channels(); k++)
            m2.at<Vec3f>(i,j)[k] = 0;
//Note that, because m2 is a float type and has 3 channels, we used Vec3f, for more info see
Vec

Mat m3(3, out, CV_32FC1, cv::Scalar(0));
```

Vec

Vec (Vector) - это шаблонный шаблон для численных значений. В отличие от c++ vector s, он обычно хранит короткие векторы (всего несколько элементов).

Способ определения Vec следующий:

```
typedef Vec<type, channels> Vec< channels>< one char for the type>;
```

где type - один из uchar, short, int, float, double а символы для каждого типа - b, s, i, f, d соответственно.

Например, Vec3b указывает беззнаковый вектор символов из 3-х каналов. Каждый индекс в RGB-изображении находится в этом формате.

```
Mat rgb = imread('path/to/file', CV_LOAD_IMAGE_COLOR);
cout << rgb.at<Vec3b>(0,0); //The output is [r g b] values as ASCII character.
// To print integer values of RED value
cout << (int)rgb.at<Vec3b>(0,0)[0]; //The output will be an integer in [0, 255].
```

В классе Vec определены следующие операторы

```
v1 = v2 + v3
v1 = v2 - v3
v1 = v2 * scale
v1 = scale * v2
v1 = -v2
v1 += v2 and other augmenting operations
```

```
v1 == v2, v1 != v2
```

Для получения дополнительной информации см. [Ссылку](#)

Прочитайте Основные структуры онлайн: <https://riptutorial.com/ru/opencv/topic/9099/основные-структуры>

глава 15: Показать изображение OpenCV

Examples

Основное чтение и отображение изображения

```
import cv2

image_path= #put your image path here

#use imread() function to read image data to variable img.
img = cv2.imread(image_path)

#display image data in a new window with title 'I am an image display window'
cv2.imshow('I am an image display window',img)

#wait until user hits any key on keyboard
cv2.waitKey(0)

#close any windows opened by opencv
cv2.destroyAllWindows()
```

Чтобы управлять размером окна дисплея на экране, добавьте следующие команды перед командой `cv2.imshow`:

```
window_width=800 #size of the display window on the screen
window_height=600

#open an empty window with a title.
#The flag cv2.WINDOW_NORMAL allows the window to be scaleable.
cv2.namedWindow('I am an image display window', cv2.WINDOW_NORMAL)

#scale the image display window to desired size
cv2.resizeWindow('I am an image display window', window_width, window_height)
```

см. [документы openCV](#) для получения более подробной информации

Чтение MJPEG с IP-камеры

```
import cv2
import numpy as np
import urllib

stream=urllib.urlopen('http://96.10.1.168/mjpg/video.mjpg')
bytes=''
while True:
    bytes+=stream.read(1024)
    a = bytes.find('\xff\xd8') # JPEG start
    b = bytes.find('\xff\xd9') # JPEG end
    if a!=-1 and b!=-1:
        jpg = bytes[a:b+2] # actual image
        bytes= bytes[b+2:] # other informations
```



```
# decode to colored image ( another option is cv2.IMREAD_GRAYSCALE )
img = cv2.imdecode(np.fromstring(jpg, dtype=np.uint8),cv2.IMREAD_COLOR)
cv2.imshow('Window name',img) # display image while receiving data
if cv2.waitKey(1) ==27: # if user hit esc
    exit(0) # exit program
```

Каждый JPEG начинается с 0xff 0xd8 и заканчивается 0xff 0xd9 . Между ними есть реальный образ. Подробная информация в [этом ответе SO](#)

Показать изображение OpenCV Java

Основное изображение для чтения из java

```
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.imgcodecs.Imgcodecs;

//Load native library
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
//Mat object used to host the image
Mat imageArray;
//Read image file from file system
imageArray=Imgcodecs.imread("path/to/image");
```

Если вы хотите просматривать изображения, вы не можете использовать imshow, потому что OpenCV-java также не имеет этого метода. Вместо этого вы можете написать следующий метод.

```
private static BufferedImage ConvertMat2Image(Mat imgContainer{
    MatOfByte byteMatData = new MatOfByte();
    //image formatting
    Imgcodecs.imencode(".jpg", imgContainer,byteMatData);
    // Convert to array
    byte[] byteArray = byteMatData.toArray();
    BufferedImage img= null;
    try {
        InputStream in = new ByteArrayInputStream(byteArray);
        //load image
        img= ImageIO.read(in);
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
    return img;
}
```

Вы можете просмотреть объект результата в JFrame, JLabel (значок jlabel) и т. Д.

Прочитайте [Показать изображение OpenCV онлайн](#):

<https://riptutorial.com/ru/opencv/topic/3306/показать-изображение-opencv>

глава 16: Рисование фигур (линия, круг, ... и т. Д.) В C ++

Вступление

В OpenCV можно рисовать многочисленные фигуры, такие как точка, линия, круг, ... и т. Д. Для заполнения фигуры может быть необязательно. Следующий код не требует пояснений, который показывает, как рисуются фигуры.

Examples

Образец рисования фигур

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

int main( int argc, char** argv )
{
    // First create a black image.
    cv::Mat image(500,500, CV_8UC3, cv::Scalar(0,0,0));

    // Check if the image is created successfully.
    if( !image.data ){
        std::cout << "Could not open or find the image" << std::endl ;
        exit(EXIT_FAILURE);
    }

    //#####( Draw Line )#####
    cv::Point p1(100,100), p2(200,100);
    cv::Scalar colorLine(0,255,0); // Green
    int thicknessLine = 2;

    cv::line(image, p1, p2, colorLine, thicknessLine);

    //#####( Draw Circle )#####
    // unfilled circle
    cv::Point centerCircle1(250,250);
    int radiusCircle = 30;
    cv::Scalar colorCircle1(0,0,255);
    int thicknessCircle1 = 2;

    cv::circle(image, centerCircle1, radiusCircle, colorCircle1, thicknessCircle1);

    // filled circle
    cv::Point centerCircle2(400,100);
    cv::Scalar colorCircle2(0,100,0);
```

```

cv::circle(image, centerCircle2, radiusCircle, colorCircle2, CV_FILLED);

//#####( Draw Rectangle )#####
// unfilled
cv::Point p3(400,400), p4(450,450);
cv::Scalar colorRectangle1(0,0,255);
int thicknessRectangle1 = 3;

cv::rectangle(image, p3, p4, colorRectangle1,thicknessRectangle1);

// filled
cv::Point p5(100,400), p6(150,450);
cv::Scalar colorRectangle2(255,0,255);

cv::rectangle(image, p5, p6, colorRectangle2, CV_FILLED);

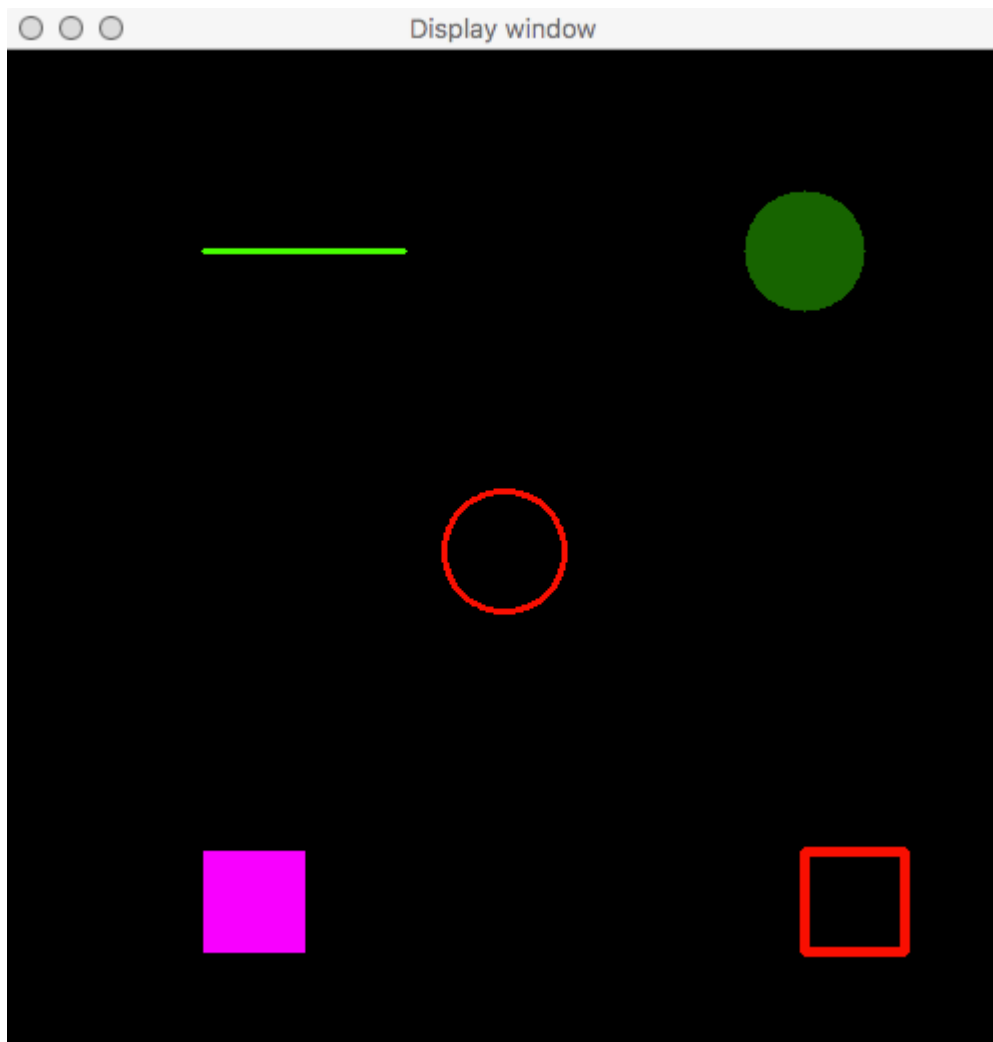
//#####( Draw Shapes on Image )#####
cv::namedWindow( "Display window", cv::WINDOW_AUTOSIZE );
cv::imshow( "Display window", image );

cv::waitKey(0);

return 0;
}

```

Выход



OpenCV 3.2 Mac с компилятором g ++

```
g++ main2.cpp -o main `pkg-config --cflags --libs opencv`
```

Прочитайте Рисование фигур (линия, круг, ... и т. Д.) В C ++ онлайн:

<https://riptutorial.com/ru/opencv/topic/9749/рисование-фигур--линия--круг-----и-т--д---в-с-plusplus>

глава 17: Сборка и компиляция opencv 3.1.0-dev для Python2 в Windows с использованием CMake и Visual Studio

замечания

Создание и компиляция `opencv 3.1.0-dev` для получения доступа к **несвободным модулям** может стать головной болью для некоторых людей, особенно на Windows-машине. В отличие от Ubuntu, настройка `opencv` для Windows занимает некоторое время и требует установки нескольких зависимостей `pf` перед сборкой и компиляцией.

Программы, которые вы должны скачать и установить, прежде чем идти дальше на любом этапе, следующие:

1. [Python 2.7.x](#) или [Python 3.xx](#)
2. [CMake](#)

Если вы собираетесь загрузить Python для Win32, вы также должны скачать CMake для Win32, даже если вы используете 64-битную машину.

Рекомендуется загружать 32-битные программы, потому что некоторые библиотеки Python поддерживаются только для 32-разрядных машин, поэтому, чтобы не бояться неприятностей, просто установите все в 32-разрядной версии.

3. [Сообщество Visual Studio 2013](#)
4. [Numpy](#) для Python2.7 Win32

После установки всех вышеуказанных зависимостей **перезагрузите** компьютер, и вы будете готовы перейти к следующему шагу.

Шаг 2:

Если вы не являетесь тем человеком, который предпочитает читать, вы можете посмотреть этот [учебник](#). Учебное пособие перенесет вас отсюда до конца этой документации.

Вам нужно будет получить `opencv` и `opencv_contrib` из `github`. Вы можете найти оба:

1. [OpenCV](#)
2. [opencv_contrib](#)

Создайте каталог с именем `opencv-3.1.0`, где в этом директоре вы создадите еще два каталога для *сборки* и один для *источников* . После извлечения вы загрузите два загруженных zip-файла в исходный файл.

Например, ваш каталог `opencv-3.1.0` находится в диске C, поэтому у вас будет три пути:

1. `C:\opencv-3.1.0`
2. `C:\opencv-3.1.0\build`
3. `C:\opencv-3.1.0\sources`

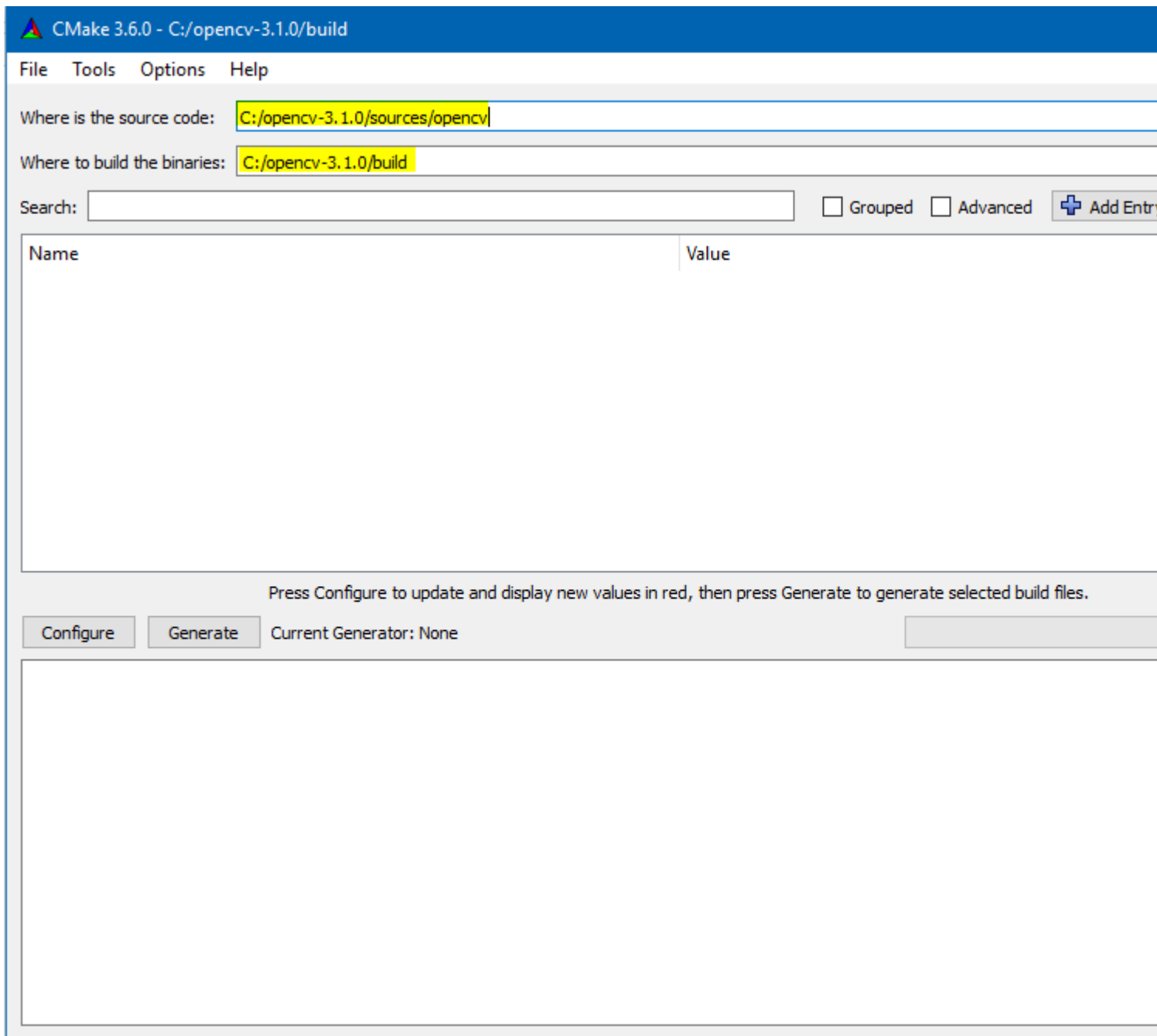
Третий каталог будет содержать два пути:

1. `C:\opencv-3.1.0\sources\opencv`
2. `C:\opencv-3.1.0\sources\opencv_contrib`

Теперь это делается с подготовкой. Давайте сделаем некоторые полезные вещи.

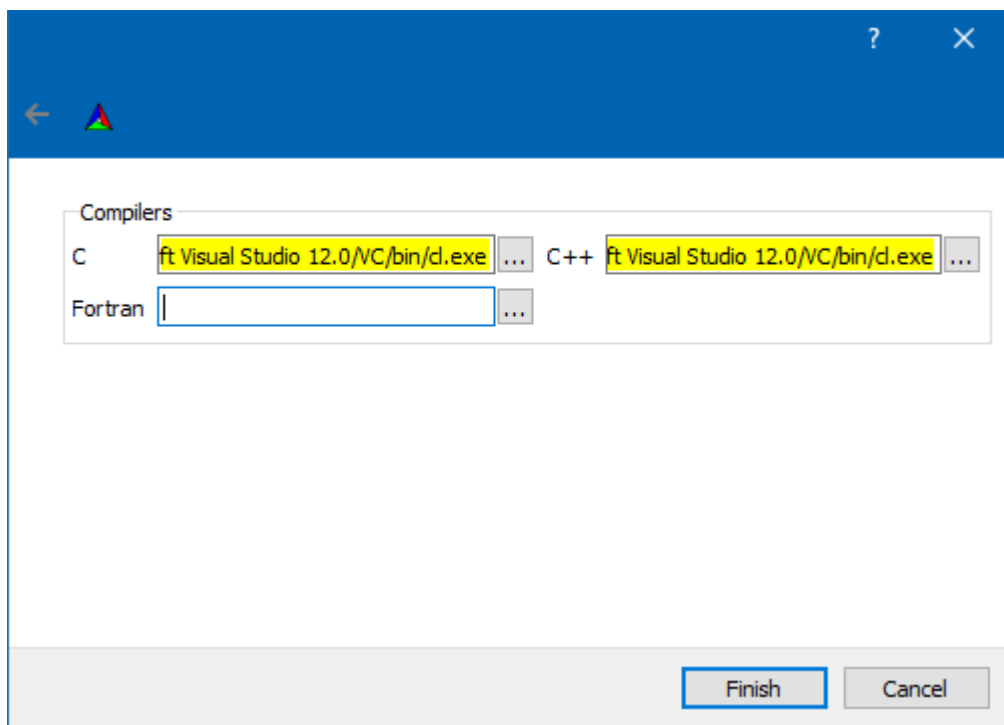
Шаг 3:

Запустите CMake как администратор. Появится окно, подобное этому, и вам нужно будет предоставить два каталога для источников, а другой - для того, где будет скомпилирован opencv. Ниже изображение может помочь вам лучше, чем слова.

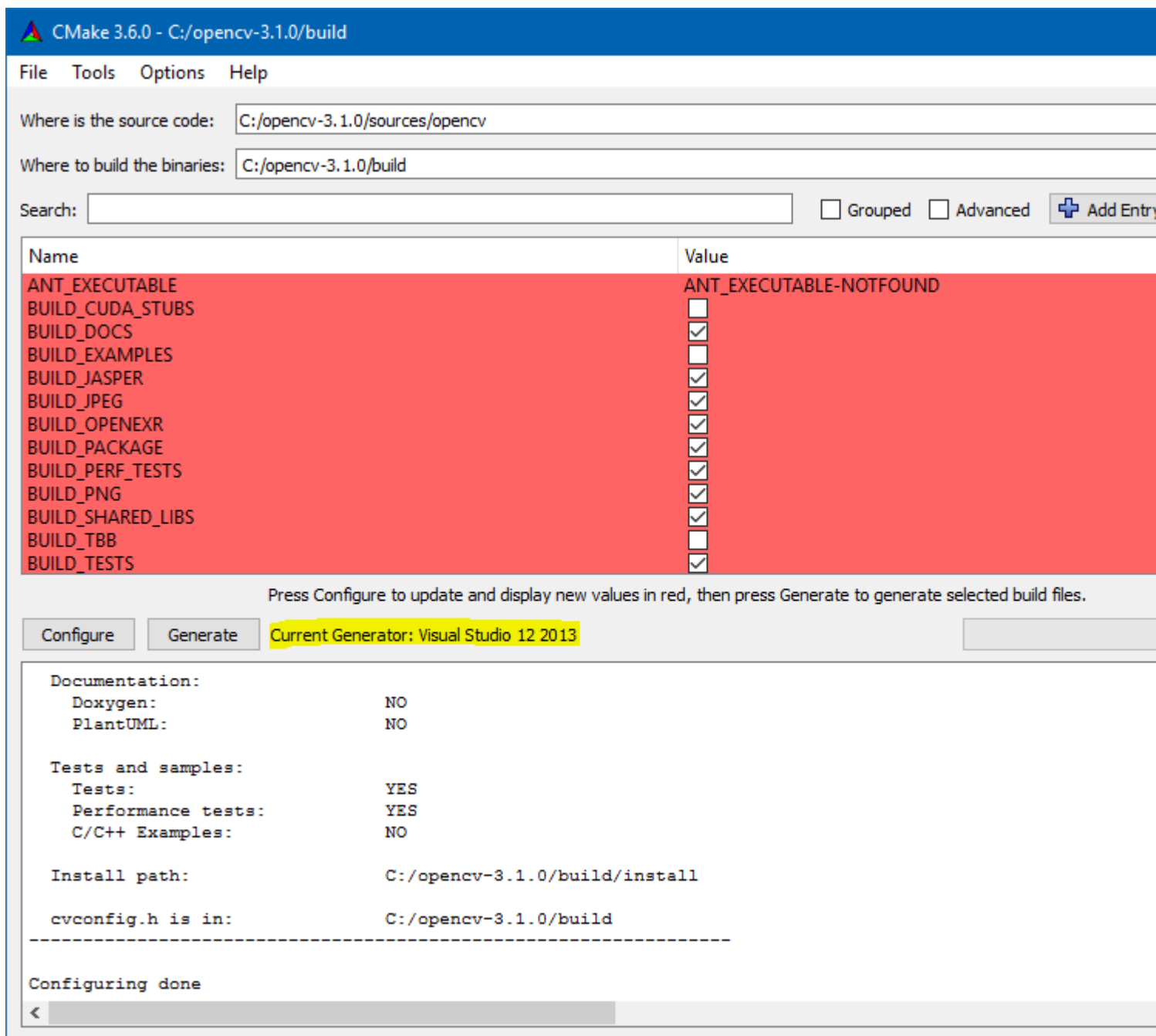


Затем нажмите **configure**, и вам будет предложено предоставить генераторы; т.е. составителей; для opencv. Вы должны предоставить `cl.exe` расположенную в Microsoft Visual Studio 2013. Нажмите « `cl.exe` **собственные генераторы** », и появится всплывающее окно, подобное следующему:

Пути будут выглядеть примерно так: `C:/Program Files (x86)/Microsoft Visual Studio 12.0/VC/bin/cl.exe` . Укажите свой путь для полей C и C ++. Нажмите «Готово» и подождите, пока не будет выполнена настройка. Если вы правильно выполнили все предыдущие шаги, вы должны получить нулевые ошибки.



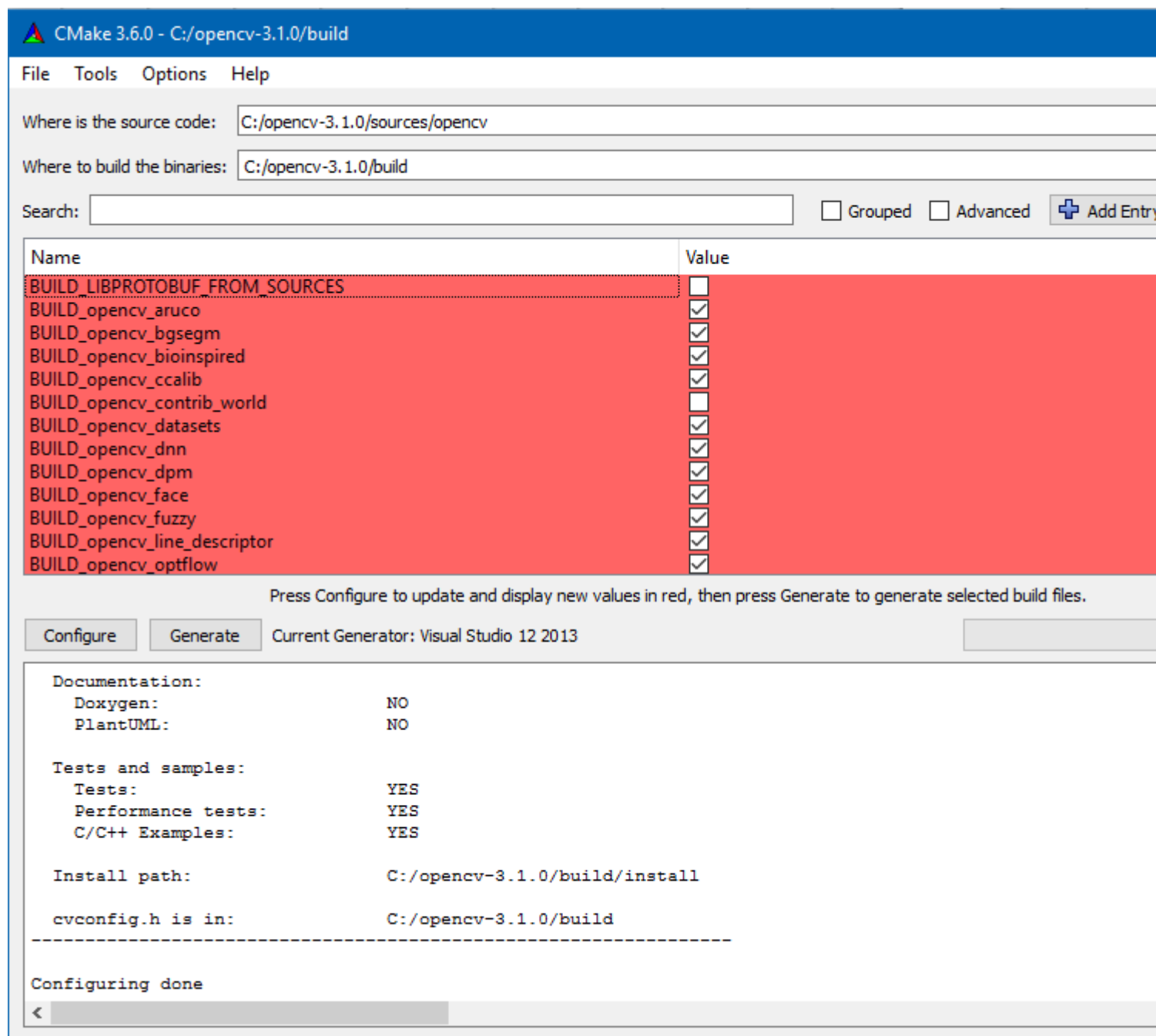
После завершения настройки CMake вы увидите новые элементы, появляющиеся в окне CMake, которые выделены красным цветом. Это будет что-то вроде:



Проверьте сборки, которые вам нужны, щелкнув маленькую квадратную коробку. Найдите строку `OPENCV_EXTRA_MODULES_PATH` и `OPENCV_EXTRA_MODULES_PATH` каталог модулей в `opencv_contrib` в каталоге источников.

MATLAB_MEXEXT_SCRIPT_	
MATLAB_MEX_SCRIPT_	
MATLAB_ROOT_DIR_	
OPENCV_CONFIG_FILE_INCLUDE_DIR	<input checked="" type="checkbox"/>
OPENCV_EXTRA_MODULES_PATH	C:/opencv-3.1.0/build
OPENCV_HAL_HEADERS	<input checked="" type="checkbox"/>
OPENCV_HAL_LIBS	
OPENCV_WARNINGS_ARE_ERRORS	<input type="checkbox"/>
PVAPI_INCLUDE_PATH	PVAPI_INCLUDE_PATH-NOTFOUND
PYTHON2_EXECUTABLE	C:/Python27/python.exe
PYTHON2_INCLUDE_DIR	C:/Python27/include
PYTHON2_INCLUDE_DIR2	

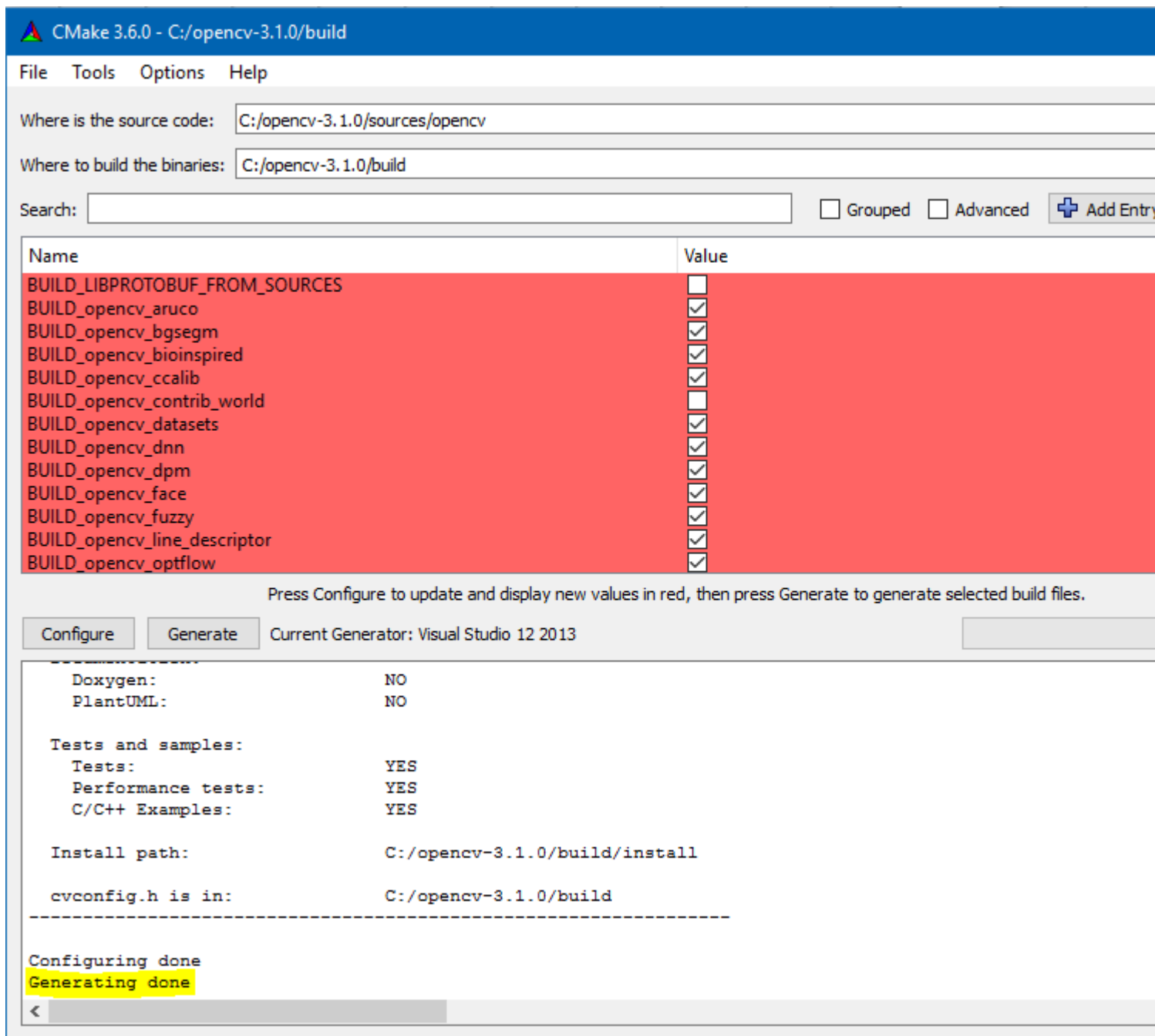
После того, как вы закончите все, что вам нужно, и укажите путь для дополнительных модулей, нажмите кнопку configure еще раз для обновления. Ранее выделенные строки больше не будут выделены, а новые поля будут выделены красным цветом.



Также установите флажки для того, что вам нужно построить.

Убедитесь, что **BUILD_opencv_contrib_world** и **BUILD_opencv_world** оба **бесконтрольно**. Вероятно, есть ошибка, при которой происходит ошибка, когда какой-либо из последних проверяется.

В конце этого шага нажмите « **Создать** », и вы сделаете это с помощью CMake, и вы можете закрыть его. Если ошибок нет, вы получите сообщение в конце нижней панели, в котором говорится, что **Generating done**.



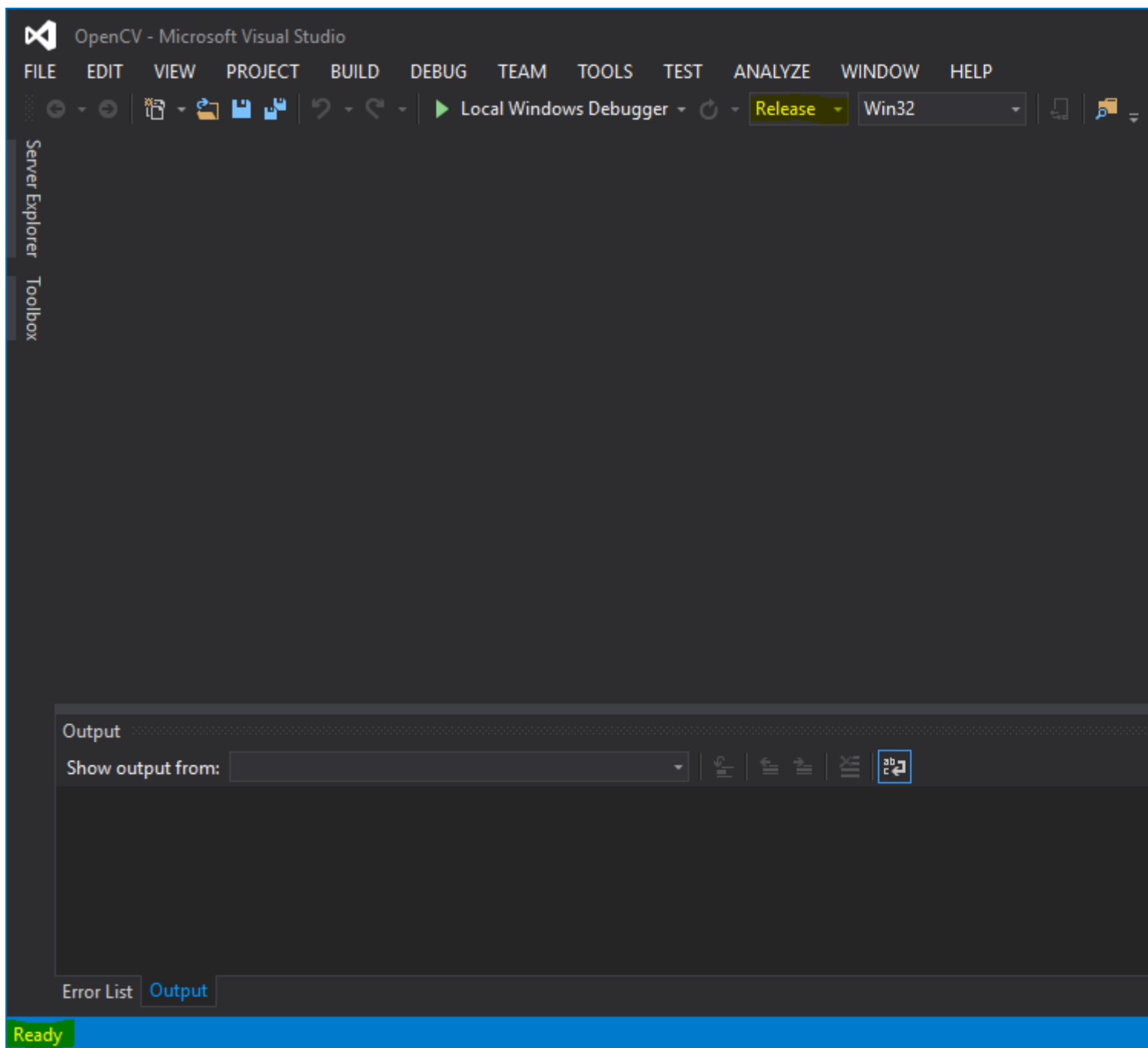
Шаг 4:

Откройте каталог сборки, расположенный в `opencv-3.1.0`, и вы найдете в нем множество новых папок и файлов. Это была пустая папка в начале этого процесса.

Вы будете иметь дело только с файлом `opencv.sln` и ничего не делаете с остальными файлами. Откройте этот файл с версией, которая использовалась при компиляции в CMake на предыдущем шаге. Это должен быть `visual Microsoft 2013`.

Name	Date modified	Type	Size
samples	7/30/2016 8:52 PM	File folder	
test-reports	7/30/2016 8:38 PM	File folder	
unix-install	7/30/2016 8:46 PM	File folder	
win-install	7/30/2016 8:46 PM	File folder	
ALL_BUILD.vcxproj	7/30/2016 8:52 PM	VC++ Project	88 KB
ALL_BUILD.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB
cmake_install.cmake	7/30/2016 8:52 PM	CMAKE File	7 KB
cmake_uninstall.cmake	7/30/2016 8:38 PM	CMAKE File	2 KB
CMakeCache.txt	7/30/2016 8:46 PM	Text Document	244 KB
CMakeVars.txt	7/30/2016 8:46 PM	Text Document	407 KB
CPackConfig.cmake	7/30/2016 8:46 PM	CMAKE File	10 KB
CPackSourceConfig.cmake	7/30/2016 8:46 PM	CMAKE File	10 KB
CTestTestfile.cmake	7/30/2016 8:52 PM	CMAKE File	1 KB
custom_hal.hpp	7/30/2016 8:38 PM	C/C++ Header	1 KB
cvconfig.h	7/30/2016 8:38 PM	C/C++ Header	5 KB
INSTALL.vcxproj	7/30/2016 8:52 PM	VC++ Project	7 KB
INSTALL.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB
OpenCV.sln	7/30/2016 8:53 PM	Microsoft Visual S...	948 KB
opencv_modules.vcxproj	7/30/2016 8:52 PM	VC++ Project	28 KB
opencv_modules.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB
opencv_perf_tests.vcxproj	7/30/2016 8:52 PM	VC++ Project	24 KB
opencv_perf_tests.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB
opencv_tests.vcxproj	7/30/2016 8:52 PM	VC++ Project	26 KB
opencv_tests.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB
OpenCVConfig.cmake	7/30/2016 8:46 PM	CMAKE File	19 KB
OpenCVConfig-version.cmake	7/30/2016 8:38 PM	CMAKE File	1 KB
OpenCVModules.cmake	7/30/2016 8:53 PM	CMAKE File	47 KB
PACKAGE.vcxproj	7/30/2016 8:52 PM	VC++ Project	7 KB
PACKAGE.vcxproj.filters	7/30/2016 8:52 PM	VC++ Project Filte...	1 KB

Когда вы откроете файл `.sln`, будьте терпеливы, так как требуется время, чтобы подготовить все для строительства. Когда **Ready** устойчив (не меняется), вы можете начать строить свои цели. Начните строительство, пронумерованное на изображении ниже. Также убедитесь, что `Solution Configuration - Release` **not** `Debug`.



Шаг 5:

По завершении построения вам нужно будет скопировать и вставить пару файлов из каталога сборки в каталог `Python27`.

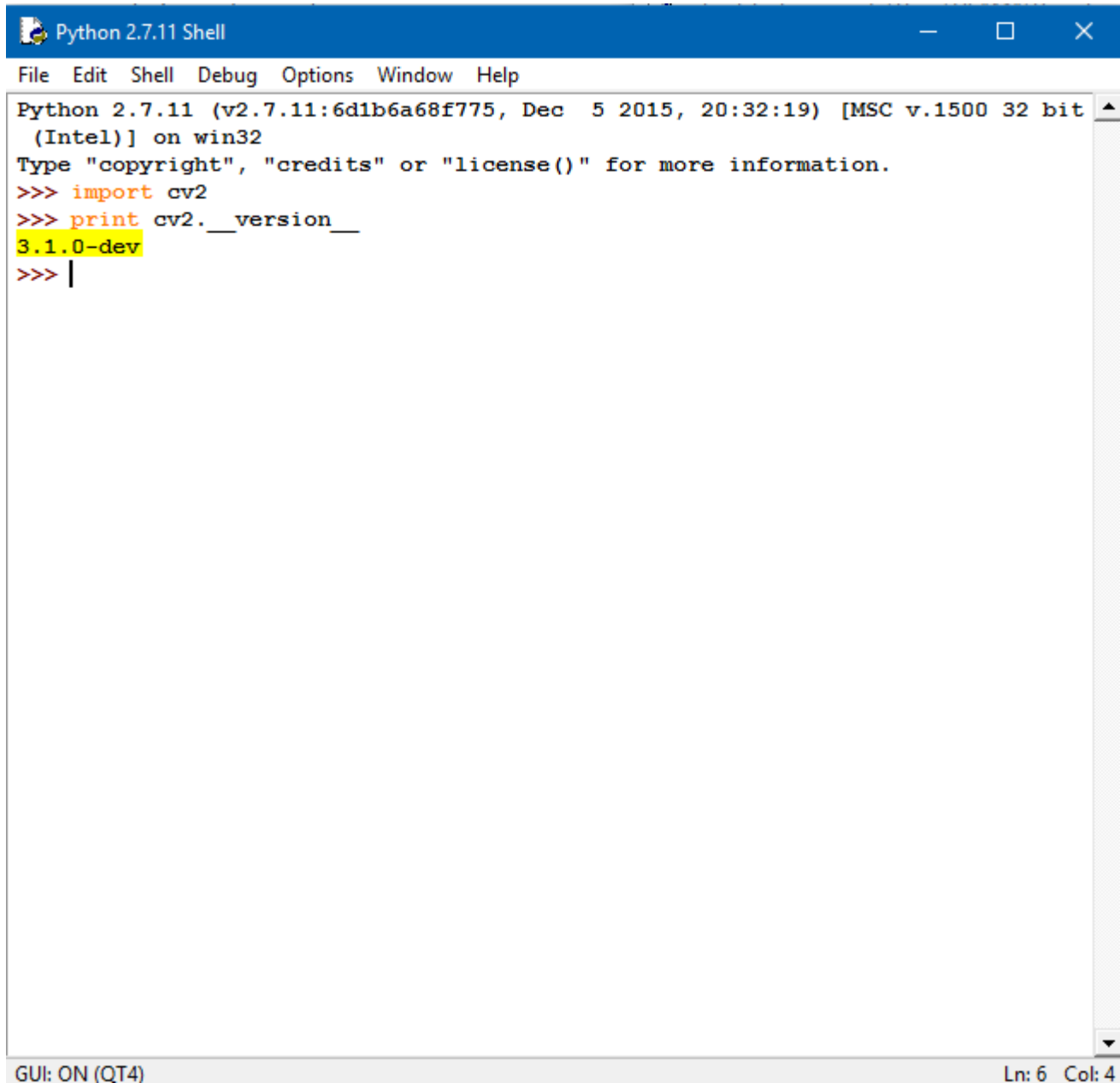
Найдите файл `cv2.pyd` и скопируйте его в каталог `site-packages` в `Python27`. `cv2.pyd` должен присутствовать в `C:\opencv-3.1.0\build\lib\Release`. После этого скопируйте **только** файлы `.dll` внутри `C:\opencv-3.1.0\build\bin\Release` в родительский каталог `Python27` в этом месте `C:\Python27`.

В конце этого шага перезагрузите компьютер.

Проверка:

Откройте IDLE и внутри оболочки оболочки Python:

```
>>> import cv2
>>> print cv2.__version__
3.1.0-dev
```



Examples

Чтение изображения и преобразование в оттенки серого

```
import cv2
import numpy as np

img = cv2.imread('<your_image>')
```

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imshow('image', img)
cv2.imshow('gray', gray)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Прочитайте Сборка и компиляция opencv 3.1.0-dev для Python2 в Windows с использованием CMake и Visual Studio онлайн: <https://riptutorial.com/ru/opencv/topic/6100/сборка-и-компиляция-opencv-3-1-0-dev-для-python2-в-windows-с-использованием-cmake-и-visual-studio>

глава 18: Создание видео

Вступление

Всякий раз, когда вы работаете с видеопотоками, вы можете в конечном итоге захотеть сохранить результат обработки изображения в виде нового видеофайла. Для простых видеовыходов вы можете использовать встроенный класс VideoWriter OpenCV, предназначенный для этого. Полезно рассмотреть некоторые концепции перед их использованием. Эти понятия представляют собой кодек, т.е. декодер и FourCC (четырёхсимвольный код).

Examples

Создание видео с помощью OpenCV (Java)

```
VideoWriter videoWriter;
videoWriter = new VideoWriter(outputFile, VideoWriter.fourcc('x', '2', '6', '4'),
                             fps, frameSize, isRGB);
//We have stated that we will use x264 as codec with FourCC
//For writing, we add the following method and it will write the image we give as parameter in
this call.
public void Write(Mat frame) {
    if(videoWriter.isOpened()==false){
        videoWriter.release();
        throw new IllegalArgumentException("Video Writer Exception: VideoWriter not
opened,"
                                     + "check parameters.");
    }
    //Write video
    videoWriter.write(frame);
}

//With Video Capture for example, we can read images from the camera and write the same video

VideoCapture videoCapture = new VideoCapture(0);
Size frameSize = new Size((int) videoCapture.get(Videoio.CAP_PROP_FRAME_WIDTH), (int)
videoCapture.get(Videoio.CAP_PROP_FRAME_HEIGHT));
VideoWriter videoWriter = new VideoWriter("test.avi", VideoWriter.fourcc('x', '2', '6', '4'),
                             videoCapture.get(Videoio.CAP_PROP_FPS), frameSize, true);
while (videoCapture.read(mat)) {
    videoWriter.write(mat);
}
videoCapture.release();
videoWriter.release();
```

Прочитайте Создание видео онлайн: <https://riptutorial.com/ru/opencv/topic/9196/создание-видео>

глава 19: Установка OpenCV

Вступление

Установка OpenCV в Linux, Mac OS и Windows

Examples

Установка OpenCV на Ubuntu

Ссылка источника

Откройте терминал и напишите следующие команды.

1-Обновление и обновление пакета вашей системы ubuntu:

```
sudo su  
  
sudo apt-get -y update  
  
sudo apt-get -y upgrade  
  
sudo apt-get -y dist-upgrade  
  
sudo apt-get -y autoremove
```

2-Установка зависимостей:

```
sudo apt-get install libopencv-dev
```

3-Build Tools для OpenCV Исходный код:

```
sudo apt-get install build-essential checkinstall cmake pkg-config
```

4-графические библиотеки ввода-вывода для OpenCV:

```
sudo apt-get install libtiff5-dev libjpeg-dev libjasper-dev libpng12-dev zlib1g-dev  
libopenexr-dev libgdal-dev
```

5-видео-библиотеки ввода-вывода для OpenCV:

```
sudo apt-get install libavcodec-dev libavformat-dev libmp3lame-dev libswscale-dev  
libdc1394-22-dev libxine2-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-  
dev v4l-utils libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-  
dev libxvidcore-dev libx264-dev x264 yasm
```

Библиотеки 6-параллелизма и линейной алгебры:

```
sudo apt-get install libtbb-dev libeigen3-dev
```

7-графические библиотеки пользовательского интерфейса:

```
sudo apt-get install libqt4-dev libgtk2.0-dev qt5-default
```

```
sudo apt-get install libvtk6-dev
```

8-Java Установка:

```
sudo apt-get install ant default-jdk
```

9-Python Установка:

```
sudo apt-get install python-dev python-tk python-numpy python3-dev python3-tk python3-numpy  
python-matplotlib
```

```
sudo apt-get install python-opencv
```

```
sudo apt-get install doxygen
```

10-Загрузить исходный код OPENCV от Github:

```
wget https://github.com/opencv/opencv/archive/3.2.0.zip
```

11-Decompress OPENCV Почтовый индекс:

```
unzip 3.2.0.zip
```

12-Удалить файл OPENCV Zip:

```
rm 3.2.0.zip
```

13-Build OPENCV:

```
mv opencv-3.2.0 opencv
```

```
cd opencv
```

```
mkdir build
```

```
cd build
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=ON -D  
BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON -D  
INSTALL_PYTHON_EXAMPLES=ON -D BUILD_DOC=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON  
-D WITH_EIGEN=ON -D FORCE_VTK=TRUE -D WITH_VTK=ON ..
```

```
make -j4
```

sudo make install

```
sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
sudo ldconfig
```

14-Finished Проверьте номер версии OpenCV:

```
pkg-config --modversion opencv
pkg-config --cflags opencv
```

Прочитайте Установка OpenCV онлайн: <https://riptutorial.com/ru/opencv/topic/8934/установка-opencv>

глава 20: Функции рисования в Java

Examples

Нарисовать прямоугольник на изображении

```
public class DrawRectangle {  
  
    public static void main(String[] args) {  
        //Load native library  
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);  
        //image container object  
        Mat goruntuDizisi=new Mat();  
        //Read image in file system  
        goruntuDizisi=Imgcodecs.imread("C:\\image.jpg");  
        //Draw rectangle  
        //Parameters: mat object for drawing, point coordinates (x1,y1,x2,y2) and color BGR  
        Imgproc.rectangle(goruntuDizisi, new Point(10,100), new Point(100,200),new  
        Scalar(76,255,0));  
        Imgcodecs.imwrite("C:\\Yeni_kiz_kulesi.jpg", goruntuDizisi);  
        System.out.println("Writed");  
    }  
}
```

Прочитайте Функции рисования в Java онлайн: <https://riptutorial.com/ru/opencv/topic/6153/функции-рисования-в-java>

кредиты

S. No	Главы	Contributors
1	Начало работы сopencv	Arijit , bburns.km , Berriel , Community , Elizabeth , hackhisass , jlarsch , John Hany , K D , MD. Nazmul Kibria , mesutpiskin , snb , StephenG , Sunreef , winseybash , Yassie , zeeshan khan
2	Доступ к пикселям	Adi Shavit , brian , cagatayodabasi , Ehsan Ab , Elizabeth , smttsp , Sunreef
3	Загрузка и сохранение различных форматов мультимедиа	Adi Shavit , cagatayodabasi , Jav_Rock , Lakshya Kejriwal , MD. Nazmul Kibria
4	Инициализация OpenCV в Android	David Miguel
5	Использование VideoCapture с OpenCV Python	jlarsch , mesutpiskin
6	Использование каскадных классификаторов в Java	Edward Shen
7	Каскадные классификаторы	Arijit , MD. Nazmul Kibria , mesutpiskin
8	Контрастность и яркость в C ++	Ehsan Ab , MD. Nazmul Kibria
9	Модификация содержимого изображения	Adi Shavit , DivyaMaheswaran , smttsp
10	Обнаружение BLOB	MD. Nazmul Kibria , Sebastian
11	Обнаружение кромок	cmastudios , Ehsan Ab , K D , m3h0w , Sounak

12	Обнаружение объекта	K D , mesutpiskin
13	Обработка изображения	cagatayodabasi , Dan Mašek , Elizabeth , jlarsch , Shubham Batra , Sunreef , Utkarsh Sinha
14	Основные структуры	smttsp
15	Показать изображение OpenCV	Aleksandar , Elizabeth , jlarsch , mesutpiskin , smttsp
16	Рисование фигур (линия, круг, ... и т. Д.) В C ++	CroCo
17	Сборка и компиляция opencv 3.1.0-dev для Python2 в Windows с использованием CMake и Visual Studio	Tes3awy
18	Создание видео	mesutpiskin
19	Установка OpenCV	amorenew
20	Функции рисования в Java	mesutpiskin