

OpenCV Tutorial C++

[Home](#) [OpenCV Lessons](#) [Reference Books](#) [About me](#)

Basics of OpenCV API

Header files

These are some of important OpenCV header files for C++ interface. As a beginner, you will need few of these header files for your application. In my following lessons, I will include only necessary header files to my example programs. If you are not sure what to include, include them all. No any penalty incurred for including all this header file except for a fact that the length of your source code will increased by few lines than necessary.

- `#include "opencv2/core/core.hpp"`
- `#include "opencv2/flann/miniflann.hpp"`
- `#include "opencv2/imgproc/imgproc.hpp"`
- `#include "opencv2/photo/photo.hpp"`
- `#include "opencv2/video/video.hpp"`
- `#include "opencv2/features2d/features2d.hpp"`
- `#include "opencv2/objdetect/objdetect.hpp"`
- `#include "opencv2/calib3d/calib3d.hpp"`
- `#include "opencv2/ml/ml.hpp"`
- `#include "opencv2/highgui/highgui.hpp"`
- `#include "opencv2/contrib/contrib.hpp"`
- `#include "opencv2/core/core_c.h"`
- `#include "opencv2/highgui/highgui_c.h"`
- `#include "opencv2/imgproc/imgproc_c.h"`

Namespace

All OpenCV classes and functions are in cv namespace. So, you have to do one of following

- Add the `'using namespace cv'` line just after including your header files (I have used this method in all my sample programs)

e.g.

```
#include "opencv2/core/core.hpp"
using namespace cv;
int main()
{
    Mat frame = cvQueryFrame( capture );
    imshow( "Video", frame );
}
```

- append the `cv::` specifier at the beginning of every OpenCV classes, functions and data structures in your source code

e.g.

```
#include "opencv2/core/core.hpp"

int main()
{
    cv::Mat frame = cvQueryFrame( capture );
    cv::imshow( "Video", frame );
}
```

Data Types for Arrays

Data type of an array defines the number of bits allocated for each element of array (pixels in an image) and how the value is represented using those bits. Any array elements should have one of following data types.

For single channel arrays :

SITE MAP

[Home](#)

OpenCV Lessons

.. [What is OpenCV?](#)
.. [Installing & Configuring v](#)
.. [Basics of OpenCV API](#)
.. [Read & Display Image](#)
.. [Capture Video from File o](#)
.. [Write Image & Video to Fi](#)
.. [Filtering Images](#)
..... [Change Brightness of In](#)
..... [Change Contrast of Ima](#)
..... [Histogram Equalization](#)
..... [Smooth / Blur Images](#)
.. [How to Add Trackbar](#)
.. [How to Detect Mouse Clic](#)
.. [Rotate Image & Video](#)
.. [Color Detection & Object](#)
.. [Shape Detection &Trackir](#)

Reference Books

About Me

GOOGLE+ FOLLOWERS

OpenCV Tutorials

Follow



639 have us in circles

782

FACEBOOK FOLLOWERS

[Like](#) [Share](#) 2,256 people
what your friends think

SEARCH THIS BLOG

- **CV_8U** (8 bit unsigned integer)
- **CV_8S** (8 bit signed integer)
- **CV_16U** (16 bit unsigned integer)
- **CV_16S** (16 bit signed integer)
- **CV_32S** (32 bit signed integer)
- **CV_32F** (32 bit floating point number)
- **CV_64F** (64 bit float floating point number)

e.g. : Here I have illustrated a single channel array with 8 bit unsigned integers. As the datatype of this array is 8 bit unsigned integers, each element should have a value from 0 to 255.

54	58	255	8	0
45	24	25	214	23
85	124	85	23	55
22	78	25	21	0
52	52	36	127	47

Single Channel Array

For multi channel arrays :

We can define all of above data types for multi channel arrays (supports up to 512 channels). Here I am going to show you how to define CV_8U data type for multi channel arrays.

- CV_8UC1 (single channel array with 8 bit unsigned integers)
- CV_8UC2 (2 channel array with 8 bit unsigned integers)
- CV_8UC3 (3 channel array with 8 bit unsigned integers)
- CV_8UC4 (4 channel array with 8 bit unsigned integers)
- CV_8UC(n) (n channel array with 8 bit unsigned integers (n can be from 1 to 512))

e.g. 1 : Here I have illustrated a 3 channel array with 8 bit unsigned integers. As the datatype is 8 bit unsigned integers, each element should have a value from 0 to 255. Because this is a 3 channel array, array consists of tuples with 3 elements. The first tuple is {54, 0, 34}, second tuple is {58, 78, 185} and so on.

54	58	255	8	0		
45	0	78	51	100	74	
85	47	34	185	207	21	36
22	20	148	52	24	147	123
52	36	250	74	214	278	41
	158	0	78	51	247	255
		72	74	136	251	74

3 Channel Arrays

e.g. 2 : Here I have illustrated a 2 channel array with 8 bit signed integers. As the datatype is 8 bit signed integers, each element should have a value from -128 to 127. Because this is a 2 channel array, array consists of tuples with 2 elements. The first tuple is {-85, -127}, second tuple is {25, 23} and so on.

-85	25	120	127	-128	
48	-127	23	48	-54	100
-50	76	52	24	-47	23
0	64	74	-14	78	41
72	-3	-78	51	47	55
	23	74	127	51	74

2 Channel Array

Note : CV_8U = CV_8UC1 = CV_8UC(1)

Example Usage :

- Mat img1(3, 5, CV_32F); //3 x 5 single-channel array with 32 bit floating point numbers
- Mat img2(23, 53, CV_64FC(5)); //23 x 53 5-channel array with 64 bit floating point numbers
- Mat img3(Size(100, 200), CV_16UC2); //100 x 200 2-channel array with 16 bit unsigned integers

Remember :

Some OpenCV functions can handle only a subset of above data types. So, be careful, when using OpenCV functions.

Bit Depths for IplImage (C style)

- **IPL_DEPTH_<bit_depth>(S|U|F)**
 - Here possible values for <bit_depth> are 1,8,16,32 and 64
 - S = Signed
 - U = Unsigned
 - F = Float
 - 1 bit depth images should be unsigned
 - 8 bit depth images should be signed or unsigned
 - 16 bit depth images should be signed or unsigned
 - 32 bit depth images should be signed or float
 - 64 bit depth images should be float
- E.g.:
 - IPL_DEPTH_1U (1 bit depth and unsigned)
 - IPL_DEPTH_8U (8 bit depth and unsigned)
 - IPL_DEPTH_16U
 - IPL_DEPTH_32F (32 bit depth and float)
 - IPL_DEPTH_8S
 - IPL_DEPTH_16S (16 bit depth and signed)
 - IPL_DEPTH_32S
 - IPL_DEPTH_64F

Bit depth means the number of bits allocated for a pixel. For example, IplImage with IPL_DEPTH_8U uses 8 bit unsigned integer per each pixel. That means each pixel can hold 0 to 255 integer numbers.

IPL_DEPTH_8U, IPL_DEPTH_8S, IPL_DEPTH_16S, IPL_DEPTH_32S, IPL_DEPTH_32F and IPL_DEPTH_64F are currently supported by IplImage data structure.

Next Tutorial : Read & Display Image

Previous Tutorial : Installing & Configuring with Visual Studio

Posted by Sermal Fernando



+2 Recommend this on Google

Is This Helpful : Yes (4) No (0)

26 comments:



Johnny June 6, 2013 at 5:08 AM

Thanks for the tutorial! I ran into a problem with this, however:

After following the steps in the previous tutorial, the header file cv.h was still not recognized by Visual Studio (2010 Express).

I had to change cv.h to
C:\opencv\build\include\opencv\cv.h
to successfully build the project.

Regards

[Reply](#)

[Replies](#)

arslan July 24, 2013 at 7:45 PM