OpenCV Overview

Nate Kent
nkent@iastate.edu

January 28, 2016

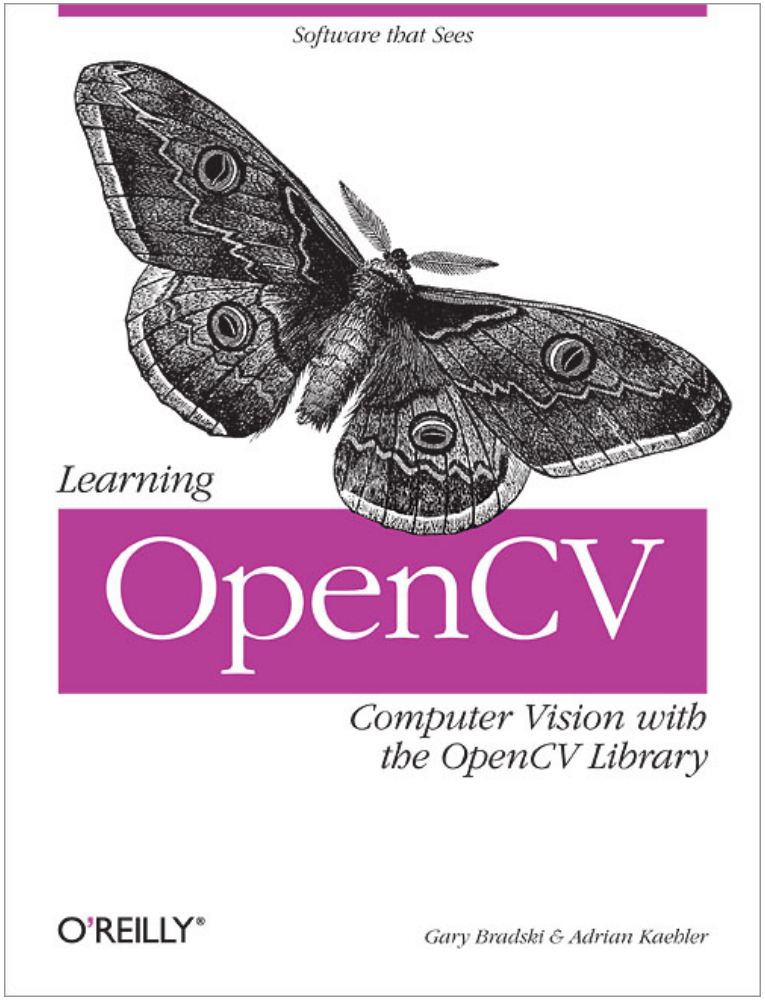
Table Of Contents

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
6. Sample Programs

Class Information

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
6. Sample Programs

Book - Learning OpenCV



Which Version?

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
6. Sample Programs

Differences

V1: Old version. Written in C.

V2: Recent version. Written in C++ with interfaces and wrappers for other language. Most tutorials will be focused on this version. Use this version unless you have a good reason to use one of the other two.

V3: Released June 2015. Uses newer C++ features with new algorithms. Small differences in API from V2.

Interfaces and Wrappers

Interfaces

- Python
- Java
- MATLAB/OCTAVE (v2.5 and after)

Wrappers

- C#
- Perl
- Ch
- Ruby

Tutorials

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
6. Sample Programs

Where To Find Tutorials

A huge number of tutorials are available on the OpenCV Website.

docs.opencv.org/doc/tutorials/tutorials.html

Installing OpenCV

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
6. Sample Programs

Windows

This method requires the latest Microsoft Visual Studio IDE.

1. Download OpenCV binaries
2. Extract along the build path
3. Set the OpenCV environment variable. This can be done in the command prompt:

```
setx -m OPENCV_DIR D:\OpenCV\Build\x86\vc10
setx -m OPENCV_DIR D:\OpenCV\Build\x64\vc10

setx -m OPENCV_DIR D:\OpenCV\Build\x86\vc11
setx -m OPENCV_DIR D:\OpenCV\Build\x64\vc11
```

OSX

1. Download the OpenCV source code
2. Install CMake
3. Open the terminal and navigate to the OpenCV folder
4. In the terminal, run the following commands:

```
$ cmake -G "Unix Makefiles"
$ make -j8 && make install
```

Linux

Do it through your package manager. I believe almost all distros have OpenCV available.

Building With OpenCV

Tutorials for your specific IDE are available online.
Warning: I've only ever used OpenCV's `C` and `C++` interfaces.

How It All Works

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
   - An Example Program
   - The `Mat` Class
   - Windows And Trackbars
   - Example Operations
   - Working With Videos And Cameras
6. Sample Programs

A Complete Example Program

Link!

The Mat Class

Mat is simply a header — not the actual data!

```
// Creates just the header parts
Mat A, C;
// Here we'll know the method used (allocate matrix)
A = imread(argv[1], CV_LOAD_IMAGE_COLOR);
// Use the copy constructor
Mat B(A);
// Assignment operator
C = A;
```

Creating Submatrices

These don't allocate any new data. They simply point the header to portions of a matrix that is already in memory.

```
// Using a rectangle
Mat D (A, Rect(10, 10, 100, 100) );
// Using row and column boundaries
Mat E = A(Range::all(), Range(1,3));
```

Duplicating Mats

These actually allocate a new matrix.

```
// Clone method
Mat F = A.clone();
// Copy method
Mat G;
A.copyTo(G);
```

Explicitly Creating Mats

```
// Constructor
Mat M(2,2, CV_8UC3, Scalar(0,0,255));
// Via initializers
int sz[3] = {2,2,2};
Mat L(3,sz, CV_8UC(1), Scalar::all(0));
// Create function
M.create(4,4, CV_8UC(2));
// Matlab style
Mat E = Mat::eye(4, 4, CV_64F);
Mat O = Mat::ones(2, 2, CV_32F);
Mat Z = Mat::zeros(3,3, CV_8UC1);
```

Creating And Showing In A Named Window

```
Mat image;

// Create the window
namedWindow( "Display Window", WINDOW_AUTOSIZE);

// Display in the window
imshow( "Display Window", image );

// Wait for the user to press a key
waitKey(0);
```

Creating Trackbars

```
String name = "Trackbar Name";
String windowName = "Display Window";
int value; // Initial position
int count; // Max position
void* userdata; // Userdata

void Callback( int, void* );

// Create our trackbar
createTrackbar(name, windowName, value,
                    count, Callback, userdata);
```

Drawing On The Image

There are many functions used to draw.

http://docs.opencv.org/modules/core/doc/drawing_functions.html

Thresholding

A source matrix can be the destination matrix!

```
Mat src, dst;
double threshold;
double maxval;
int type = THRESH_BINARY; // Many different types

threshold(src, dst, threshold, maxval, type);
```

Mathematical Operations On Mat

```
Mat srcA, srcB, dst;
Mat mask;
int depth; // Many defined

// Addition
add(srcA, srcB, mask, depth);

// Finding the determinant
determinant(srcA);

// Flip the matrix
// 0 is vertical, > 0 horizontal, < 0 both
int flipcode;
flip(srcA, dst, flipcode);
```

Working With Videos And Cameras

```
// Try to open based on video file
VideoCapture cap(arg);

// Try to open a camera based on index
cap.open(0);

// Process each frame like a normal image
Mat frame;
for(;;) {
    // Put the frame in the matrix
    cap >> frame;

    // ... do your thing
}

// Write the last frame to a file
imwrite(frame, "lastframe.jpg");
```

Sample Programs

1. Class Information
2. Which Version?
3. Tutorials
4. Installing OpenCV
5. How It All Works
   - An Example Program
   - The `Mat` Class
   - Windows And Trackbars
   - Example Operations
   - Working With Videos And Cameras
6. Sample Programs

Sample Code Location

Sample code is downloaded along with the rest of OpenCV for Windows and OSX. Linux users may have to install a separate package. They can be found in: `$OPENCV_HOME/samples`

Let's Do Some Samples

bgfg_gmg and **bgfg_segm**: Background filters
convexhull*: Creating convex hull around a set of points
demhist*: Creates a histogram of the image
drawing: Basic drawing commands
edge: Canny edge detection
fback and **lkdemo**: Optical flow
fitellipse: Fit an ellipse around a shape
gencolors*: Generate a set of *n* easily distinguishable colors
houghcircles and **houghlines**: Detect circles and lines
image: Open and display an image
kalman: Create an use a Kalman filter
minarea: Find minimum enclosing box and circle
morphology2: Erosion and dilation
squares: Square detection
starter_video and **video_dmtx**: Load, modify, display, and save videos

convexhull

```
// Lines 28 to 40
vector<Point> points;

for( i = 0; i < count; i++ )
{
    Point pt;
    pt.x = rng.uniform(img.cols/4, img.cols*3/4);
    pt.y = rng.uniform(img.rows/4, img.rows*3/4);

    points.push_back(pt);
}

vector<int> hull;
convexHull(Mat(points), hull, true);
```

demhist

```
// Lines 39 to 46
Mat dst, hist;
image.convertTo(dst, CV_8U, a, b);
imshow("image", dst);

calcHist(&dst, 1, 0, Mat(), hist, 1, &histSize, 0);
Mat histImage = Mat::ones(200, 320, CV_8U)*255;
```

gencolors

```
// Lines 21 to 23
vector<Scalar> colors;
theRNG() = (uint64)time(0);
generateColors( colors, colorsCount );
```

Questions?