

A very brief Introduction to OpenCV

Introduction

- Open Source Computer Vision (OpenCV) library aims at real time computer vision applications.
 - Object Identification
 - Segmentation
 - Face Recognition
 - Motion Tracking
 - Mobile Robotics

Download and Setup

- Download
 - You can download the library from the homepage: <http://opencv.org/>
 - The current stable versions are 3.0/2.4.11, but you may choose any version > 2.1 to do the homework.
- Setup
 - We will demonstrate how to set up OpenCV 3.0 in Visual Studio 2013.
 - For users of other OSs, please refer to:
 - Linux: http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html
 - Ubuntu: <http://milq.github.io/install-opencv-ubuntu-debian/>
 - Mac OS X: <http://blogs.wcode.org/2014/10/howto-install-build-and-use-opencv-macosx-10-10/>

Resources

- Tutorial
 - <http://docs.opencv.org/doc/tutorials/tutorials.html>
 - http://docs.opencv.org/master/d9/df8/tutorial_root.html
 - <http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/index.html>
- Documents
 - <http://docs.opencv.org>
 - <http://docs.opencv.org/master/index.html>
 - `$OpencvDirectory/sources/doc/tutorials/tutorials.markdown`

Useful Functions: Image reading

- `cv::Mat`
 - Matrix structure for the OpenCV
 - `Mat Mat::clone()`
 - Duplicate a image of itself and return it.
- `cv::Mat cv::imread(filename, color_flag=1)`
 - Load a image by the filename. `Color_flag` specifies the color type of the image.
- `bool cv::imwrite(filename, img_mat)`
 - Save a image with the filename.

Useful Functions: User Interface

- `void cv::namedWindow(window_name, flags=WINDOW_AUTOSIZE)`
 - Create a window with the `window_name`.
- `void cv::imshow(window_name, img_mat)`
 - Show the `img_mat` in the window with that `window_name`.
- `int cv::waitKey(delay=0)`
 - `delay(ms)`: how long the window will be shown. 0 means forever.
- `void cv::destroyWindow(window_name)`
 - Destroy the window with the `window_name`.

Useful Functions: Image Processing

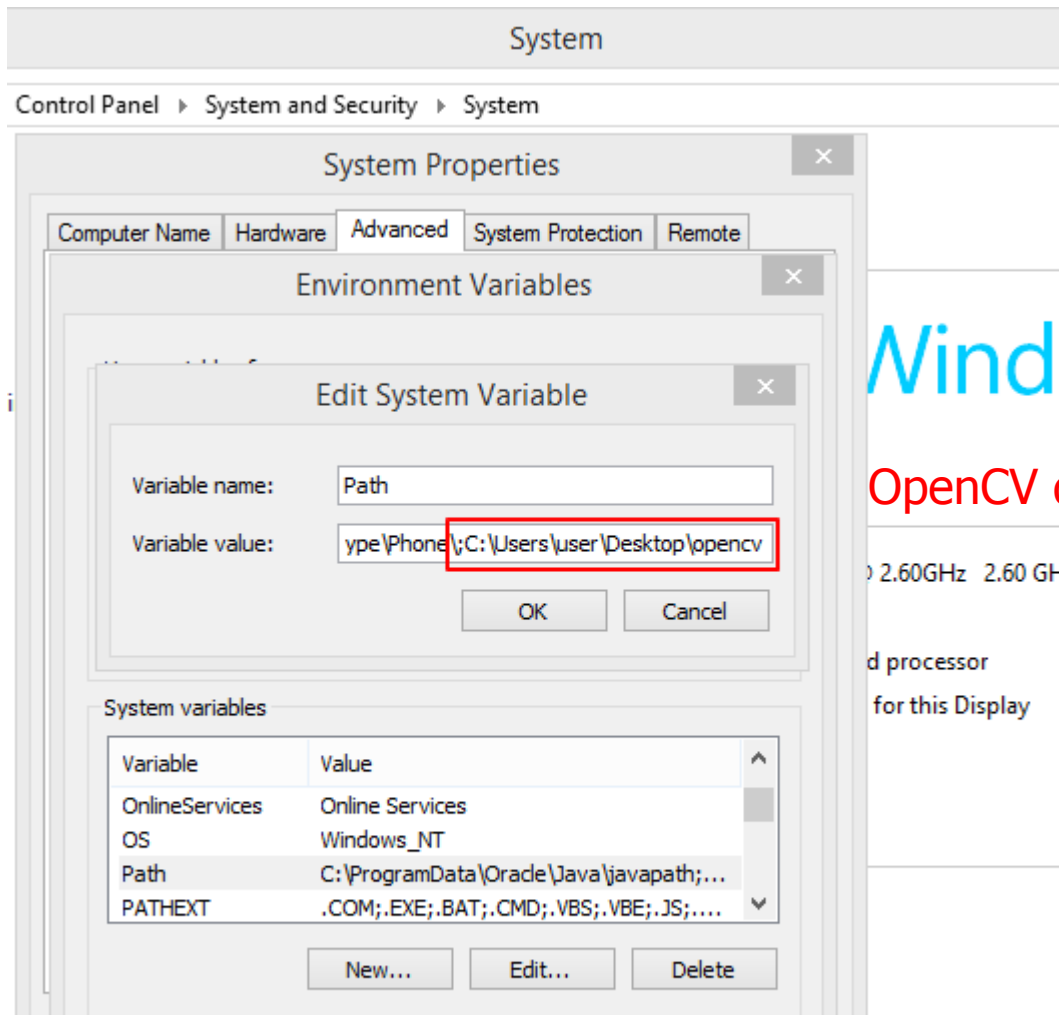
- `void GaussianBlur(src, dst, ksize, sigmaX, sigmaY)`
 - Blurs an image using a Gaussian filter.
- `double threshold(src, dst, thresh, maxval, type)`
 - Applies a fixed-level threshold to each array element.
- `void erode(src, dst, kernel)`
 - Erodes an image by using a specific kernel.
- `void dilate(src, dst, kernel)`
 - Dilates an image by using a specific kernel.

Useful Functions

- If you're familiar with OpenCV, you should try the following functions:
- `cvFindContours()`
- `cvMoments()`
- `cvGetCentralMoment()`

System Configurations

- Environment Variables



Windows

OpenCV directory path

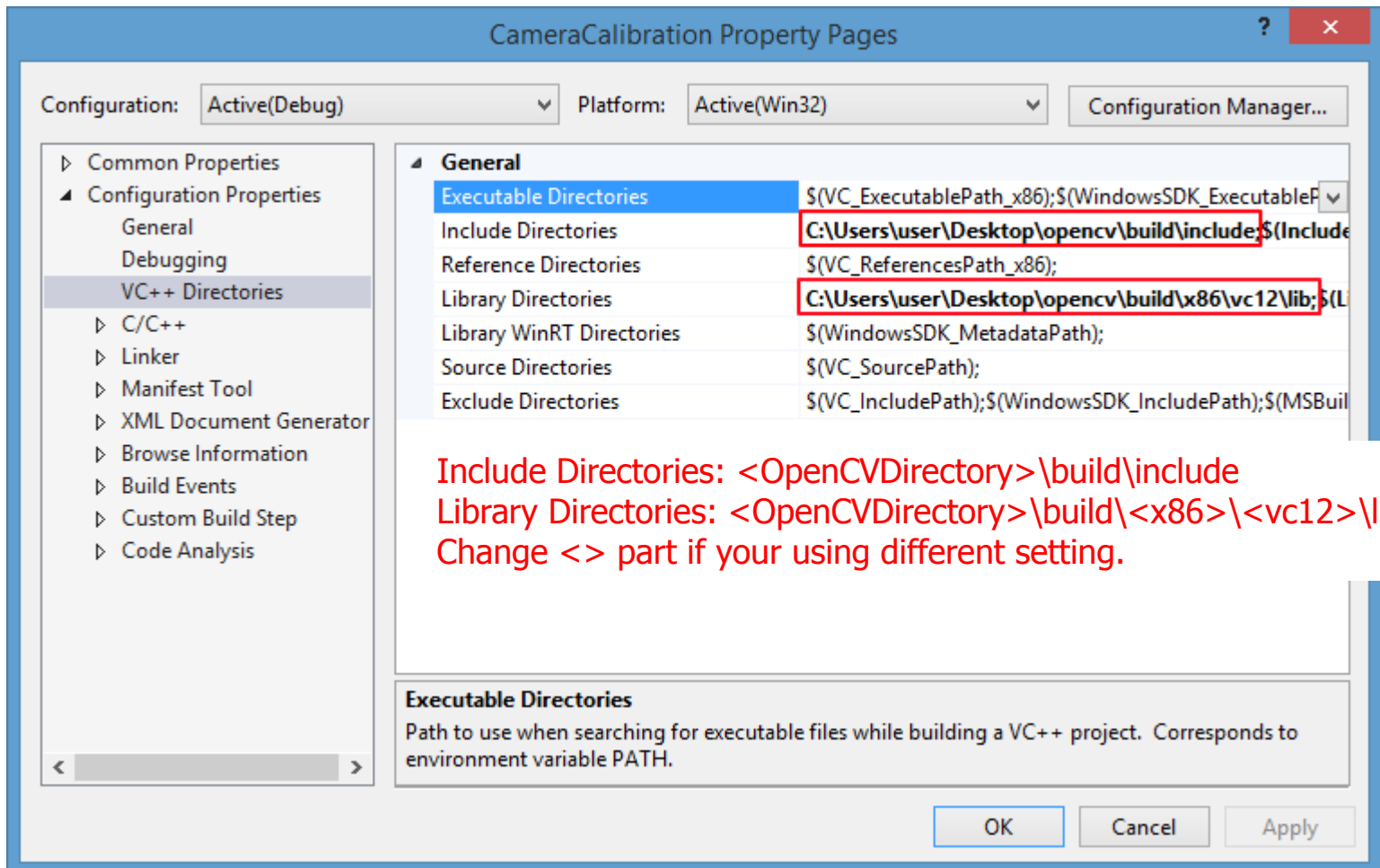
2.60GHz 2.60 GHz

processor

for this Display

Visual Studio Configurations

- Directory Path



Visual Studio Configurations

- Linker->Input: opencv_world300d.lib (debug)
opencv_world300.lib (release)

