**INSTALLATION**

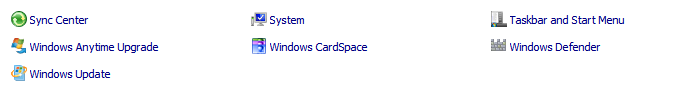
* **OpenCV 2.4.11 and C++ Using Windows 7**
* **Visual Studio 2013 (Community Edition)**
* **Precompiled binaries**

1) Download and install Visual Studio 2013 Community Edition (yes, it’s free, choosing all default options will work fine)

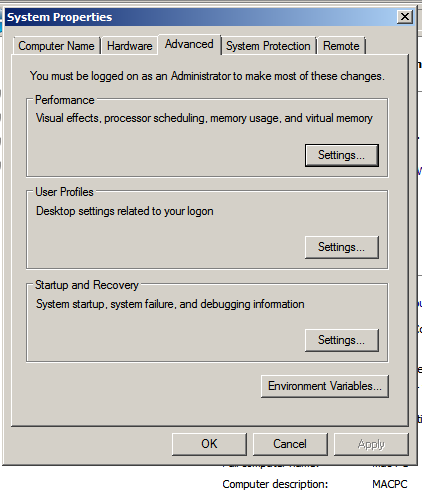
2) Download OpenCV 2.4.11

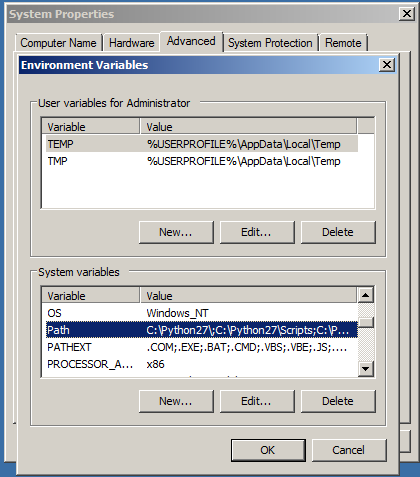
3) Make a folder "C:\OpenCV-2.4.11" and extract OpenCV 2.4.11 to there

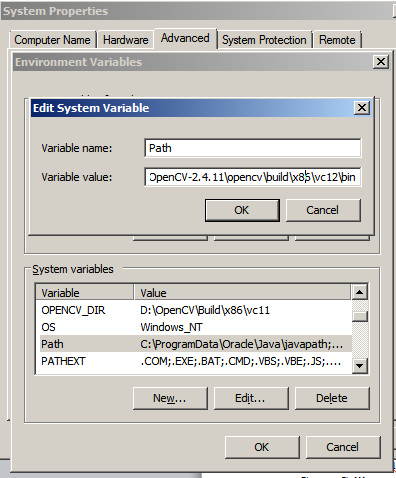
4a) Add the bin directory to the operating system PATH: C:\OpenCV-2.4.11\opencv\build\x86\vc12\bin 4b) Pull up a Command Prompt and verify the bin directory is now in PATH, then reboot

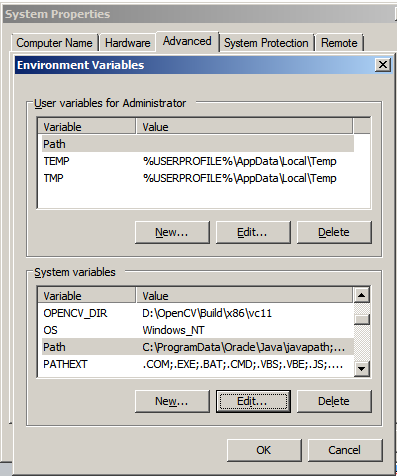








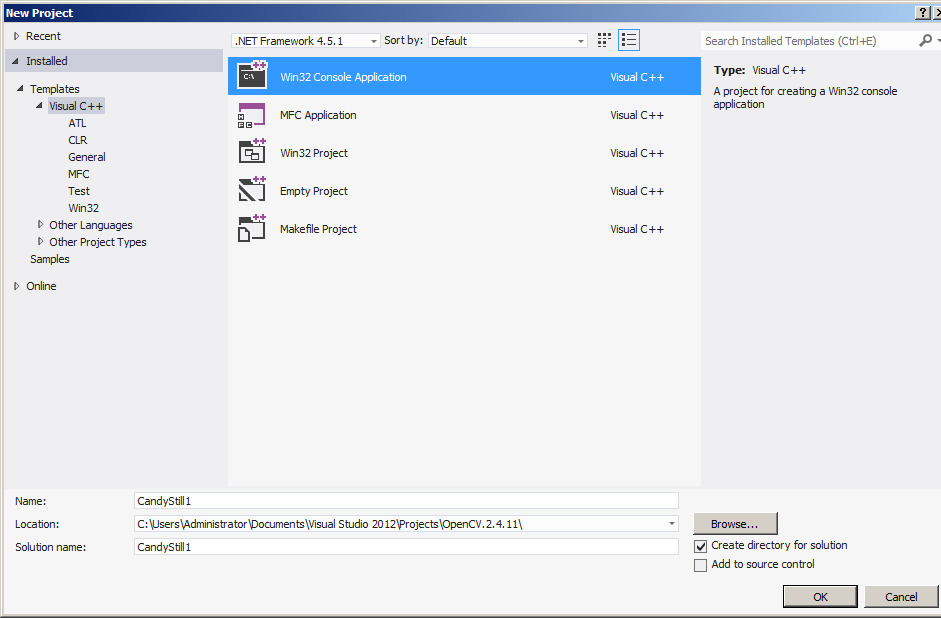


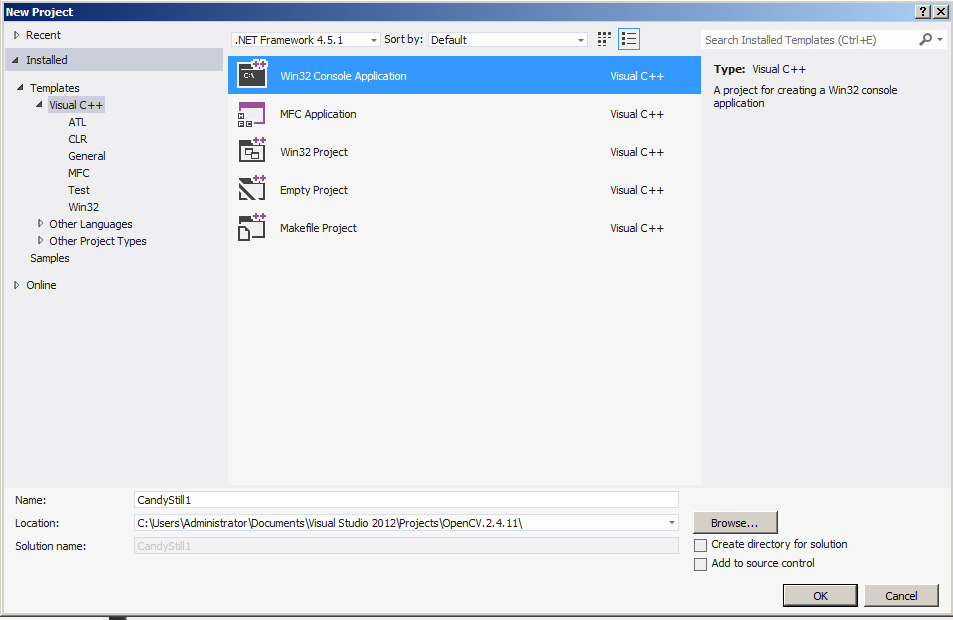


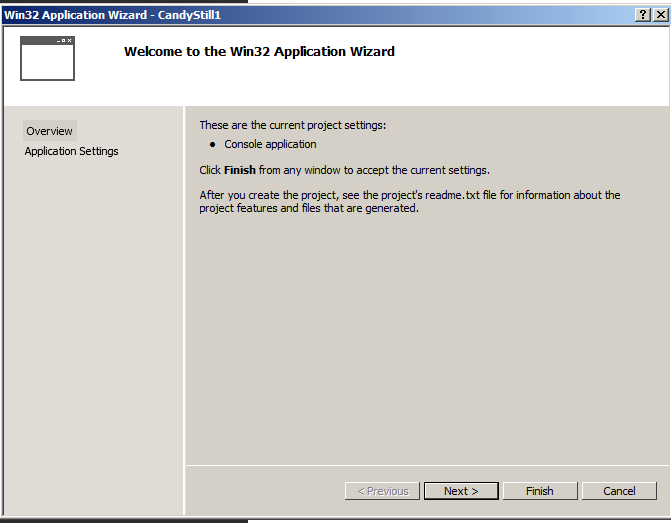
5) We are going to use: CannyStill.cpp (uses a still image) CannyWebcam.cpp (uses a webcam) RedBallTracker.cpp (tracks a red ball, uses a webcam)

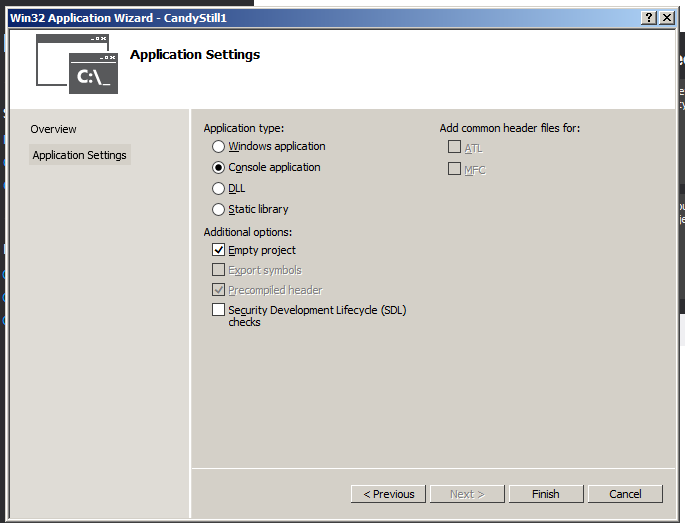
If you are going through this for the first time I suggest CannyStill.cpp

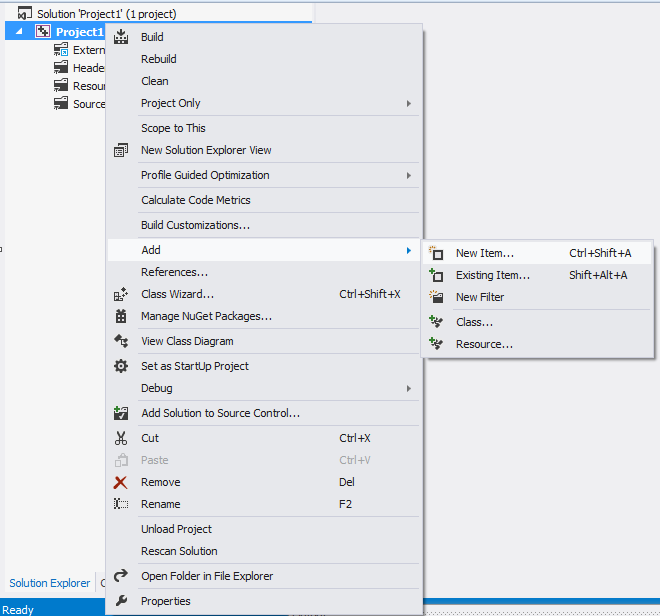
6a) Start Visual Studio 2013, make a new project 6b) Choose Visual C++, Win32 Console Application, name as you prefer, ex "CannyStill1", set preferred location, uncheck "Create directory for solution" and "Add to source control", choose OK 6c) On the “Welcome to the Win32 Application Wizard” screen choose Next 6d) On the "Application Settings" screen, uncheck "Precompiled Header" and "Security Development", check "Empty Project", and verify "Console application" radio button is checked, then choose Finish



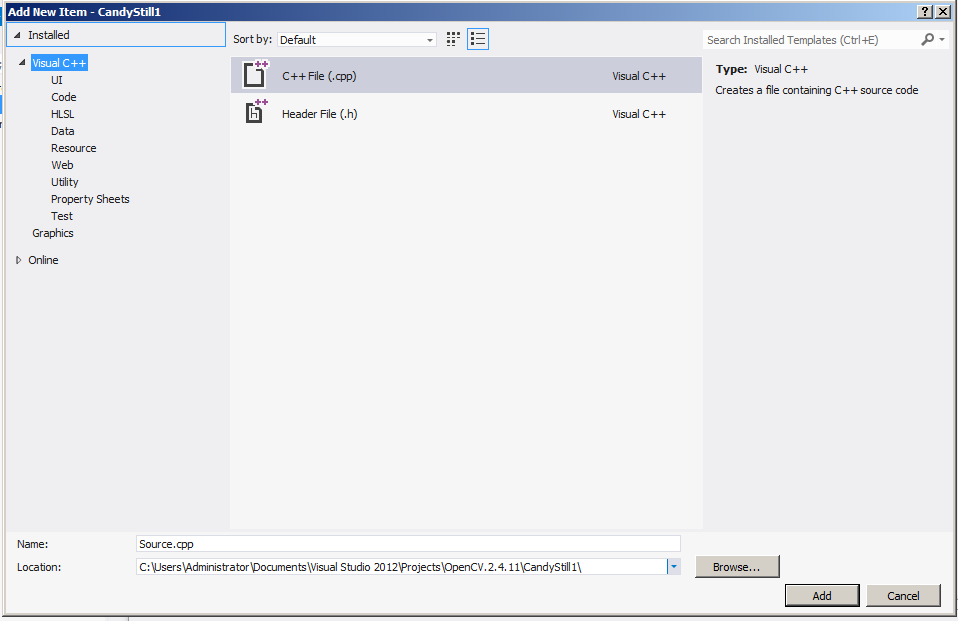


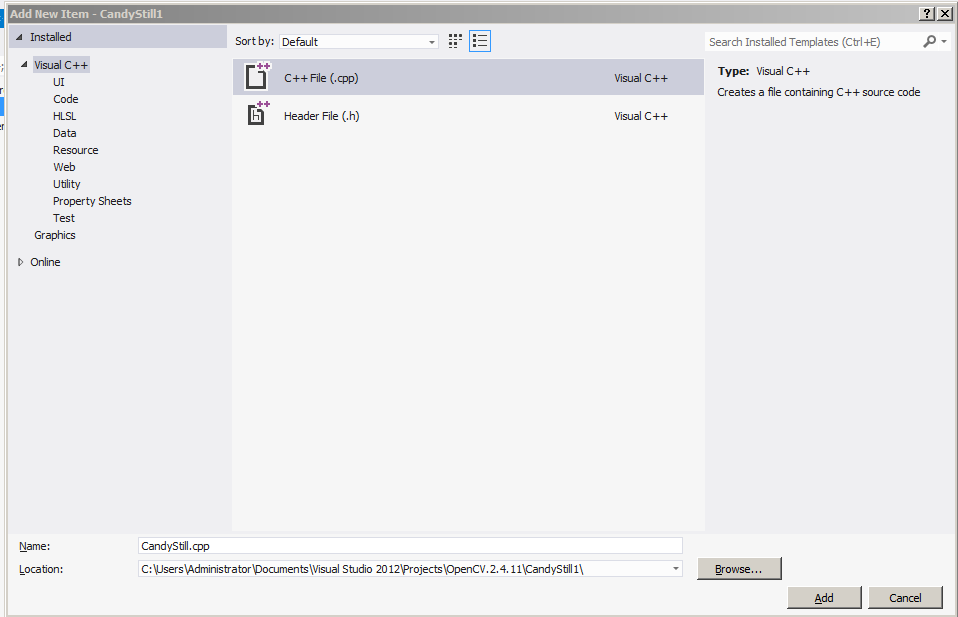






Add New item





7a) Right click in Solution Explorer, choose Add -> New Item 7b) Choose "C++ File", name the C++ file as preferred, ex. "Project1.cpp", choose "Add" 7c) Copy/paste the entire code from your chosen example into the .cpp file (At this point Visual Studio will underline many of the lines of code with red because we have not yet informed Visual Studio as to the location of OpenCV, subsequent steps will resolve this)

CODE

// Project1.cpp

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

#include<opencv2/imgproc/imgproc.hpp>

#include<iostream>

///////////////////////////////////////////////////////////////////////////////////////////////////

int main() {

cv::Mat matOriginal; // input image

cv::Mat matGrayscale; // grayscale of input image

cv::Mat matBlurred; // intermediate blured image

cv::Mat matCanny; // Canny edge image

matOriginal = cv::imread("image.jpg"); // open image

if (matOriginal.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(matOriginal, matGrayscale, CV\_BGR2GRAY); // convert to grayscale

cv::GaussianBlur(matGrayscale, // input image

matBlurred, // 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(matBlurred, // input image

matCanny, // output image

100, // low threshold

200); // high threshold

// declare windows

cv::namedWindow("Original", CV\_WINDOW\_AUTOSIZE); // note: you can use CV\_WINDOW\_NORMAL which allows resizing the window

cv::namedWindow("Canny", CV\_WINDOW\_AUTOSIZE); // or CV\_WINDOW\_AUTOSIZE for a fixed size window matching the resolution of the image

// CV\_WINDOW\_AUTOSIZE is the default

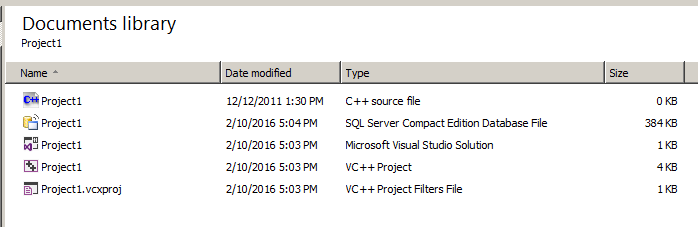
cv::imshow("Original", matOriginal); // show windows

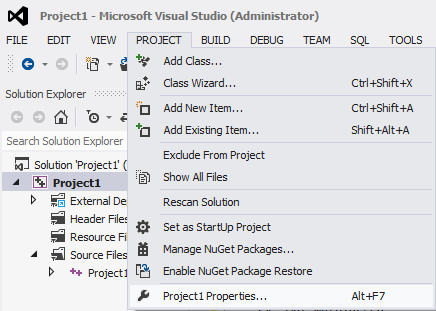
cv::imshow("Canny", matCanny);

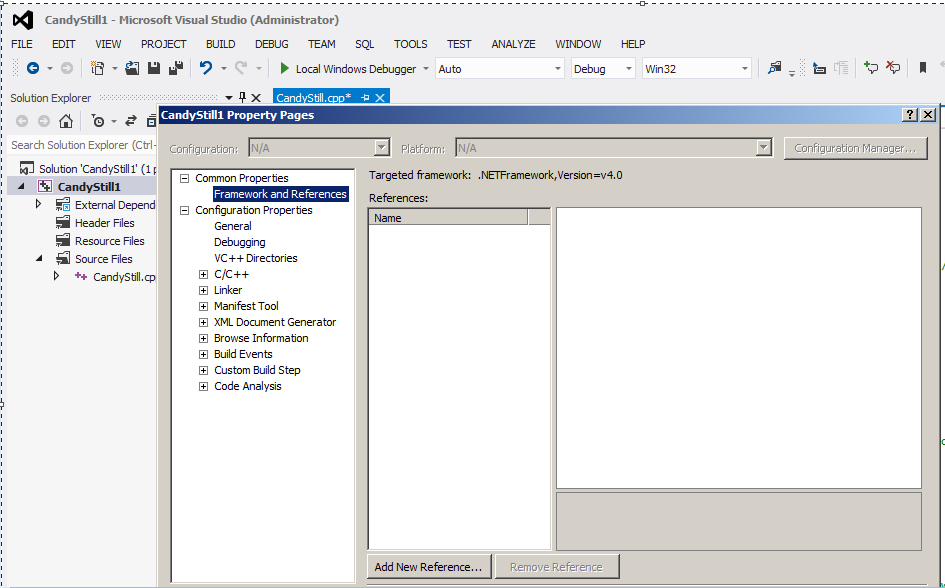
cv::waitKey(0); // hold windows open until user presses a key

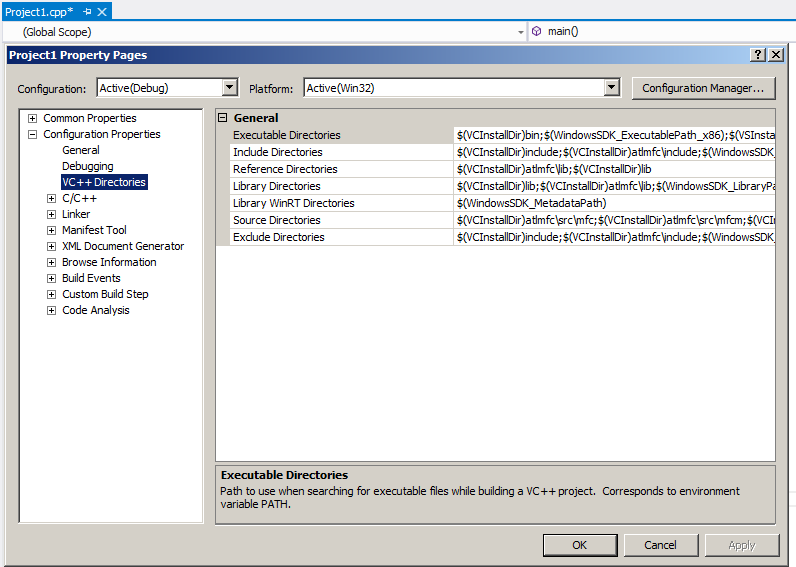
return(0);

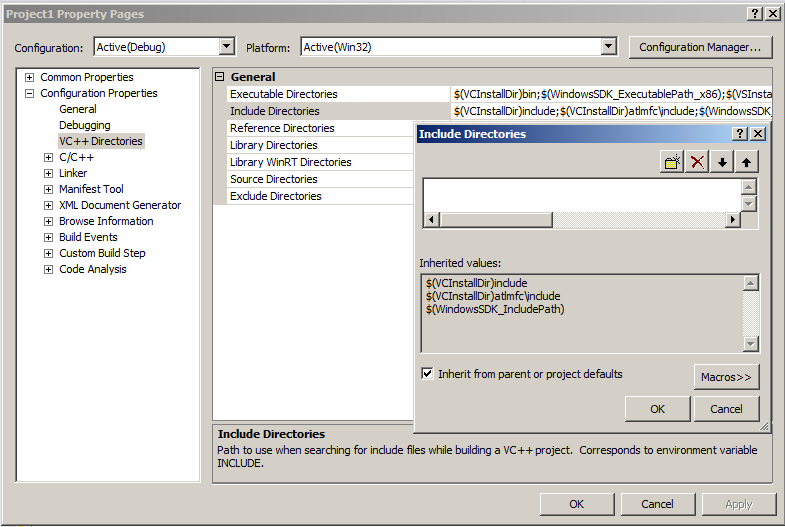
}



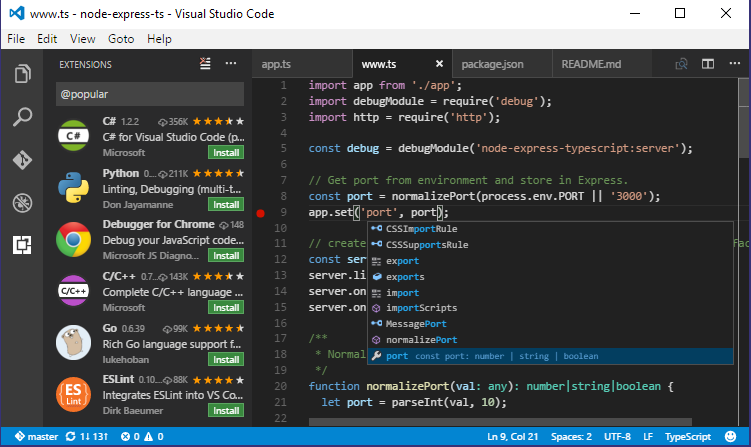




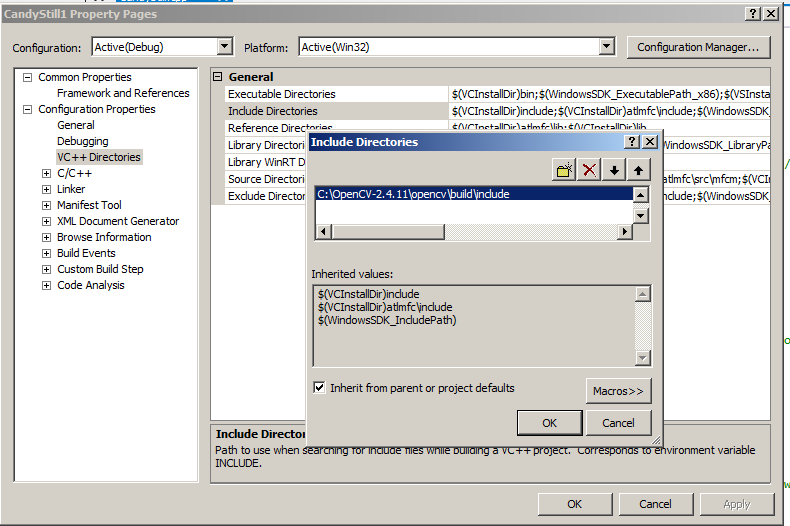




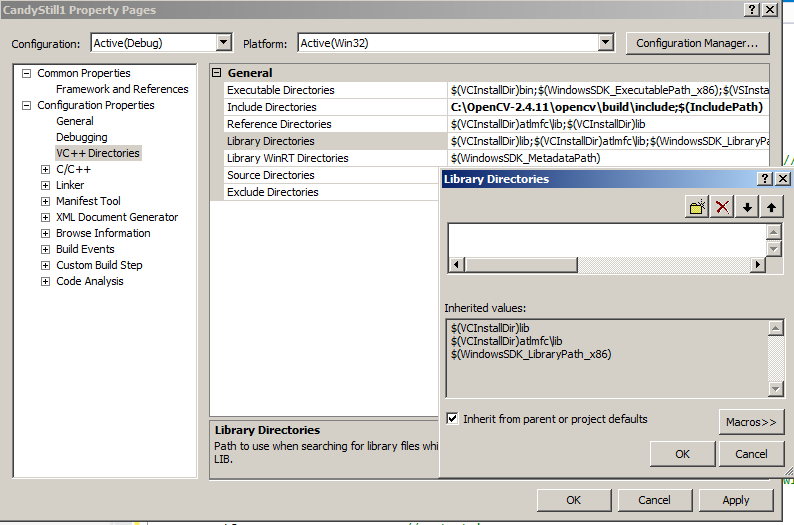
8) If you are using an example with a still image (i.e. Project1.cpp), copy any JPEG image into the project directory and rename it "image.jpg".

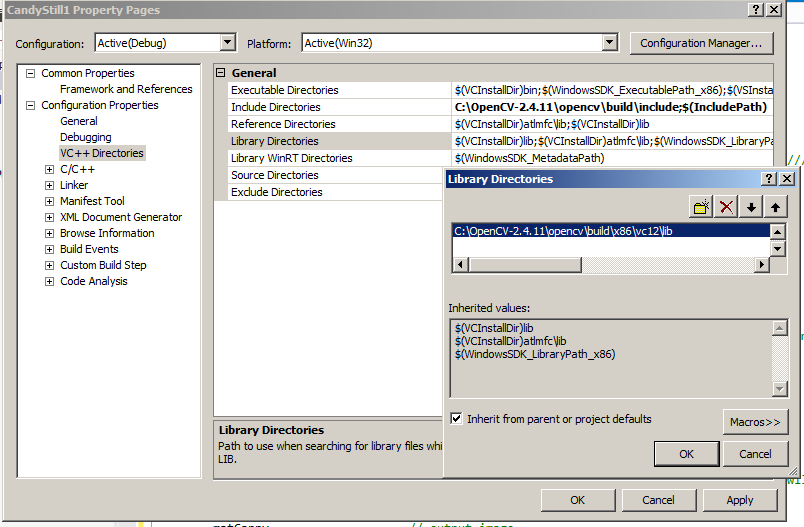


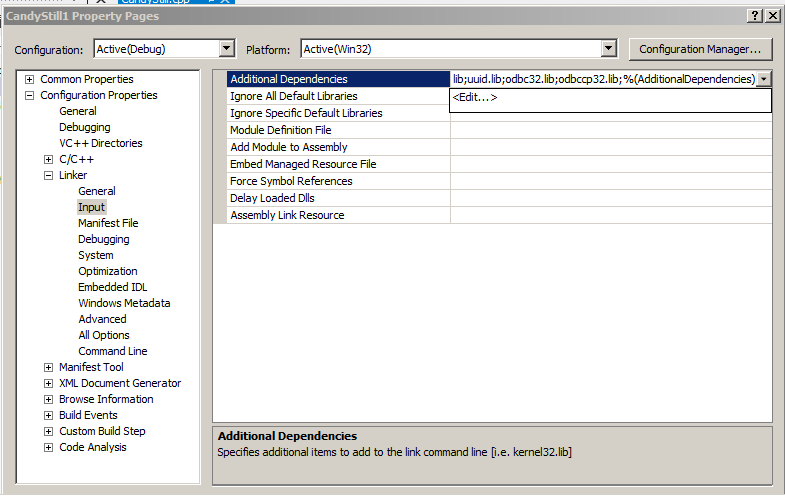
9) In VS go to: Project -> Properties -> Configuration Properties -> VC++ Directories -> Include Directories add the include directory: C:\OpenCV-2.4.11\opencv\build\include



10) In VS go to: Project -> Properties -> Configuration Properties -> VC++ Directories -> Library Directories: add the library directory: C:\OpenCV-2.4.11\opencv\build\x86\vc12\lib

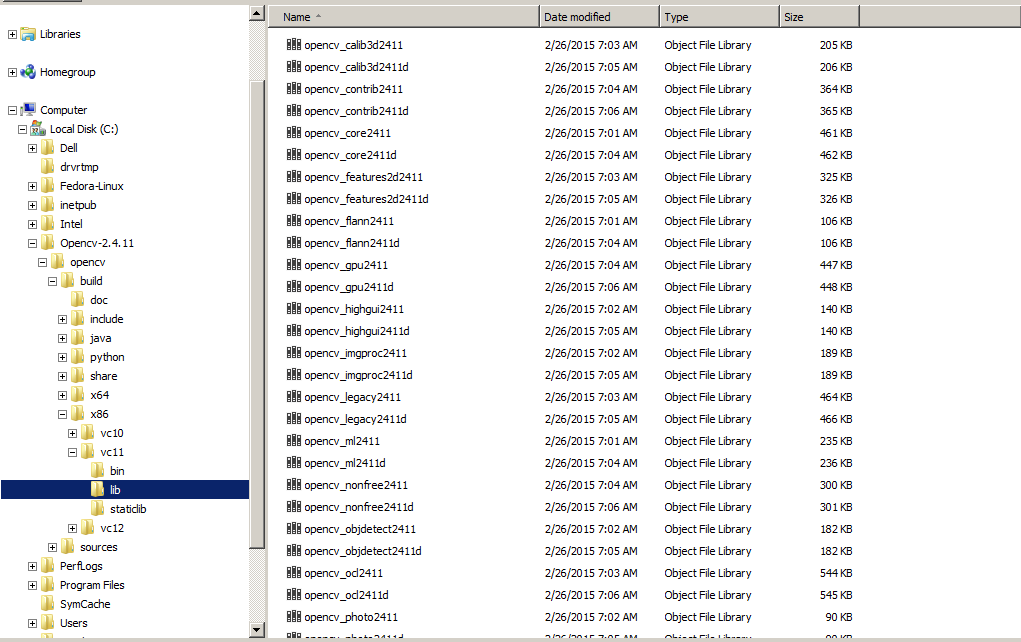






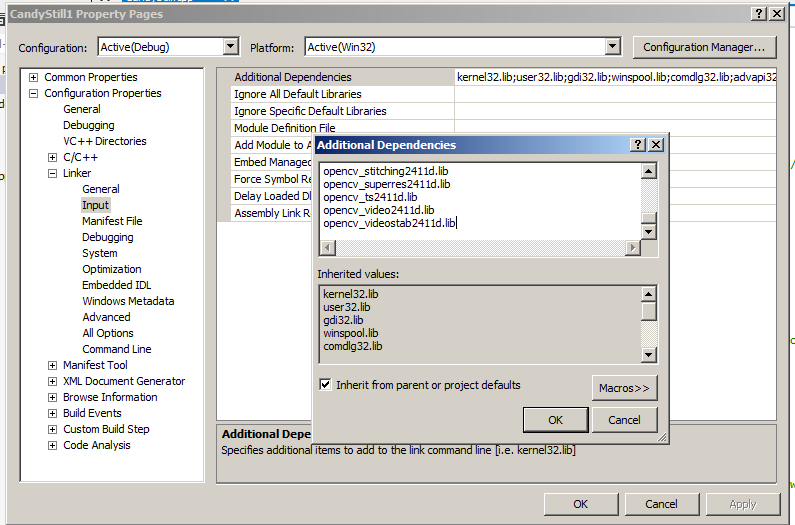
11) If you currently do not have Windows 7 configured to allow viewing / editing of file extensions, go to: Start -> Control Panel -> View by: Large icons -> Folder Options -> View tab -> uncheck "Hide extensions for known file types" (if you already have viewing file extensions enabled then skip this step).

12) In Windows Explorer (not within Visual Studio), navigate to the lib directory: C:\OpenCV-2.4.11\opencv\build\x86\vc12\lib



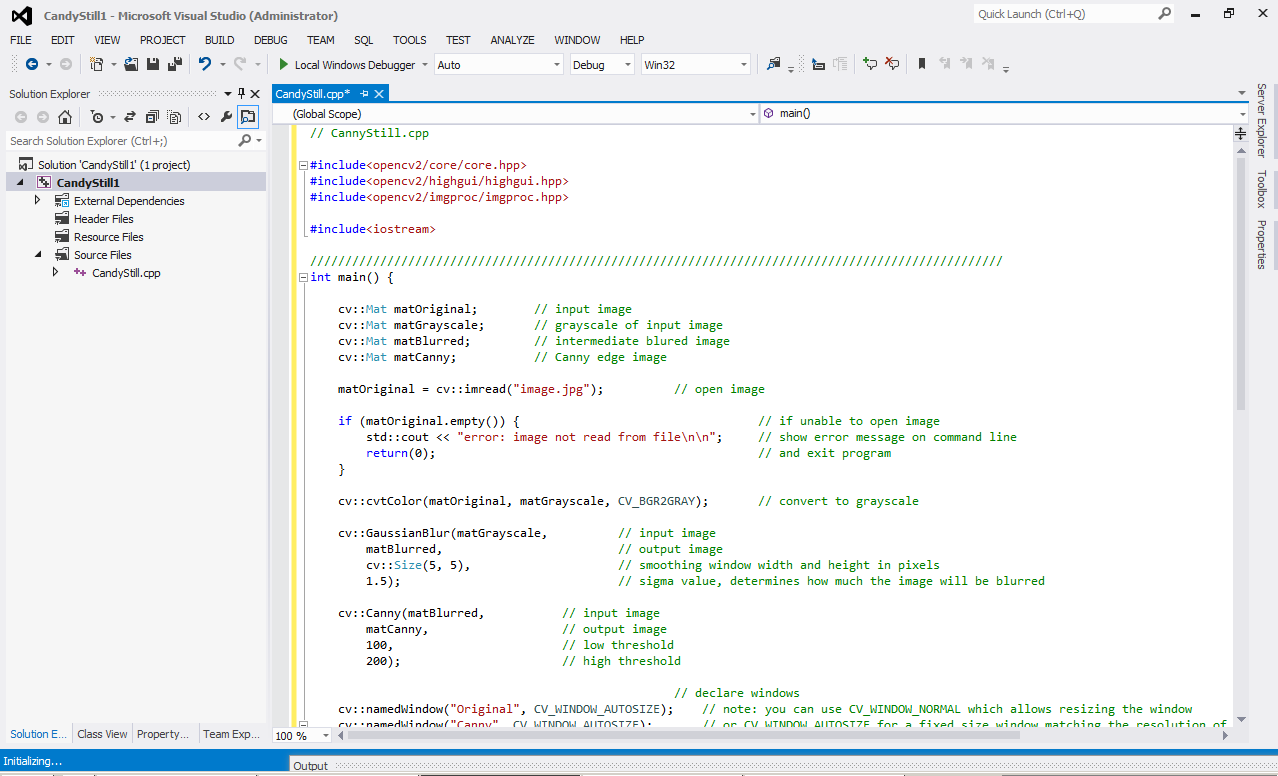
Verify the debug libs (ending with a 'd') listed are the same as this list:

opencv\_calib3d2411d.lib opencv\_contrib2411d.lib opencv\_core2411d.lib opencv\_features2d2411d.lib opencv\_flann2411d.lib opencv\_gpu2411d.lib opencv\_highgui2411d.lib opencv\_imgproc2411d.lib opencv\_legacy2411d.lib opencv\_ml2411d.lib opencv\_nonfree2411d.lib opencv\_objdetect2411d.lib opencv\_ocl2411d.lib opencv\_photo2411d.lib opencv\_stitching2411d.lib opencv\_superres2411d.lib opencv\_ts2411d.lib opencv\_video2411d.lib opencv\_videostab2411d.lib



Then in Visual Studio copy/paste this list of libs into: Project -> Properties -> Configuration Properties -> Linker -> Input -> Additional Dependencies

13) In the Visual Studio toolbar, verify that "Solution Configurations" and "Solution Platforms" are set to "Debug" and "Win32", respectively



14) Run the program, either without debugging (choose Debug, then the hollow green arrow, or press Ctrl+F5) or with debugging (solid green arrow or press F5)