Multimedia
(Lab 01)

Spring, 2020

Yong Ju Jung (정용주)

미래창조과학부 SW중심대학
가천대학교 소프트웨어학과
Gachon University Department of Software

[Lab01-1] Introduction to OpenCV

- Install Visual C++ and OpenCV library in your computer

- Load an image and display it on your screen

What is OpenCV

- OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.

- Online documentation : https://docs.opencv.org/4.2.0/
- Tutorial : https://docs.opencv.org/4.2.0/d3/d81/tutorial_contrib_root.html

OpenCV

- Install Visual C++ and OpenCV library in your computer
  - OpenCV4.2 – vc15 (Visual Studio2017), vc14 (VS2015)
  - OpenCV4.0 – vc15 (Visual Studio2017), vc14 (VS2015)
  - OpenCV3.4 – vc15 (VS2017), vc14 (VS2015)
  - OpenCV3.2 – vc14 (VS2015)
  - OpenCV3.1 – vc14 (VS2015), vc12 (VS2013)
  - OpenCV3.0 – vc12, vc11 (VS2012)

  - If you are using x86 computer, you have to install OpenCV3.0 prebuild or you have to build the source code by yourself.

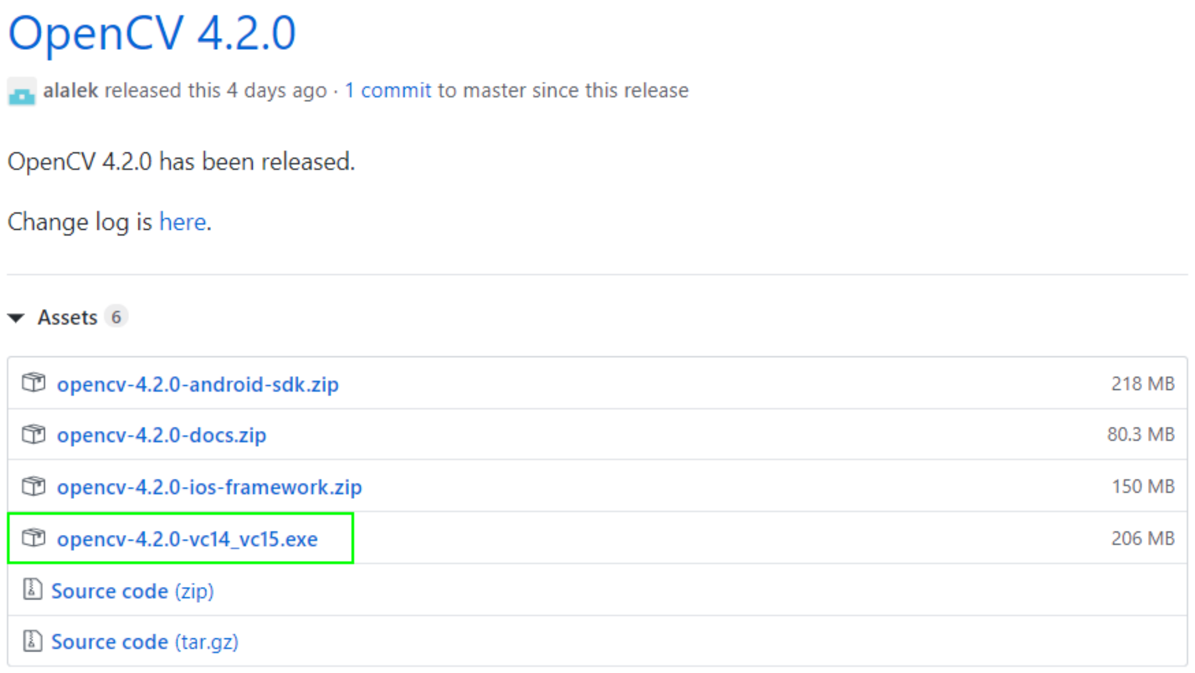
OpenCV Tutorials

- https://docs.opencv.org/4.2.0/d3/d81/tutorial_contrib_root.html
- **Introduction to OpenCV**
    - You will learn how to setup OpenCV on your computer!
- **The Core Functionality (core module)**
    - Here you will learn the about the basic building blocks of the library. A must read and know for understanding how to manipulate the images on a pixel level.

How to setup OpenCV

How to setup OpenCV

- **Step 1: Download and extract the pre-built library**
  - Release files for 4.2.0 are listed at https://github.com/opencv/opencv/releases/tag/4.2.0.



OpenCV 4.2.0

alalek released this 4 days ago · 1 commit to master since this release

OpenCV 4.2.0 has been released.

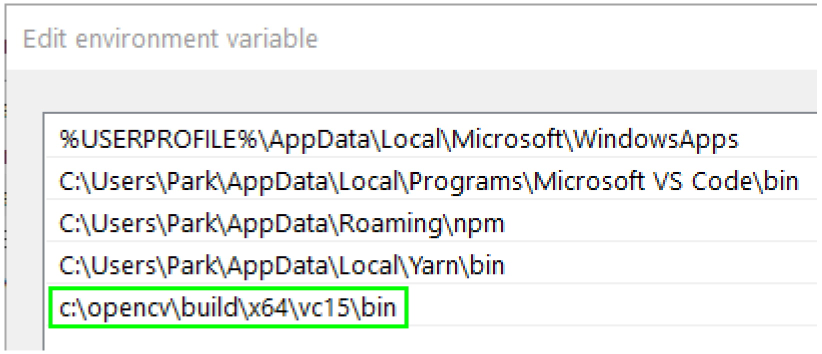
Change log is here.

Assets 6

opencv-4.2.0-android-sdk.zip	218 MB
opencv-4.2.0-docs.zip	80.3 MB
opencv-4.2.0-ios-framework.zip	150 MB
opencv-4.2.0-vc14_vc15.exe	206 MB
Source code (zip)	
Source code (tar.gz)	

How to setup OpenCV

- **Step 2: Add to path**
  - Add opencv's bin directory to path.



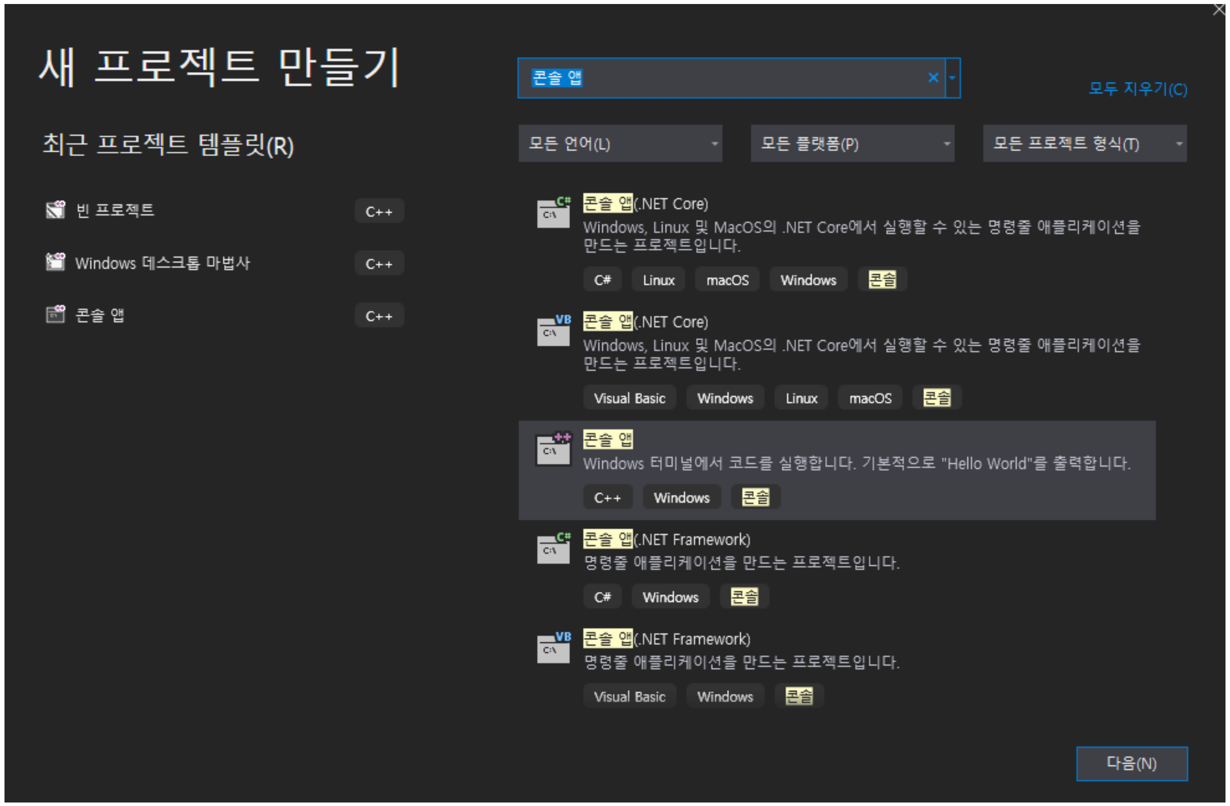
Edit environment variable

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\Park\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\Park\AppData\Roaming\npm
C:\Users\Park\AppData\Local\Yarn\bin
c:\opencv\build\x64\vc15\bin
```

환경변수에 이걸 써줘야함. v14인지 v15인

How to setup OpenCV

- **Step 3: Create a project in Visual Studio 2019**



새 프로젝트 만들기

최근 프로젝트 템플릿(R)

빈 프로젝트	C++
Windows 데스크톱 마법사	C++
콘솔 앱	C++

콘솔 앱

모두 지우기(C)

모든 언어(L) · 모든 플랫폼(P) · 모든 프로젝트 형식(T)

콘솔 앱(.NET Core)
Windows, Linux 및 MacOS의 .NET Core에서 실행할 수 있는 명령줄 애플리케이션을 만드는 프로젝트입니다.
C# · Linux · macOS · Windows · 콘솔

콘솔 앱(.NET Core)
Windows, Linux 및 MacOS의 .NET Core에서 실행할 수 있는 명령줄 애플리케이션을 만드는 프로젝트입니다.
Visual Basic · Windows · Linux · macOS · 콘솔

콘솔 앱
Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.
C++ · Windows · 콘솔

콘솔 앱(.NET Framework)
명령줄 애플리케이션을 만드는 프로젝트입니다.
C# · Windows · 콘솔

콘솔 앱(.NET Framework)
명령줄 애플리케이션을 만드는 프로젝트입니다.
Visual Basic · Windows · 콘솔

다음(N)

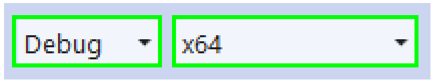
Choose **“콘솔 앱”**

How to setup OpenCV

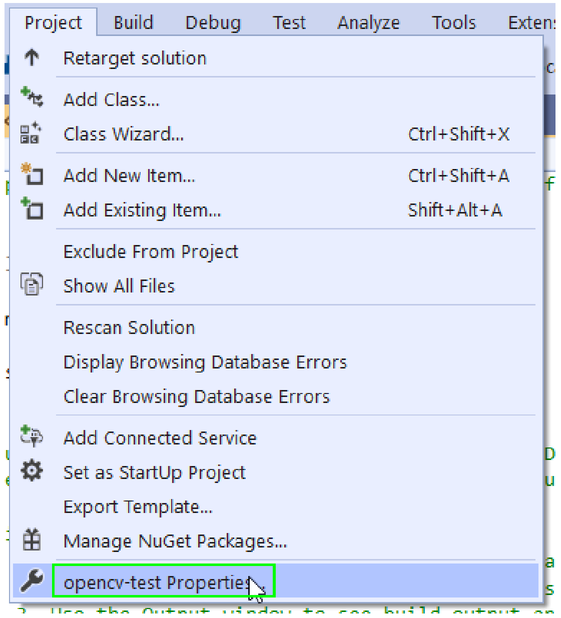
- **Set platform target to x64** — Pre-built binaries are built for x64 Windows platforms.
- **Add to Include Directories** — Tell the compiler how the OpenCV library looks. This is done by providing a path to the header files (build/include).
- **Add to Library Directories** — Tell the linker where it can find the lib files for different modules.
- **Add Additional Dependencies** — List .lib files for different modules. Note that we're only going to list a single all-in-one file named opencv_world.

How to setup OpenCV

- **Set platform target to x64**

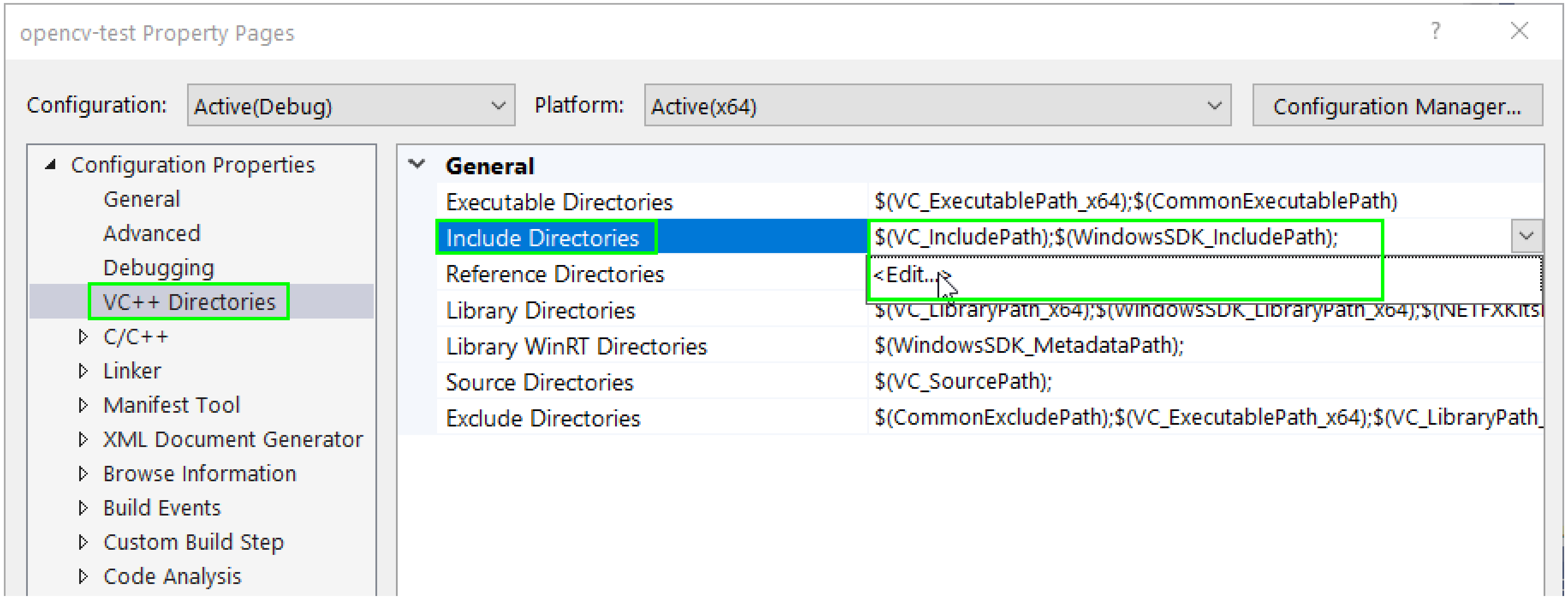


- Now, go to **Project** → ***YourProjectName*** Properties in the menu.



How to setup OpenCV

- **Add to Include Directories**
  - select **VC++ Directories** page on the left and click on **Include Directories** row.



opencv-test Property Pages

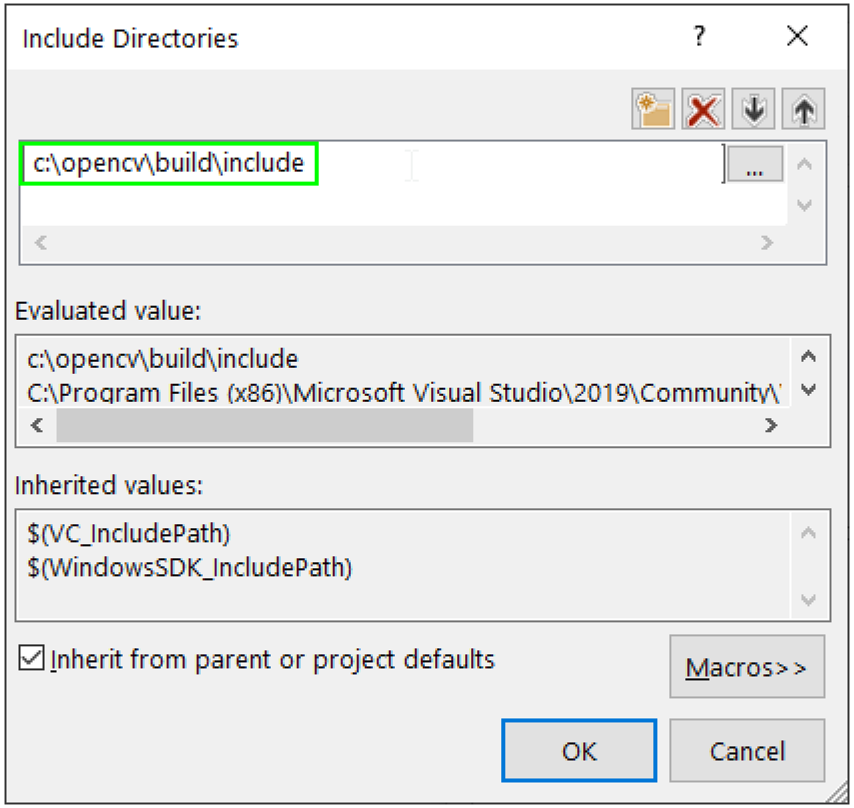
Configuration: Active(Debug) Platform: Active(x64) Configuration Manager...

General

Executable Directories	$(VC_ExecutablePath_x64);$(CommonExecutablePath)
Include Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);
Reference Directories	<Edit...>
Library Directories	$(VC_LibraryPath_x64);$(WindowsSDK_LibraryPath_x64);$(NETFXKitsI
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(CommonExcludePath);$(VC_ExecutablePath_x64);$(VC_LibraryPath_

How to setup OpenCV

- In Include Directories window, add c:\opencv\build\include.



c:\opencv\build\include

Evaluated value:

```
c:\opencv\build\include
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\
```

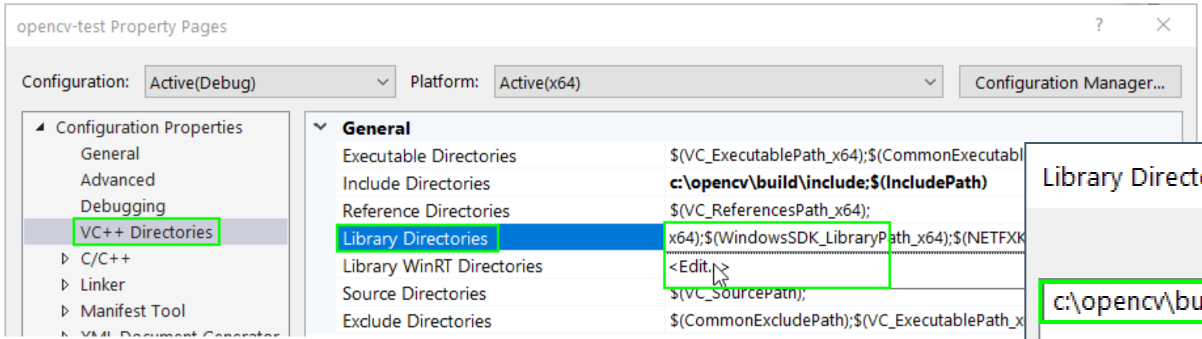
Inherited values:

```
$(VC_IncludePath)
$(WindowsSDK_IncludePath)
```

☑ Inherit from parent or project defaults

How to setup OpenCV

- In the same tab, look for **Library Directories**
- Add **c:\opencv\build\x64\vc15\lib**



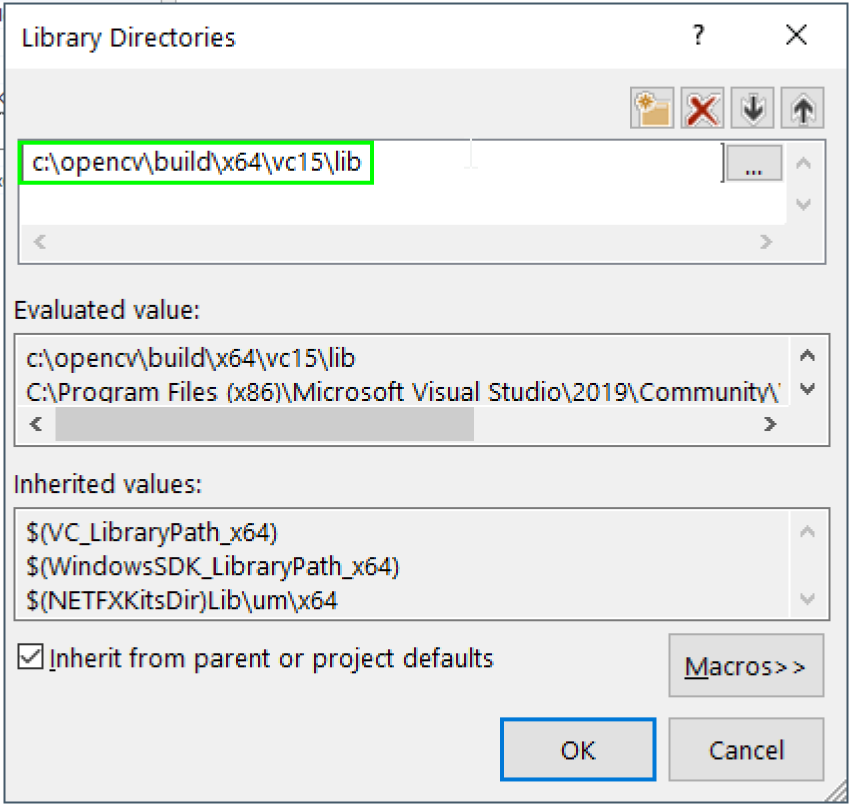
opencv-test Property Pages

Configuration: Active(Debug) Platform: Active(x64) Configuration Manager...

- Configuration Properties
  - General
  - Advanced
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
  - Manifest Tool

General

Executable Directories	$(VC_ExecutablePath_x64);$(CommonExecutabl
Include Directories	**c:\opencv\build\include;$(IncludePath)**
Reference Directories	$(VC_ReferencesPath_x64);
Library Directories	x64);$(WindowsSDK_LibraryPath_x64);$(NETFXK
Library WinRT Directories	<Edit...>
Source Directories	$(VC_SourcePath);
Exclude Directories	$(CommonExcludePath);$(VC_ExecutablePath_x



Library Directories

```
c:\opencv\build\x64\vc15\lib
```

Evaluated value:

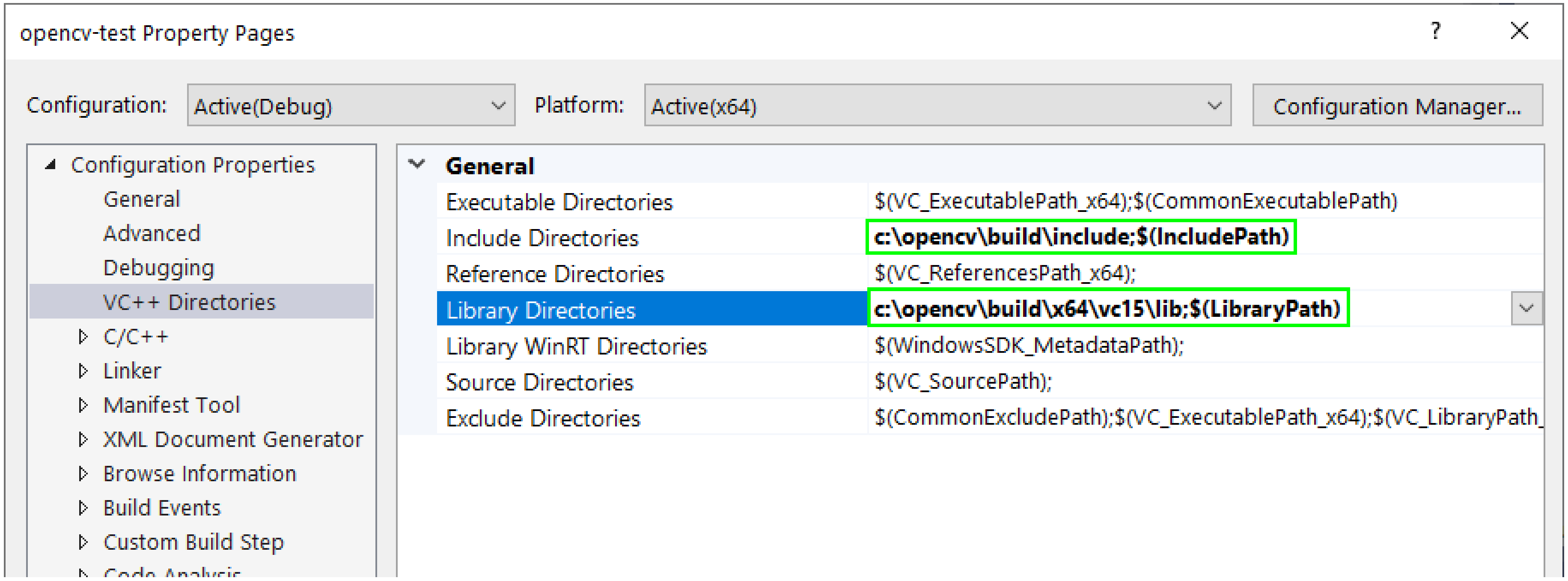
```
c:\opencv\build\x64\vc15\lib
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\
```

Inherited values:

```
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)
$(NETFXKitsDir)Lib\um\x64
```

☑ Inherit from parent or project defaults

- The VC++ Directories tab should look like below:



opencv-test Property Pages

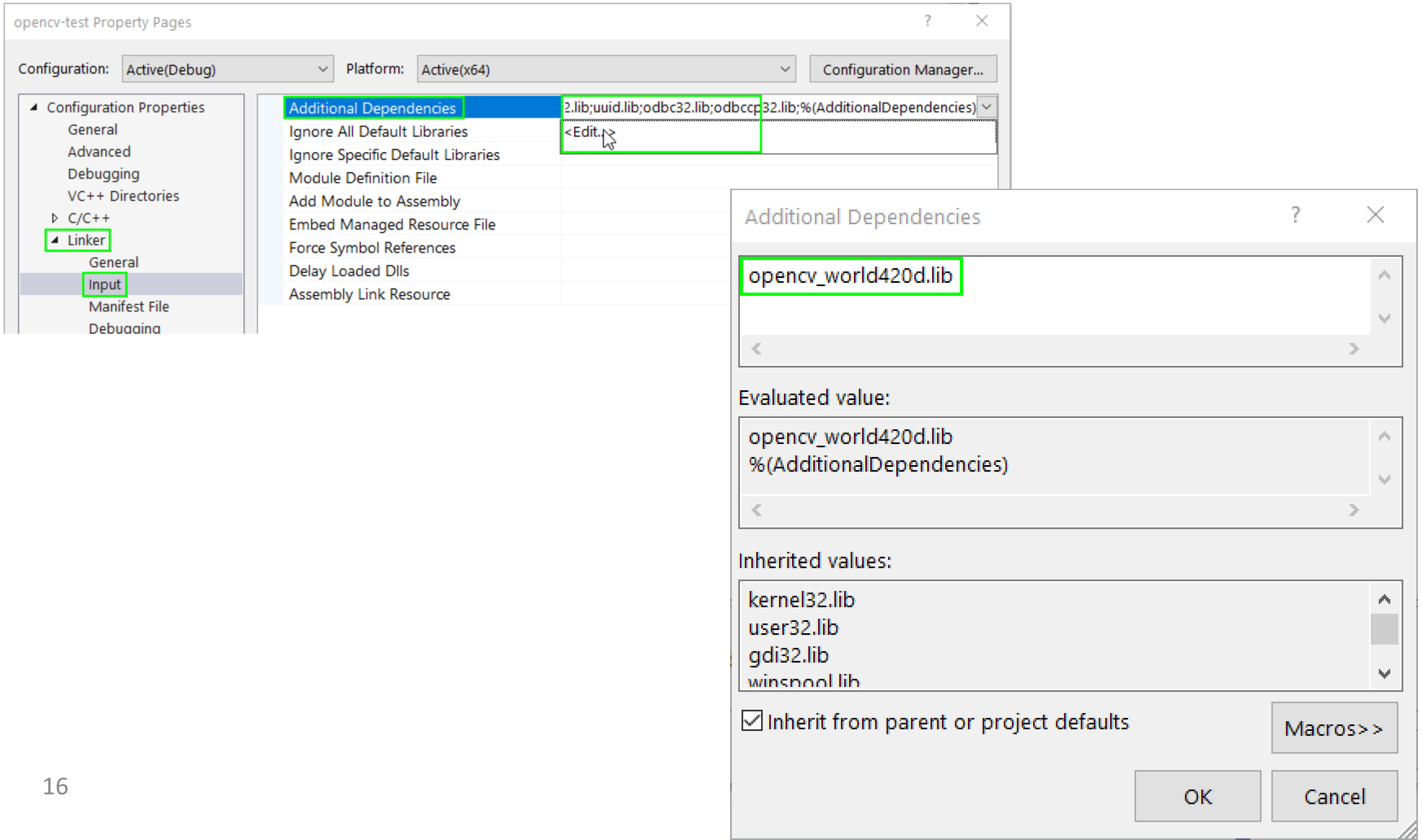
Configuration: Active(Debug)
Platform: Active(x64)

General

Executable Directories	$(VC_ExecutablePath_x64);$(CommonExecutablePath)
Include Directories	**c:\opencv\build\include;$(IncludePath)**
Reference Directories	$(VC_ReferencesPath_x64);
Library Directories	**c:\opencv\build\x64\vc15\lib;$(LibraryPath)**
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(CommonExcludePath);$(VC_ExecutablePath_x64);$(VC_LibraryPath_

How to setup OpenCV

- **Add Additional Dependencies**



opencv-test Property Pages

Configuration: Active(Debug) Platform: Active(x64) Configuration Manager...

- Configuration Properties
  - General
  - Advanced
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging

Additional Dependencies	2.lib;uuid.lib;odbc32.lib;odbccp32.lib;%(AdditionalDependencies)
Ignore All Default Libraries	<Edit...>
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies

```
opencv_world420d.lib
```

Evaluated value:

```
opencv_world420d.lib
%(AdditionalDependencies)
```

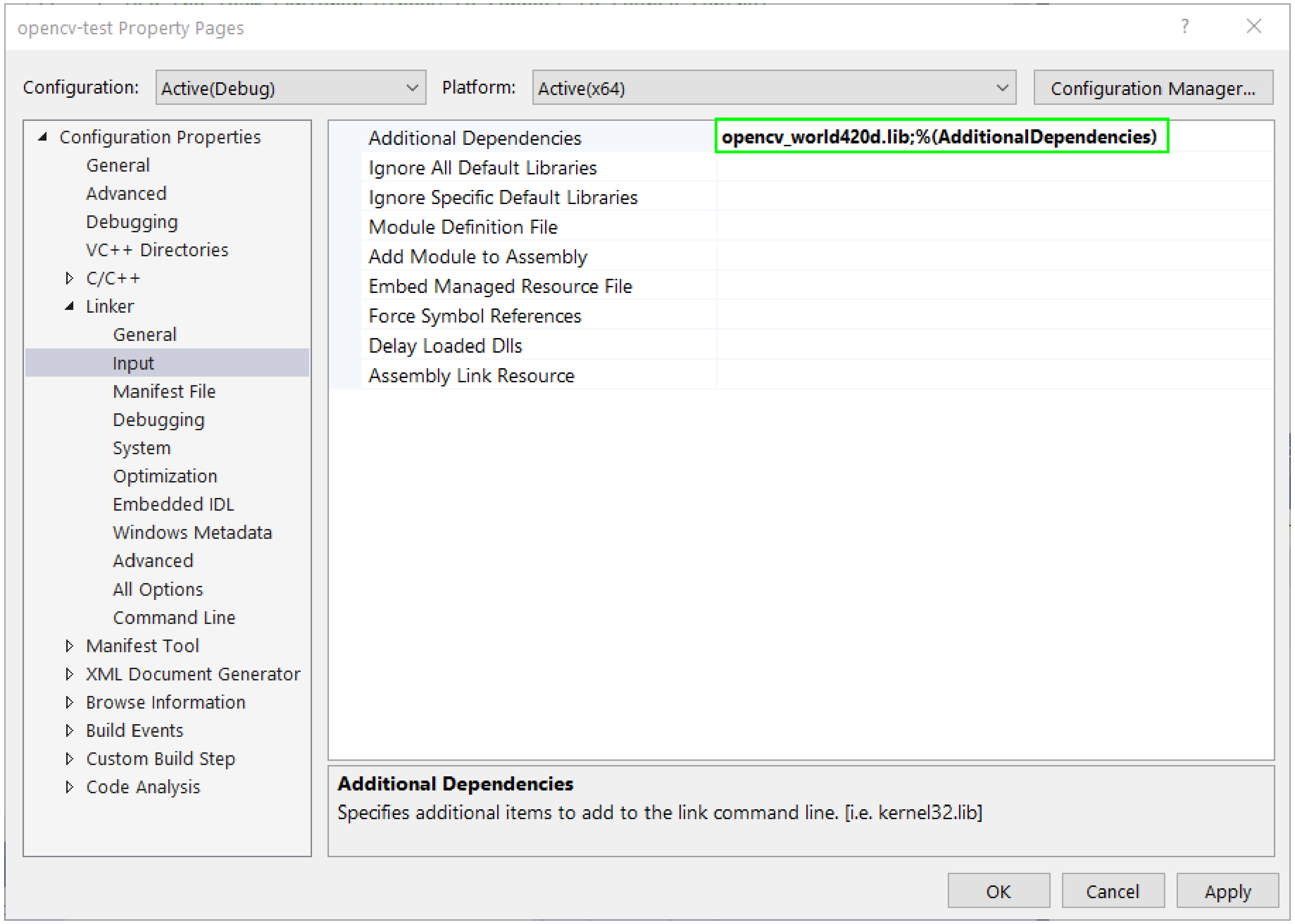
Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
```

[✓] Inherit from parent or project defaults

How to setup OpenCV

- The Linker tab should look like below:



opencv-test Property Pages

Configuration: Active(Debug)   Platform: Active(x64)   Configuration Manager...

- Configuration Properties
  - General
  - Advanced
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Windows Metadata
    - Advanced
    - All Options
    - Command Line
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

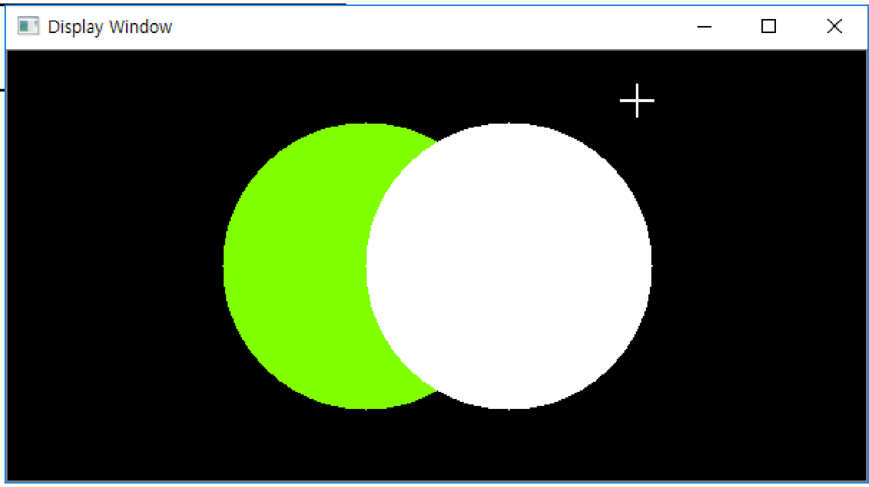
Additional Dependencies	**opencv_world420d.lib;%(AdditionalDependencies)**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

OK   Cancel   Apply

How to setup OpenCV

- **Step 4: Check out demo code!**



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

using namespace cv;
using namespace std;

int main()
{
   Mat image = Mat::zeros(300, 600, CV_8UC3);

   circle(image, Point(250, 150), 100, Scalar(0, 255, 128), -100);  //2차원 어레이로 이미지 표현

   circle(image, Point(350, 150), 100, Scalar(255, 255, 255), -100);

   imshow("Display Window", image);  //윈도우 타이틀
   waitKey(0);  //키 누르면 화면 꺼지게.
   return 0;
}
```

****여기까지 작동하도록 설치해놓기

Overview of OpenCV

Modular Structure

- OpenCV has a **modular structure**, which means that the package includes several shared or static libraries. The following modules are available:
  - **Core functionality** - a compact module defining basic data structures, including the dense multi-dimensional array ‘Mat’ and basic functions used by all other modules.
  - **Image processing** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
  - **video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
  - **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
  - **features2d** - salient feature detectors, descriptors, and descriptor matchers.
  - **objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
  - **highgui** - an easy-to-use interface to simple UI capabilities.
  - **videoio** - an easy-to-use interface to video capturing and video codecs.
  - **gpu** - GPU-accelerated algorithms from different OpenCV modules.
  - ... some other helper modules

API Concepts

- **cv Namespace**
  - All the OpenCV classes and functions are placed into the cv namespace. Therefore, to access this functionality from your code, use the **cv::** specifier or using namespace cv;

```
#include "opencv2/core.hpp"
...
cv::Mat H = cv::findHomography(points1, points2,
CV_RANSAC, 5);
...
```

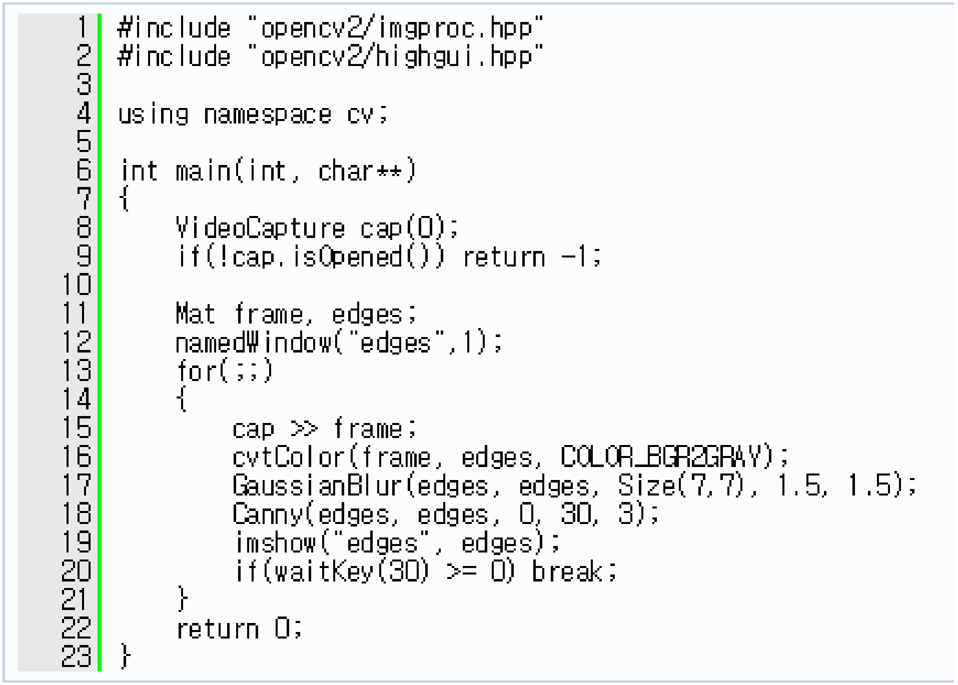
  - or :

```
#include "opencv2/core.hpp"
using namespace cv;   //이렇게 정의해주면 cv:: 매번 안해줘도 됨.
...
Mat H = findHomography(points1, points2, CV_RANSAC, 5 );
...
```

- **Automatic Memory Management**
  - First of all, std::vector, Mat, and other data structures used by the functions and methods have destructors that deallocate the underlying memory buffers when needed. This means that the destructors do not always deallocate the buffers as in case of Mat. They take into account possible data sharing. A destructor *decrements the reference counter* associated with the matrix data buffer. The buffer is deallocated *if and only if the reference counter reaches zero*, that is, when no other structures refer to the same buffer.

  - Similarly, when a Mat instance is copied, no actual data is really copied. Instead, the reference counter is incremented to memorize that there is another owner of the same data.
    - There is also the **Mat::clone** method that creates a full copy of the matrix data. See the example in the next slide:

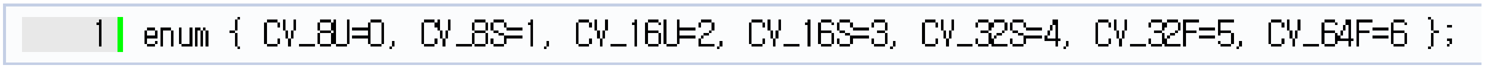
- **Automatic Allocation of the Output Data**
  - OpenCV deallocates the memory automatically, as well as automatically allocates the memory for output function parameters most of the time.
  - The size and type of the output arrays are determined from the size and type of input arrays.



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

using namespace cv;

int main(int, char**)
{
    VideoCapture cap(0);
    if(!cap.isOpened()) return -1;

    Mat frame, edges;
    namedWindow("edges",1);
    for(;;)
    {
        cap >> frame;
        cvtColor(frame, edges, COLOR_BGR2GRAY);
        GaussianBlur(edges, edges, Size(7,7), 1.5, 1.5);
        Canny(edges, edges, 0, 30, 3);
        imshow("edges", edges);
        if(waitKey(30) >= 0) break;
    }
    return 0;
}
```

Data Types

- A limited fixed set of primitive data types
- That is, array elements should have one of the following types:
  - 8-bit unsigned integer (uchar)
  - 8-bit signed integer (schar)
  - 16-bit unsigned integer (ushort)
  - 16-bit signed integer (short)
  - 32-bit signed integer (int)
  - 32-bit floating-point number (float)
  - 64-bit floating-point number (double)



```
enum { CV_8U=0, CV_8S=1, CV_16U=2, CV_16S=3, CV_32S=4, CV_32F=5, CV_64F=6 };
```

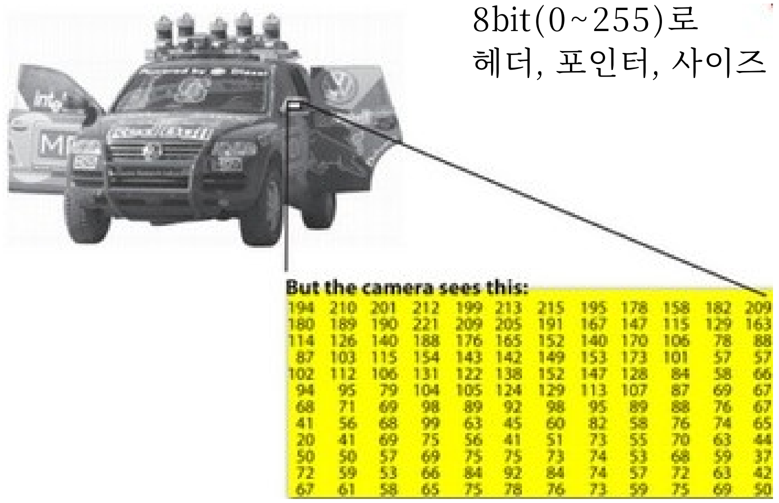
- Examples of array data construction

```
Mat mtx(3, 3, CV_32F); // make a 3x3 floating-point matrix
Mat cmtx(10, 1, CV_64FC2); // make a 10x1 2-channel floating-point
                           // matrix (10-element complex vector)
Mat img(Size(1920, 1080), CV_8UC3); // make a 3-channel (color) image
                                    // of 1920 columns and 1080 rows.
Mat grayscale(image.size(), CV_MAKETYPE(image.depth(), 1)); // make a 1-channel image of
                                                            // the same size and same
                                                            // channel type as img
```

cv:Mat

- In the below image, you can see that the mirror of the car is nothing more than a matrix containing all the intensity values of the pixel points.
- All images inside a computer world are numerical matrices and other information describing the matrix itself.
- *OpenCV* is a computer vision library whose main focus is to process and manipulate this information.
- Therefore, the first thing you need to be familiar with is how OpenCV stores and handles images.

8bit(0~255)로
헤더, 포인터, 사이즈 등 실제 데이터의 정보를 표현??


But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	74	65
20	41	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

- *Mat* is basically a class with two data parts:
  - The **matrix header** (containing information such as the size of the matrix, the method used for storing, at which address is the matrix stored, and so on)
  - and **a pointer** to the matrix containing the pixel values (taking any dimensionality depending on the method chosen for storing).
  - The matrix header size is constant, however the size of the matrix itself may vary from image to image and usually is larger by orders of magnitude.

- You need to know about *Mat* is that you no longer need to manually allocate its memory and release it as soon as you do not need it.
- Most of the OpenCV functions will allocate its output data automatically.

- OpenCV uses a reference counting system.
- The idea is that each *Mat* object has its own header, however the matrix may be shared between two instances of them by having their matrix pointers point to the same address.
- Moreover, the copy operators **will only copy the headers and the pointer** to the large matrix, **not the data itself**.

**메모리 A에 잡고 초기화 후 C에 A를 복사하는 코드 - > C에는 A의 포인터만 연결해준 거라 A,C는 같은 곳을 가리키고 세개가 하나의 데이터를 reference해주게 됨.

```
Mat A, C;                          // creates just the header parts
A = imread(argv[1], IMREAD_COLOR); // here we'll know the method used (allocate matrix)
                                      여기서 실제로 메모리를 잡게 됨.
Mat B(A);                                 // Use the copy constructor

C = A;                                    // Assignment operator
```

- All the above objects, in the end, point to the same single data matrix.
- Their headers are different, however, and making a modification using any of them will affect all the other ones as well.

- To create a region of interest (*ROI*) in an image you just create a new header with the new boundaries:

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

- When the counter reaches zero the matrix too is freed.

- Sometimes you will want to copy the matrix itself too, so OpenCV provides the **Mat::clone()** and **Mat::copyTo()** functions.

```
Mat F = A.clone();
Mat G;
A.copyTo(G);
```

Practice

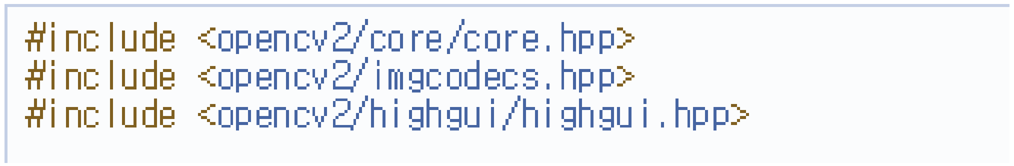
- **[Lab01-2]**
  - Load an image (using cv::imread)
  - Create a named OpenCV window (using cv::namedWindow)
  - Display an image in the OpenCV window (using cv::imshow)

http://docs.opencv.org/3.2.0/db/deb/tutorial_display_image.html

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

#include <iostream>
#include <string>

using namespace cv;

using namespace std;

int main( int argc, char** argv )
{
    string imageName("../data/HappyFish.jpg"); // by default   //이미지 절대경로
    if( argc > 1)
    {
        imageName = argv[1];
    }

    Mat image;

    image = imread(imageName.c_str(), IMREAD_COLOR); // Read the file

    if( image.empty() )                      // Check for invalid input
    {
        cout <<  "Could not open or find the image" << std::endl ;
        return -1;
    }
                    //Display window는 이름이라 아래랑 똑같아야함
    namedWindow( "Display window", WINDOW_AUTOSIZE ); // Create a window for display.

    imshow( "Display window", image );                // Show our image inside it.

    waitKey(0); // Wait for a keystroke in the window
    return 0;
}
```

Tip: include opencv2/opencv.hpp

- Instead the inclusion of each module’s header file,



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

- only one include statement is enough

#include opencv2/opencv.hpp