

# 실험 10

## OpenCV와 카메라를 활용한 기본적인 영상처리

---

2025. 10. 29. (Wed)

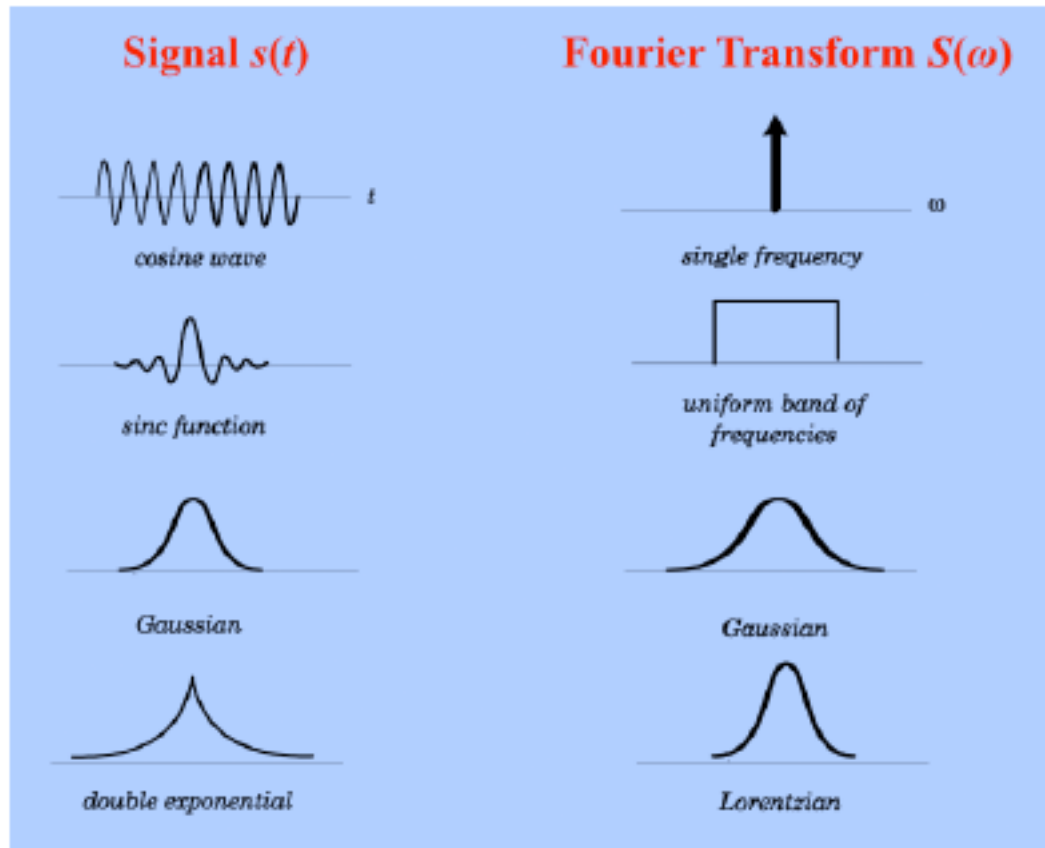
# 실험 목표

---

- 영상처리 라이브러리인 OpenCV를 통해 영상처리법을 체험해본다
- Gaussian filter 와 Sobel filter를 통해 이미지의 주파수 대역 특성을 파악한다.
- OpenCV에서 제공하는 기능인 Gaussian filter를 활용하여 새로운 응용을 해본다

# 영상의 주파수 정보

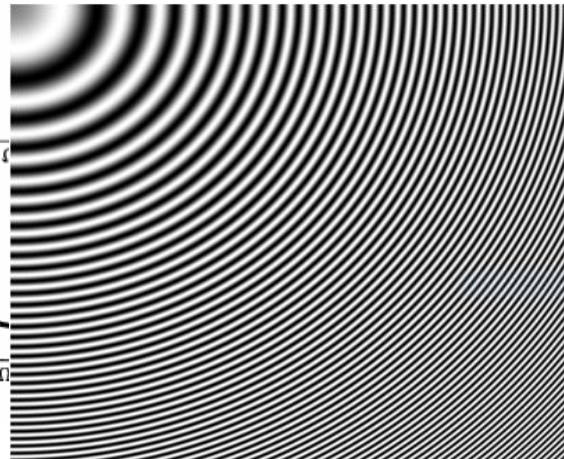
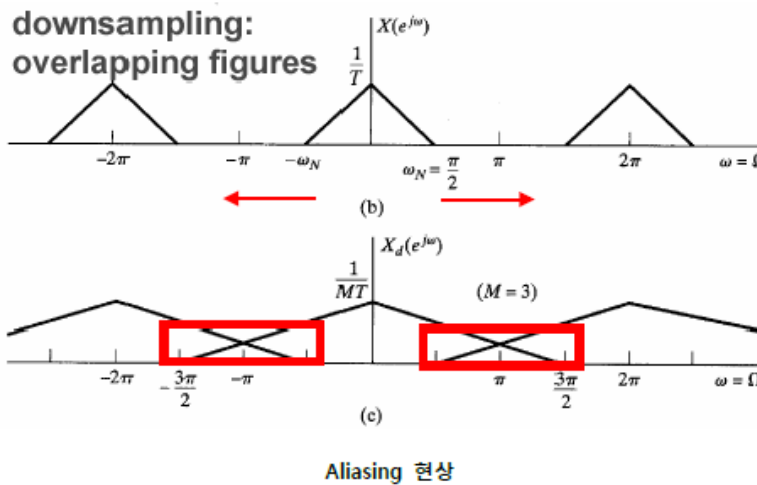
- 푸리에 변환을 통해 영상을 이루는 이미지의 정보를 주파수 평면상에 나타낼 수 있다.



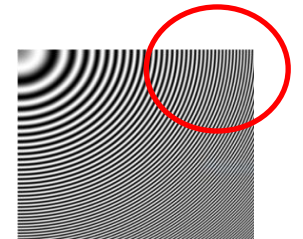
시간 축 상의 신호의 주파수 축으로의 변환

# Down-sampling & Aliasing

- 이미지 축소  
= 신호의 downsampling  
= 주파수 대역의 확대
- Aliasing : 확대된 주파수 대역이 겹치는 현상



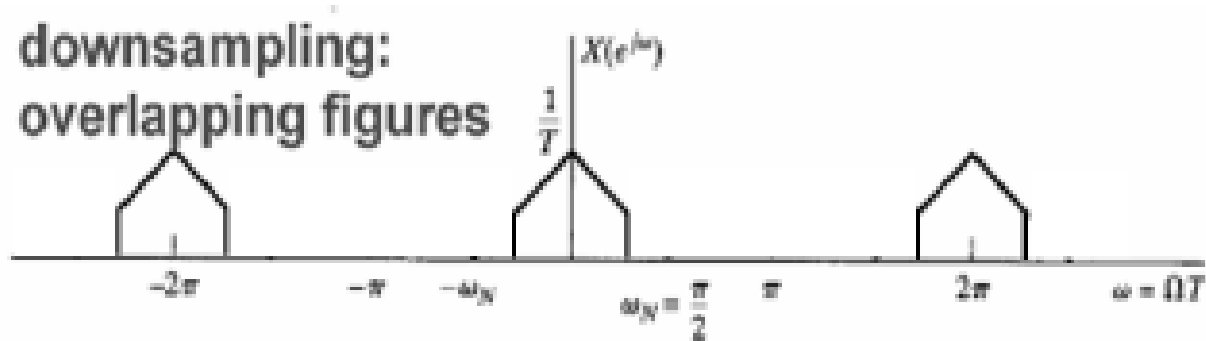
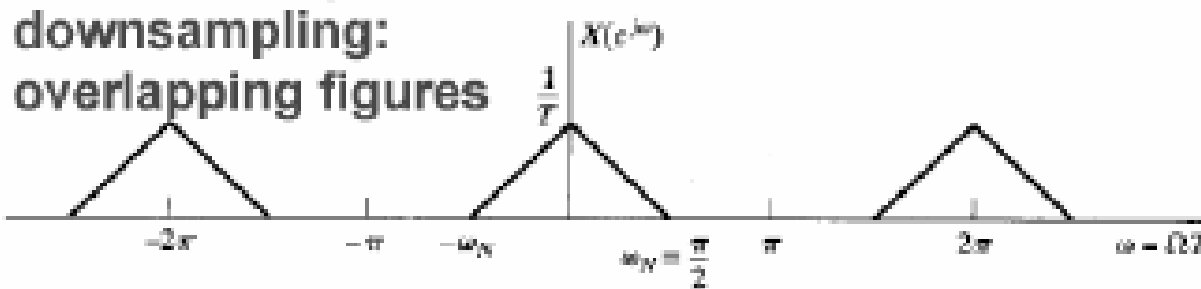
원본에는 없는 무늬가  
축소본에 발생



이미지 원본 및 축소본

# Gaussian Filter

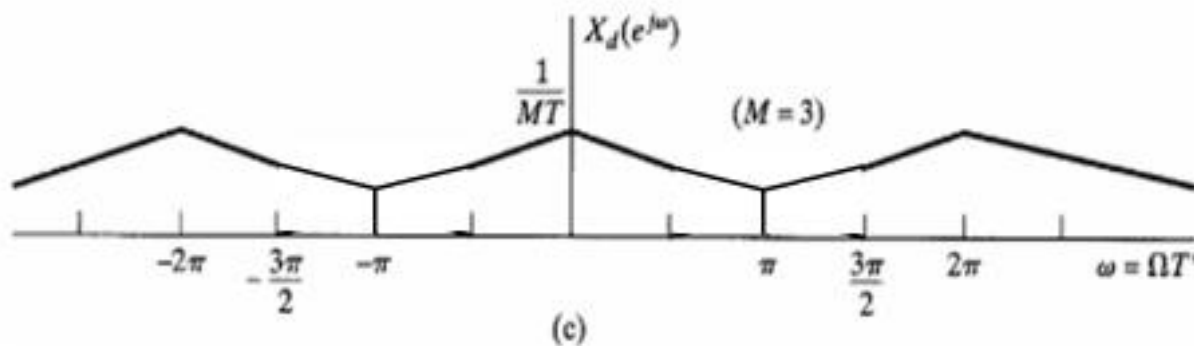
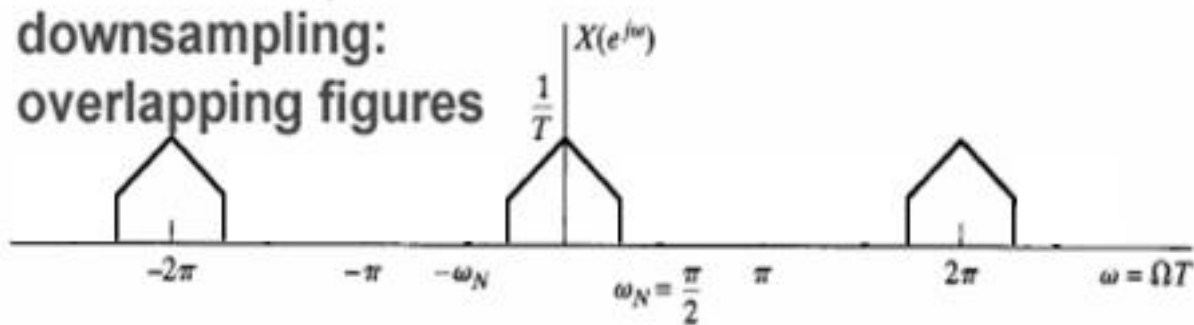
- 대표적 저주파 필터 (Low-pass filter)
- 하나의 픽셀을 중심으로 정해진 크기( $N \times M$ )의 필터 적용
- 고주파 신호를 무시



저주파 신호 유지, 고주파 신호 무시

# Gaussian Filter

- Gaussian Filter 적용 후 이미지 축소하여 aliasing 방지



Aliasing의 방지

# Sobel Filter

- 이미지의 밝기 gradient를 검출하여 모서리 검출
- 가로축과 세로축의 gradient 사용



# 실험 내용

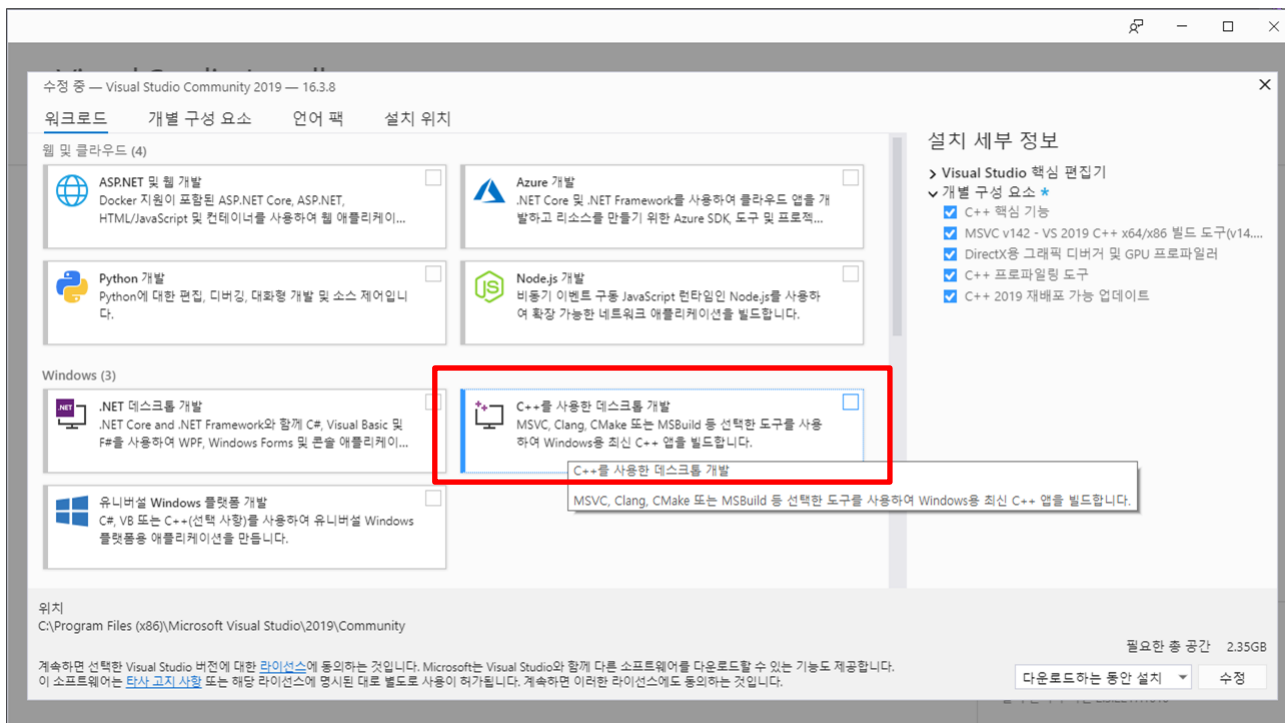
---

- 실험 1: OpenCV 세팅 및 기본 동작 확인
- 실험 2: 카메라 영상 출력
- 실험 3: Gaussian Filter 사용
- 실험 4: Sobel Filter 사용



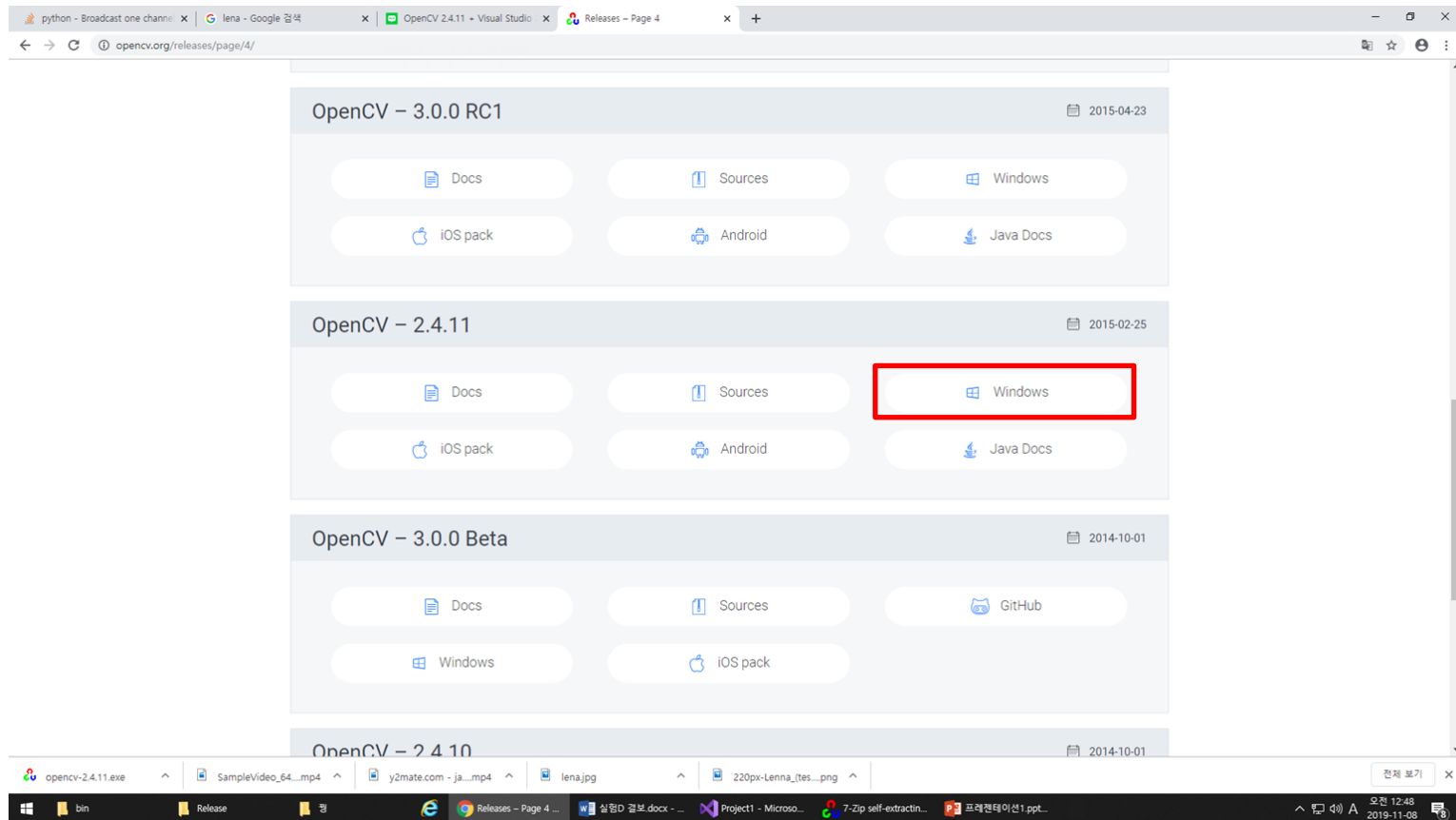
# Visual Studio 설치

- Visual Studio Community 다운
- <https://visualstudio.microsoft.com/ko/free-developer-offers/>
- 워크로드 “C++을 사용한 데스크톱 개발” 선택 후 설치



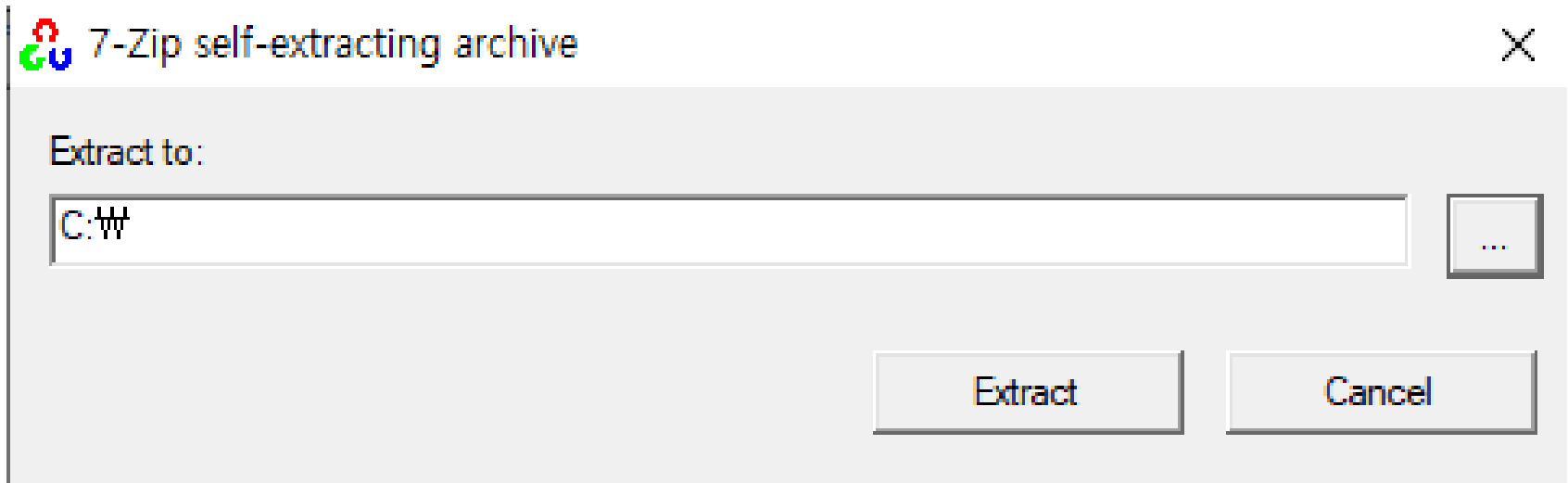
# 실험 1: OpenCV 세팅, 기본동작 확인

- <https://opencv.org/releases/page/8/> 에서 OpenCV 2.4.11 다운로드 및 설치 파일 실행



# 실험 1: OpenCV 세팅, 기본동작 확인

- 설치 경로는 C:\



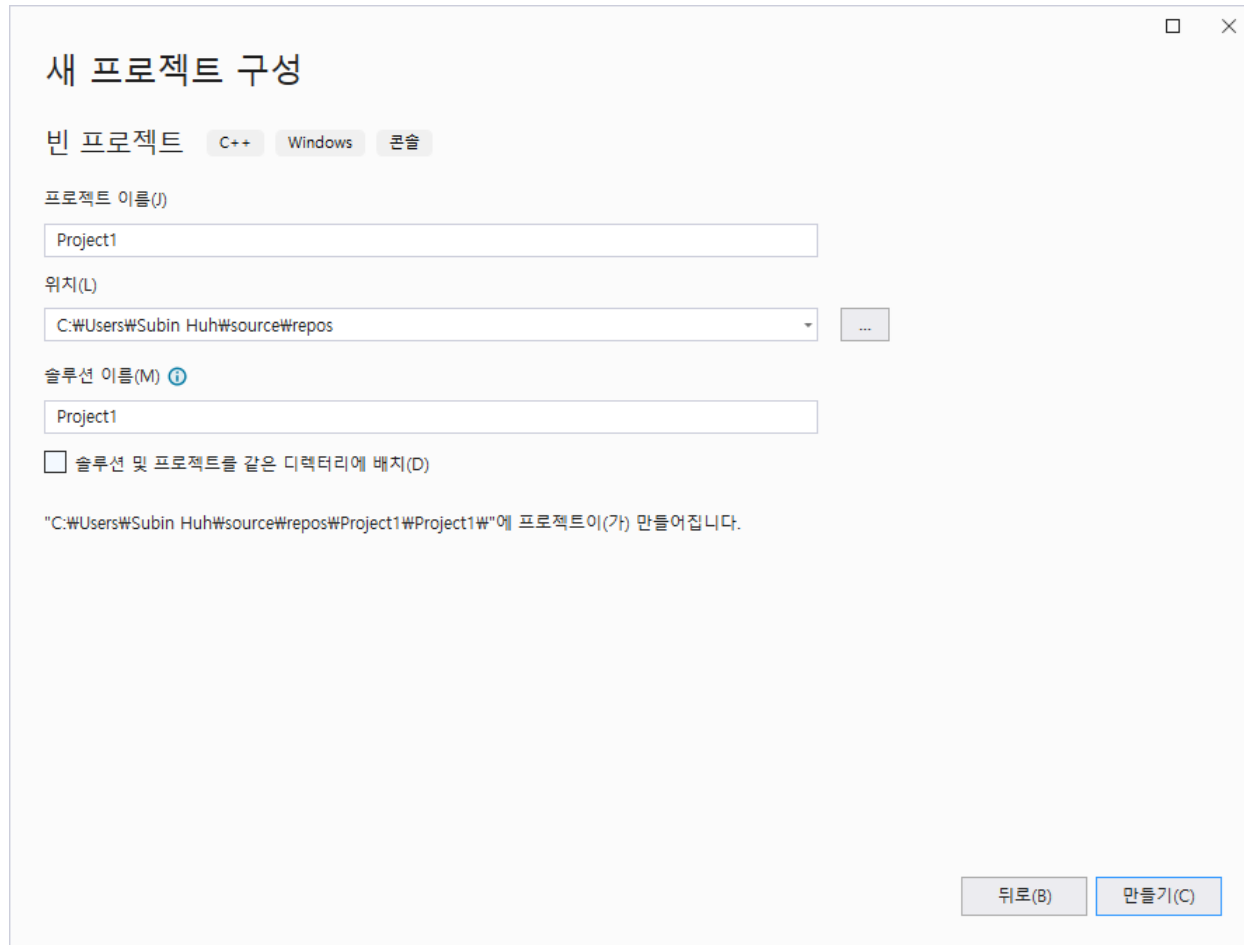
# 실험 1: OpenCV 세팅, 기본동작 확인

- Visual Studio > 새 프로젝트 만들기
  - 빈 프로젝트 만들기



# 실험 1: OpenCV 세팅, 기본동작 확인

- Visual Studio > 새 프로젝트 만들기
  - 빈 프로젝트 만들기



새 프로젝트 구성

빈 프로젝트 C++ Windows 콘솔

프로젝트 이름(I)

Project1

위치(L)

C:\Users\Subin Huh\source\repos

솔루션 이름(M) ⓘ

Project1

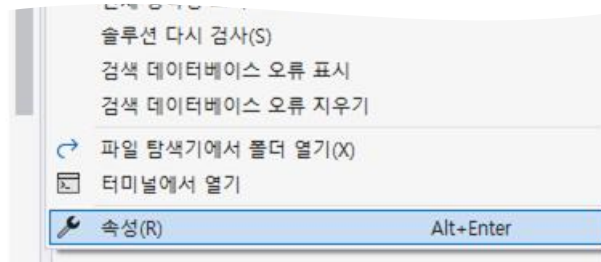
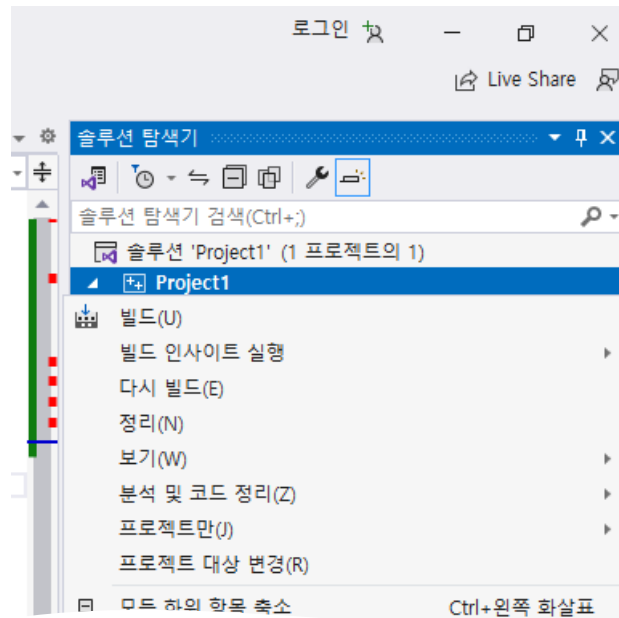
☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

"C:\Users\Subin Huh\source\repos\Project1\Project1"에 프로젝트이(가) 만들어집니다.

뒤로(B) 만들기(C)

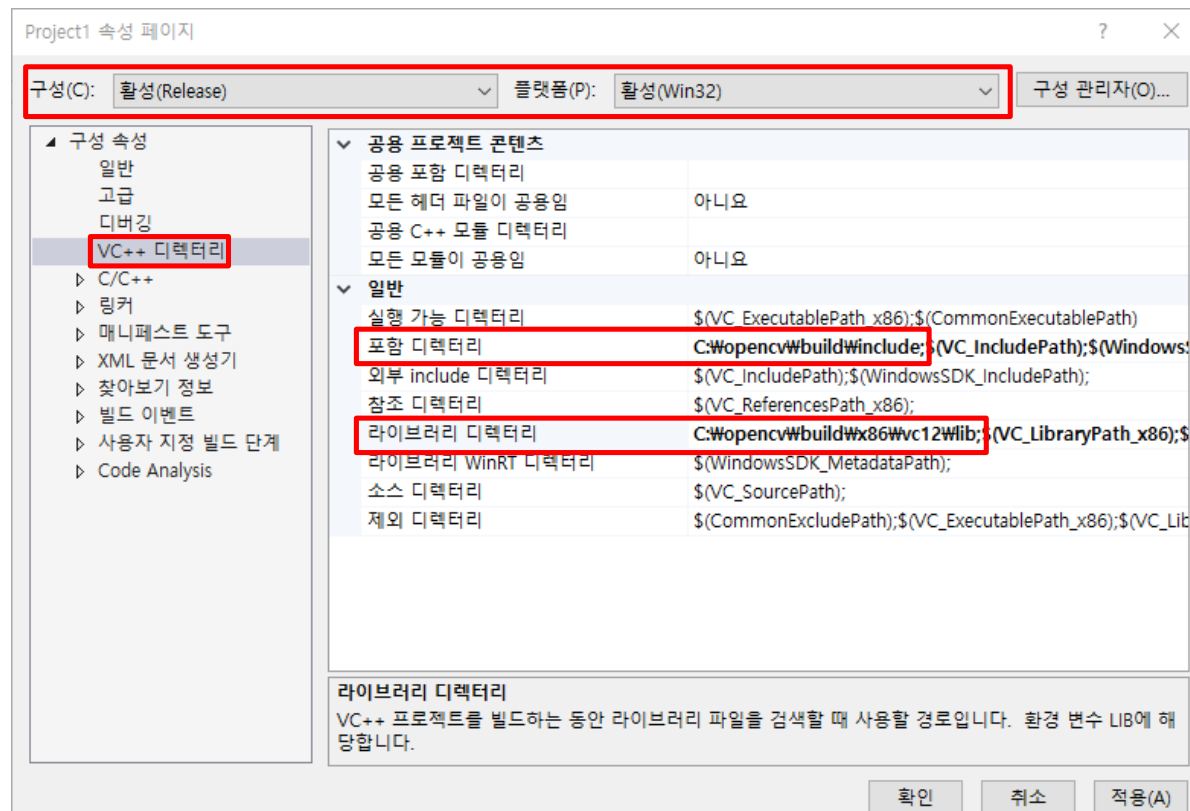
# 실험 1: OpenCV 세팅, 기본동작 확인

- 프로젝트 > 속성
  - 외부 라이브러리인 OpenCV를 사용하기 위한 절차



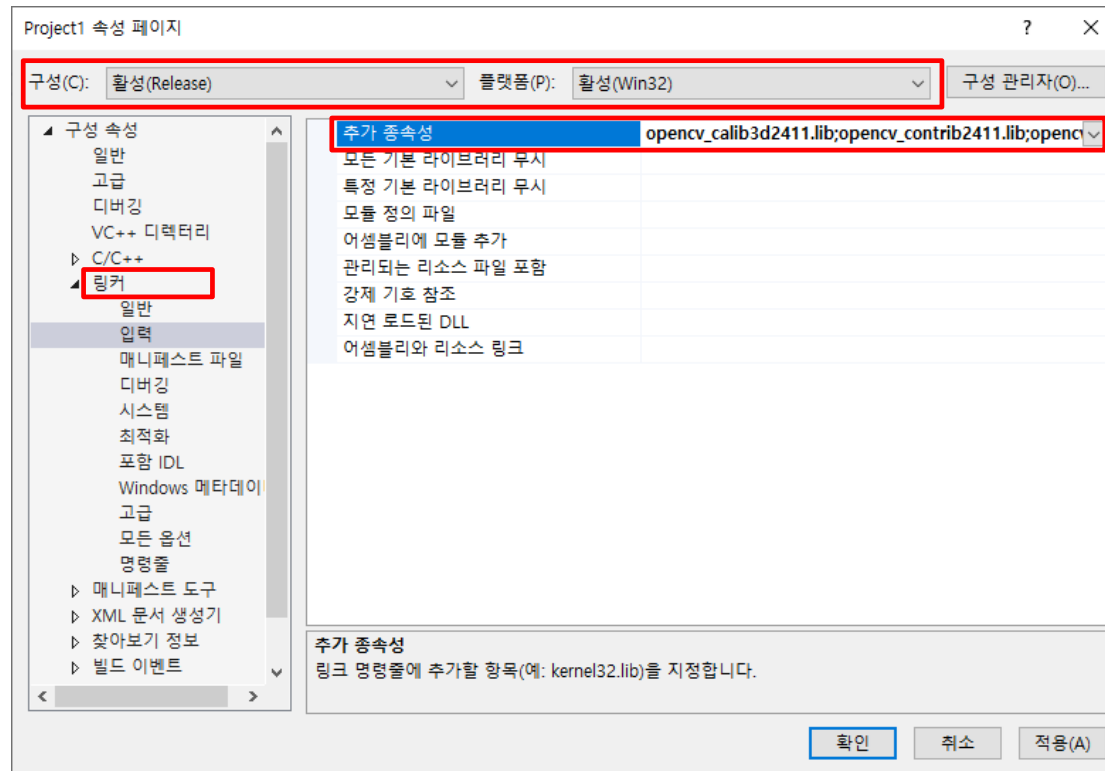
# 실험 1: OpenCV 세팅, 기본동작 확인

- 구성: Release / 플랫폼: x86(Win32) 설정
- VC++ 디렉터리 > 아래와 같이 openCV 경로 추가
  - C:\opencv\build\include;
  - C:\opencv\build\x86\vc12\lib;
  - 경로 사이에 띄어쓰기 없음



# 실험 1: OpenCV 세팅, 기본동작 확인

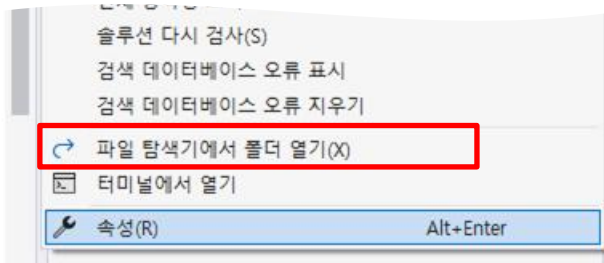
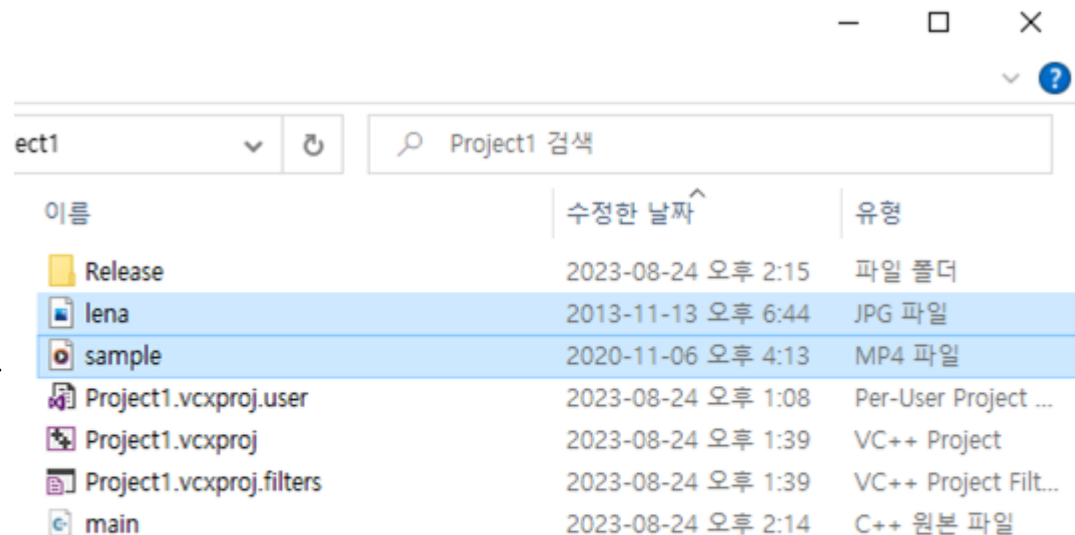
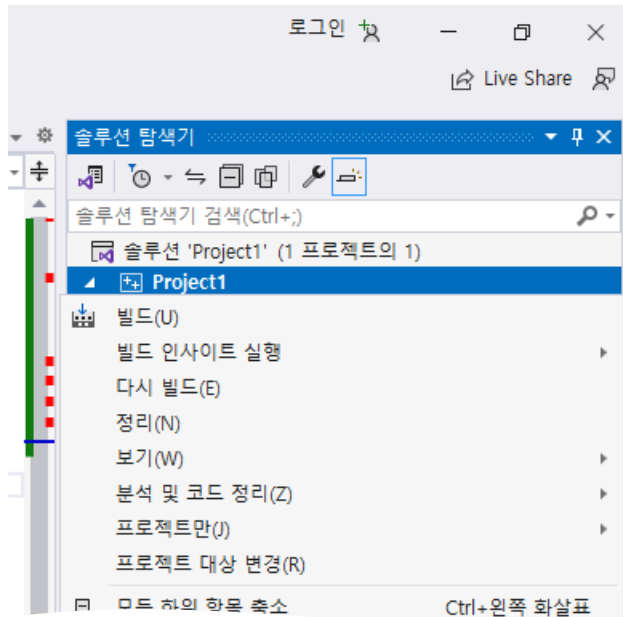
- 링커 > 입력 > 추가 종속성 추가 후 확인
- 기존 항목 지우지 않고 아래 항목 추가
  - opencv\_calib3d2411.lib;opencv\_contrib2411.lib;opencv\_core2411.lib;opencv\_features2d2411.lib;opencv\_flann2411.lib;opencv\_gpu2411.lib;opencv\_highgui2411.lib;opencv\_imgproc2411.lib;opencv\_ml2411.lib;opencv\_nonfree2411.lib;





# 실험 1: OpenCV 세팅, 기본동작 확인

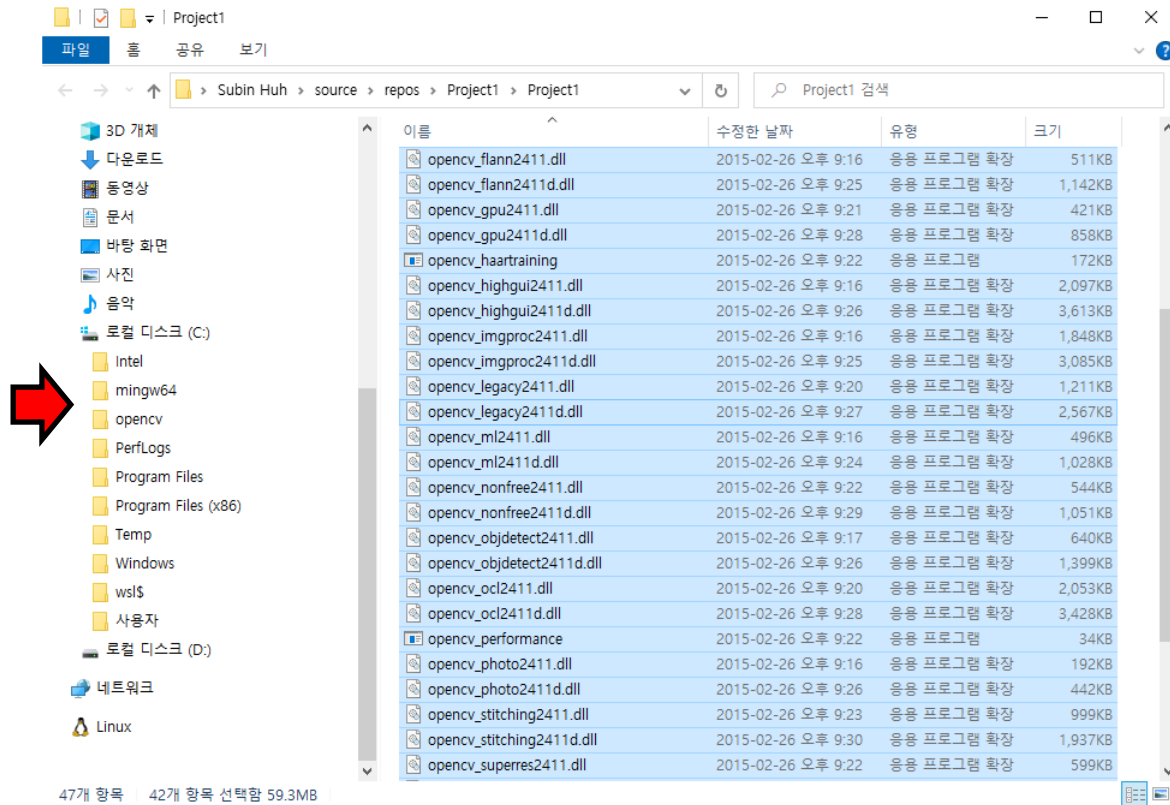
- 프로젝트명 우클릭 > 파일 탐색기에서 폴더 열기
- 제공된 이미지 파일 및 동영상 파일 붙여 넣기



# 실험 1: OpenCV 세팅, 기본동작 확인

- C:\opencv\build\x86\vc12\bin에 있는 모든 파일 복사
- 프로젝트 폴더에 붙여 넣기

이름	수정한 날짜	유형	크기
opencv_annotation	2015-02-26 오후...	응용 프로그램	48KB
opencv_calib3d2411.dll	2015-02-26 오후...	응용 프로그램 확장	961KB
opencv_calib3d2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,751KB
opencv_contrib2411.dll	2015-02-26 오후...	응용 프로그램 확장	1,146KB
opencv_contrib2411d.dll	2015-02-26 오후...	응용 프로그램 확장	2,357KB
opencv_core2411.dll	2015-02-26 오후...	응용 프로그램 확장	2,089KB
opencv_core2411d.dll	2015-02-26 오후...	응용 프로그램 확장	3,438KB
opencv_createsamples	2015-02-26 오후...	응용 프로그램	173KB
opencv_features2d2411.dll	2015-02-26 오후...	응용 프로그램 확장	706KB
opencv_features2d2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,433KB
opencv_ffmpeg2411.dll	2013-12-20 오후...	응용 프로그램 확장	10,289KB
opencv_flann2411.dll	2015-02-26 오후...	응용 프로그램 확장	511KB
opencv_flann2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,142KB
opencv_gpu2411.dll	2015-02-26 오후...	응용 프로그램 확장	421KB
opencv_gpu2411d.dll	2015-02-26 오후...	응용 프로그램 확장	858KB
opencv_haartraining	2015-02-26 오후...	응용 프로그램	172KB
opencv_highgui2411.dll	2015-02-26 오후...	응용 프로그램 확장	2,097KB
opencv_highgui2411d.dll	2015-02-26 오후...	응용 프로그램 확장	3,613KB
opencv_imgproc2411.dll	2015-02-26 오후...	응용 프로그램 확장	1,848KB
opencv_imgproc2411d.dll	2015-02-26 오후...	응용 프로그램 확장	3,085KB
opencv_legacy2411.dll	2015-02-26 오후...	응용 프로그램 확장	1,211KB
opencv_legacy2411d.dll	2015-02-26 오후...	응용 프로그램 확장	2,567KB
opencv_ml2411.dll	2015-02-26 오후...	응용 프로그램 확장	496KB
opencv_ml2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,028KB
opencv_nonfree2411.dll	2015-02-26 오후...	응용 프로그램 확장	544KB
opencv_nonfree2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,051KB
opencv_objdetect2411.dll	2015-02-26 오후...	응용 프로그램 확장	640KB
opencv_objdetect2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,399KB
opencv_ocl2411.dll	2015-02-26 오후...	응용 프로그램 확장	2,053KB
opencv_ocl2411d.dll	2015-02-26 오후...	응용 프로그램 확장	3,428KB
opencv_performance	2015-02-26 오후...	응용 프로그램	34KB
opencv_photo2411.dll	2015-02-26 오후...	응용 프로그램 확장	192KB
opencv_photo2411d.dll	2015-02-26 오후...	응용 프로그램 확장	442KB
opencv_stitching2411.dll	2015-02-26 오후...	응용 프로그램 확장	999KB
opencv_stitching2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,937KB
opencv_superres2411.dll	2015-02-26 오후...	응용 프로그램 확장	599KB
opencv_superres2411d.dll	2015-02-26 오후...	응용 프로그램 확장	1,128KB



Project1

파일 | 홈 | 공유 | 보기

Subin Huh > source > repos > Project1 > Project1

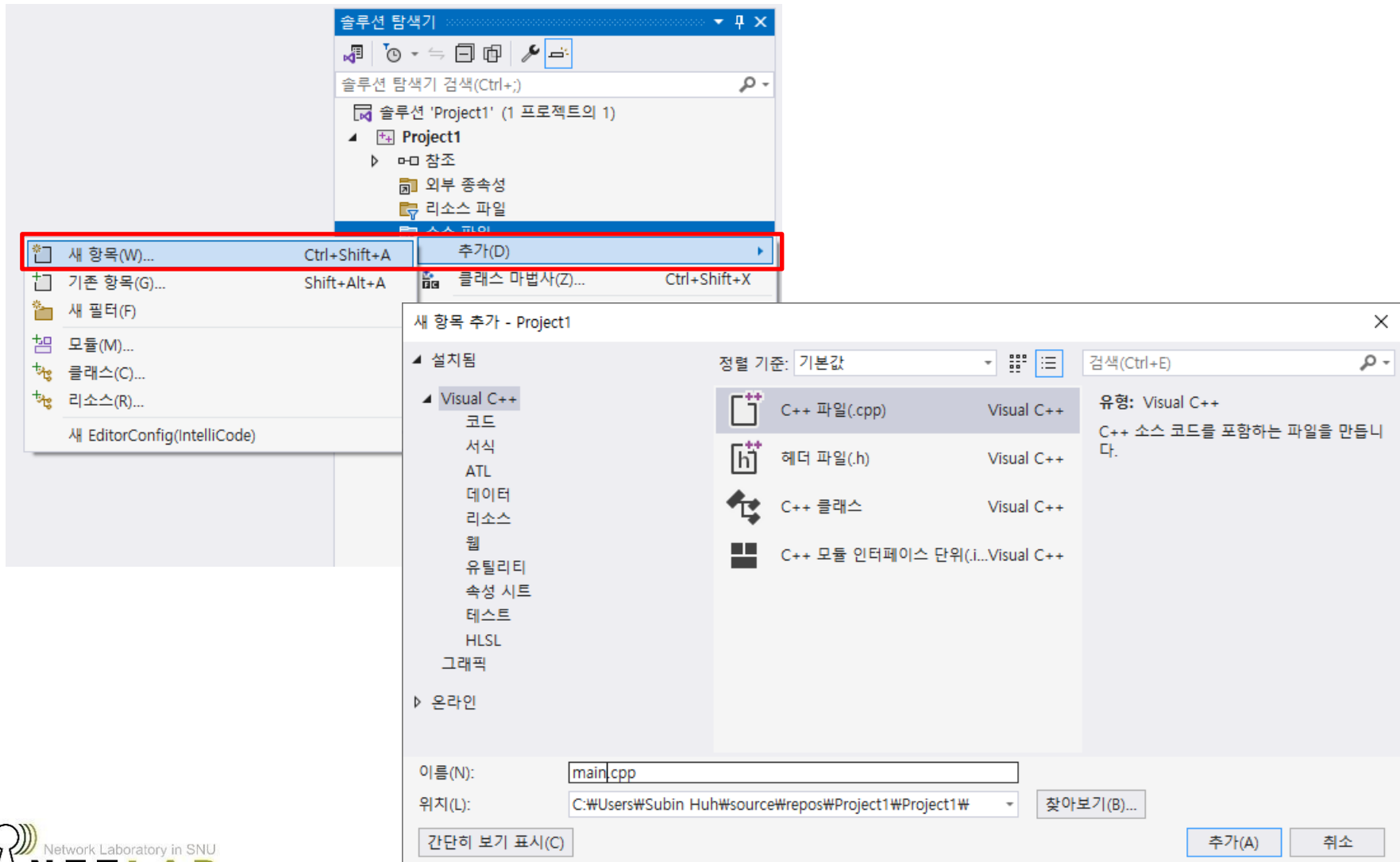
Project1 검색

이름	수정한 날짜	유형	크기
opencv_flann2411.dll	2015-02-26 오후 9:16	응용 프로그램 확장	511KB
opencv_flann2411d.dll	2015-02-26 오후 9:25	응용 프로그램 확장	1,142KB
opencv_gpu2411.dll	2015-02-26 오후 9:21	응용 프로그램 확장	421KB
opencv_gpu2411d.dll	2015-02-26 오후 9:28	응용 프로그램 확장	858KB
opencv_haartraining	2015-02-26 오후 9:22	응용 프로그램	172KB
opencv_highgui2411.dll	2015-02-26 오후 9:16	응용 프로그램 확장	2,097KB
opencv_highgui2411d.dll	2015-02-26 오후 9:26	응용 프로그램 확장	3,613KB
opencv_imgproc2411.dll	2015-02-26 오후 9:16	응용 프로그램 확장	1,848KB
opencv_imgproc2411d.dll	2015-02-26 오후 9:25	응용 프로그램 확장	3,085KB
opencv_legacy2411.dll	2015-02-26 오후 9:20	응용 프로그램 확장	1,211KB
opencv_legacy2411d.dll	2015-02-26 오후 9:27	응용 프로그램 확장	2,567KB
opencv_ml2411.dll	2015-02-26 오후 9:16	응용 프로그램 확장	496KB
opencv_ml2411d.dll	2015-02-26 오후 9:24	응용 프로그램 확장	1,028KB
opencv_nonfree2411.dll	2015-02-26 오후 9:22	응용 프로그램 확장	544KB
opencv_nonfree2411d.dll	2015-02-26 오후 9:29	응용 프로그램 확장	1,051KB
opencv_objdetect2411.dll	2015-02-26 오후 9:17	응용 프로그램 확장	640KB
opencv_objdetect2411d.dll	2015-02-26 오후 9:26	응용 프로그램 확장	1,399KB
opencv_ocl2411.dll	2015-02-26 오후 9:20	응용 프로그램 확장	2,053KB
opencv_ocl2411d.dll	2015-02-26 오후 9:28	응용 프로그램 확장	3,428KB
opencv_performance	2015-02-26 오후 9:22	응용 프로그램	34KB
opencv_photo2411.dll	2015-02-26 오후 9:16	응용 프로그램 확장	192KB
opencv_photo2411d.dll	2015-02-26 오후 9:26	응용 프로그램 확장	442KB
opencv_stitching2411.dll	2015-02-26 오후 9:23	응용 프로그램 확장	999KB
opencv_stitching2411d.dll	2015-02-26 오후 9:30	응용 프로그램 확장	1,937KB
opencv_superres2411.dll	2015-02-26 오후 9:22	응용 프로그램 확장	599KB

47개 항목 | 42개 항목 선택함 59.3MB

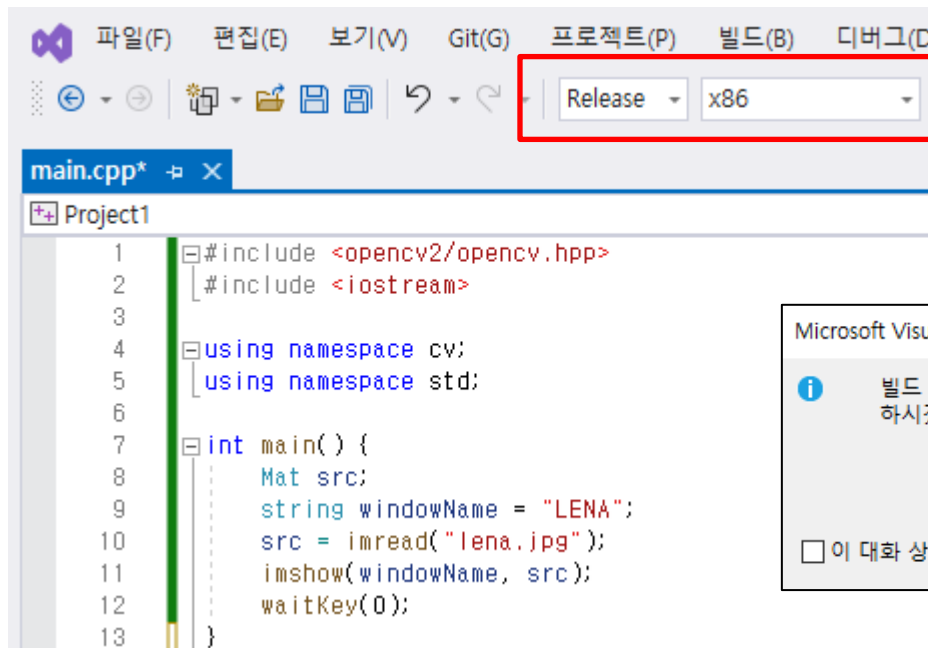
# 실험 1: OpenCV 세팅, 기본동작 확인

- 소스 파일 우클릭 > 추가 > 새 항목 > main.cpp 추가



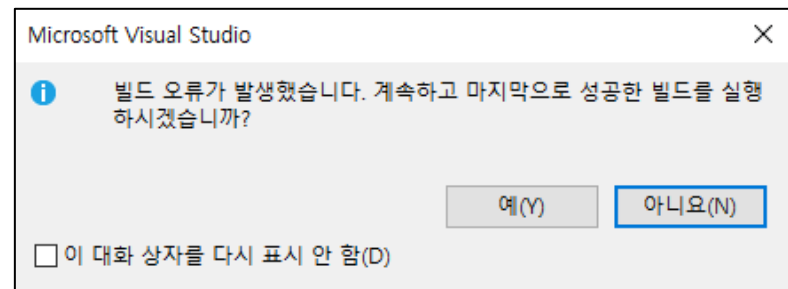
# 실험 1: OpenCV 세팅, 기본동작 확인

- 아래 코드 작성
- 구성, 플랫폼 설정
  - 각각 Release, x86(Win32)
- 빌드 및 실행
  - Ctrl + F5 누르기 > 에러 메시지에 '아니요' 누르기



The screenshot shows the Microsoft Visual Studio interface. The 'Build' menu is open, and the 'Release' configuration and 'x86' platform are selected. Below the menu, the code editor shows a C++ file named 'main.cpp' with the following code:

```
1 #include <opencv2/opencv.hpp>
2 #include <iostream>
3
4 using namespace cv;
5 using namespace std;
6
7 int main() {
8     Mat src;
9     string windowName = "LENA";
10    src = imread("lena.jpg");
11    imshow(windowName, src);
12    waitKey(0);
13 }
```



# 실험 1: OpenCV 세팅, 기본동작 확인

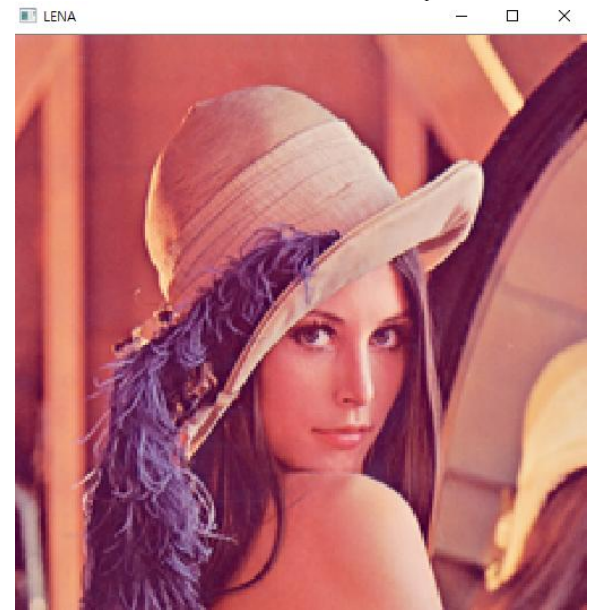
- 아래 코드로 제공된 lena 이미지 출력하기
- waitKey(N) : N ms 정지
- N=0일 때에는 키보드 입력 있을 때까지 정지

```
#include <opencv2\opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;

void main(void) {
    Mat src;
    string windowName = "LENA";
    src = imread("lena.jpg");

    imshow(windowName, src);

    waitKey(0);
}
```



# 실험 2: 샘플 영상 출력

## ■ 아래 코드로 샘플 영상 출력하기

```
1  #include <opencv2/opencv.hpp>
2  #include <iostream>
3
4  using namespace cv;
5  using namespace std;
6
7  int main() {
8      char keychar;
9      Mat frame;
10     VideoCapture vid_in = VideoCapture("sample.mp4");
11
12     if (vid_in.isOpened())
13     {
14         cout << "Error: Unable to open camera!" << endl;
15     }
16
17     while (1)
18     {
19         vid_in.read(frame);
20         if (frame.empty())
21         {
22             cout << "Error: Disconnection happened!" << endl;
23         }
24         imshow("CAM", frame);
25         keychar = waitKey(5);
26         if (keychar == 'c' || keychar == 'C')
27         {
28             break;
29         }
30     }
31 }
```

동영상에서 1개 이미지  
읽어서 frame에 저장하기

영상 재생 중 c키 누르면  
종료됨

# 실험 3: Gaussian Filter 사용

- 아래 코드로 이미지에 Gaussian filter 적용
- kernel\_size(홀수) 조절하며 blur의 정도 변경 가능

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;

void main(void){
    Mat src, dst;
    string winName1 = "Original";
    string winName2 = "Blurred";
    int kernel_size = 3;

    src = imread("lena.jpg");
    dst = src.clone();

    if (kernel_size % 2 == 0)
    {
        kernel_size += 1;
    }
    imshow(winName1, src);
```

```
GaussianBlur(src, dst, Size(kernel_size, kernel_size), 0, 0);
imshow(winName2, dst);

    waitKey(0);
}
```

# 실험 3: Gaussian Filter 사용

- kernel\_size를 0.1초마다 증가시키면서 이미지가 점점 흐릿하게 보이도록 statement 추가하기
- 힌트 : for 문, waitKey(100), kernel\_size+=2

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;

void main(void)
{
    Mat src, dst;
    string winName1 = "Original";
    string winName2 = "Blurred";
    int kernel_size = 3;

    src = imread("lena.jpg");
    dst = src.clone();

    if (kernel_size % 2 == 0)
    {
        kernel_size += 1;
    }
    imshow(winName1, src);

    /*
    Insert your code
    */

    waitKey(0);
}
```



# 실험 4: Sobel Filter 사용

## ■ 아래 코드로 입력 이미지에 Sobel filter 적용

```
#include <opencv2\opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;

void main(void)
{
    Mat src, dst;
    string winName1 = "Original";
    string winName2 = "Edge";
    int scale = 1; //Sobel filter 연산결과에 곱할 값
    int delta = 0; //Sobel filter 연산결과에 더할 값
    int kernel_size = 3; //Sobel filter의 크기 3x3
    int ddepth = CV_16S; //Filter 연산시 overflow를 방지하기 위한
    형변환할 대상

    src = imread("lena.jpg");
    dst = src.clone();

    /* edge detect start */
    cvtColor(src, dst, CV_BGR2GRAY); /// 편의상 흑백이미지로 변환

    Mat grad_x, grad_y, grad;
    Mat abs_grad_x, abs_grad_y;

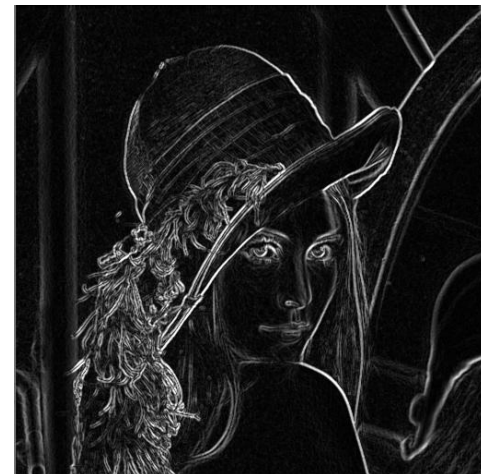
    /// X방향에 대해 Sobel filter를 적용하여 grad_x에 저장
    Sobel(dst, grad_x, ddepth, 1, 0, kernel_size, scale, delta,
    BORDER_DEFAULT);
    /// Y방향에 대해 Sobel filter를 적용 grad_y에 저장
    Sobel(dst, grad_y, ddepth, 0, 1, kernel_size, scale, delta,
    BORDER_DEFAULT);
```

```
/// Sobel filter를 사용하면 음수가 발생할 수 있다 픽셀 값은 0-255의
값이므로 아래 함수를 이용하여 절대값을 구한다. 결과는 abs_grad_*에 저장
convertScaleAbs(grad_x, abs_grad_x);
convertScaleAbs(grad_y, abs_grad_y);

/// X와 Y방향에 대한 Sobel filter의 결과를 0.5의 가중치를 두고
합친다. 결과는 grad에 저장. 이 함수에 대한 설명은 추가 실험 참고.
addWeighted(abs_grad_x, 0.5, abs_grad_y, 0.5, 0, grad);
/* edge detect end */

imshow(winName2, grad);

waitKey(0);
}
```



결과 예시

# 실험 4: Sobel Filter 사용

- 샘플 영상에 Sobel filter 적용하도록 코드 추가
- 변경: VideoCapture(0) > VideoCapture("sample.mp4")

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;

void main(void)
{
    char keychar;
    Mat frame;
    VideoCapture vid_in = VideoCapture(0);

    if (!vid_in.isOpened())
    {
        cout << "Error: Unable to open camera!" << endl;
    }

    while(1)
    {
        vid_in.read(frame);

        /* edge detect start */
        /* edge detect end */

        if (frame.empty())
        {
            cout << "Error: Disconnection happened!" << endl;
        }

        imshow("CAM", frame);

        keychar = waitKey(5);
        if (keychar == 'c' || keychar == 'C')
        {
            break;
        }
    }
}
```

# 실험 검사 항목

---

## ■ 실험 1

- 이미지가 정상적으로 출력되는지 확인

## ■ 실험 2

- 입력 동영상이 정상적으로 출력되는지 확인

## ■ 실험 3

- Gaussian filter 적용 결과가 정상적으로 출력되는지 확인
- 100ms 간격으로 blurring이 심화되는지 확인

## ■ 실험 4

- 이미지 및 동영상에 대해서 Sobel filter 적용 결과가 정상적으로 출력되는지 확인

# 결과보고서 항목

---

- 모든 실험 과정 및 결과 정리
- 추가실험 A (코드 첨부 필수)
  - 제공된 “alias\_test.jpg” 이미지를 단순 축소한 것과 Gaussian filter를 적용한 후에 축소한 것을 비교하여 Gaussian filter가 만들어낸 효과에 대해 주파수 관점에서 설명
  - 사용할 축소 함수 : `resize(src, dst, Size(), 0.5, 0.5, 0)`
- 추가실험 B (코드 첨부 필수)
  - 실험3 두 번째 항목의 코드와 실험4에서 사용한 `addWeighted()` 함수를 사용해서 이미지 내 모서리를 강화하는 image sharpening 구현
  - 원본 이미지와 Sobel filter 적용 결과의 가중합
  - 교재 10주차 추가 실험B 설명 참고!

# 결과보고서 항목

## ■ 추가 실험 교재 설명

### 6. 추가 실험

- A. "alias\_test.jpg" 이미지를 불러온다. 그리고 `resize(src, dst, Size(), 0.5, 0.5, 0)` 함수를 이용하여 이미지가 절반으로 축소하여 `dst`에 저장한다. 여기서 함수에 들어갈 parameter는 첫 번째와 두 번째만 (`src, dst`) 빼고 나머지는 그대로 유지한다. 네 번째와 다섯 번째 parameter는 각각 `x`와 `y`방향으로 얼마나 `resizing`할 지 결정하는 `factor`이다.

그리고 원본 이미지에 Gaussian 블러링을 한 다음 축소를 한 것과 비교하여 어떻게 다른지를 보고서에 작성하고 코드를 첨부한다.

- B. 실험3의 두 번째 항목의 코드에 실험4의 `addWeighted()` 함수를 추가하여 image sharpening (이미지의 엷지를 강화)하는 작업을 한다. 결과는 100ms마다 바뀔 것이다. 발생하는 현상을 보고서에 작성하고 코드를 첨부한다.

참고로 `addWeighted()`는 두 이미지에 각각 일정한 가중치를 두어 합해주는 함수이다. 함수의 argument와 그에 대한 제한 조건은 다음과 같다.

`addWeighted(first image, weight of first image, second image, weight of second image, offset(usually 0), output Mat data)`

Constraint:  $\text{weight of first image} + \text{weight of second image} = 1$