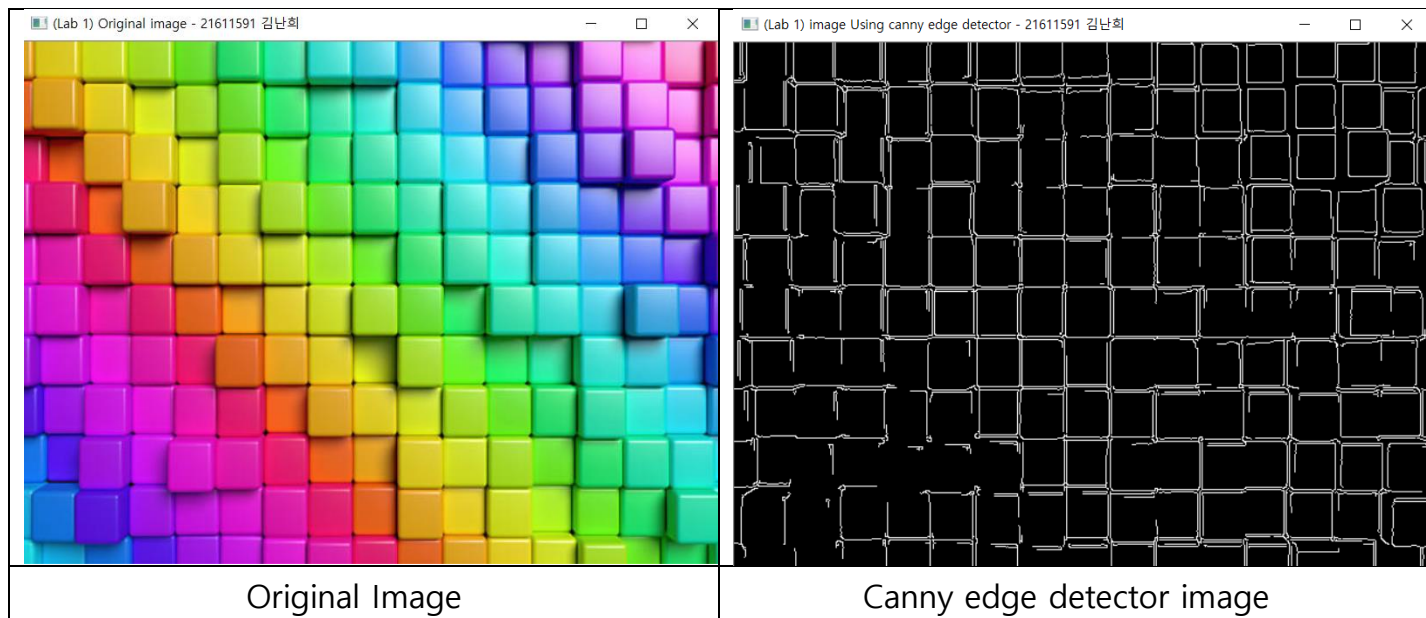


# [Computer Vision Programming] Lab 5. Canny Edge Detector

전자공학과 21611591 김난희

## Lab. 1: Simple Canny Edge Detector

-결과 사진



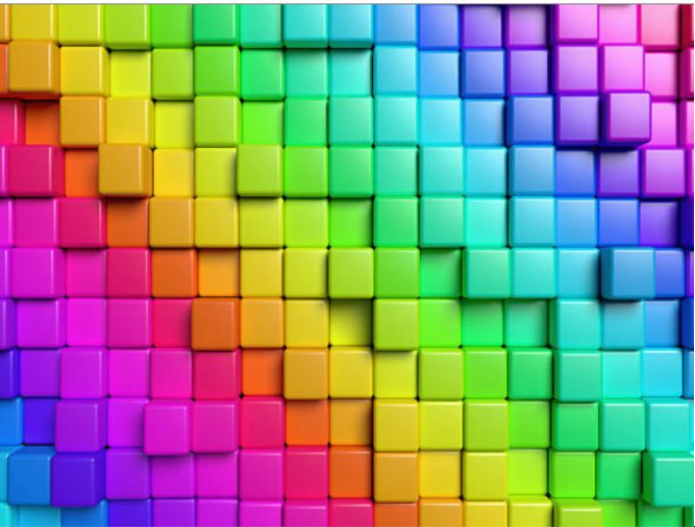
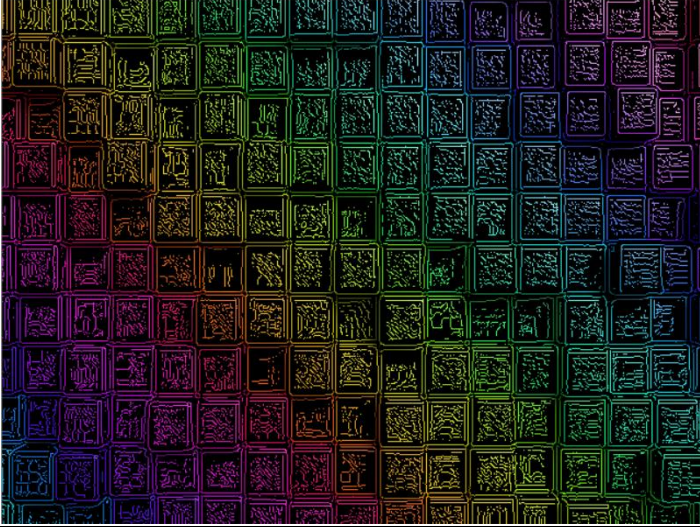
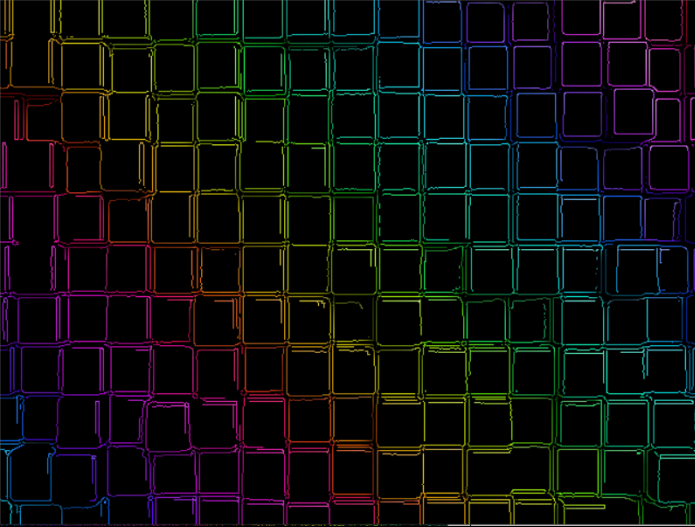
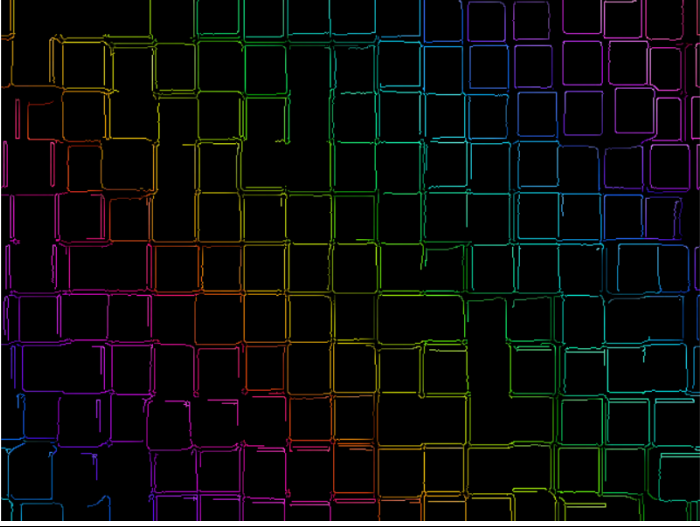
Edge만 검출하여 나온 사진을 볼 수 있다.

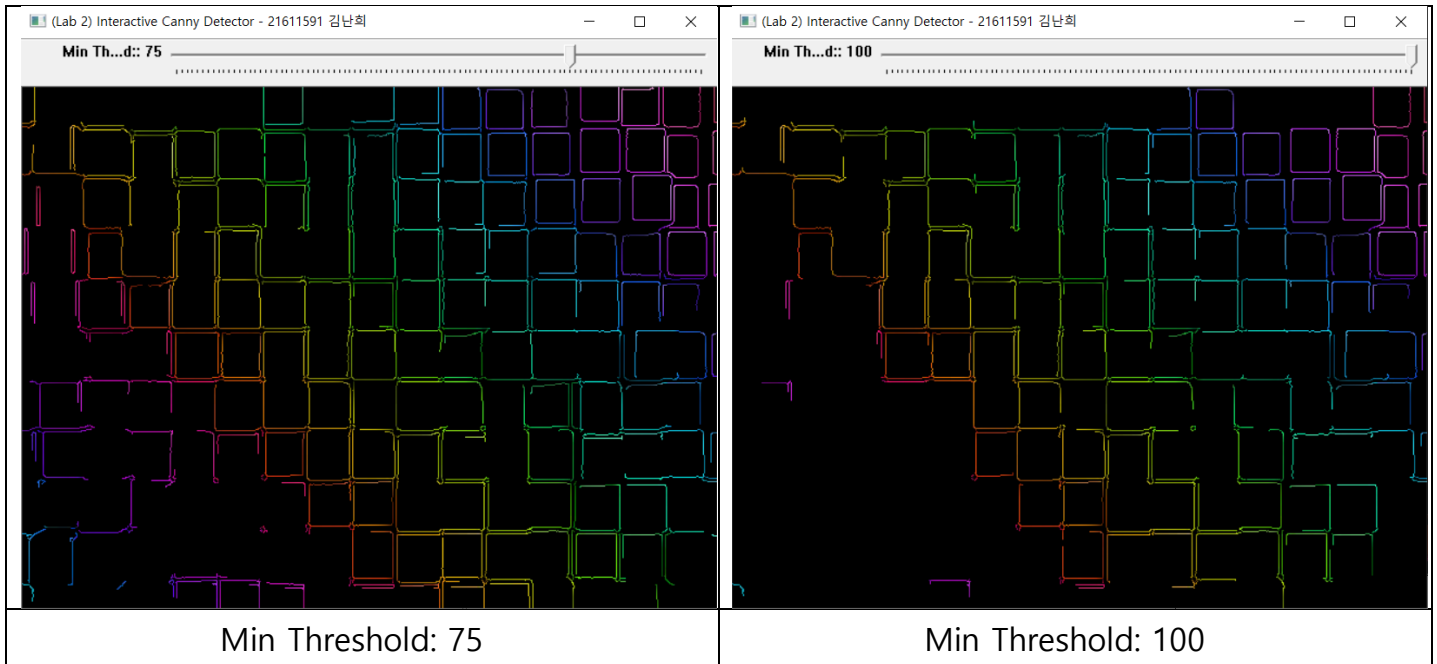
-소스 코드

```
1  #include "opencv2/imgproc/imgproc.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3
4  using namespace cv;
33 int main()
34 {
35     //***** ***** Lab.1 Simple Canny Edge Detector ***** *****//
36
37     // Load an image
38     src = imread("colorful stock.jpg");
39     resize(src, src, Size(640, 480));
40     //namedWindow("Original image", CV_WINDOW_AUTOSIZE);
41     imshow("(Lab 1) Original image - 21611591 김난희", src); //show original image
42
43     Mat gray, edge, draw; //gray 이미지, edge 결과 이미지, edge 결과를 gray로 바꿀 이미지
44     cvtColor(src, gray, CV_BGR2GRAY); //원본 이미지를 gray 이미지로 바꿈
45     Canny(gray, edge, 100, 200, 3); //@input, output, low threshold(linking), high threshold, aperture
46     edge.convertTo(draw, CV_8U); //to convert gray scale 1채널 unsigned char
47     //namedWindow(" image", CV_WINDOW_AUTOSIZE);
48     imshow("(Lab 1) image Using canny edge detector - 21611591 김난희", draw);
49 }
```

# Lab. 2: Interactive Canny Detector

-결과 사진

	
Original Image	Min Threshold: 0
	
Min Threshold: 25	Min Threshold: 50



Lab 1에서 edge를 해당하는 색상으로 표현한 것이다.

Threshold 값을 올릴수록 edge가 약한 부분은 사라져서 잘 나타나지 않는 것을 볼 수 있다. Threshold 값이 0일 때 edge가 아닌 배경 색도 나타난다. Image마다 배경 색이 없어지는 Threshold 값은 다르다.

#### -소스 코드

```

6 // Global variables of Lab.2
7 Mat src, src_gray; //원본 이미지, gray 전환 이미지
8 Mat dst, detected_edges; //결과 이미지, edge 이미지
9 //int edgeThresh = 1; //필요없는 변수
10 int lowThreshold; //현재 slicing 위치
11 int const max_lowThreshold = 100; //slicing 단위(범위: 0~100)
12 int ratio = 3; //lowThreshold = highThreshold
13 int kernel_size = 3; //aperture
14 char* window_name = "(Lab 2) Interactive Canny Detector - 21611591 김난희";
15
16 void CannyThreshold(int, void*)
17 {
18     // @function CannyThreshold // @brief Tracker callback - Canny thresholds input with a ratio 1:3
19
20     // Reduce noise with a kernel 3x3 @input, output, kernel size
21     blur(src_gray, detected_edges, Size(3, 3));
22
23     // Canny detector @input, output, 최소 threshold1, 최대 threshold2, aperture
24     Canny(detected_edges, detected_edges, lowThreshold, lowThreshold*ratio, kernel_size);
25
26     // Using Canny's output as a mask, we display our result
27     dst = Scalar::all(0); //fill zero matrix //검은색 배경
28     src.copyTo(dst, detected_edges); // mask: detected_edges //배경 위에 edge 씌움
29
30     //show result about colorful edge detect
31     imshow(window_name, dst);
32 }

```

Main function 내부



```

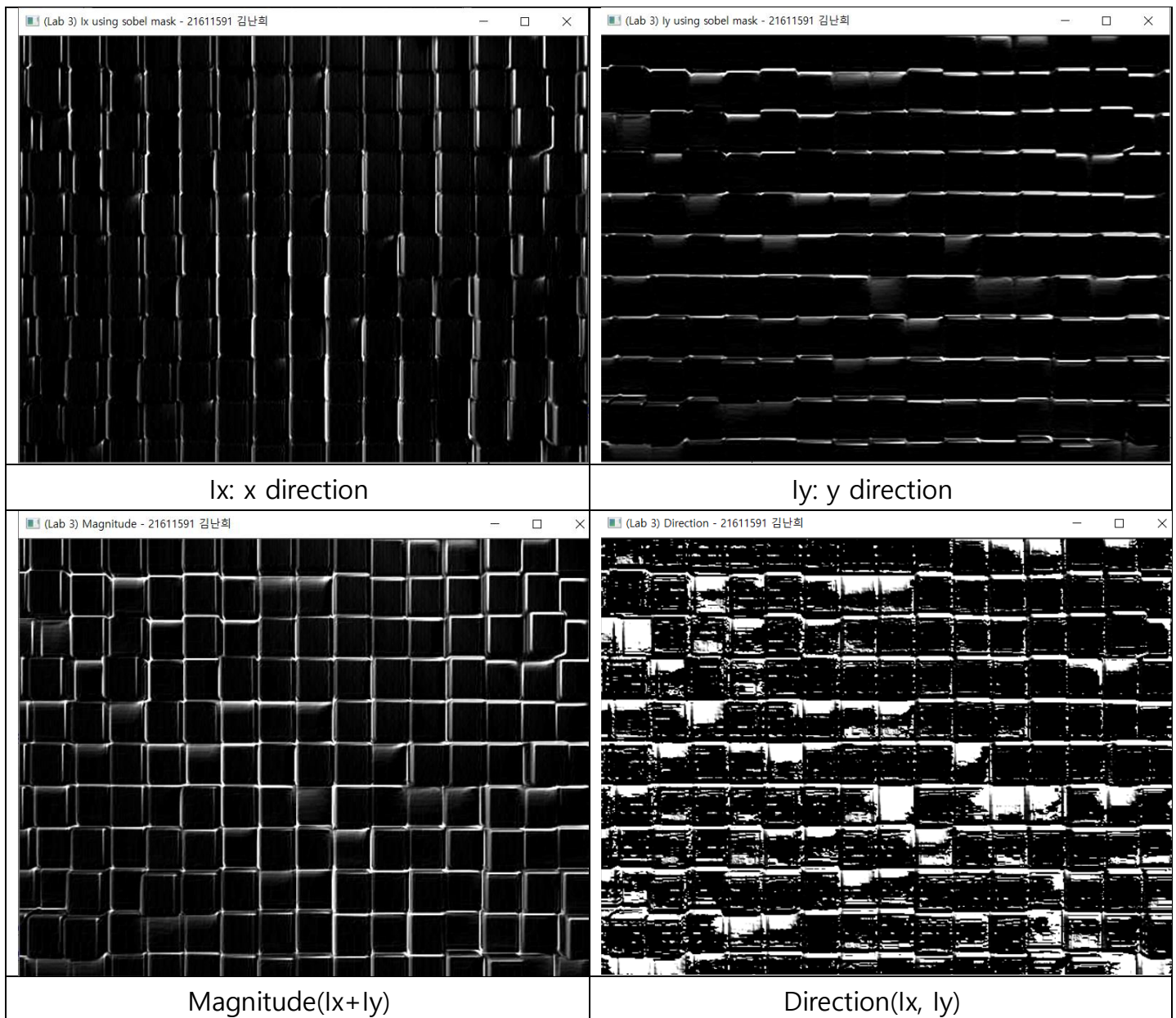
51 //***** ***** Lab.2 Interactive Canny Detector ***** **//
52
53 // Create a matrix of the same type and size as src (for dst)
54 dst.create(src.size(), src.type());
55
56 // Convert the image to grayscale @input, output, BGR to GRAY
57 cvtColor(src, src_gray, CV_BGR2GRAY);
58
59 // Create a window
60 namedWindow(window_name, CV_WINDOW_AUTOSIZE);
61 // Create a Trackbar for user to enter threshold
62 // @Trackbar이름, Window이름, 현재 slicing 위치, slicing 단위, 적용할 함수
63 createTrackbar("Min Threshold:", window_name, &lowThreshold, max_lowThreshold, CannyThreshold);
64
65 // Show the image //thread
66 CannyThreshold(0, 0);

```

## Lab. 3: Sobel Edge

-결과 사진

Lena 이미지 사용하는 것을 권장했지만, edge 방향성이 잘 드러나도록 다른 이미지를 선택하였습니다.



가로 방향 미분을 하면 세로 방향으로 edge가 검출되고, 세로 방향 미분을 하면 가로 방향 edge가 검출되는 것을 볼 수 있다. Magnitude는 각 방향 성분의 크기를 구한 것이다. Direction은 edge가 변하는 방향으로 각도를 구할 수 있는데, 그러한 결과 그림이 나타났다.

#### -소스 코드

```
68 //***** Lab.3 Sobel Edge *****//
69
70 // Sobel Filter @input, output, depth(출력이미지 깊이조합,8비트:절단유도체, dx, dy
71 Mat sobelMag, sobelDir, sobelX, sobelY;
72
73 Sobel(gray, sobelX, CV_8U, 1, 0); //x direction
74 imshow("(Lab 3) Ix using sobel mask - 21611591 김난희", sobelX);
75
76 Sobel(gray, sobelY, CV_8U, 0, 1); //y direction
77 imshow("(Lab 3) Iy using sobel mask - 21611591 김난희", sobelY);
78
79 //magnitude(sobelX, sobelY, sobelMag);
80 sobelMag = abs(sobelX) + abs(sobelY); //magnitude
81 imshow("(Lab 3) Magnitude - 21611591 김난희", sobelMag);
82
83 sobelX.convertTo(sobelX, CV_32F); //phase 계산을 위해 float형변환
84 sobelY.convertTo(sobelY, CV_32F);
85 sobelMag.convertTo(sobelMag, CV_32F);
86
87 //cartToPolar(sobelX, sobelY, sobelMag, sobelDir);
88 phase(sobelX, sobelY, sobelDir); //@input x, input y, output angle
89 imshow("(Lab 3) Direction - 21611591 김난희", sobelDir); //direction
90
91 // Wait until user exit program by pressing a key
92 waitKey(0);
93 return 0;
94 }
```

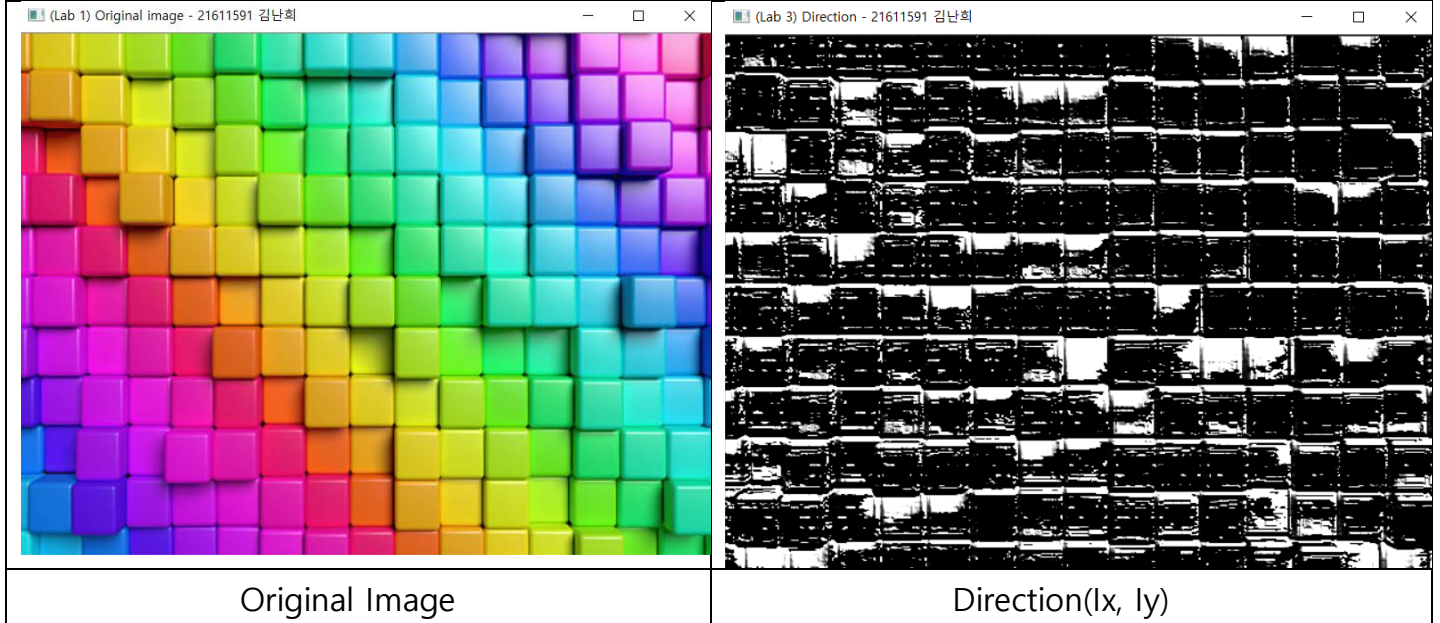
#### Discussion

첫 번째와 두 번째 Lab은 강의자료에 잘 나와있어서 그대로 따라하여 쉬웠다. 세 번째 Lab도 이론적으로 이해는 어느정도 하였으나, direction을 구하는 과제에서 시행착오를 겪었다.

Direction을 구하는 것은 edge의 수직 성분으로(기울기에 따라, 변하는 방향으로) 화살표로 표시를 하고 싶었다. 이번 Lab의 범위를 넘는 것 같아, 이론적으로 나와있는  $\text{atan}(Iy/Ix)$  값을 구하는 방법으로 시도하였다.

$\text{atan}$  값을 구하는 방법은 Phase, atan, atan2, cartToPolar 함수 등 다양한 방법으로 구할 수 있다. 그 중에서 가장 깔끔하게 구할 수 있는 방법을 선택하였다. 그러나 이러한 방법은 모두 float형 계산이 필요하여 소스 코드 중간에 float형 전환을 해주었다. 결과로 나타난 사진을 보아 정확한 정답인지는 모르나, stock이 튀어나와 그림자가 지는 곳이 밝게 표현되었다. 의도한 바를 어느정도 잘 표현한 것이라 생각하였다.

## Discussion 참고 사진



\*\*\* 전체 소스 코드 \*\*\*

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

using namespace cv;

// Global variables of Lab.2
Mat src, src_gray; //원본 이미지, gray 전환 이미지
Mat dst, detected_edges; //결과 이미지, edge 이미지
//int edgeThresh = 1; //필요없는 변수
int lowThreshold; //현재 slicing 위치
int const max_lowThreshold = 100; //slicing 단위(범위: 0~100)
int ratio = 3; //lowThreshold = highThreshold
int kernel_size = 3; //aperture
char* window_name = "(Lab 2) Interactive Canny Detector - 21611591 김난희";

void CannyThreshold(int, void*)
{
    // @function CannyThreshold // @brief Trackbar callback - Canny thresholds input with a ratio 1:3

    // Reduce noise with a kernel 3x3 @input, output, kernel size
    blur(src_gray, detected_edges, Size(3, 3));

    // Canny detector @input, output, 최소 threshold1, 최대 threshold2, aperture
    Canny(detected_edges, detected_edges, lowThreshold, lowThreshold*ratio, kernel_size);

    // Using Canny's output as a mask, we display our result
    dst = Scalar::all(0); //fill zero matrix //검은색 배경
    src.copyTo(dst, detected_edges); // mask: detected_edges //배경 위에 edge 씌움

    //show result about colorful edge detect
    imshow(window_name, dst);
}
```

```

int main()
{
    //***** ***** Lab.1 Simple Canny Edge Detector ***** ***/

    // Load an image
    src = imread("colorful_stock.jpg");
    resize(src, src, Size(640, 480));
    //namedWindow("Original image", CV_WINDOW_AUTOSIZE);
    imshow("(Lab 1) Original image - 21611591 김난희", src); //show original image

    Mat gray, edge, draw; //gray 이미지, edge 결과 이미지, edge 결과를 gray로 바꿀 이미지
    cvtColor(src, gray, CV_BGR2GRAY); //원본 이미지를 gray 이미지로 바꿈
    Canny(gray, edge, 100, 200, 3); //@input, output, low threshold(linking), high threshold, aperture
    edge.convertTo(draw, CV_8U); //to convert gray scale 1채널 unsigned char
    //namedWindow("image", CV_WINDOW_AUTOSIZE);
    imshow("(Lab 1) image Using canny edge detector - 21611591 김난희", draw);

    //***** ***** Lab.2 Interactive Canny Detector ***** ***/

    // Create a matrix of the same type and size as src (for dst)
    dst.create(src.size(), src.type());

    // Convert the image to grayscale @input, output, BGR to GRAY
    cvtColor(src, src_gray, CV_BGR2GRAY);

    // Create a window
    namedWindow(window_name, CV_WINDOW_AUTOSIZE);
    // Create a Trackbar for user to enter threshold
    // @Trackbar이름, Window이름, 현재 slicing 위치, slicing 단위, 적용할 함수
    createTrackbar("Min Threshold:", window_name, &lowThreshold, max_lowThreshold, CannyThreshold);

    // Show the image //thread
    CannyThreshold(0, 0);

    //***** ***** Lab.3 Sobel Edge ***** ***/

    // Sobel Filter @input, output, depth(출력이미지 깊이조합,8비트:절단유도체, dx, dy
    Mat sobelMag, sobelDir, sobelX, sobelY;

    Sobel(gray, sobelX, CV_8U, 1, 0); //x direction
    imshow("(Lab 3) Ix using sobel mask - 21611591 김난희", sobelX);

    Sobel(gray, sobelY, CV_8U, 0, 1); //y direction
    imshow("(Lab 3) Iy using sobel mask - 21611591 김난희", sobelY);

    //magnitude(sobelX, sobelY, sobelMag);
    sobelMag = abs(sobelX) + abs(sobelY); //magnitude
    imshow("(Lab 3) Magnitude - 21611591 김난희", sobelMag);

    sobelX.convertTo(sobelX, CV_32F); //phase 계산을 위해 float형변환
    sobelY.convertTo(sobelY, CV_32F);
    sobelMag.convertTo(sobelMag, CV_32F);

    //cartToPolar(sobelX, sobelY, sobelMag, sobelDir);
    phase(sobelX, sobelY, sobelDir); //@input x, input y, output angle
    imshow("(Lab 3) Direction - 21611591 김난희", sobelDir); //direction

    // Wait until user exit program by pressing a key
    waitKey(0);
    return 0;
}

```

---

\*\*\*참고 사이트\*\*\*

Sobel 설명

[https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel\\_derivatives/sobel\\_derivatives.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html)

magnitude 구하는 예제

<https://blog.thelinker.xyz/17>

direction 구하는 예제

<https://stackoverflow.com/questions/7405223/sobel-edge-detection-edge-orientation>  
<https://stackoverflow.com/questions/21881950/getting-error-in-calculating-the-magnitude-and-orientation-of-a-mat-using-c-an>

direction 방향성 표시하기

<https://www.learnopencv.com/histogram-of-oriented-gradients/>