2023 로봇스터디 사전 교육 3주차

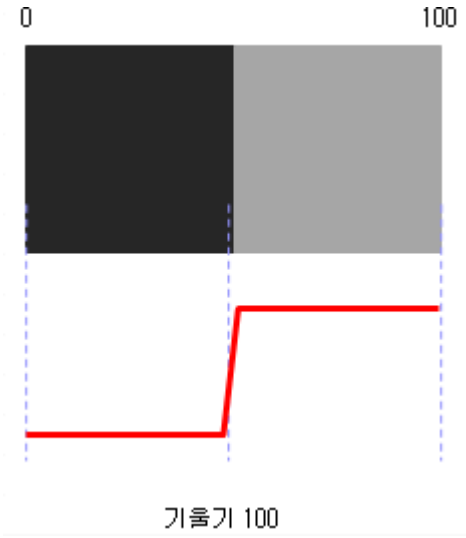
Image Processing Technic

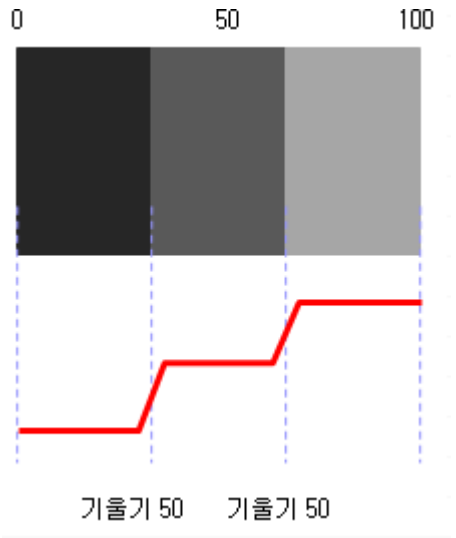
목차

1. Edge란?

1. Edge란?

- 경계선, 윤곽선
- 영상에서 밝기가 급격하게 변하는 부분 (낮은 값 -> 높은 값)
- Edge를 검출함으로써 물체의 위치, 모양, 크기, 방향성 등에 대한 정보를 쉽게 찾을 수 있다





1. Edge란?

- 픽셀 값이 급격하게 변하는 지점 -〉 Edge 부분

- 픽셀은 곡선 그래프처럼 연속공간에 있지 않으므로 미분 값이 아닌 미분 근사값으로 Edge를 찾을 수 있음
  -〉 서로 붙어 있는 픽셀 값의 차를 구하면 됨 (연속된 두 개의 픽셀 중 한쪽 값을 음수로 만든다)

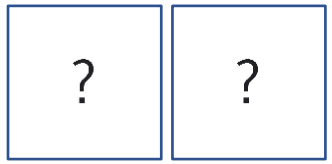
$$G_x \approx f_{x+1,y} - f_{x,y}$$

$$G_y \approx f_{x,y+1} - f_{x,y}$$

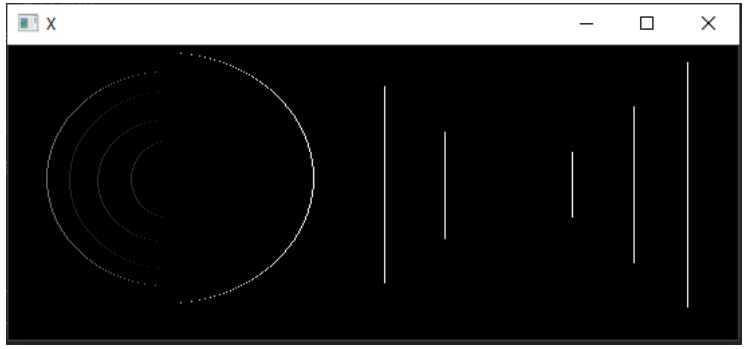
1. Edge란?

$$G_x = [-1 \quad 1]$$

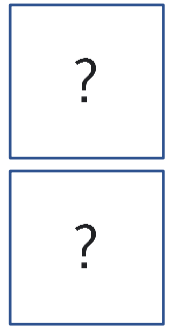
$$G_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$



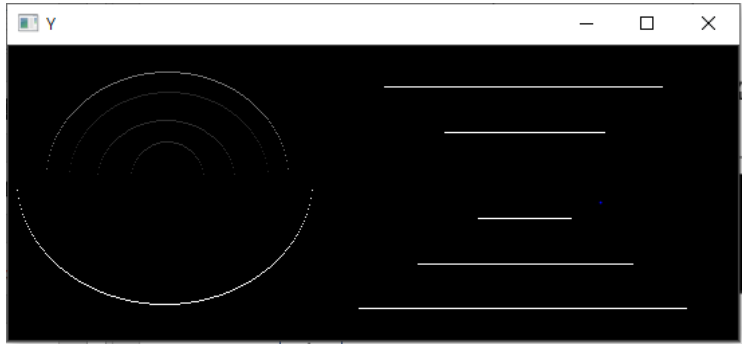
X축 필터



+



Y축 필터



=

2. Sobel Filter

2. Sobel Filter

- Sobel이 고안해낸 가장자리 검출 알고리즘

- 3X3 크기의 행렬을 사용하여 연산하였을 때 중심을 기준으로 각 방향의 값을 비교하여 픽셀 값의 변화량 검출 -〉 edge 판별 가능

-1	0	+1
-2	0	+2
-1	0	+1

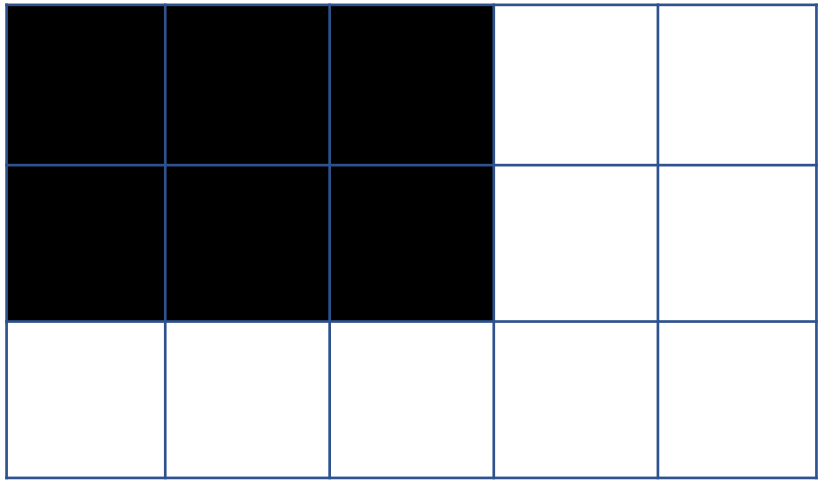
x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

- X축 필터 : 세로 성분 검출
- Y축 필터 : 가로 성분 검출

2. Sobel Filter

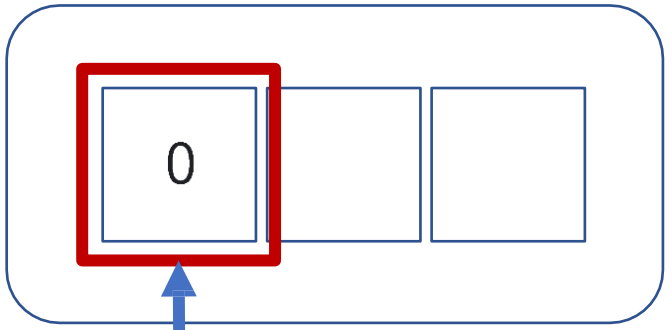


*

X축 필터

-1	0	+1
-2	0	+2
-1	0	+1

-〉 세로 Edge 검출 불가능



0	0	0	255	255
0	0	0	255	255
255	255	255	255	255

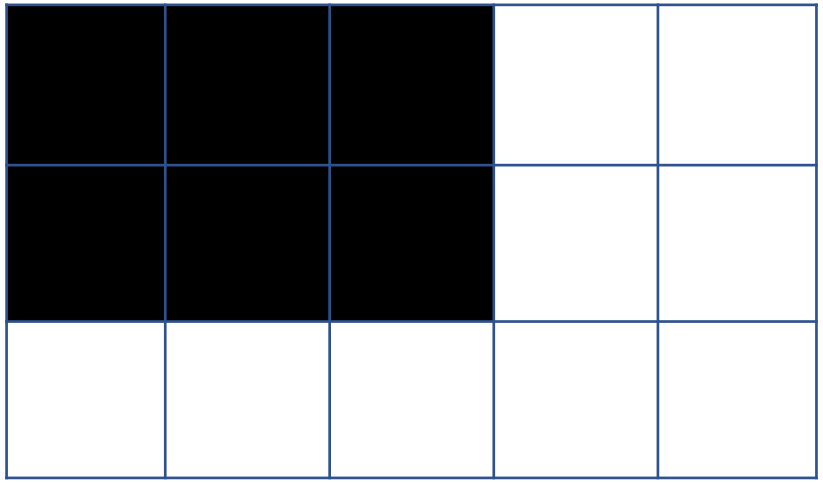
*

-1	0	+1
-2	0	+2
-1	0	+1

=

0	0	0
0	0	0
-255	0	255

2. Sobel Filter



*

X축 필터

-1	0	+1
-2	0	+2
-1	0	+1

-> 세로 Edge 검출 가능

0	0	0	255	255
0	0	0	255	255
255	255	255	255	255

*

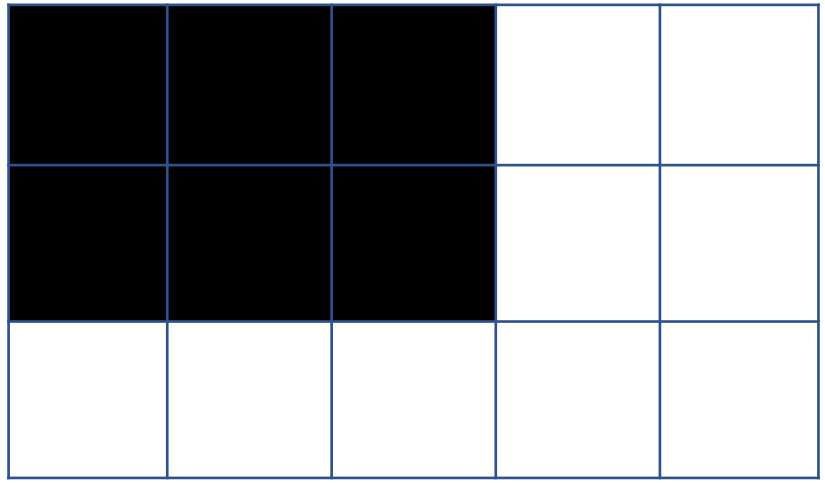
-1	0	+1
-2	0	+2
-1	0	+1

=

0	0	255
0	0	510
-255	0	255

0	765

2. Sobel Filter



*

X축 필터

-1	0	+1
-2	0	+2
-1	0	+1

0	0	0	255	255
0	0	0	255	255
255	255	255	255	255

*

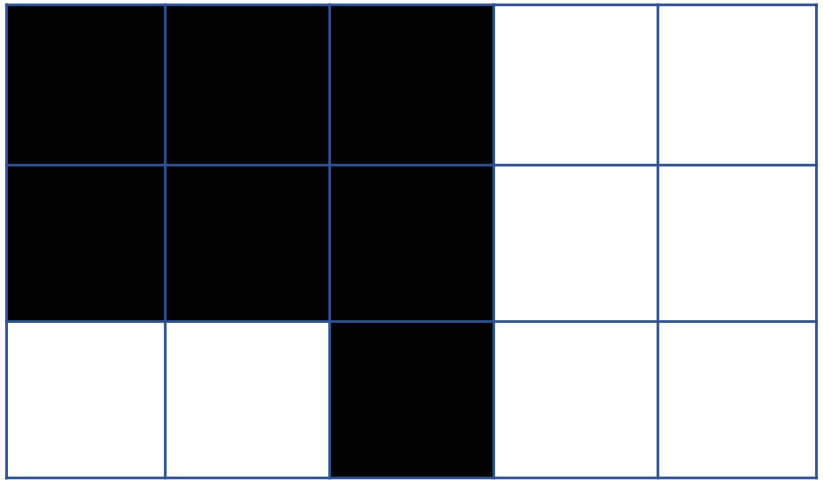
-1	0	+1
-2	0	+2
-1	0	+1

=

0	0	255
0	0	510
−255	0	255

0	765	765

2. Sobel Filter

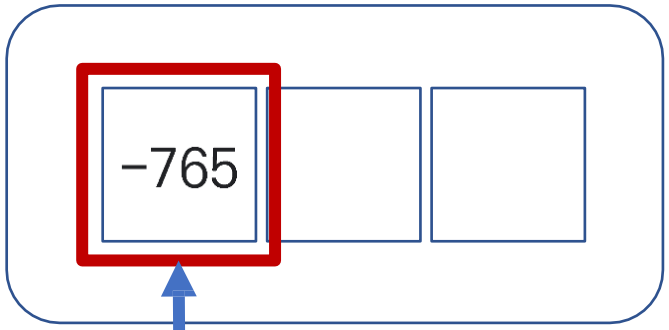


*

Y축 필터

+1	+2	+1
0	0	0
-1	-2	-1

-> 가로 Edge 검출 가능



0	0	0	255	255
0	0	0	255	255
255	255	0	255	255

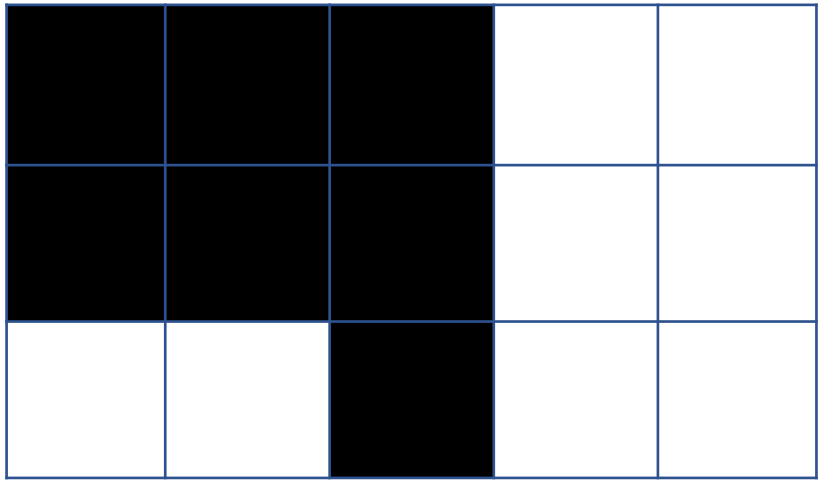
*

+1	+2	+1
0	0	0
-1	-2	-1

=

0	0	0
0	0	0
-255	-510	0

2. Sobel Filter



*

Y축 필터

+1	+2	+1
0	0	0
-1	-2	-1

0	0	0	255	255
0	0	0	255	255
255	255	0	255	255

*

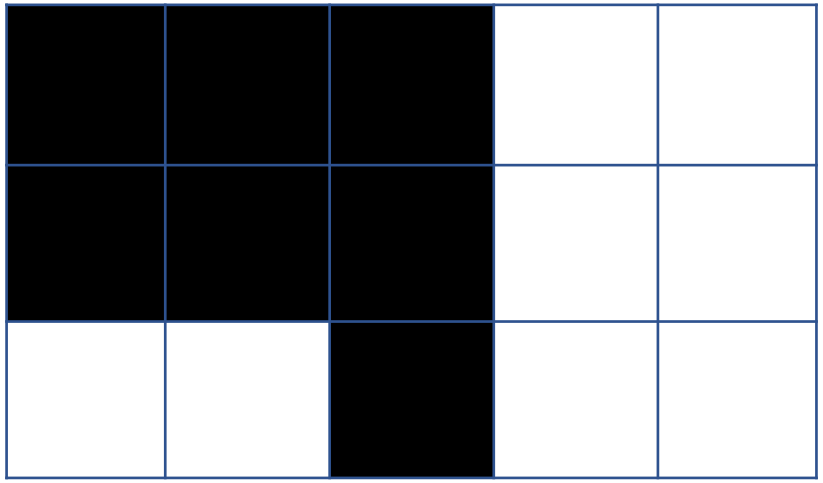
+1	+2	+1
0	0	0
-1	-2	-1

=

0	0	255
0	0	0
-255	0	-255

-765	-255	

2. Sobel Filter



*

Y축 필터

+1	+2	+1
0	0	0
-1	-2	-1

-> 가로 Edge 검출 불가능

-765	-255	0

0	0	0	255	255
0	0	0	255	255
255	255	0	255	255

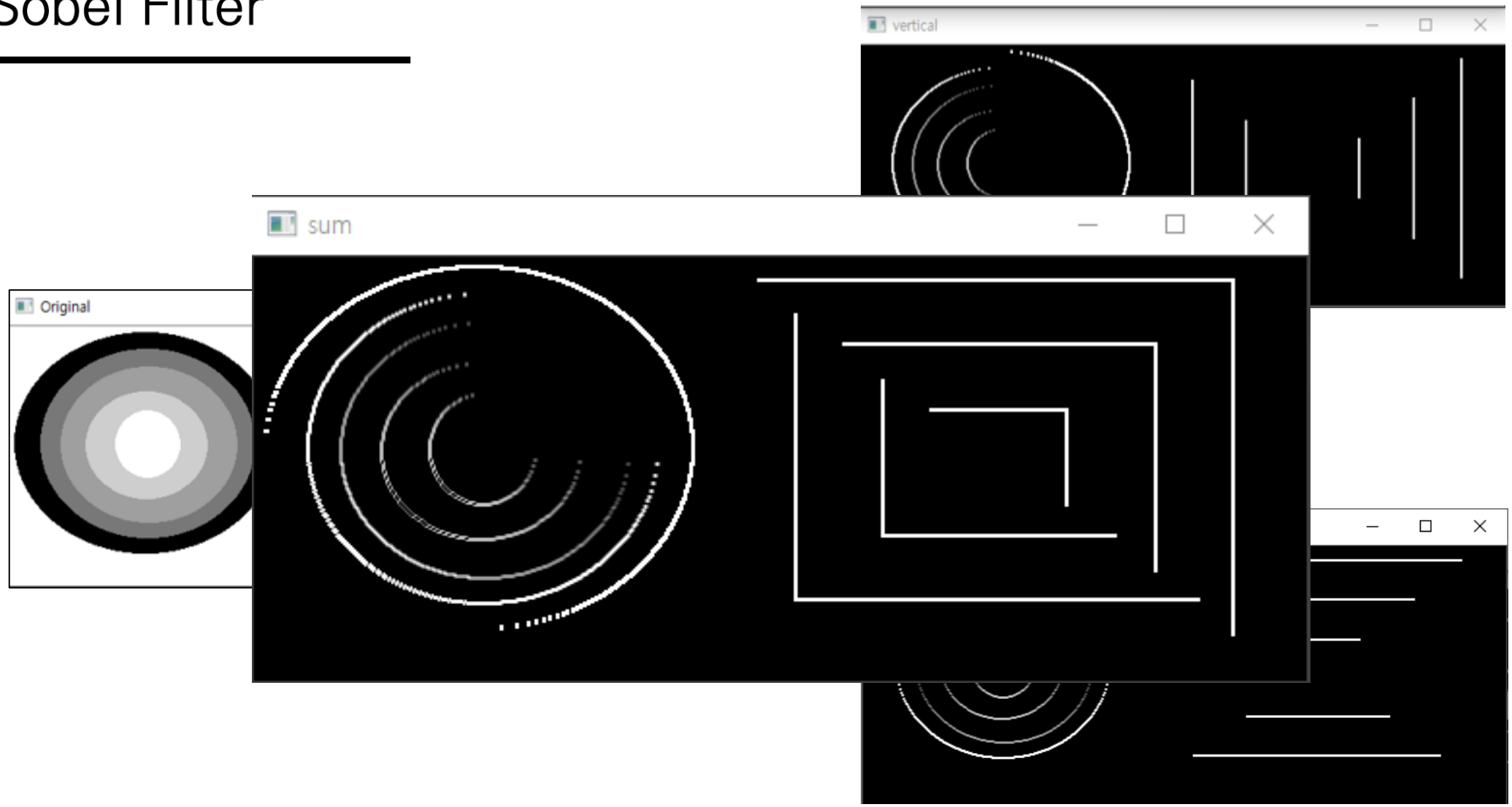
*

+1	+2	+1
0	0	0
-1	-2	-1

=

0	510	255
0	0	0
0	-510	-255

2. Sobel Filter



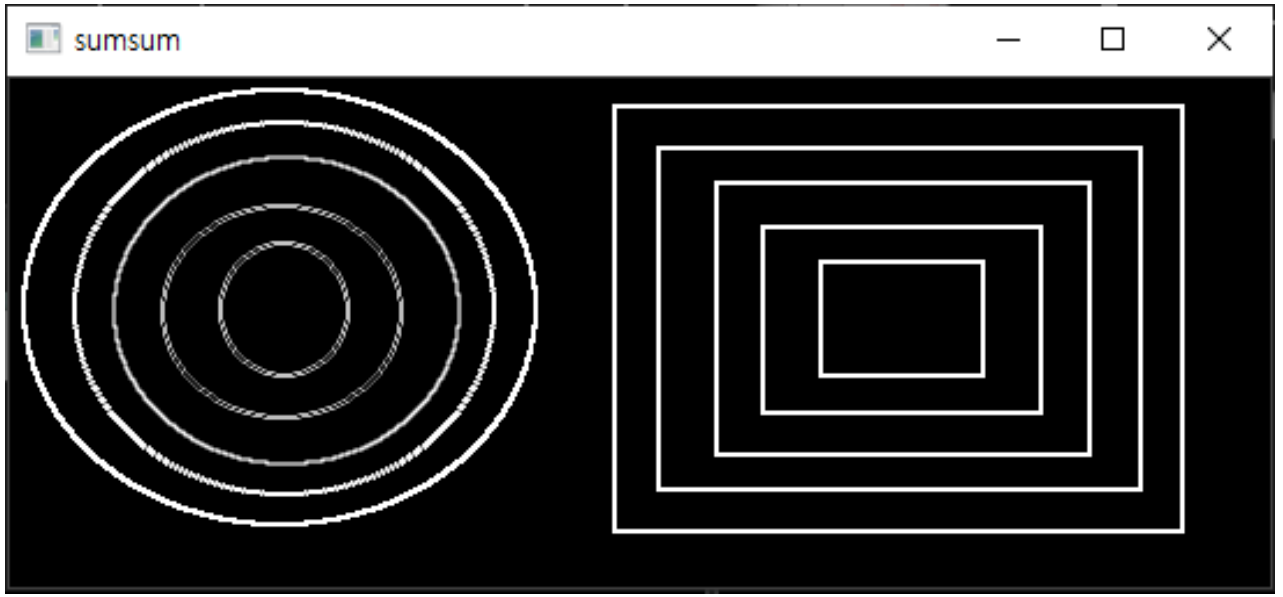
Y축 필터

2. Sobel Filter

/ 방향 edge 검출 필터

\ 방향 edge 검출 필터

2. Sobel Filter



네 개의 필터를 사용하여 위와 같은 결과 만들기

2. Sobel Filter

Sobel Filter의 특징

- 모든 방향의 Edge 추출 가능
- 돌출한 화소 값을 평균화 하므로 잡음에 대체적으로 강함
- 수직, 수평보다 대각선 방향 Edge에 더 민감하게 반응

3. Canny Edge

3. Canny Edge

Canny Edge 특징

- Edge 검출 알고리즘에서 가장 신뢰성이 높고 활용하기 간편하여 보편화 된 알고리즘
- 도형의 윤곽을 하나의 선으로 얻을 수 있음
- 다양한 환경에 적용 가능
- 이미지의 특징 추출을 위한 전처리로 많이 활용함

〈원본 이미지〉

〈Canny Edge를 사용한 Edge 검출 이미지〉

3. Canny Edge

1. 노이즈 제거

5X5 사이즈의 가우시안 필터를 이용해
이미지의 노이즈를 제거

〈블러링 된 GrayScale 이미지〉

```
# 1. 노이즈 제거 - 가우시안 블러링
gauss_filter = cv.getGaussianKernel(5, 3)    # kernal size : 5, sigma : 3
gauss_img = cv.filter2D(img, -1, gauss_filter)
```

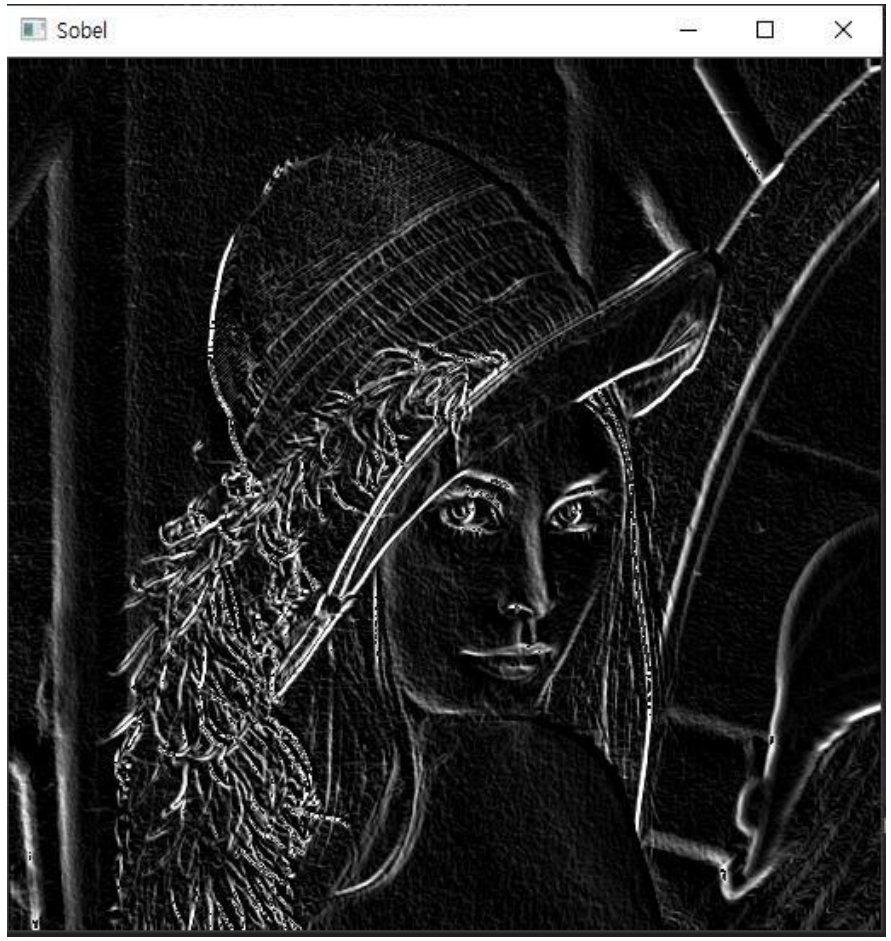
3. Canny Edge

2. Edge Gradient(기울기) 계산

소벨 필터를 사용하여 각 방향의

Edge 및 Gradient 검출

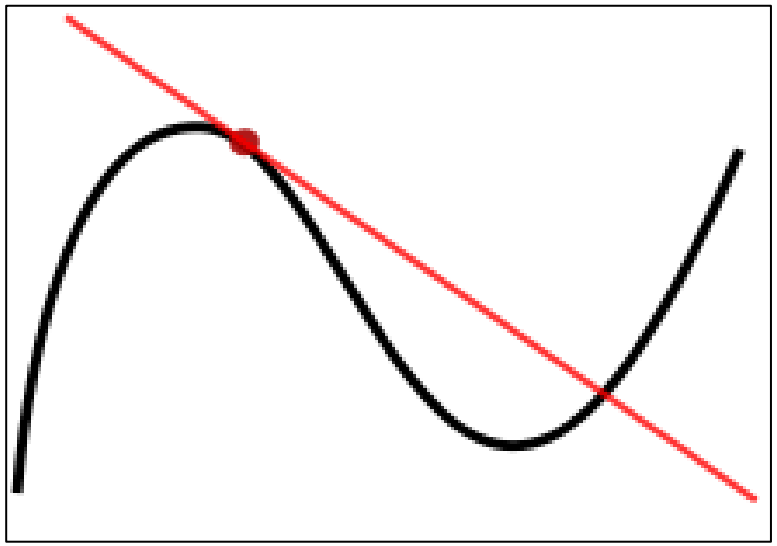
```
# 2. 경계 Gradient 방향 계산 - Sobel Filter 사용
# Sobel X축 Filter
kernel1 = np.array([[-1, 0, 1],
                    [-2, 0, 2],
                    [-1, 0, 1]])
dst1 = cv.filter2D(img, -1, kernel1)
# Sobel Y축 Filter
kernel2 = np.array([[1, 2, 1],
                    [0, 0, 0],
                    [-1, -2, -1]])
dst2 = cv.filter2D(img, -1, kernel2)
```

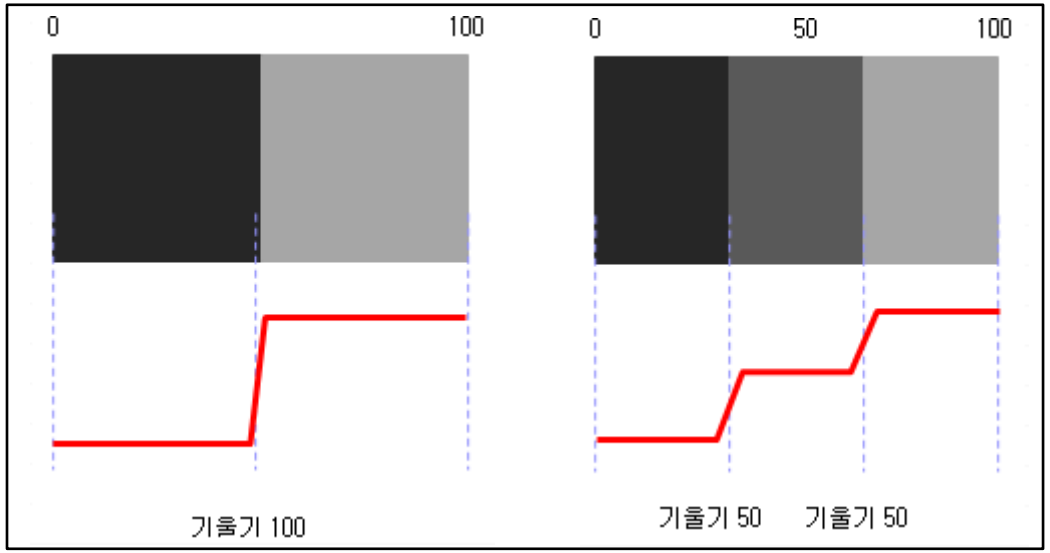


〈Sobel Filter 적용한 이미지〉

3. Canny Edge

2. Edge Gradient(기울기) 계산





함수의 점에서 미분 = 그 점에서의 접선의 기울기

픽셀에서의 미분 근사값 = 연속된 픽셀 값의 차이

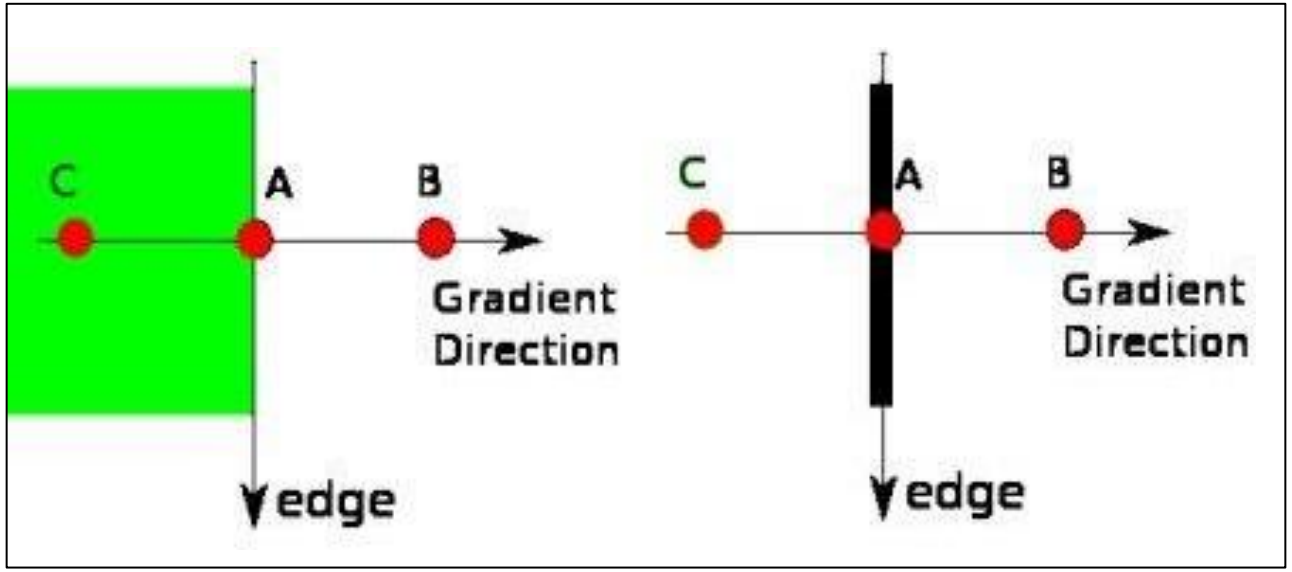
기울기 = 연속된 픽셀 값의 차이 = Gradient

함수의 그래프 미분 -〉 모든 픽셀 Sobel Filter로 Convolution 연산

3. Canny Edge

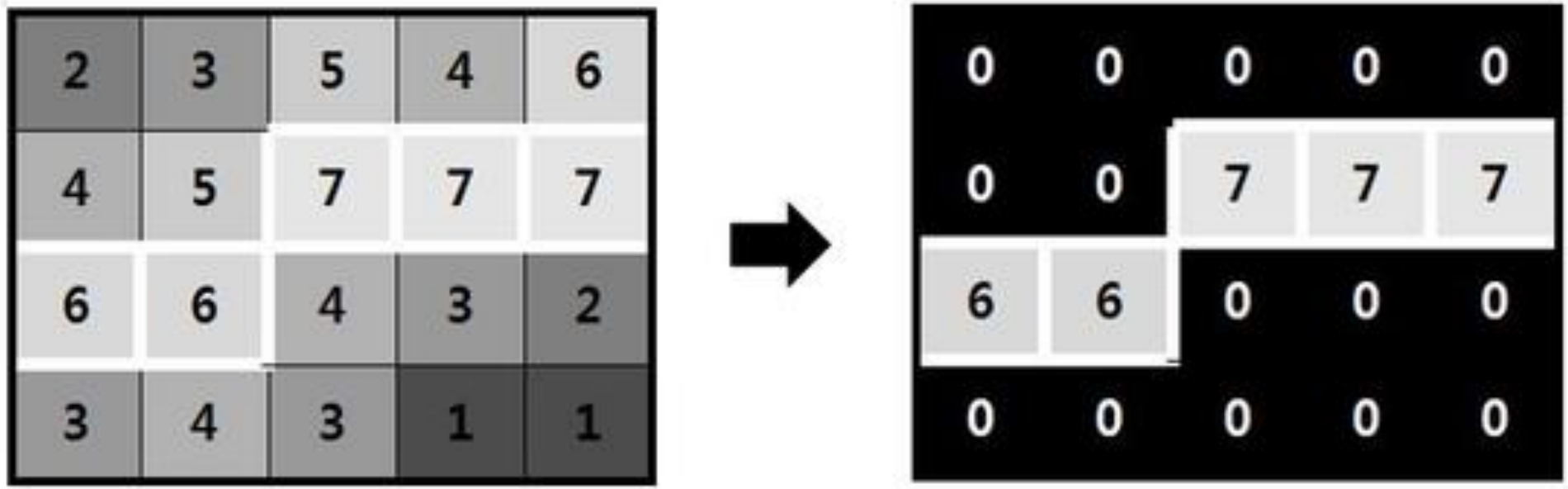
3. 비최대치 억제 (Non-Maximum Suppression)

Edge 추출에 기여도가 적은 픽셀을 제거하기 위해 이미지를 Full Scan 하여 Edge Gradient 방향 부근의 픽셀들이 지역 최대치인지 판별 -> 최대가 아닌 픽셀은 억제하여 0으로 만든다



3. Canny Edge

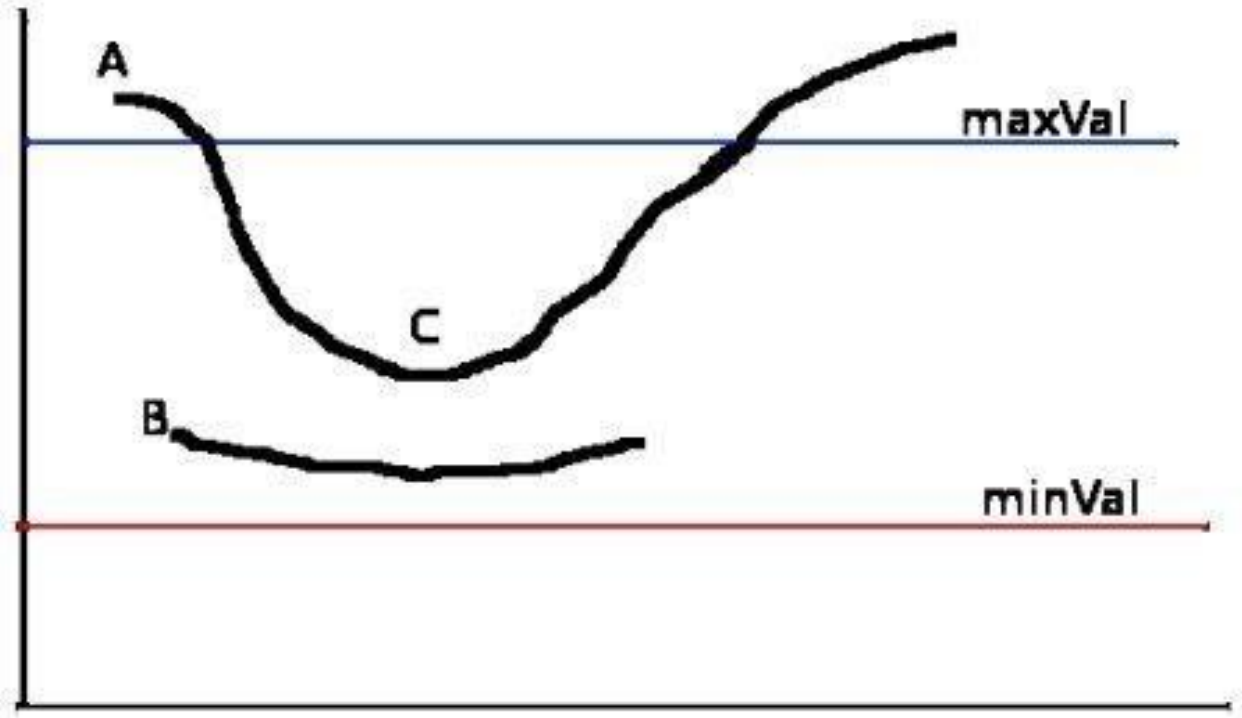
3. 비최대치 억제 (Non-Maximum Suppression)



3. Canny Edge

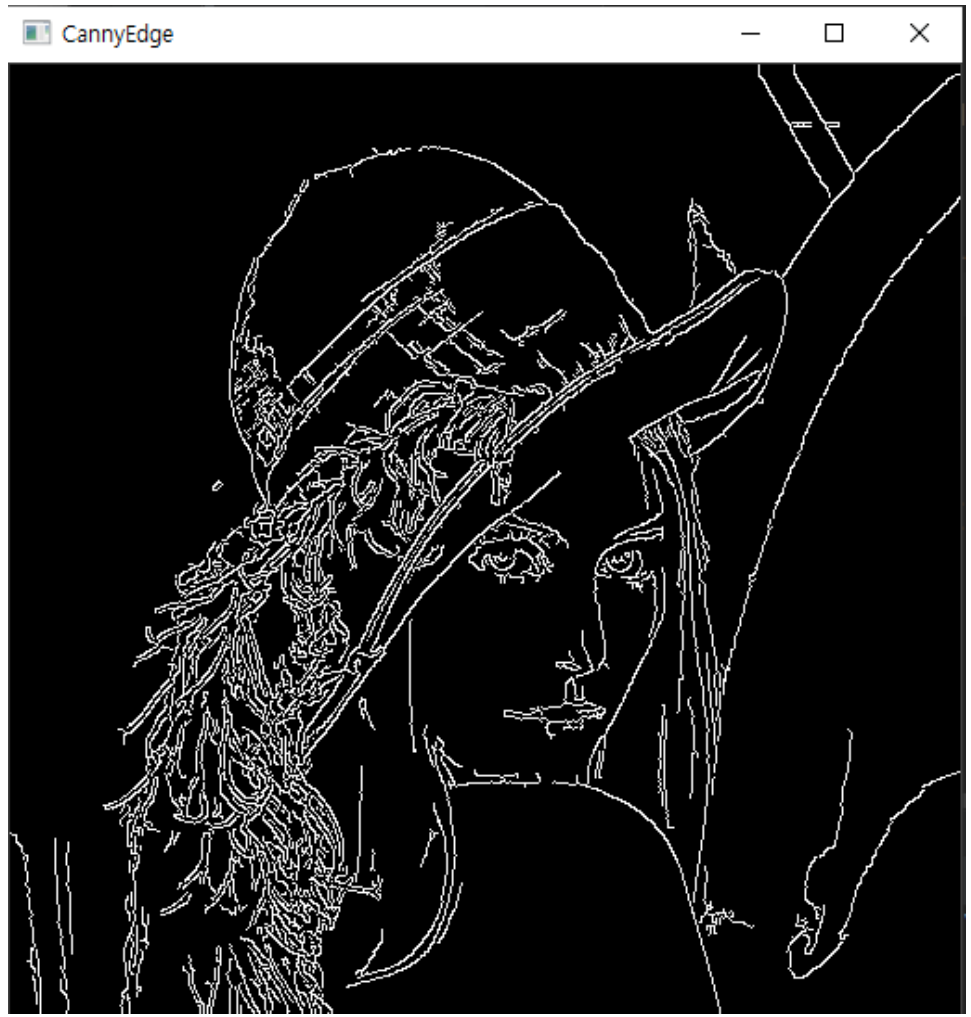
4. 이력 스레시홀딩 (Hyteresis Thresholding)

Gradient 강도가 maxVal보다 크면 Edge, minVal보다 작으면 Edge가 아니라고 판단하여 제거



3. Canny Edge

결과 영상



4. Edge Detection

Roberts Cross Filter

- 대각선 방향으로 +1과 −1을 배치시켜 사선 경계 검출 효과를 높였다
- 노이즈에 민감함

1	0
0	−1

X축 필터

0	1
−1	0

Y축 필터

Prewitt Filter

- 영상의 x축, y축의 각 방향으로 차분을 세 번 계산하여 경계를 검출

- 상하좌우 Edge는 뚜렷하지만 대각선 검출에 약함

−1	0	1
−1	0	1
−1	0	1

X축 필터

−1	−1	−1
0	0	0
1	1	1

Y축 필터

Scharr Filter

- 커널의 중심에서 멀어질수록 Edge 방향성의 정확도가 떨어지는 소벨 필터를 개선

−3	0	3
−10	0	10
−3	0	3

X축 필터

−3	−10	−3
0	0	0
3	10	3

Y축 필터

Laplacian Filter

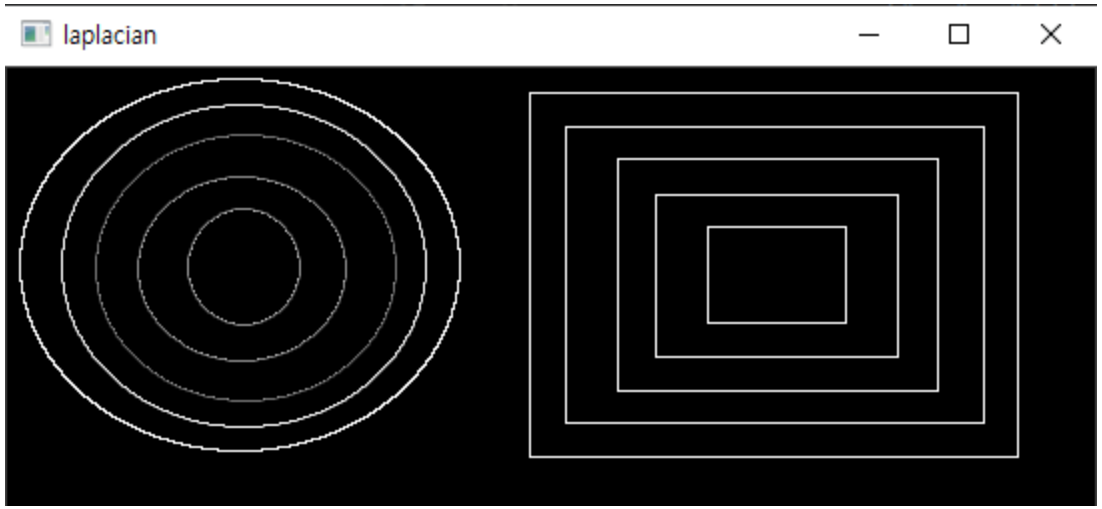
- 영상의 x축, y축에 각각 이차 미분을 수행하여 합한 수식을 행렬 형태로 만든 필터

- 가우시안 필터와 비슷한 모양의 필터

1	1	1
1	−8	1
1	1	1

가로, 세로, 양 대각선 방향으로
모두 미분연산을 수행한 필터

Laplacian Filter



모든 방향의 뚜렷한 edge 검출 가능

과제

1. 주어진 이미지를 Gray Scale로 변환
2. 이미지 Blurring 처리
3. Canny Edge로 Edge 검출
4. ROI로 차선 부분만 추출

형식	보고서 형식(한글, 워드) 또는 PPT 형식
내용	각 단계의 파이썬 코드 각 단계의 결과 화면 캡처

실습 정답 파일 비밀번호 필요하신 분은 말씀해주세요!

감사합니다