

Canny Edge Detection

1986년 J.Canny, "A Computational Approach to Edge Detection" 논문에서 발표

⇒ Good detection, Good localization, Single edge

1. Grayscale 변환

```
# 2. 그레이스케일 변환
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

컬러 이미지는 RGB의 3채널을 가져 연산이 복잡하므로

2. 노이즈 제거 -> Gaussian Blur

```
# 3. 가우시안 블러 적용
blur = cv2.GaussianBlur(gray, (5,5), 1.5)
```

3. Gradient 계산 -> Sobel 필터

Sobel 필터?

- 3X3 커널의 1차 미분 이용
- 중앙 값에 더 큰 가중치 부여 : -2 0 2

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

X축 방향 필터

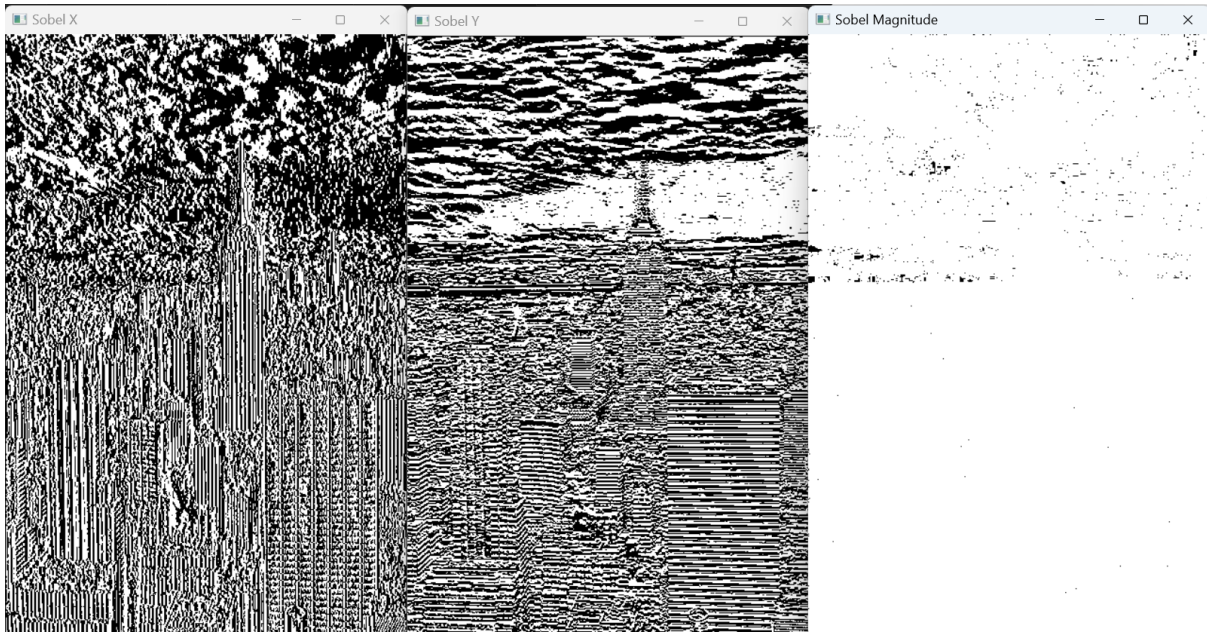
- 수직 엣지 강조
- 좌우 밝기 차이가 클 때 강한 엣지로 인식

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

Y축 방향 필터

- 수평 엣지 강조
- 위아래 밝기 차이가 클 때 강한 엣지로 인식

- Gradient 계산 -> $G = \sqrt{G_x^2 + G_y^2}$
- Magnitude 값은 주변값과 큰 차이가 없어 엣지가 잘 보이지 않음



4. 강한 엣지 추출 -> Non-Maximum Suppression(최대비억제)

Gradient 방향을 기준으로 최대값이 아닌 픽셀을 제거해(0으로 만들) 엣지를 얇게 만들

Gradient 방향은 0~180°이므로 4가지 방향으로 근사: 수평(0°), 수직(90°), 대각선(45°, 135°)

5. 강한/약한 엣지 구분 -> Hysteresis Thresholding(이중 임계값)

```
# 4. Canny Edge Detection 실행 (이중 임계값 적용)
edges = cv2.Canny(blur, 100, 200)
```

100: 최소 임계값으로 이보다 더 작은 gradient는 엣지가 아님

200: 최대 임계값으로 이보다 큰 gradient는 확실히 엣지

⇒ 100~200사이 값은 주변에 강한 엣지가 있을 경우에만 유지

