



SINCE 1947  
**SEO KYEONG**  
UNIVERSITY

# 로봇 스터디

## -Assignment 02-

컴퓨터공학과 2019314009 김시현

2022.02.21

## 1. 각 필터의 특징

3 개의 방법 모두 에지 검출 방법입니다. 미분을 이용합니다. (에지 edge: 영역의 경계)

(1) - 로버츠 공식

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

공식이 간단해 계산 속도가 빠르다

(2) - 프리윗 공식

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * \mathbf{A}$$

소벨보다 간단하여 속도가 빠르지만 로버츠보다 더 좋은 결과가 나온다.

(3) - 소벨 공식

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

제일 복잡한 만큼 매우 뚜렷한 엣지가 검출된다.

## 2. 파이썬 코드

```
import cv2
import numpy as np

def mouse_callback(event, x, y, flags, param):
    if event == 1:
        print('B: ', param[y][x][0], '\nG: ', param[y][x][1], '\nR: ', param[y][x][2])
        print('=====')

Path = 'Data/'
Name = 'lenna.tif'
src = Path + Name

#로버츠, 프리윗, 소벨은 에지 검출이기 때문에 grayscale 로 불러와줌
color_img = cv2.imread(src)
img = cv2.imread(src, cv2.IMREAD_GRAYSCALE)

#필터값
roberts_x = np.array([[ -1,  0,  0], [ 0,  1,  0], [ 0,  0,  0]])
roberts_y = np.array([[ 0,  0, -1], [ 0,  1,  0], [ 0,  0,  0]])

prewitt_x = np.array([[ -1, -1, -1], [ 0,  0,  0], [ 1,  1,  1]])
prewitt_y = np.array([[ 1,  0, -1], [ 1,  0, -1], [ 1,  0, -1]])

sobel_x = np.array([[ -1, -2, -1], [ 0,  0,  0], [ 1,  2,  1]])
sobel_y = np.array([[ 1,  0, -1], [ 2,  0, -2], [ 1,  0, -1]])

#공식에 맞게 계산&&정수화
roberts_x = cv2.convertScaleAbs(cv2.filter2D(img, -1, roberts_x))
roberts_y = cv2.convertScaleAbs(cv2.filter2D(img, -1, roberts_y))

prewitt_x = cv2.convertScaleAbs(cv2.filter2D(img, -1, prewitt_x))
prewitt_y = cv2.convertScaleAbs(cv2.filter2D(img, -1, prewitt_y))

sobel_x = cv2.convertScaleAbs(cv2.filter2D(img, -1, sobel_x))
sobel_y = cv2.convertScaleAbs(cv2.filter2D(img, -1, sobel_y))

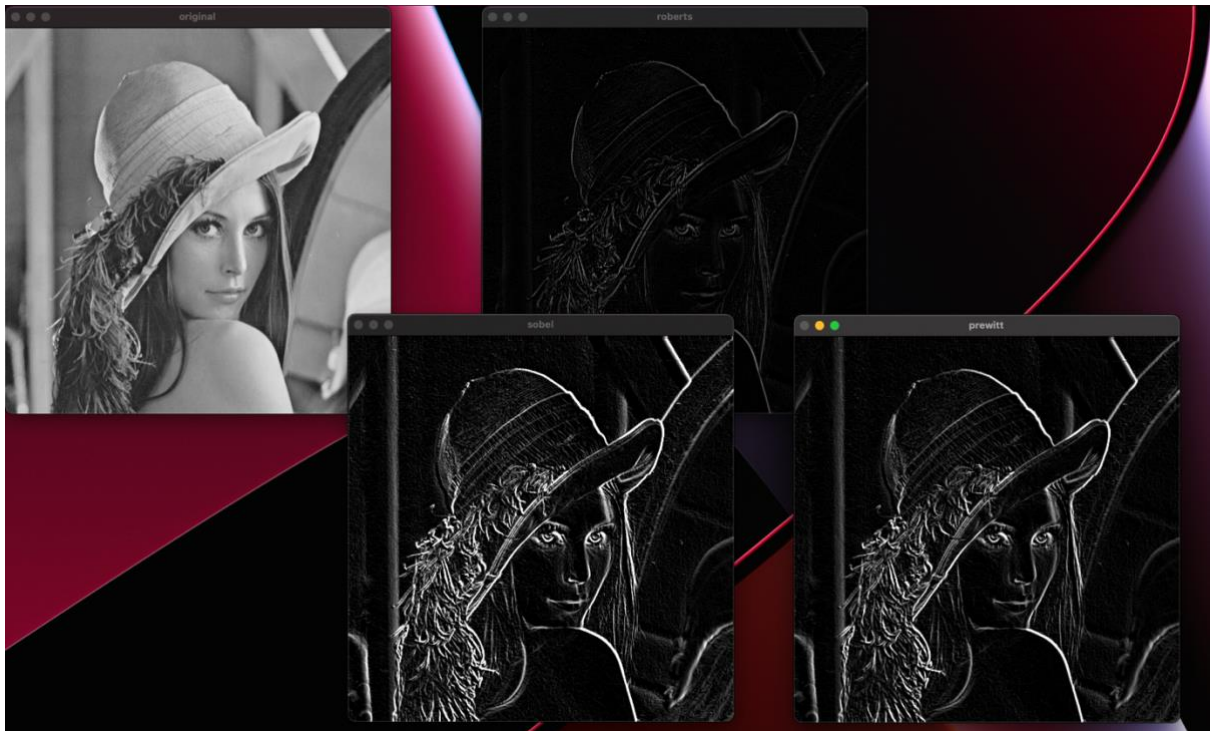
#x 와 y 값을 하나로 합쳐줌
prewitt = cv2.addWeighted(prewitt_x, 1, prewitt_y, 1, 0)
roberts = cv2.addWeighted(roberts_x, 1, roberts_y, 1, 0)
sobel = cv2.addWeighted(sobel_x, 1, sobel_y, 1, 0)

#이미지 저장
cv2.imwrite('./Data/prewitt.png', prewitt)
cv2.imwrite('./Data/roberts.png', roberts)
cv2.imwrite('./Data/sobel.png', sobel)
```

```
#imshow
cv2.imshow('original', img)
cv2.imshow('prewitt', prewitt)
cv2.imshow('roberts', roberts)
cv2.imshow('sobel', sobel)

while cv2.waitKey(33) <= 0:
    cv2.setMouseCallback('img', mouse_callback, img)
```

### 3. 결과 화면&사진



<로버츠>



<프리윗>



<소벨>