Image Processing

실습 6.

2021. 04. 11.

실습 수업 소개

- **과목 홈페이지**
  - 충남대학교 사이버 캠퍼스 (http://e-learn.cnu.ac.kr)

- **TA 연락처**
  - 신준호
  - wnsgh578@naver.com

- **튜터 연락처**
  - 한승오
  - so.h4ns@gmail.com

- **실습 중 질문사항**
  - 실시간 수업중 질문 or 메일을 통한 질문
  - 메일로 질문할 때 [IP] 를 제목에 붙여주세요

실습 수업 소개

- **실습 출석**
  - 사이버캠퍼스를 통해 Zoom 출석
  - Zoom 퇴장 전 채팅 기록[학번 이름] 남기고 퇴장
  - 위 두 기록을 통해 출석 체크 진행 예정

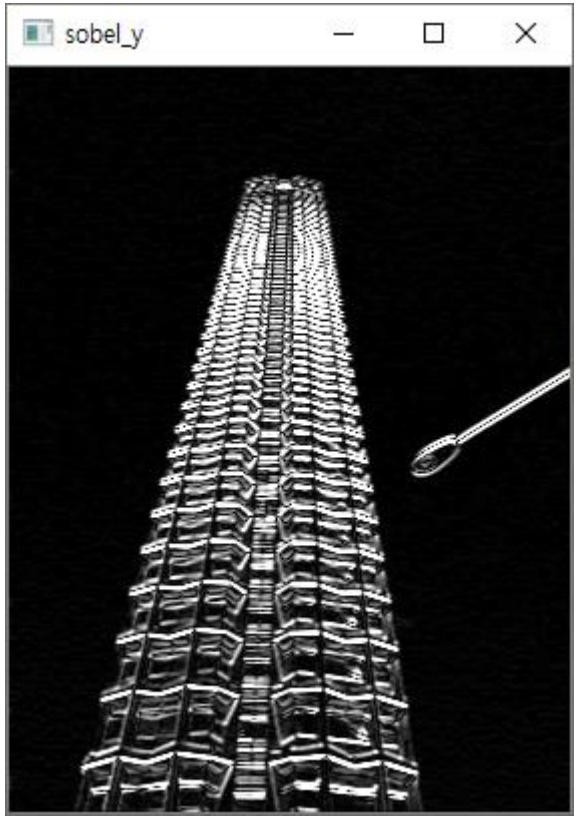
4월 실습수업 관련 공지

- **실습 진행**
  - 4월까지는 비대면 실시간 수업으로 진행
  - 기존처럼 실습 수업시간(월요일 13:00-15:00)에 zoom을 통한 수업 진행
  - 비대면으로 수업 관련 질문을 위한 office time

목 차

Edge detection

- **edge detection filter**
  - Sobel filter



Edge detection

- **edge detection filter**
  - Sobel filter

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

sobel filter_x
vertical

=

1
2
1

blurring

*

-1	0	1

1D derivative
filter (x-direction)

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

sobel filter_y
horizontal

=

-1
0
1

1D derivative
filter (y-direction)

*

1	2	1

blurring

Edge detection

- **sobel filter 실습**

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

sobel filter_x
vertical

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

sobel filter_y
horizontal

```
sobel_x
[[-1  0  1]
 [-2  0  2]
 [-1  0  1]]
sobel_y
[[-1 -2 -1]
 [ 0  0  0]
 [ 1  2  1]]
```

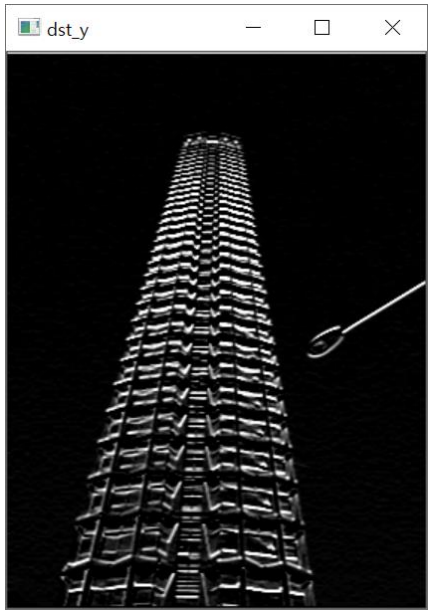
```python
import numpy as np

def get_sobel():
    derivative = np.array([[-1, 0, 1]])
    blur = np.array([[1], [2], [1]])

    x = np.dot(blur, derivative)
    y = np.dot(derivative.T, blur.T)

    return x, y

def main():
    sobel_x, sobel_y = get_sobel()
    print('sobel_x')
    print(sobel_x)

    print('sobel_y')
    print(sobel_y)

if __name__== '__main__':
    main()
```

Edge detection

- sobel filter 실습

```python
def main():
    sobel_x, sobel_y = get_sobel()

    src = cv2.imread('../imgs/sobel_test.png', cv2.IMREAD_GRAYSCALE)
    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')

    dst_x = np.clip(dst_x, 0, 255).astype(np.uint8)
    dst_y = np.clip(dst_y, 0, 255).astype(np.uint8)

    cv2.imshow('dst_x', dst_x)
    cv2.imshow('dst_y', dst_y)
    cv2.waitKey()
    cv2.destroyAllWindows()
```

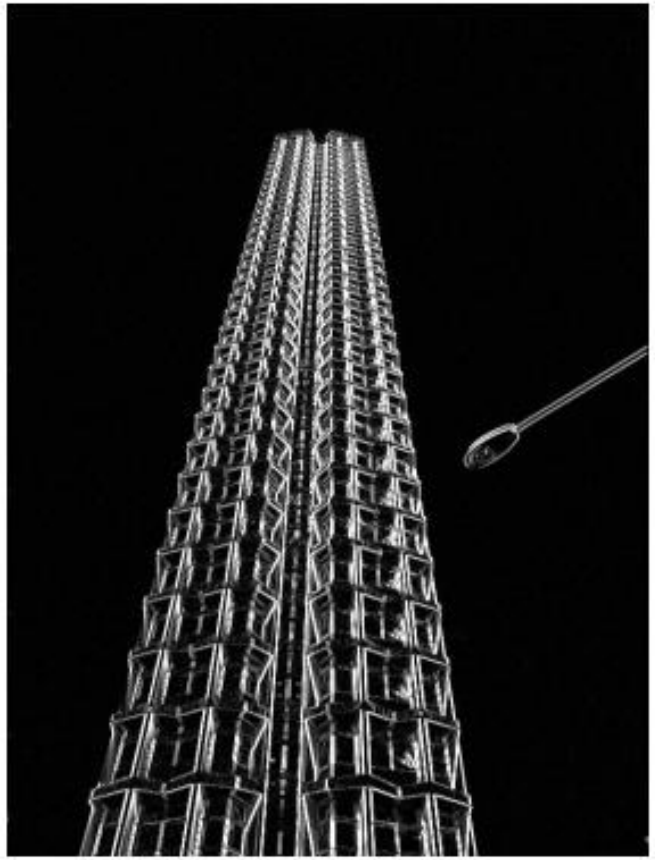
```python
import cv2
import numpy as np

# library add
import os
import sys
sys.path.append(os.path.dirname(os.path.abspath(os.path.dirname(__file__))))
from my_library.filtering import my_filtering

def get_sobel():
    derivative = np.array([[-1, 0, 1]])
    blur = np.array([[1], [2], [1]])

    x = np.dot(blur, derivative)
    y = np.dot(derivative.T, blur.T)
```



Edge detection

- **sobel filter 실습**

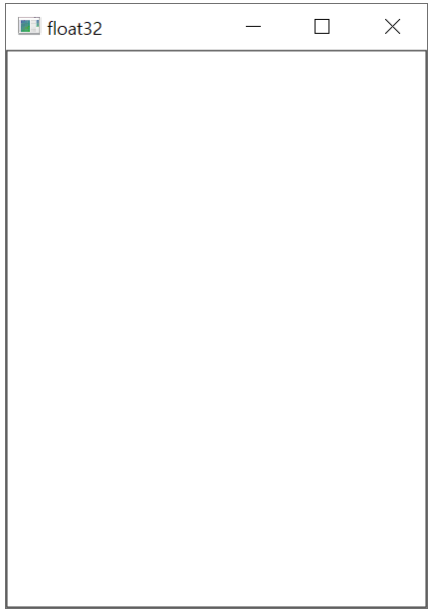
실습 결과



교수님 pdf 결과

Edge detection

- sobel filter 실습



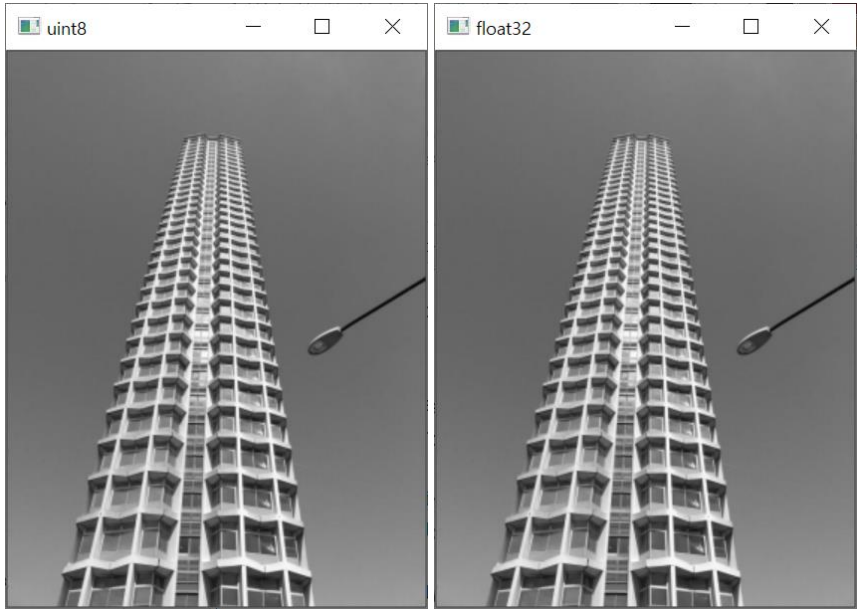

```python
import cv2
import numpy as np

def main():
    src = cv2.imread('../imgs/sobel_test.png', cv2.IMREAD_GRAYSCALE)
    src_float = src.astype(np.float32)

    cv2.imshow('uint8', src)
    cv2.imshow('float32', src_float)
    cv2.waitKey()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

Edge detection

- **sobel filter 실습**
  - Data type
    - uint8 : 0 ~ 255(0 : 검, 255 : 흰)
    - float : 0 ~ 1(0 : 검, 1 : 흰)

Edge detection

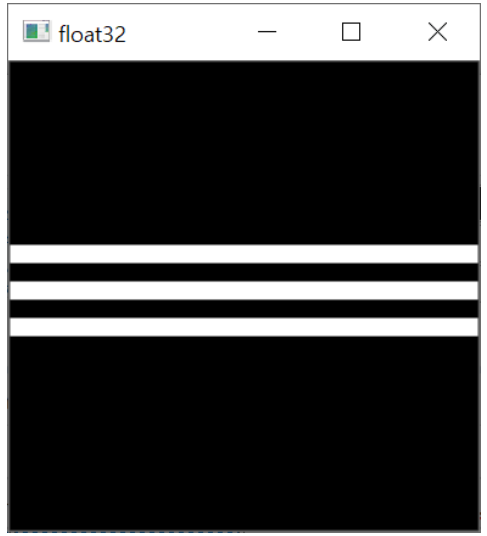
- **sobel filter 실습**
  - Data type
    - uint8 : 0 ~ 255(0 : 검, 255 : 흰)
    - float : 0 ~ 1(0 : 검, 1 : 흰)




```python
import cv2
import numpy as np

def main():
    src = cv2.imread('../imgs/sobel_test.png', cv2.IMREAD_GRAYSCALE)
    src_float = src.astype(np.float32)

    cv2.imshow('uint8', src)
    cv2.imshow('float32', src_float/255)
    cv2.waitKey()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

Edge detection

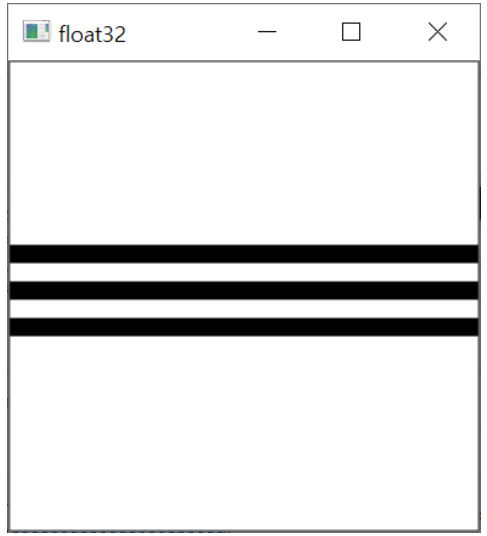
- **sobel filter 실습**
  - Data type
    - uint8 : 0 ~ 255(0 : 검, 255 : 흰)
    - float : 0 ~ 1(0 : 검, 1 : 흰)
      - 1이상은 전부 1과 같다.




```python
import cv2
import numpy as np

def main():
    src = np.zeros((256, 256))
    src[100:110, :] = 1
    src[120:130, :] = 2
    src[140:150, :] = 255

    cv2.imshow('float32', src)
    cv2.waitKey()
    cv2.destroyAllWindows()


if __name__ == '__main__':
    main()
```

Edge detection

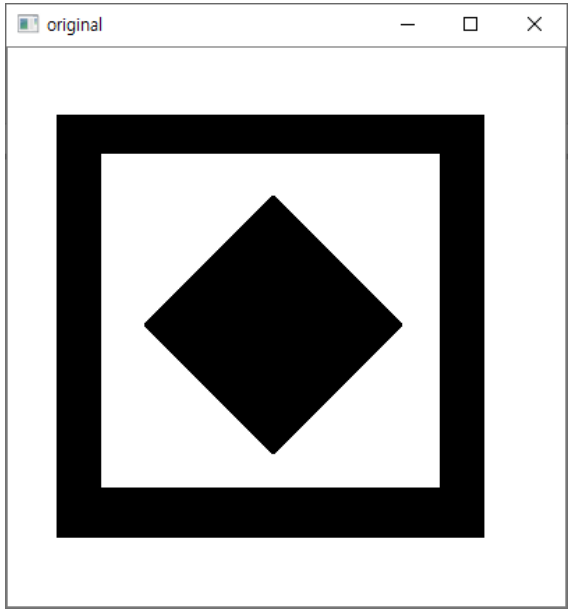
- **sobel filter 실습**
  - Data type
    - uint8 : 0 ~ 255(0 : 검, 255 : 흰)
    - float : 0 ~ 1(0 : 검, 1 : 흰)
      - 1이상은 전부 1과 같다.
      - 0 이하도 전부 0과 같다.




```python
import cv2
import numpy as np

def main():
    src = np.ones((256, 256))
    src[100:110, :] = 0
    src[120:130, :] = -1
    src[140:150, :] = -255

    cv2.imshow('float32', src)
    cv2.waitKey()
    cv2.destroyAllWindows()


if __name__ == '__main__':
    main()
```

Edge detection

- **sobel filter 실습**

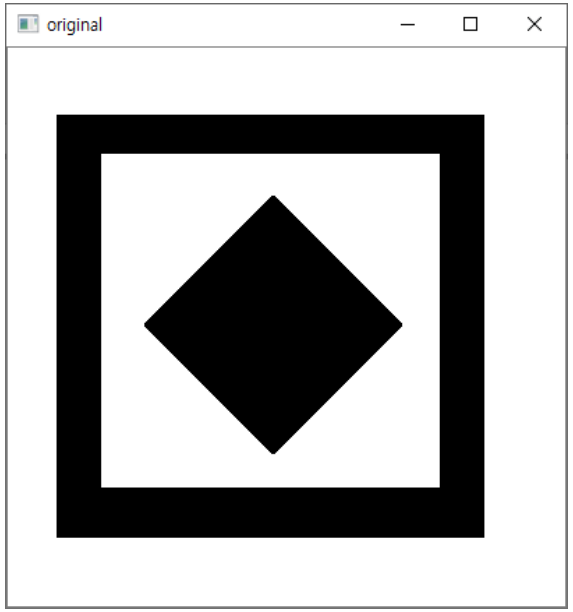


src

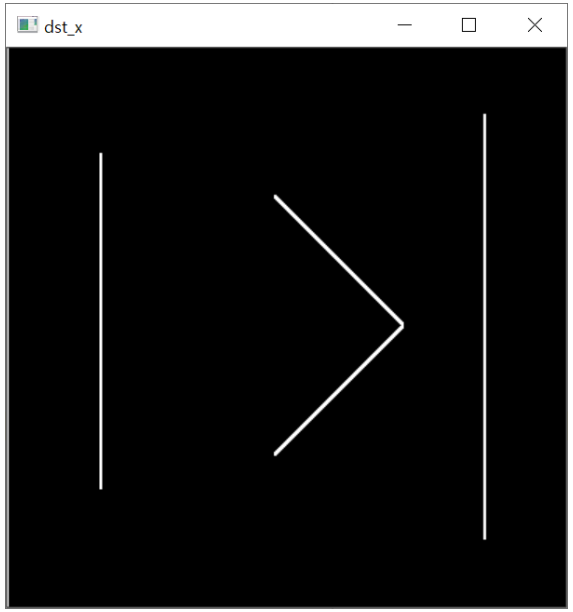
```
def main():
    sobel_x, sobel_y = get_sobel()

    src = cv2.imread('../imgs/edge_detection_img.png', cv2.IMREAD_GRAYSCALE)
    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')

    cv2.imshow('dst_x', dst_x)
    cv2.imshow('dst_y', dst_y)
    cv2.waitKey()
    cv2.destroyAllWindows()
```

Edge detection

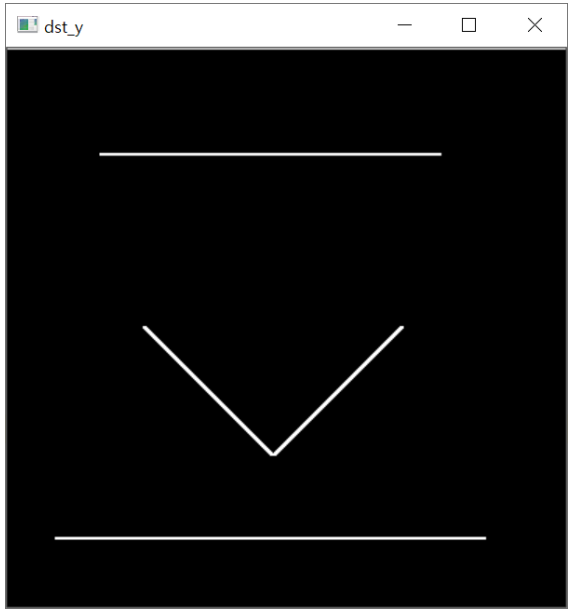
- **sobel filter 실습**



src



sobel_x



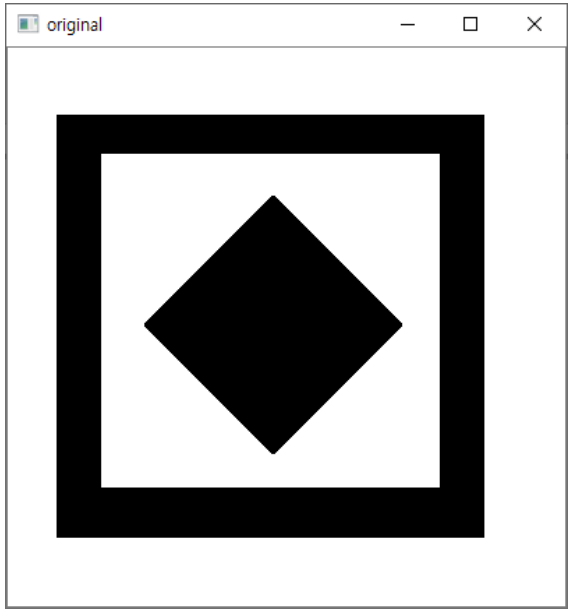
sobel_y

Edge detection

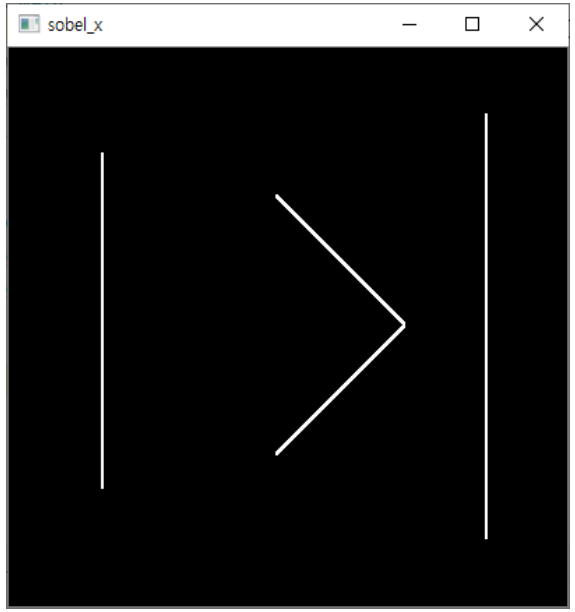
- **sobel filter 실습**

```
sobX = cv2.Sobel(src, cv2.CV_64F, 1, 0, ksize=3)
sobY = cv2.Sobel(src, cv2.CV_64F, 0, 1, ksize=3)
cv2.imshow('sobX', sobX/255)
cv2.imshow('sobY', sobY/255)
```

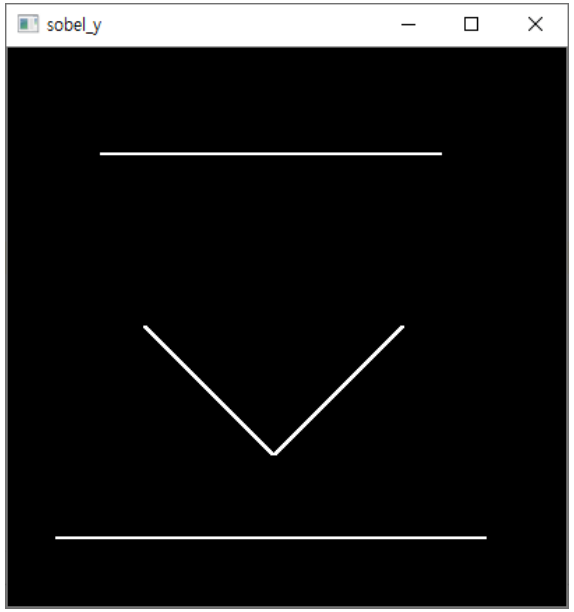
내장함수를 사용해도 같은 결과가 나옴



src



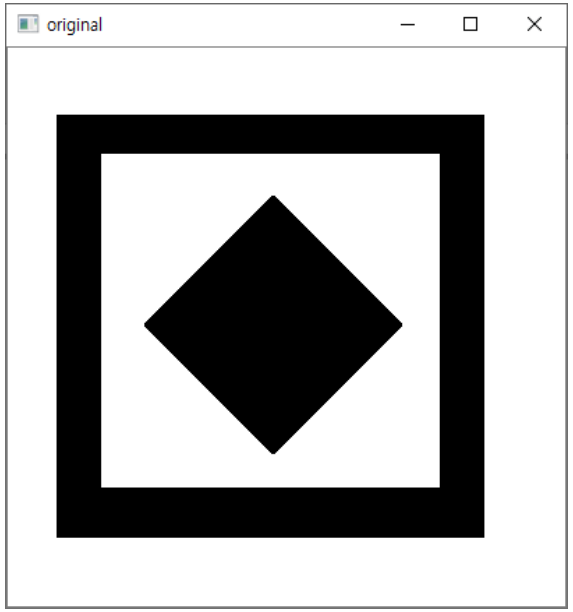
sobel_x



sobel_y

Edge detection

- **sobel filter 실습**

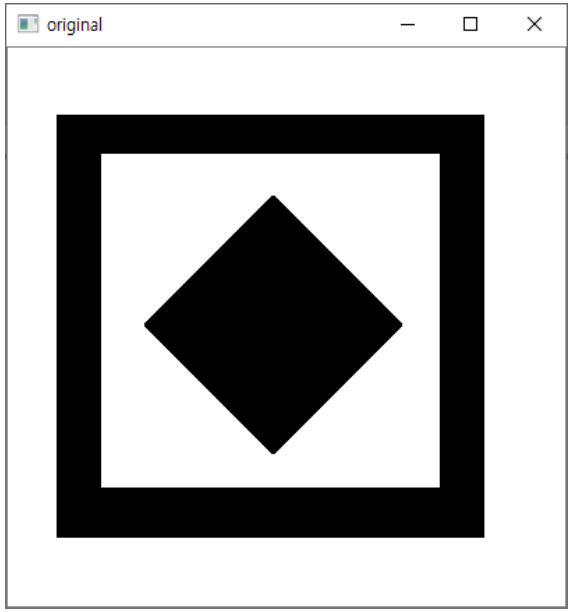


src

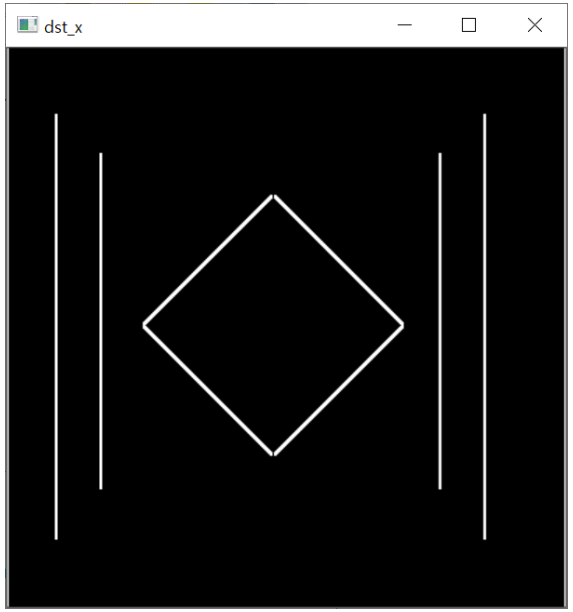
```
def main():
    sobel_x, sobel_y = get_sobel()

    src = cv2.imread('../imgs/edge_detection_img.png', cv2.IMREAD_GRAYSCALE)
    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')

    #dst_x = np.abs(dst_x)
    #dst_y = np.abs(dst_y)
    dst_x = np.sqrt(dst_x**2)
    dst_y = np.sqrt(dst_y**2)

    cv2.imshow('dst_x', dst_x)
    cv2.imshow('dst_y', dst_y)
    cv2.waitKey()
    cv2.destroyAllWindows()
```

Edge detection

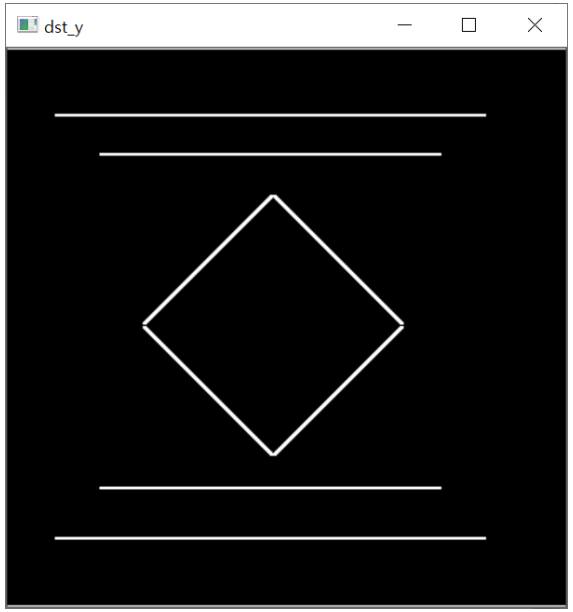
- **sobel filter 실습**



src



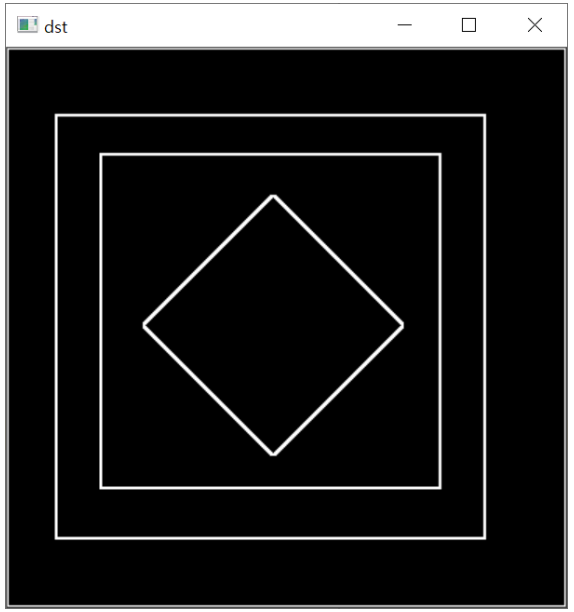
sobel_x



sobel_y

Edge detection

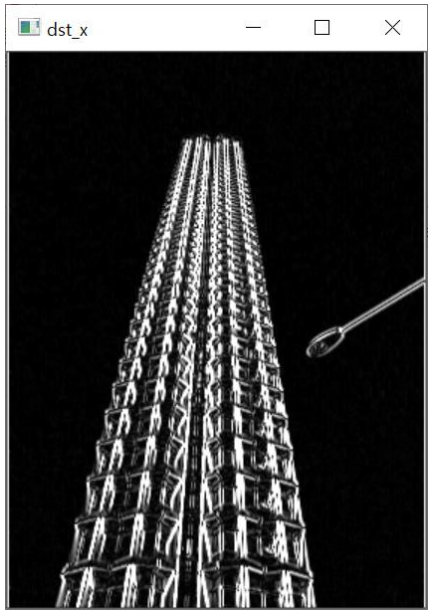
- sobel filter 실습

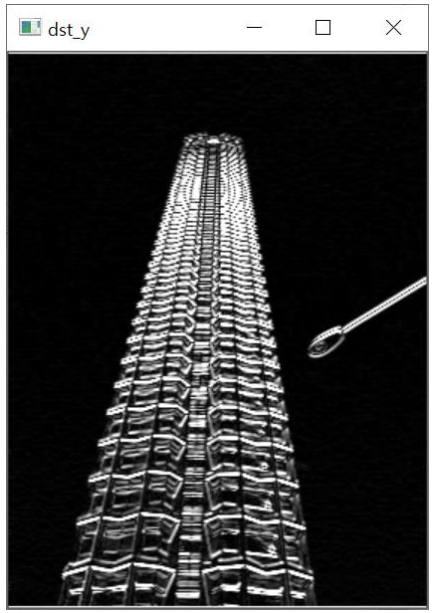



```
def main():
    sobel_x, sobel_y = get_sobel()

    src = cv2.imread('edge_detection_img.png', cv2.IMREAD_GRAYSCALE)
    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')

    dst_x = np.abs(dst_x)
    dst_y = np.abs(dst_y)
    dst = np.sqrt(dst_x**2 + dst_y**2)
    cv2.imshow('dst',dst)
    cv2.waitKey()
    cv2.destroyAllWindows()
```

Edge detection

- sobel filter 실습

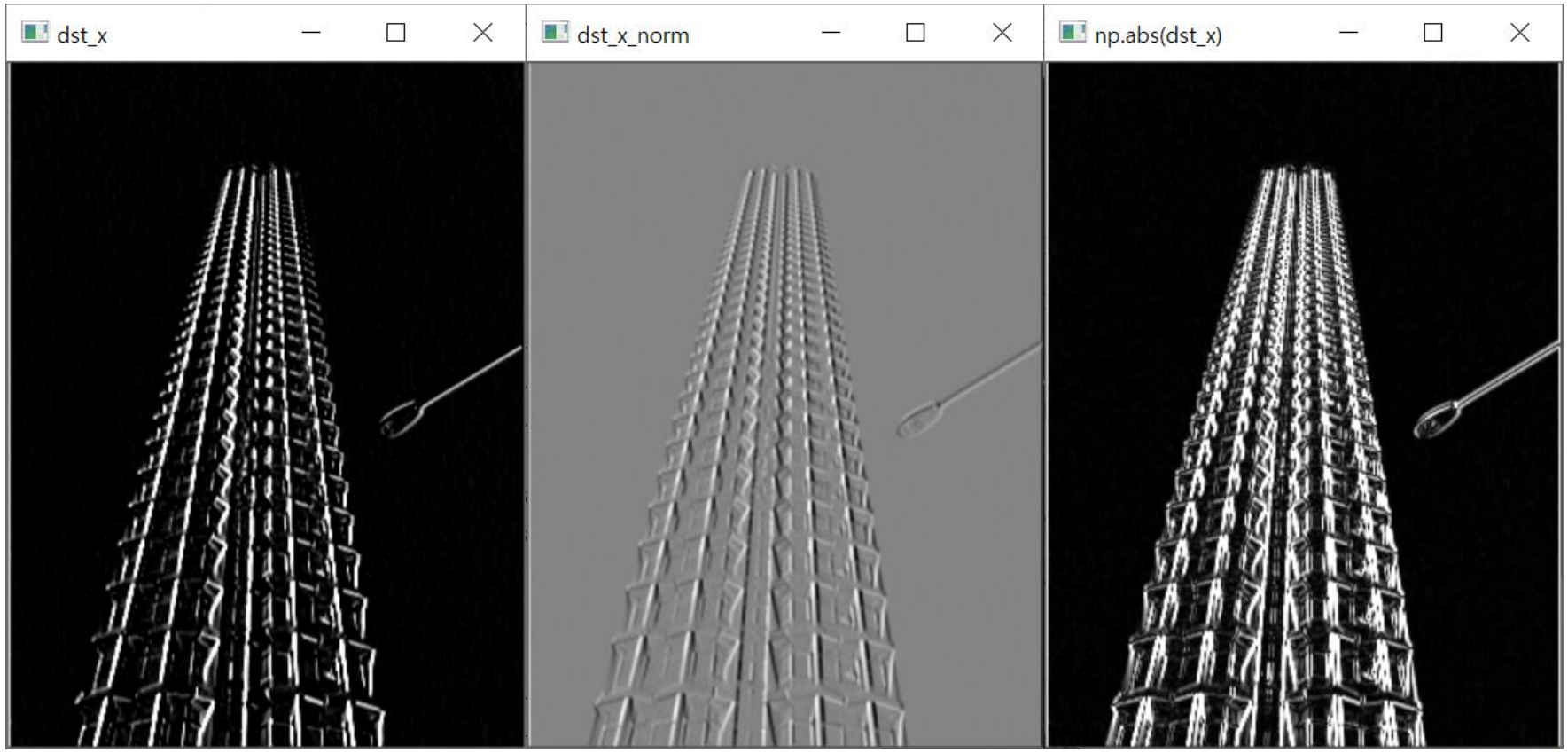




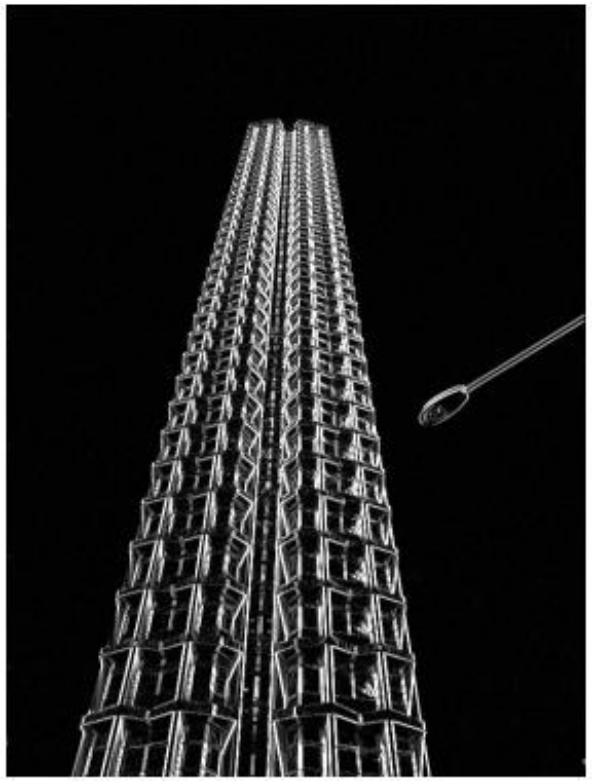

```
def main():
    sobel_x, sobel_y = get_sobel()

    src = cv2.imread('../imgs/sobel_test.png', cv2.IMREAD_GRAYSCALE)
    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')

    dst_x = np.abs(dst_x)
    dst_y = np.abs(dst_y)
    cv2.imshow('dst_x', dst_x/255)
    cv2.imshow('dst_y', dst_y/255)
    cv2.waitKey()
    cv2.destroyAllWindows()
```

Edge detection

- **sobel filter 실습**



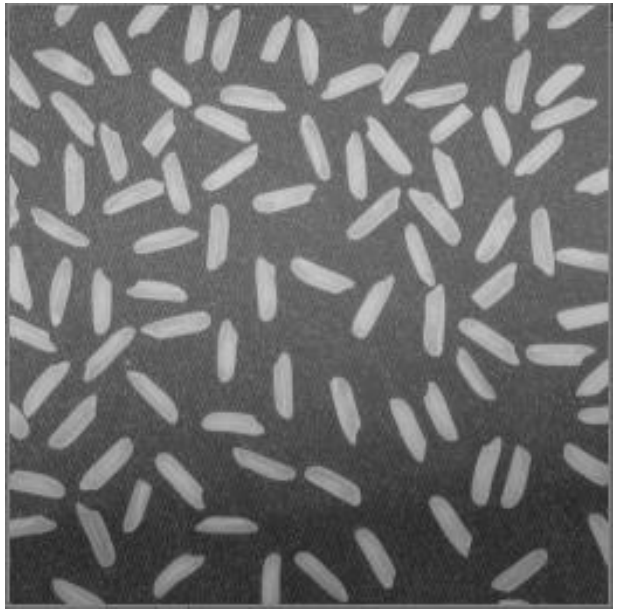
실습 결과



교수님 pdf 결과

threshold

- **threshold**
  - 영상을 흑/백으로 분류하여 처리하는 것을 이진화 라고 한다.
  - 이때 기준이 되는 임계값을 threshold value라고 한다

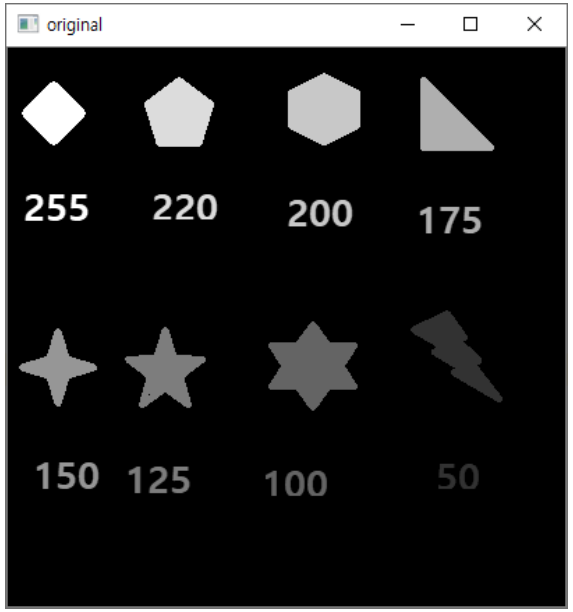


original

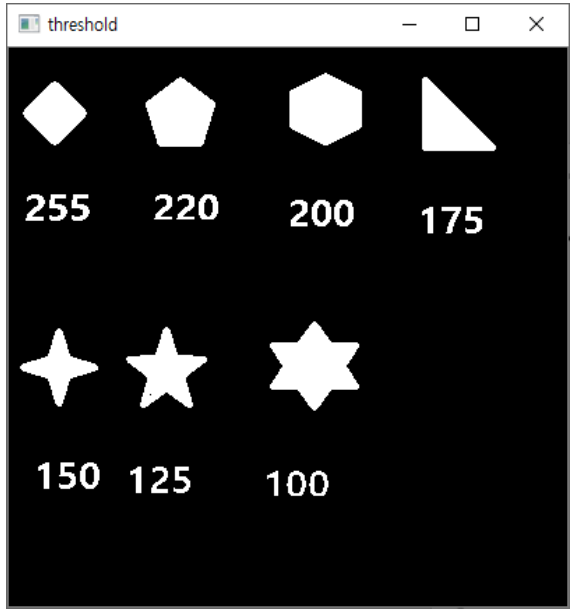
thresholding
threshold value = 131

threshold

- **threshold 실습**



original



thresholding
threshold value = 85

threshold

- **threshold 실습**

cv2.threshold(*src, thresh, maxval, type*) → retval, dst

Parameters:
- **src** – input image로 single-channel 이미지.(grayscale 이미지)
- **thresh** – 임계값
- **maxval** – 임계값을 넘었을 때 적용할 value
- **type** – thresholding type

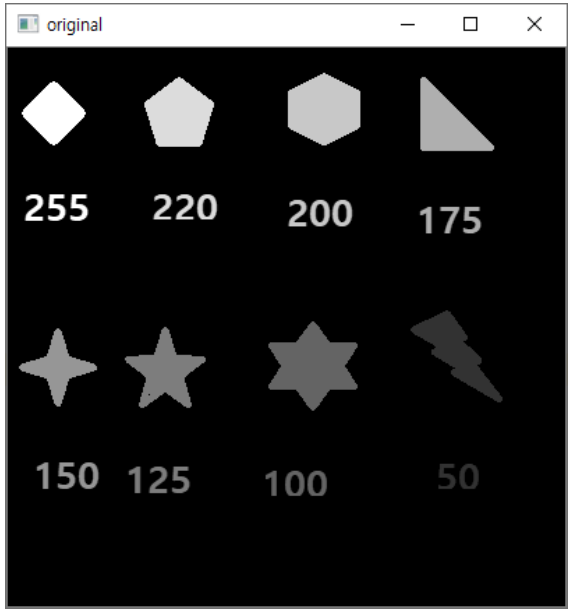
thresholding type은 아래와 같습니다.
- cv2.THRESH_BINARY
- cv2.THRESH_BINARY_INV
- cv2.THRESH_TRUNC
- cv2.THRESH_TOZERO
- cv2.THRESH_TOZERO_INV

```
C:\Users\cvlab\Pyc
ret :  150.0
```

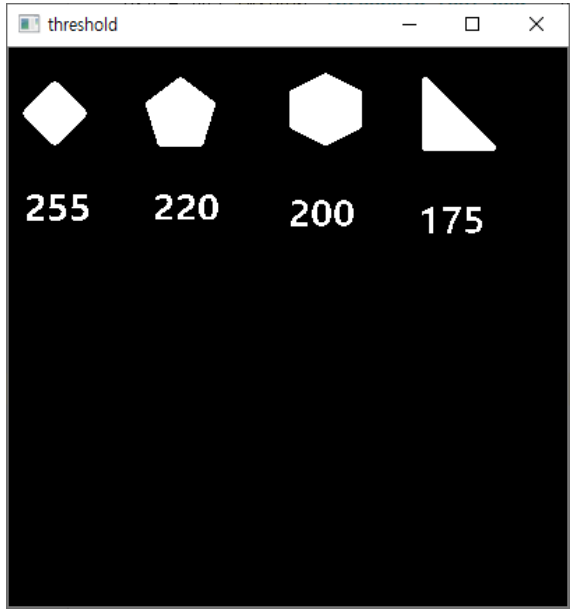
```python
import cv2
import numpy as np

def main():
    src = cv2.imread('../imgs/threshold_test.png', cv2.IMREAD_GRAYSCALE)
    ret, dst = cv2.threshold(src, 150, 255, cv2.THRESH_BINARY)

    print('ret : ', ret)
    cv2.imshow('original', src)
    cv2.imshow('threshold_test', dst)
    cv2.waitKey()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

threshold

- **threshold 실습**



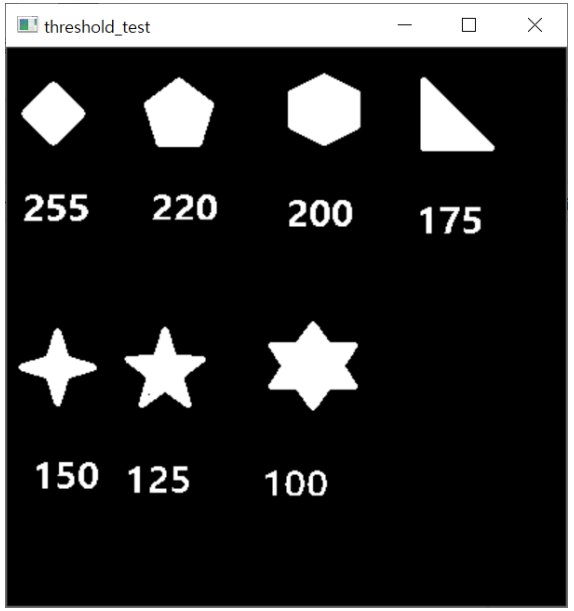
original



thresholding
threshold value = 150

threshold

- **threshold 실습**



thresholding
threshold value = 85

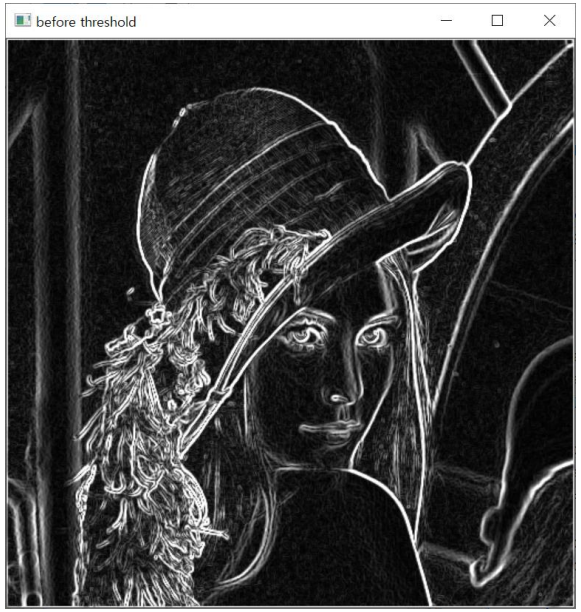
```python
def main():
    src = cv2.imread('../imgs/threshold_test.png', cv2.IMREAD_GRAYSCALE)
    #ret, dst = cv2.threshold(src, 150, 255, cv2.THRESH_BINARY)
    ret, dst = cv2.threshold(src, 0, 255, cv2.THRESH_OTSU)

    print('ret : ', ret)
    cv2.imshow('original', src)
    cv2.imshow('threshold_test', dst)
    cv2.waitKey()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

자동으로 threshold value를 정해줌

```
C:\Users\cvlab\Pycha
ret :  85.0
```

Edge detection

- sobel filter 실습

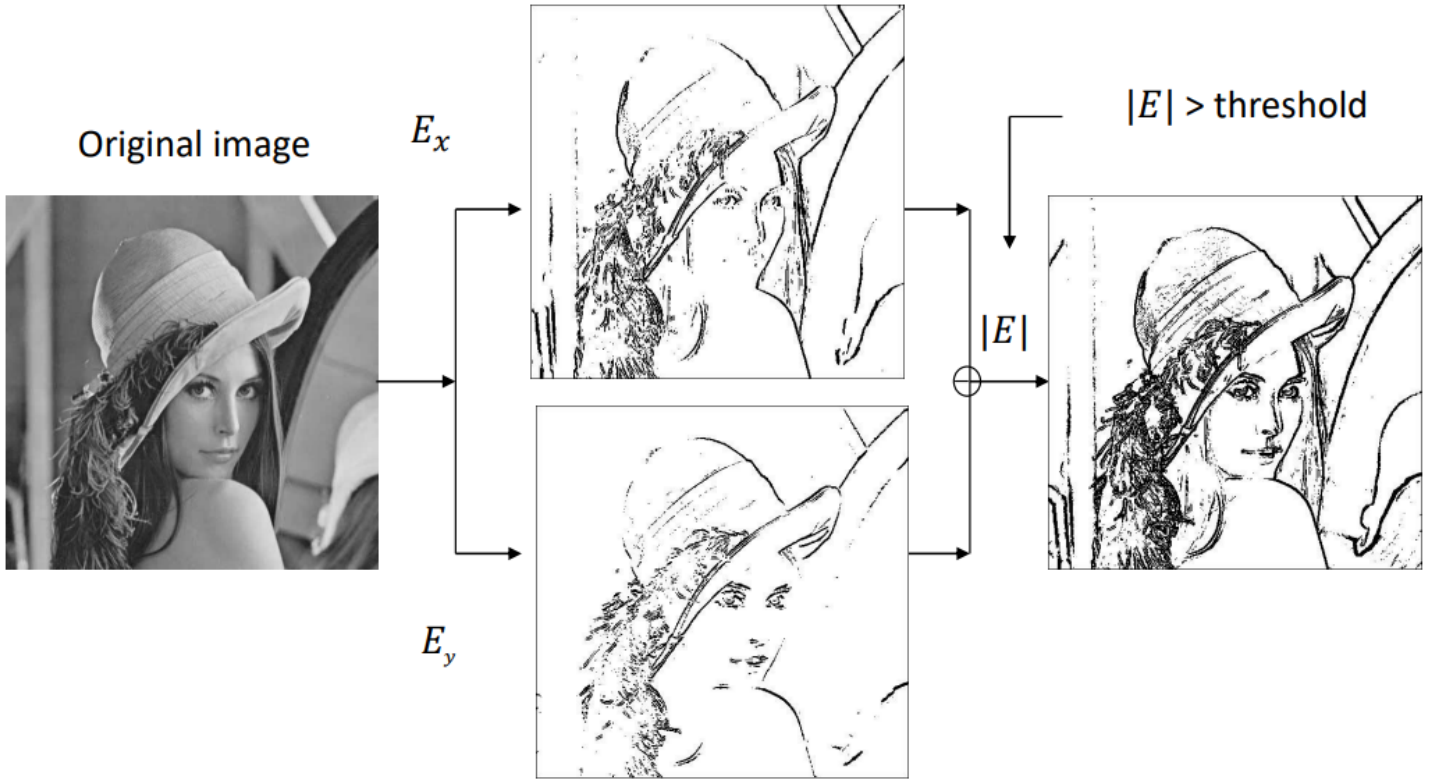



```python
import cv2
import numpy as np

# library add
import os
import sys
sys.path.append(os.path.dirname(os.path.abspath(os.path.dirname(__file__))))
from my_library.filtering import my_filtering

def get_sobel():
    derivative = np.array([[-1, 0, 1]])
    blur = np.array([[1], [2], [1]])

    x = np.dot(blur, derivative)
    y = np.dot(derivative.T, blur.T)

    return x, y

def main():
    src = cv2.imread('../imgs/Lena.png', cv2.IMREAD_GRAYSCALE)
    sobel_x, sobel_y = get_sobel()

    dst_x = my_filtering(src, sobel_x, 'zero')
    dst_y = my_filtering(src, sobel_y, 'zero')
    dst = np.abs(dst_x) + np.abs(dst_y)
    ret, dst_threshold = cv2.threshold(dst, 100, 255, cv2.THRESH_BINARY)

    print('ret : ', ret)
    cv2.imshow('before threshold', dst/255)
    cv2.imshow('after threshold', dst_threshold/255)
    cv2.waitKey()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

Edge detection

- **sobel filter 실습**

sobel
threshold value = 100



Result of Sobel operator (threshold = 100)

출처 : 교수님 이론 ppt

Derivative of Gaussian(DoG)

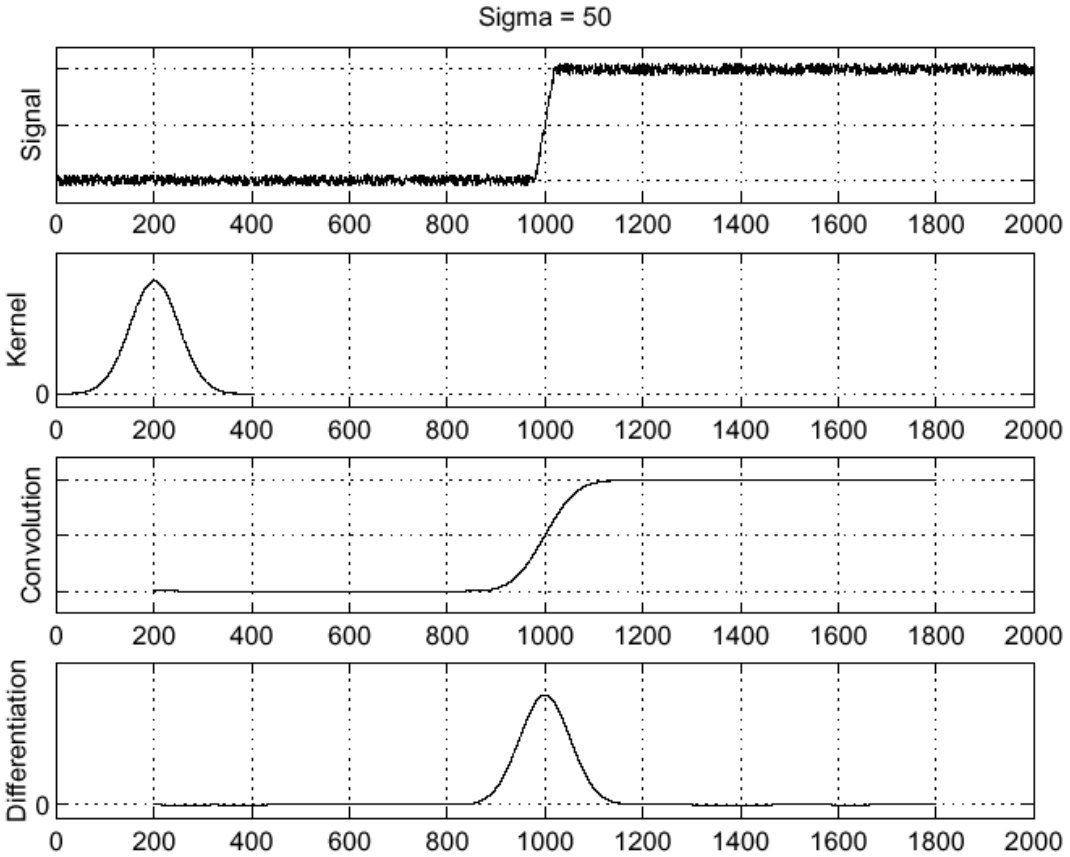
- **DoG**
  - Gaussian 필터로 noise를 감소시키고 미분 필터를 적용

I

$G(x) = e^{-\frac{x^2}{2\sigma^2}}$

$G * I$

$\frac{d}{dx}(G * I)$



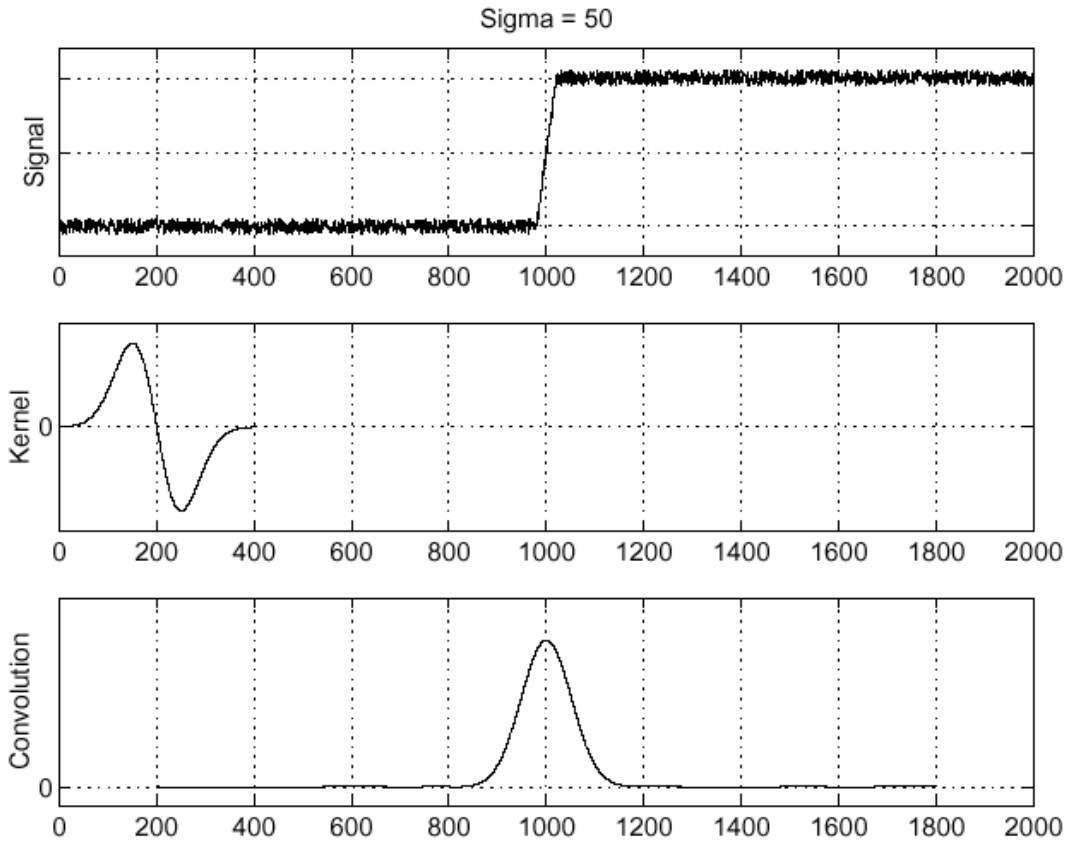
Derivative of Gaussian(DoG)

- **DoG**
  - Gaussian식을 먼저 미분하고 컨볼루션 연산을 하면 절차가 간소화된다

I

$\frac{d}{dx}G$

$\left(\frac{d}{dx}G\right) * I$



Derivative of Gaussian(DoG)

- **DoG 수행 절차** ($\frac{1}{2\pi\sigma^2}$는 없어도 됨)

1. 가우시안 분포 함수 : $G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ (원본 이미지 : I)

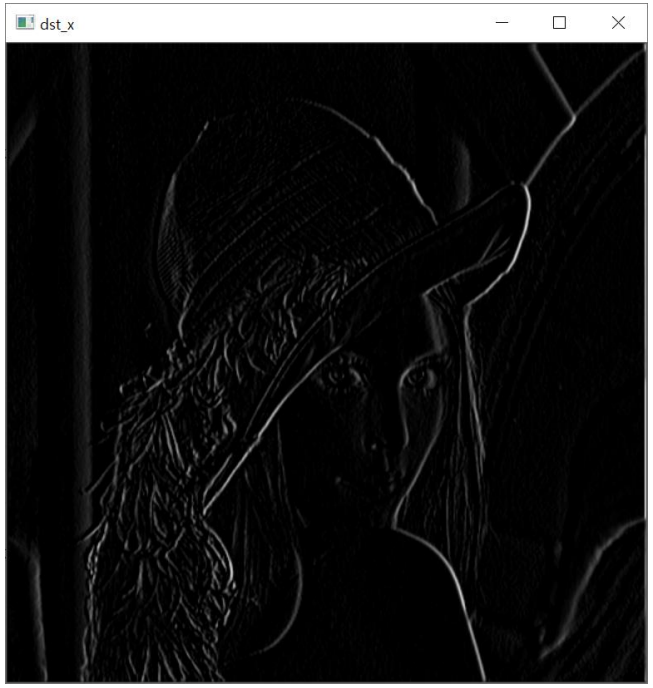
2. $\frac{d}{dx}\mathrm{G}(x,y) = -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, $\frac{d}{dy}\mathrm{G}(x,y) = -\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

3. $DoG(x) = \frac{d}{dx}\mathrm{G}(x,y) * I$

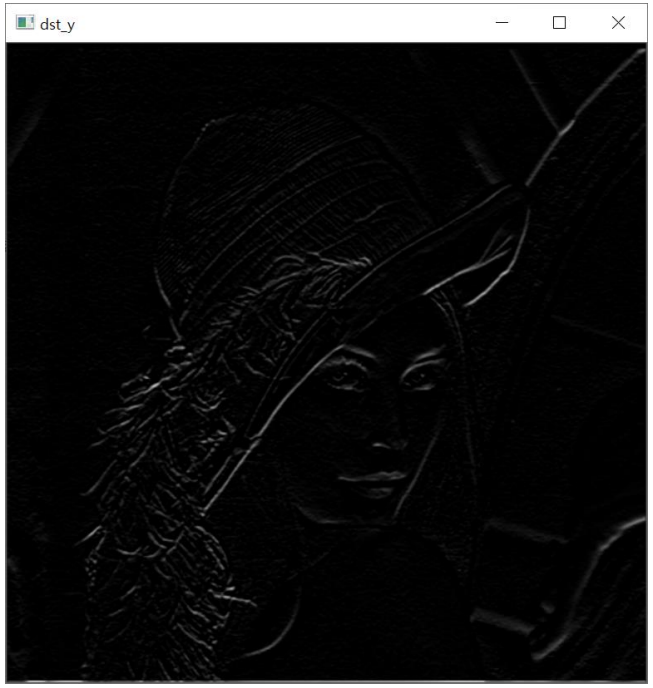
4. $DoG(x), DoG(y)$의 magnitude를 계산

과제

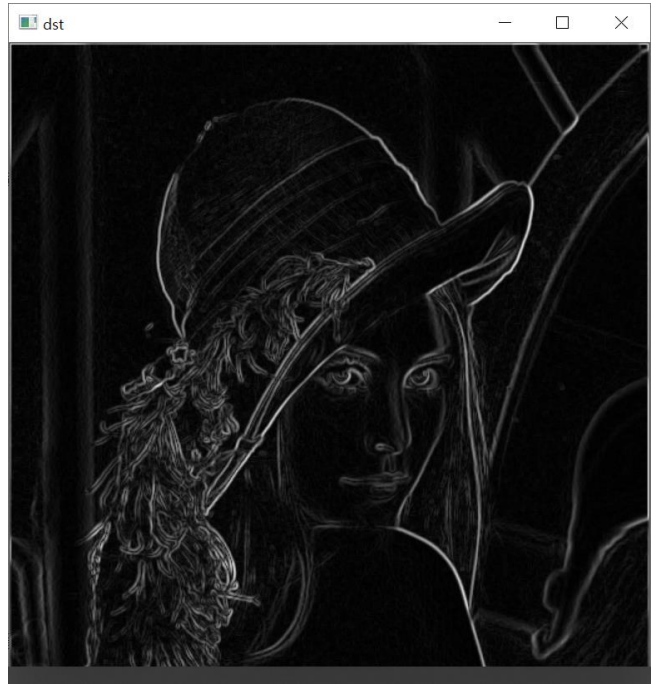
- **Derivative of Gaussian(DoG)**



DoG(x)



DoG(y)



DoG(x, y) - magnitude

과제

- **Derivative of Gaussian(DoG)**

$$\frac{d}{dx}\mathrm{G}(x,y) = -\frac{x}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{d}{dy}\mathrm{G}(x,y) = -\frac{y}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Magnitude $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$

```python
def get_DoG_filter(fsize, sigma=1):
    ########################################################
    # TODO                                                 #
    # DoG mask 완성                                         #
    ########################################################

    DoG_x = ???
    DoG_y = ???

    return DoG_x, DoG_y

########################################################
# TODO                                                 #
# DoG mask sigma값 조절해서 mask 만들기                    #
########################################################
# DoG_x, DoG_y filter 확인
x, y = get_DoG_filter(fsize=256, sigma=?)
x = ???
y = ???

dst_x = my_filtering(src, DoG_x, 'zero')
dst_y = my_filtering(src, DoG_y, 'zero')

########################################################
# TODO                                                 #
# dst_x, dst_y 를 사용하여 magnitude 계산                 #
########################################################
dst = ???
```

과제

- **my_DoG.py**
  - get_DoG_filter(fsize,sigma=1)
    - fsize:
    - sigma:
    - return DoG_x, DoG_y
  - main()
    - get_DoG_filter 함수를 통해서 mask 생성
    - dst_x, dst_y를 사용하여 magnitude를 계산
  - 보고서에 결과 5개의 이미지(DoG_x filter, DoG_y filter, dst_x, dst_y, dst) 모두 포함하여 작성

과제

- **제출 방법**
  - 코드 파일
    - 구현 결과가 포함된 python 파일(.py)
  - 보고서
    - [IP]201900000_홍길동_2주차_과제.pdf
    - 보고서 양식 사용
    - PDF 파일 형식으로 제출(pdf가 아닌 다른 양식으로 제출시 감점)
  - 제출 파일
    - [IP]201900000_홍길동_2주차_과제.zip
    - .py 파일과 pdf 보고서를 하나의 파일로 압축한 후, 양식에 맞는 이름으로 제출

출석체크

- Zoom 퇴장 전, [학번 이름]을 채팅창에 올린 후 퇴장해 주시기 바랍니다.

QnA