

Image Processing

실습 13.

2022. 05. 30.

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

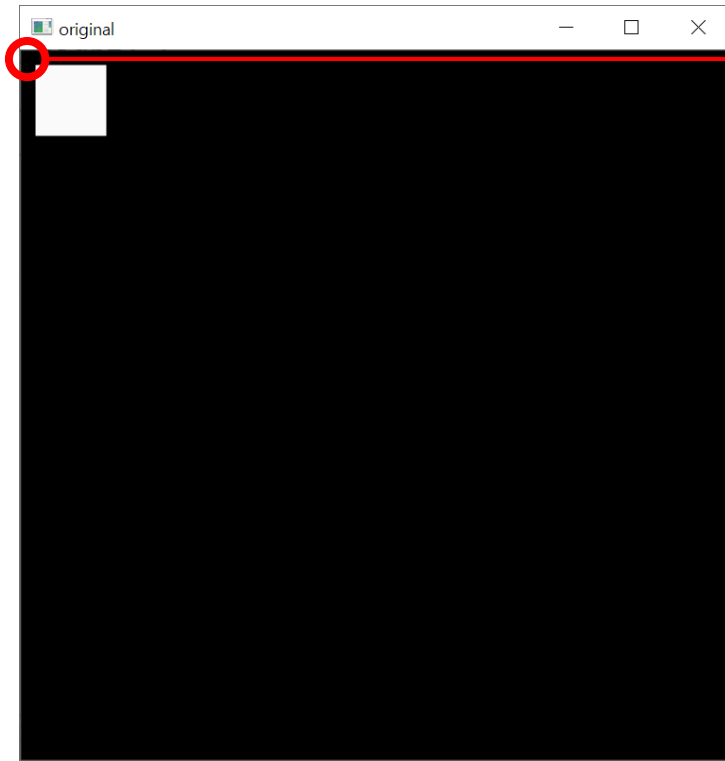
실습

- 실습
 - Translation
 - Scaling
 - Rotation
 - Shear
- 과제
 - Backward

실습

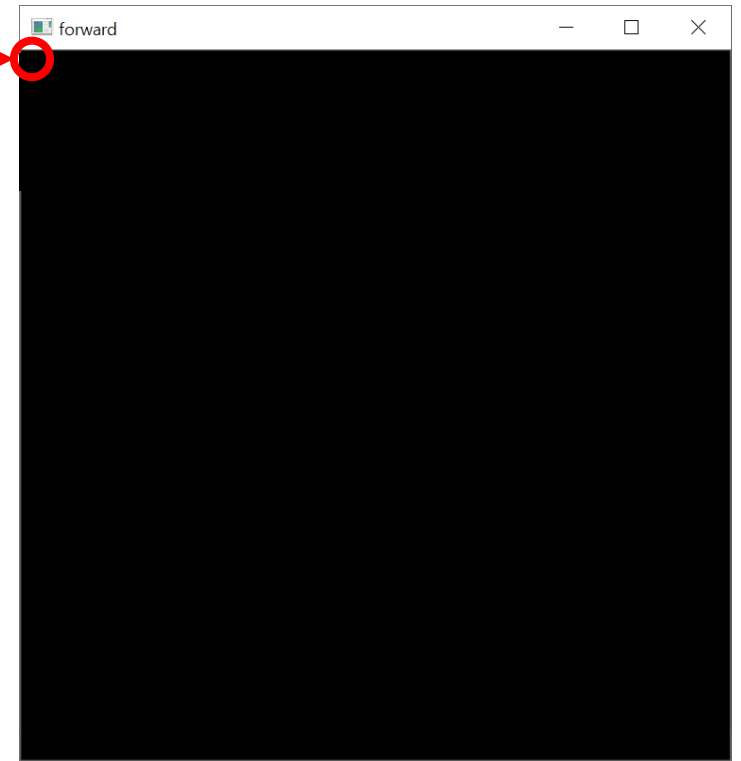
- Forward transform (x2 Scaling)
 - 원본 좌표에서 M을 적용 -> dst에서 대응되는 좌표 계산

(0, 0)



M

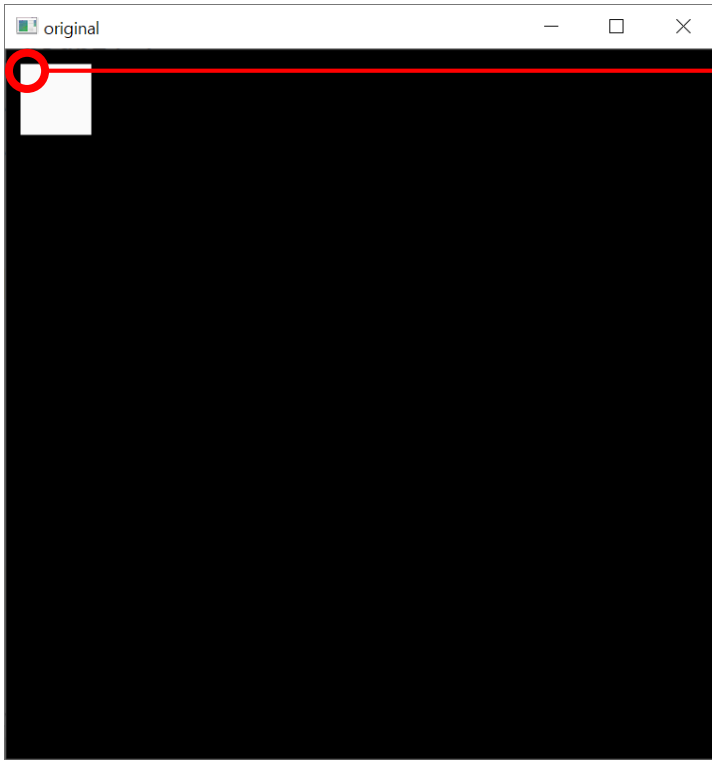
(0, 0)



실습

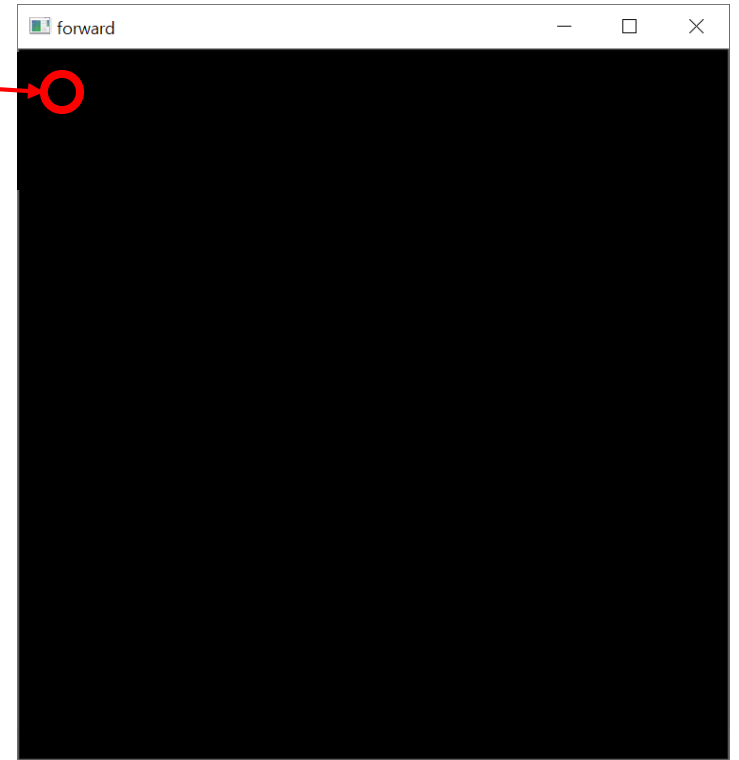
- Forward transform (x2 Scaling)
 - 원본 좌표에서 M을 적용 -> dst에서 대응되는 좌표 계산

(1, 1)



M

(2, 2)

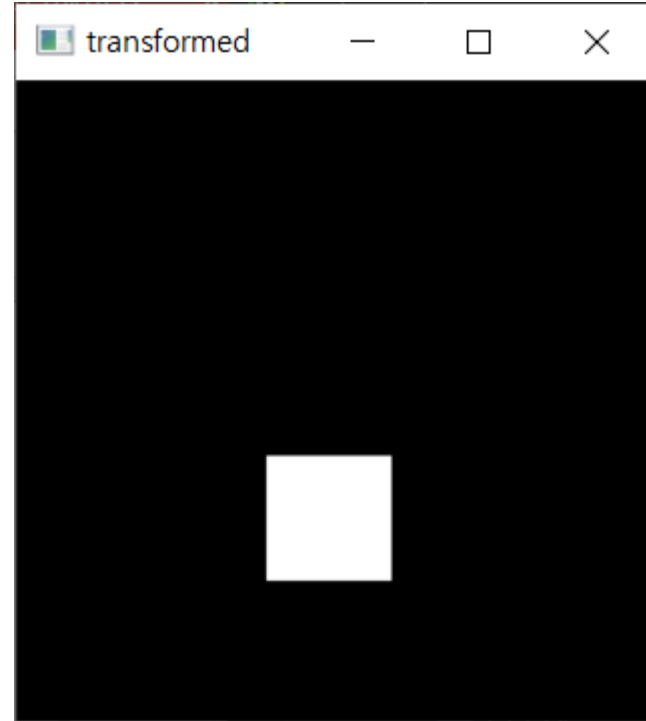
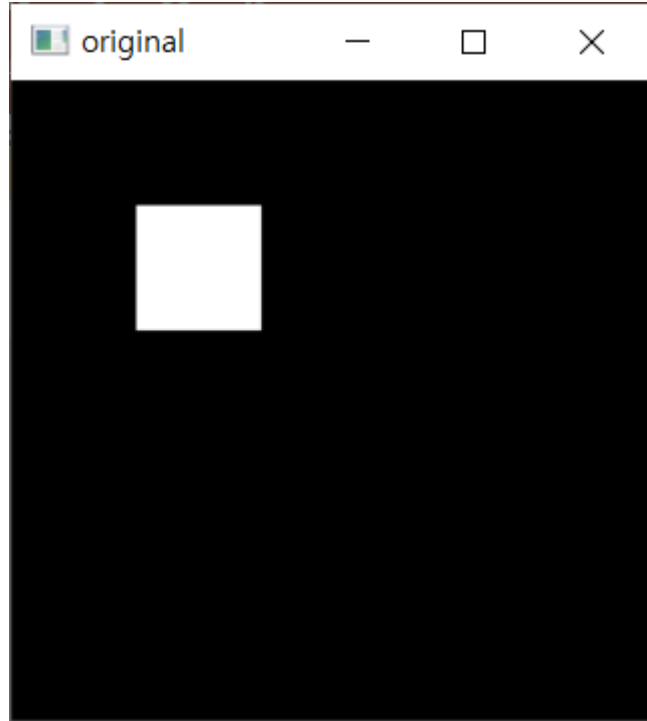
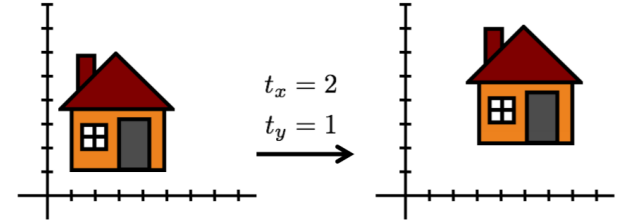


실습

- Translation(+50,+100)

$$M = \begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix}$$



실습

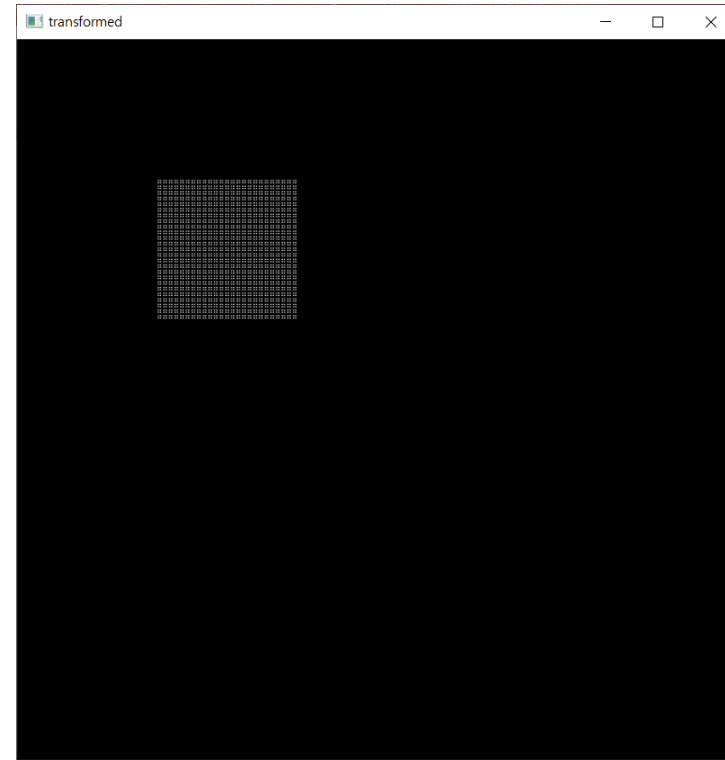
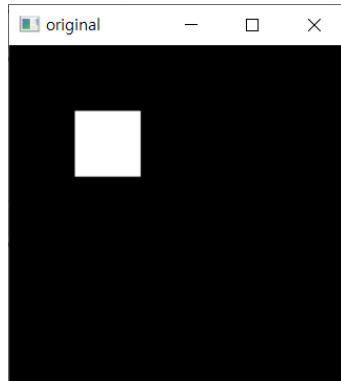
- Scaling(x2.5)

M

$$\begin{bmatrix} 2.5 & 0. & 0. \\ 0. & 2.5 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

Scale

$$M = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$



실습

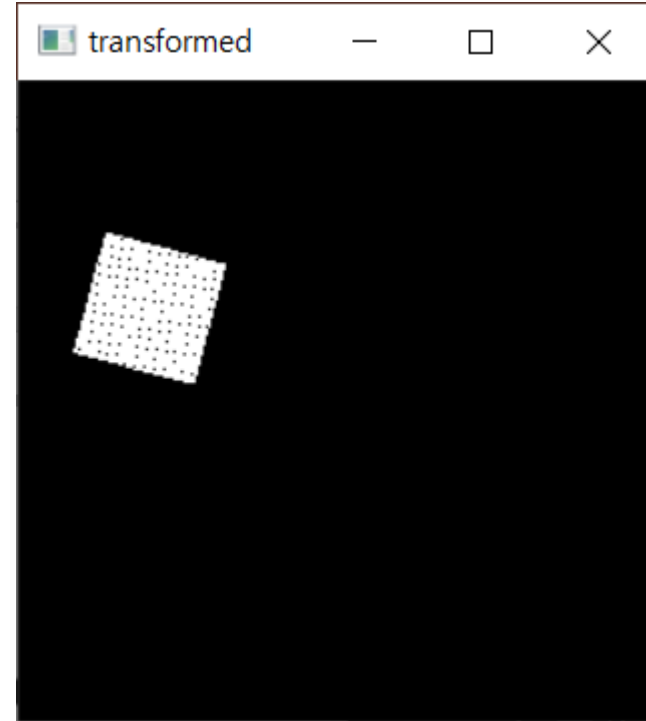
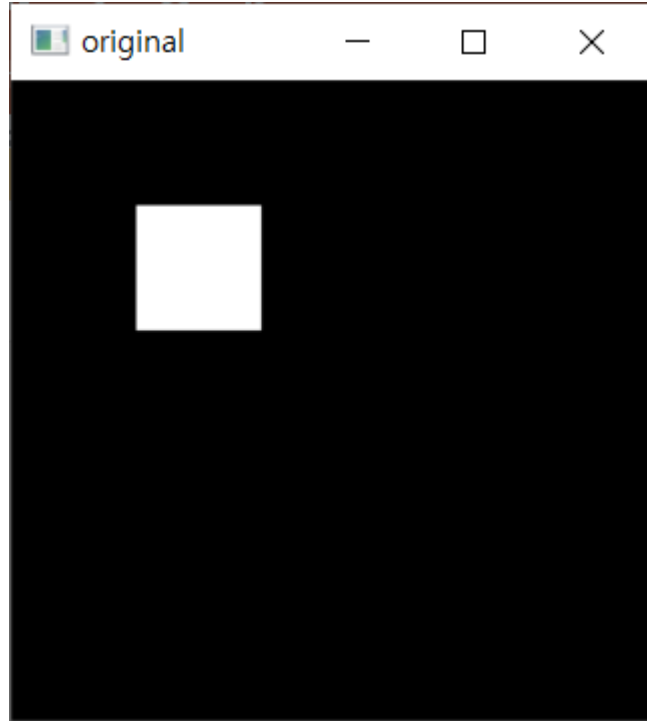
- Rotation(15도)

M

```
[[ 0.96592583 -0.25881905  0.
   0.25881905  0.96592583  0.
   0.          0.          1.]]
```

Rotate

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$



실습

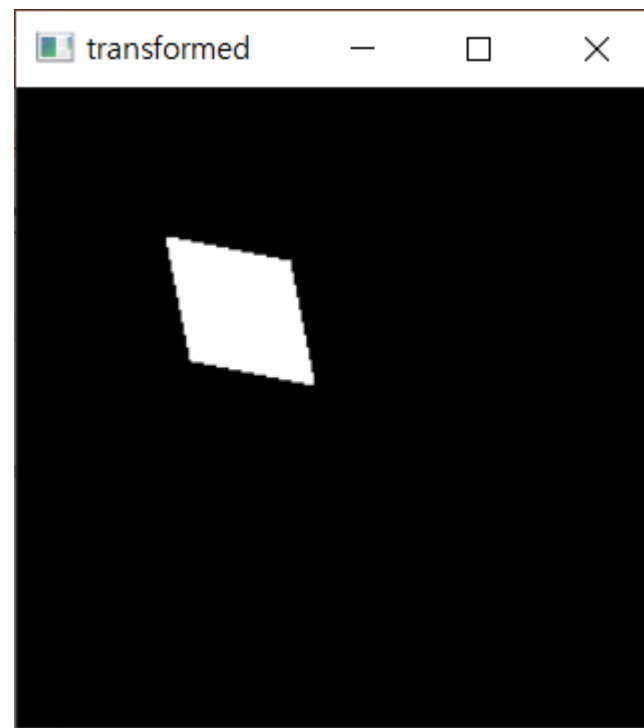
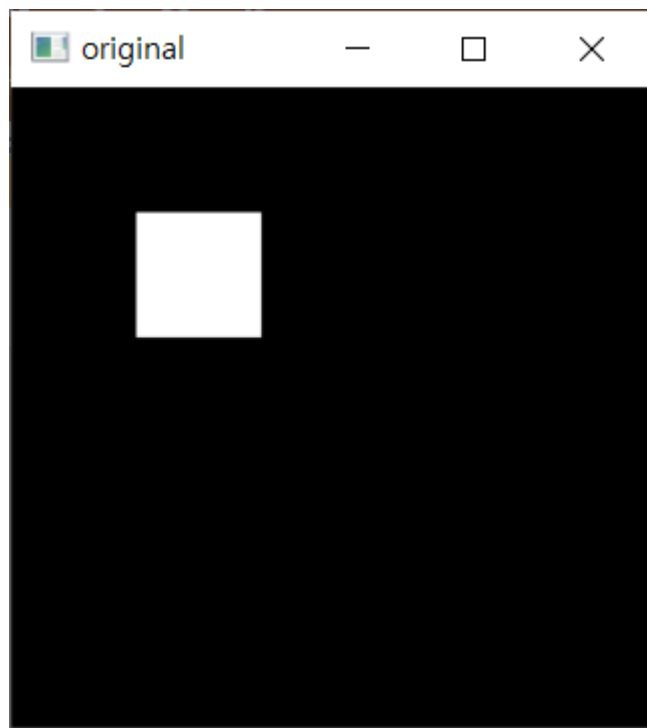
- Shear(0.2, 0.2)

M

```
[[1.  0.2 0. ]  
 [0.2 1.  0. ]  
 [0.  0.  1. ]]
```

Shear

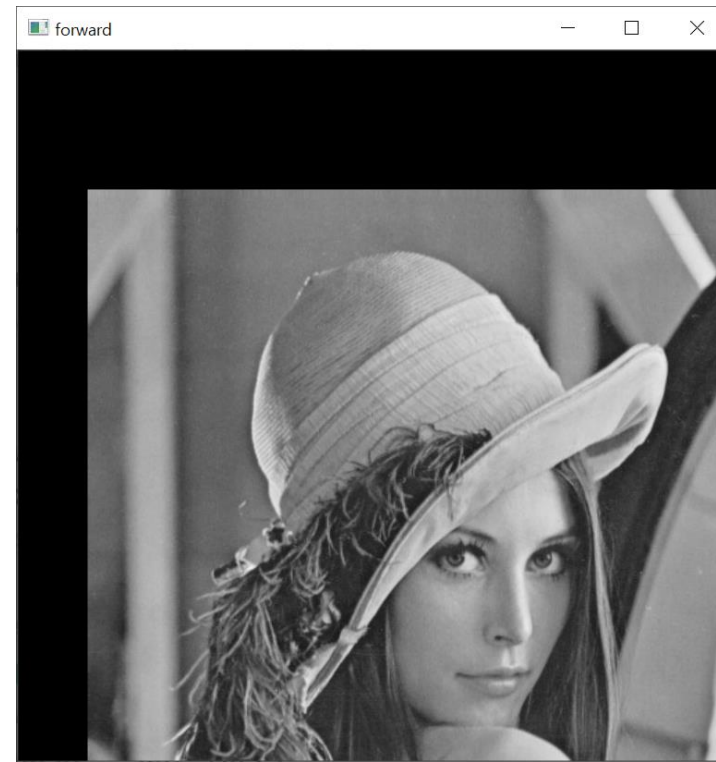
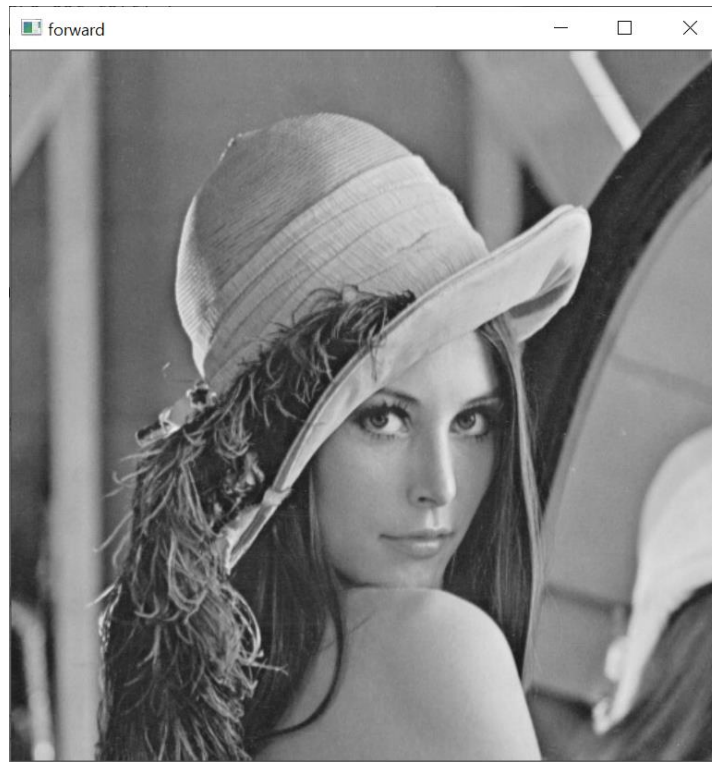
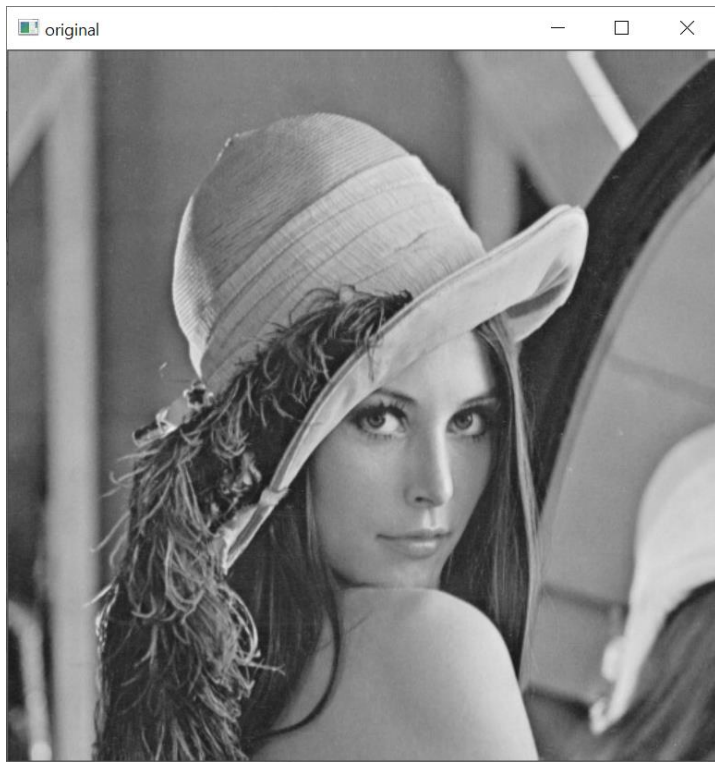
$$\mathbf{M} = \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix}$$



실습

- Translation(+50,+100)

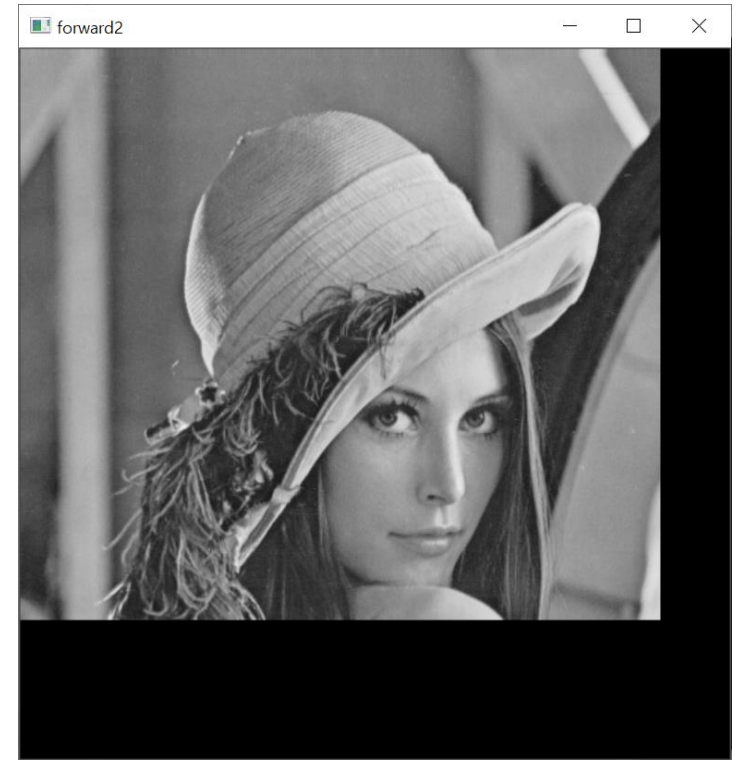
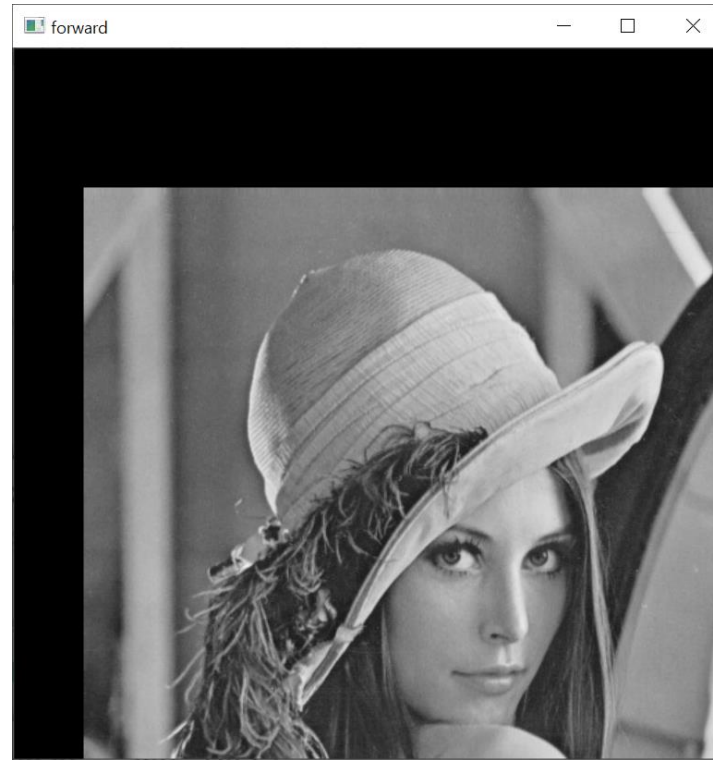
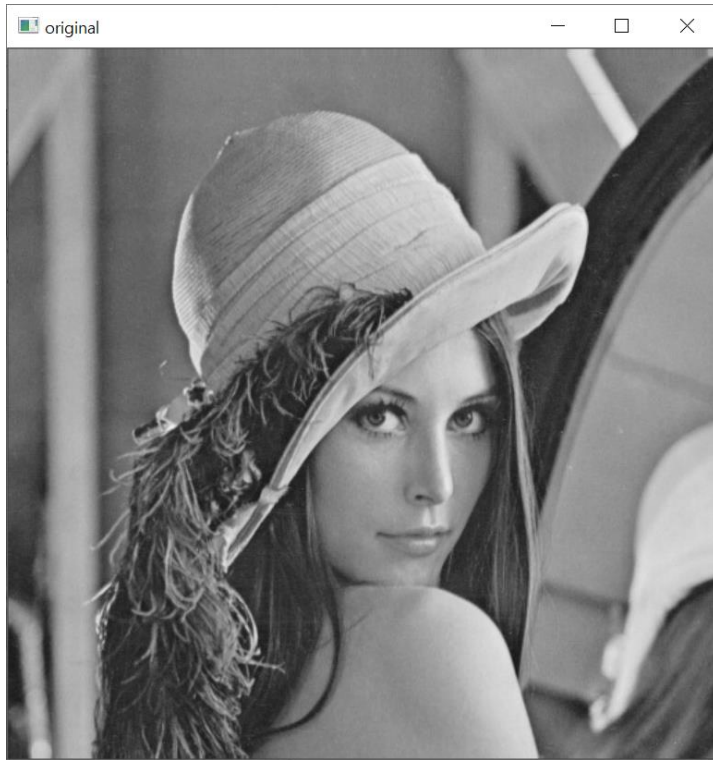
$$M = \begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$



실습

- Translation(+50,+100)

$$\begin{array}{l} M \\ \begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{l} M \text{ inv} \\ \begin{bmatrix} 1. & 0. & -50. \\ 0. & 1. & -100. \\ 0. & 0. & 1. \end{bmatrix} \end{array}$$



실습

```
if __name__ == '__main__':
    #img = np.zeros((256, 256)).astype('uint8')
    #img[50:100, 50:100] = 255

    img = cv2.imread('Lenna.png', cv2.IMREAD_GRAYSCALE)

    transform_mat = np.array([[1, 0, 50],
                              [0, 1, 100],
                              [0, 0, 1]])

    dst = transform(transform_mat, img)

    cv2.imshow('original', img)
    cv2.imshow('transformed', dst)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

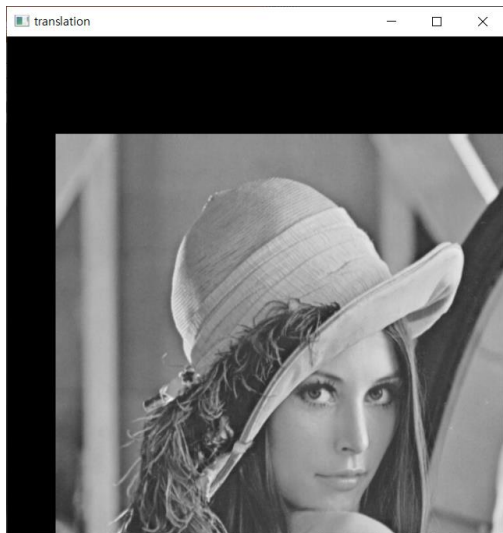
```
def transform(transform_mat, img):
    #####
    # TODO #
    # Forward transform 구현 #
    # dst의 크기는 scale이 1 이상일 때에만 때에만 변화하도록 함 #
    # -> 구현 편의를 위함 #
    # 각 픽셀마다 dst에 대응되는 좌표를 계산해 transform 수행 #
    #####
    src_h, src_w = img.shape
    y_scale = transform_mat[1, 1]
    x_scale = transform_mat[0, 0]

    dst_h = ???
    dst_w = ???
    dst = np.zeros((dst_h, dst_w), img.dtype)

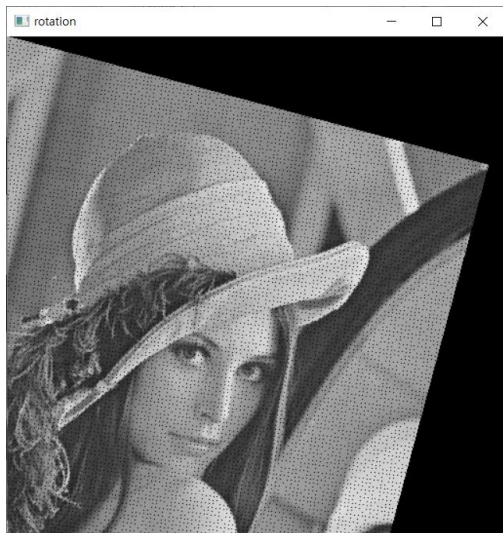
    for y in range(src_h):
        for x in range(src_w):
            dst[???, ???] = img[y, x]

    return dst
```

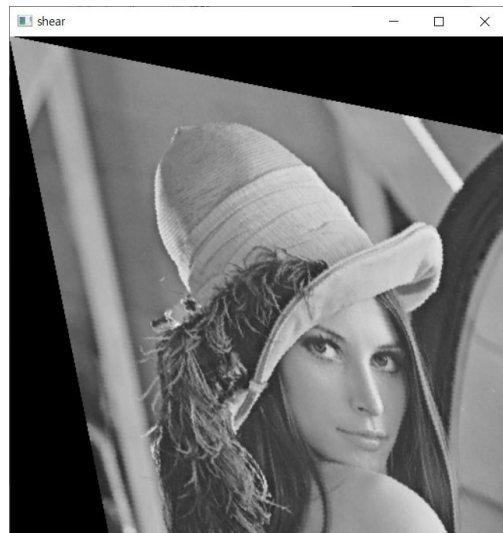

실습



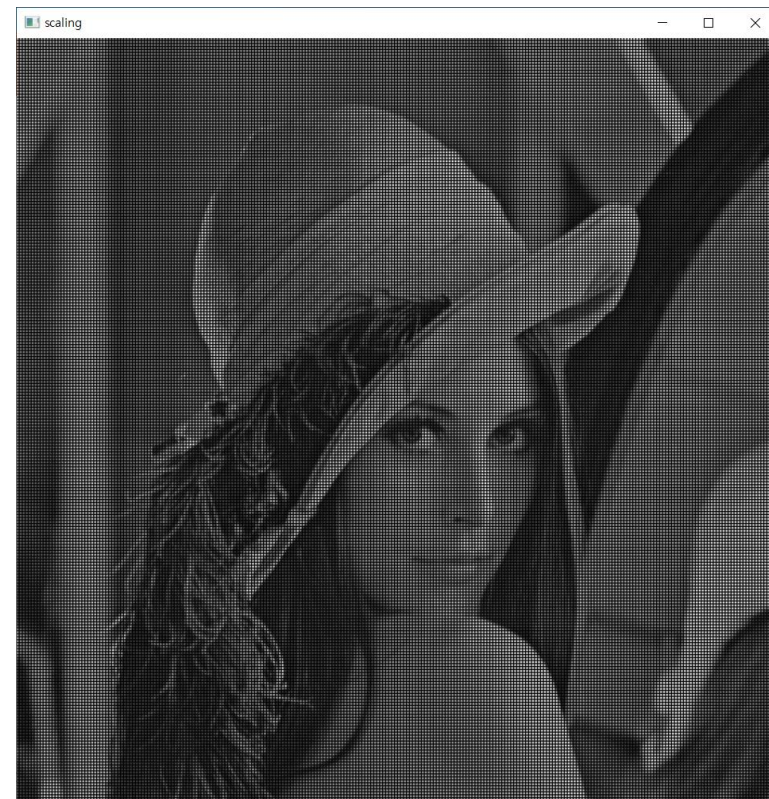
Translation



Rotation



Shear

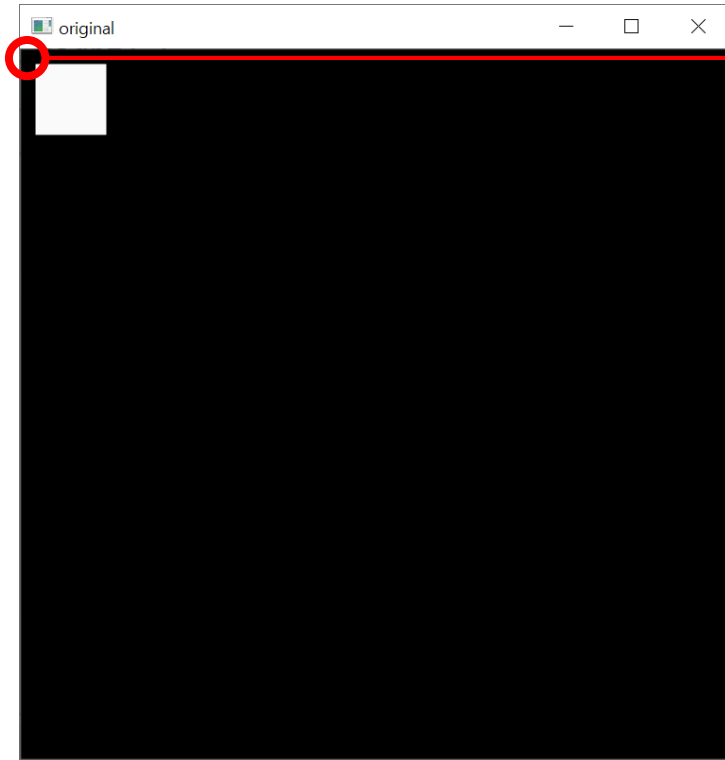


Scaling

과제

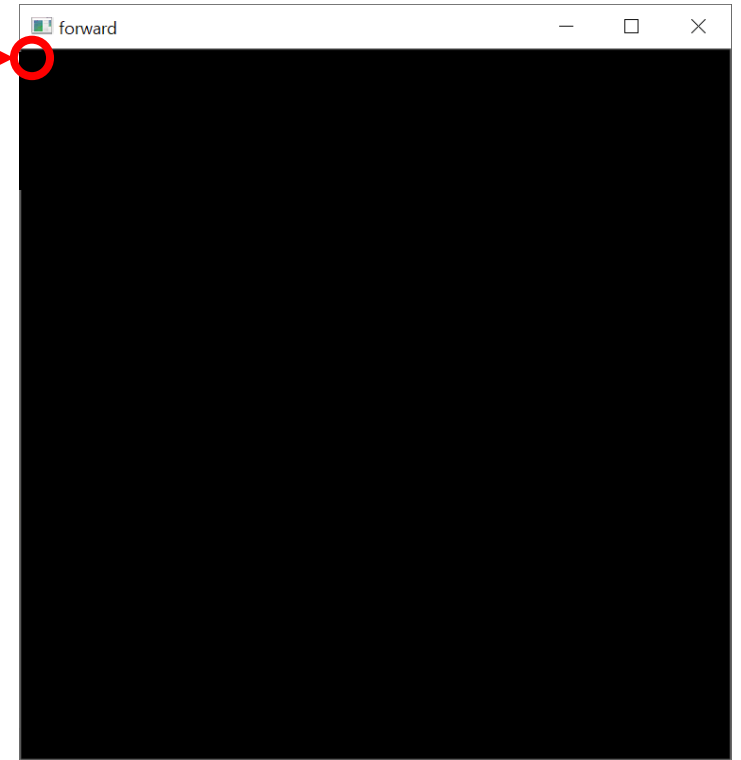
- Forward (x2 Scaling)

(0, 0)



M

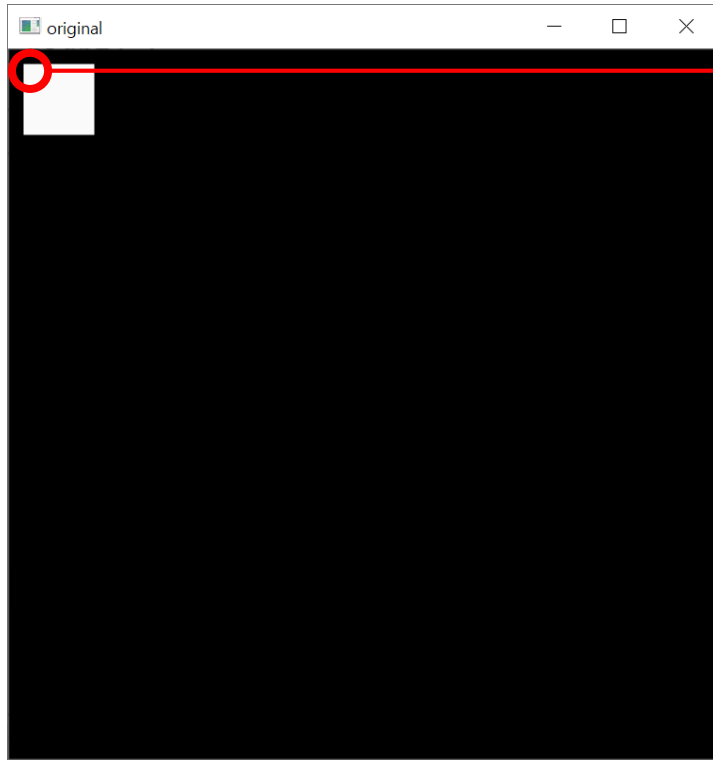
(0, 0)



과제

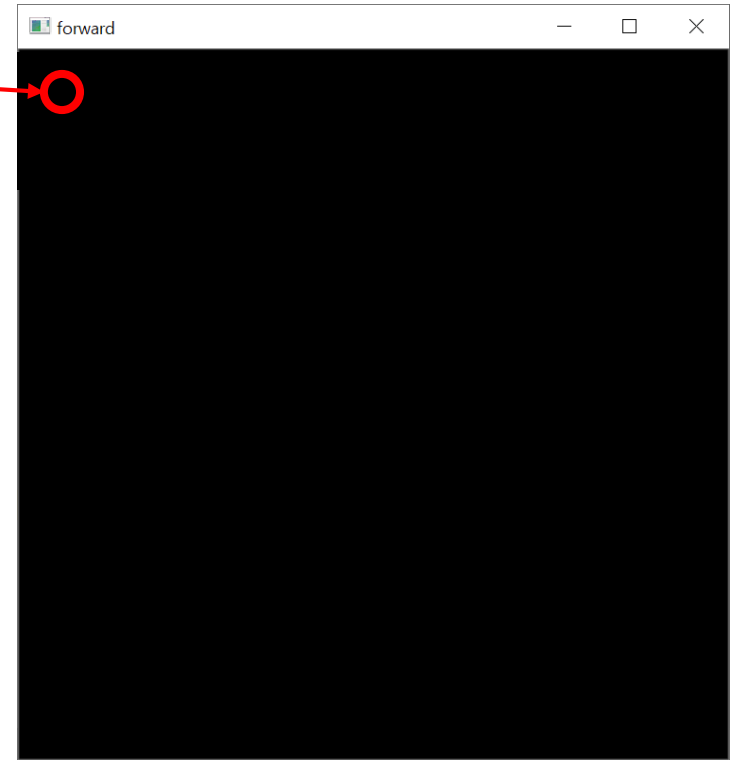
- Forward (x2 Scaling)

(1, 1)



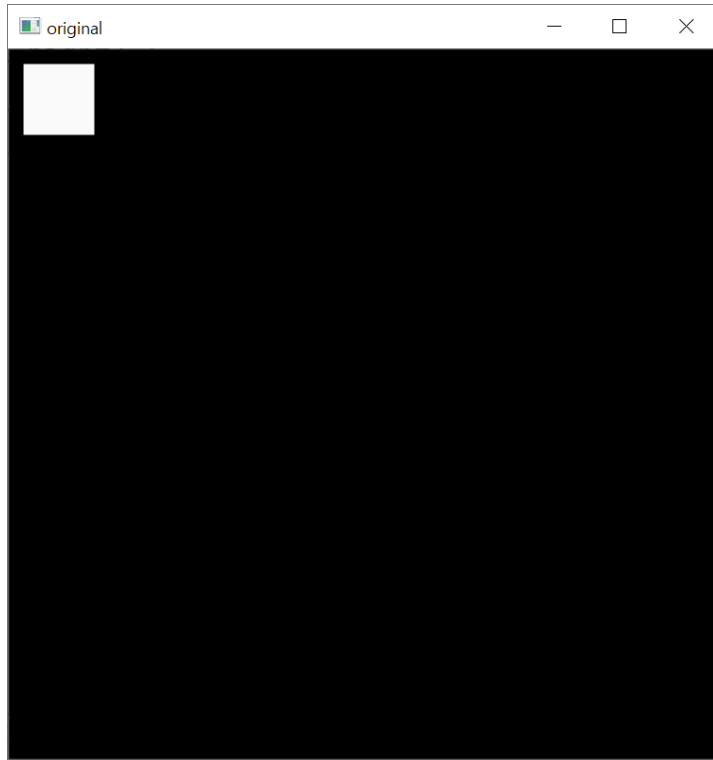
M

(2, 2)

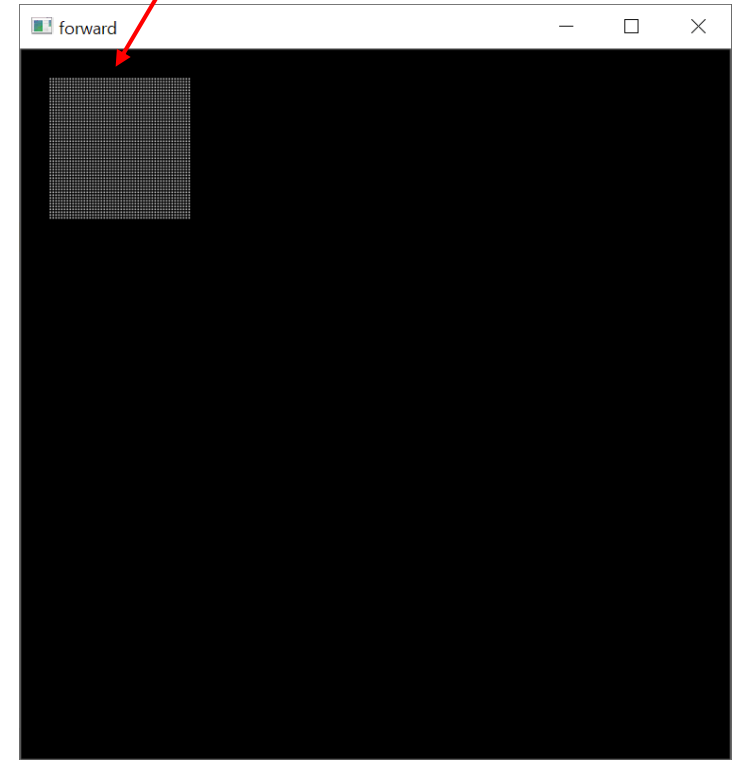


과제

- Forward (x2 Scaling)



→ M →



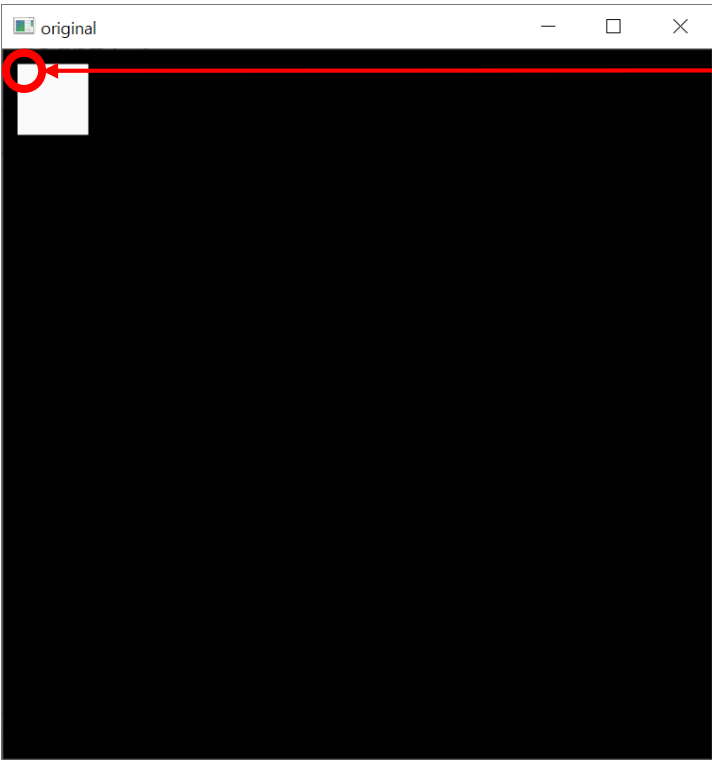
구멍이 생기는 이유는?

과제

- Backward ((x2, x2) Scaling)

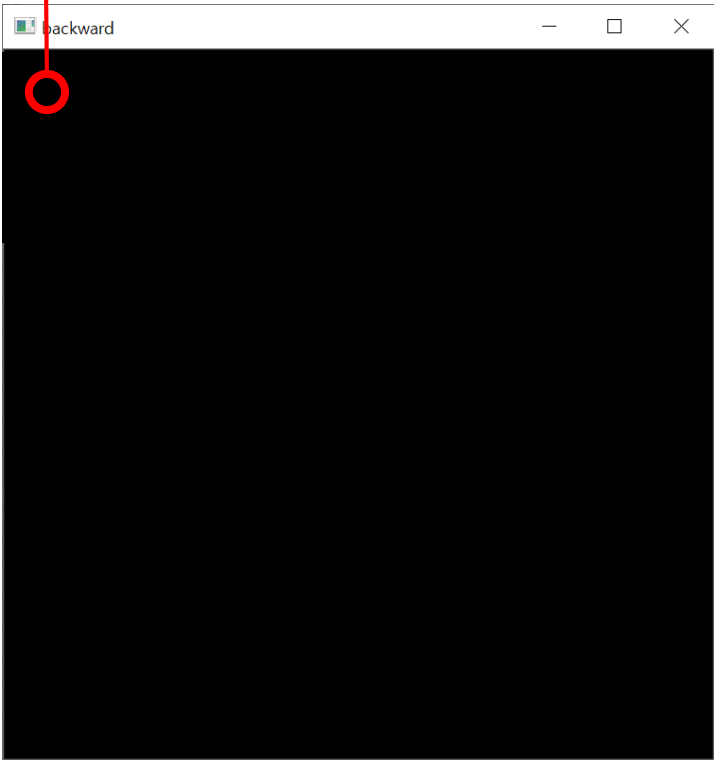
M	M inv
$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0. & 0. \\ 0. & 0.5 & 0. \\ 0. & 0. & 1. \end{bmatrix}$

(1, 1)



inv(M)

(2, 2)



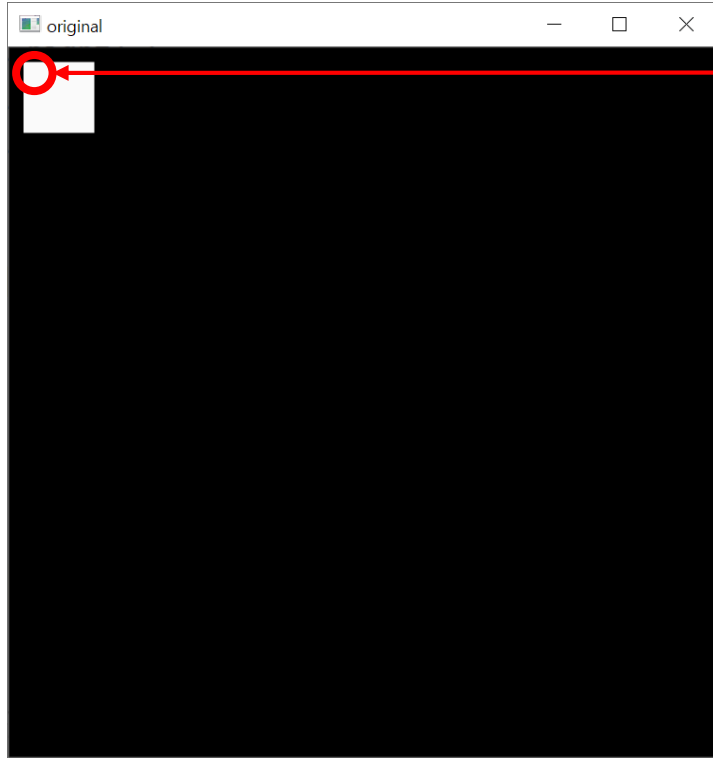
inv(M)

과제

• Backward ((x2, x2) Scaling)

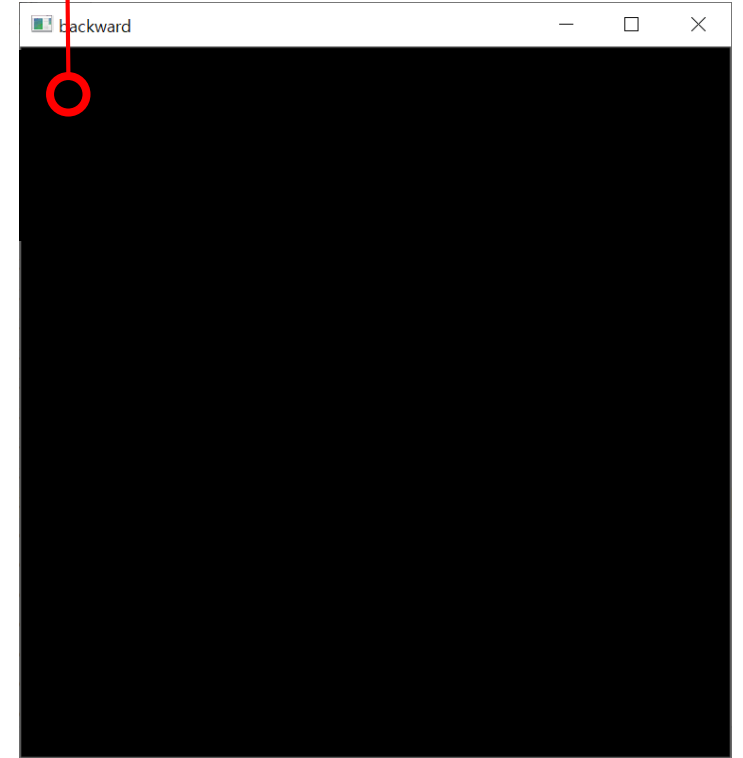
(1.5, 1.5) -> 존재하지 않는 좌표 -> bilinear interpolation

M	M inv
$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0. & 0. \\ 0. & 0.5 & 0. \\ 0. & 0. & 1. \end{bmatrix}$



inv(M)

(3, 3)



inv(M)

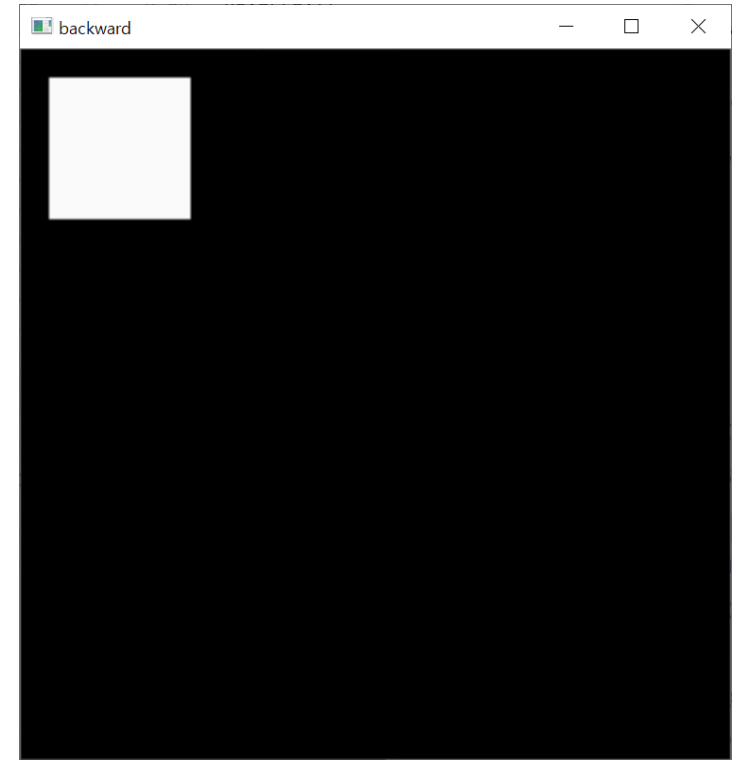
과제

- Backward ((x2, x2) Scaling)

M	M inv
$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0. & 0. \\ 0. & 0.5 & 0. \\ 0. & 0. & 1. \end{bmatrix}$



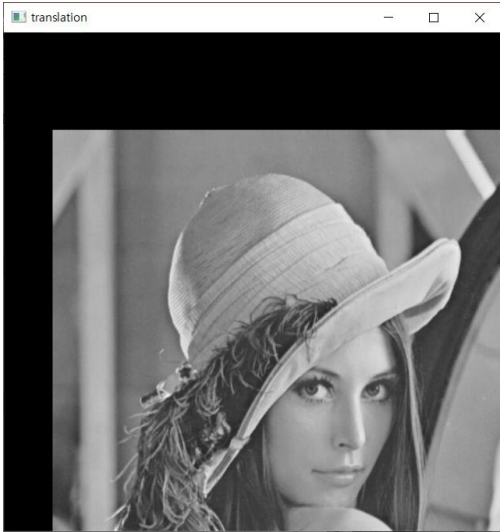
← **inv(M)** ←



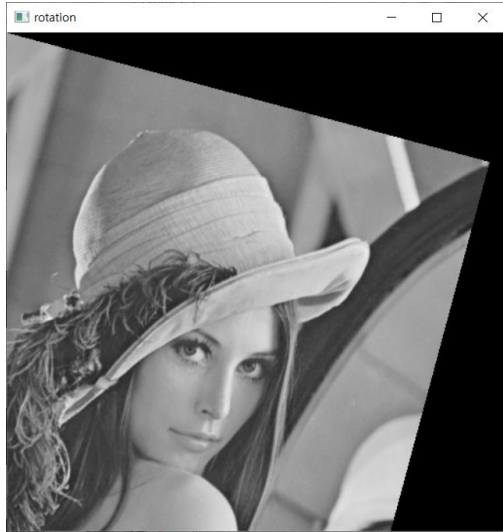
우리가 아는 Bilinear Interpolation은
Backward 방식의 Scaling이라고 볼 수 있다.

과제

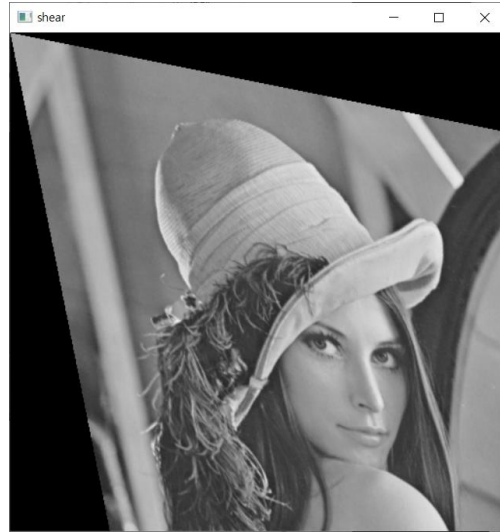
- Backward warping 결과물



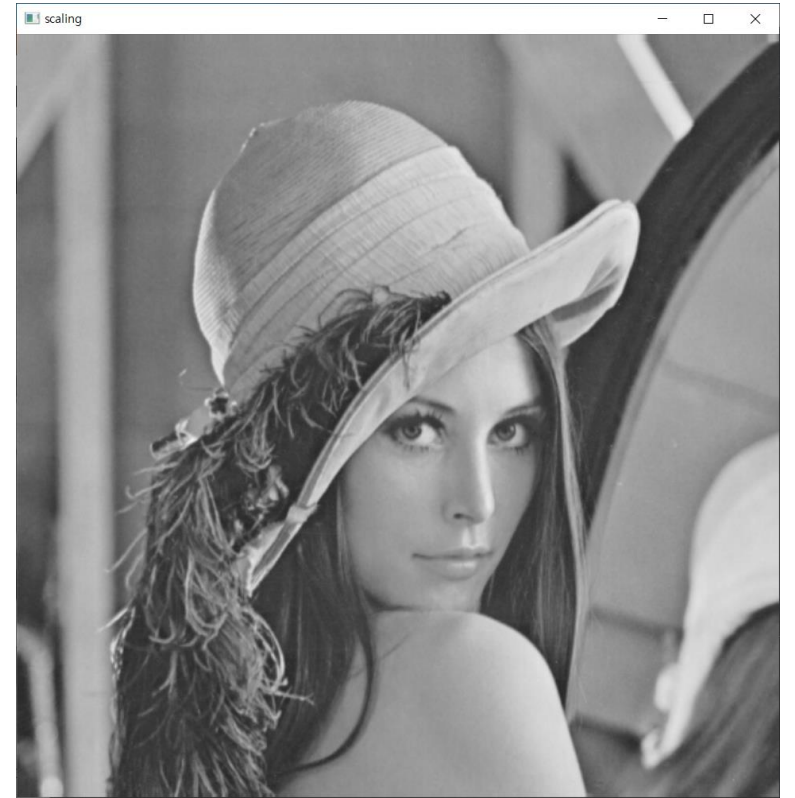
Translation



Rotation



Shear



Scaling

과제

• 제출 방법

- 코드 파일
 - 구현 결과가 포함된 python 파일(.py)
- 보고서
 - [IP]201900000_홍길동_13주차_과제.pdf
 - 보고서 양식 사용
 - PDF 파일 형식으로 제출(pdf가 아닌 다른 양식으로 제출시 감점)
- 제출 파일
 - [IP]201900000_홍길동_13주차_과제.zip
 - .py 파일과 pdf 보고서를 하나의 파일로 압축한 후, 양식에 맞는 이름으로 제출

QnA