Image Processing
실습 13주차

김 대 현

Department of Computer Science and Engineering

Chungnam National University, Korea

실습 소개

- **과목 홈페이지**
  - 충남대학교 사이버 캠퍼스 (https://dcs-lcms.cnu.ac.kr/login?redirectUrl=https://dcs-lcms.cnu.ac.kr/)
- **TA 연락처**
  - 김대현
  - 공대 5호관 531호 컴퓨터비전 연구실
  - Email: seven776484@gmail.com
    - [IP]을 이메일 제목에 붙여주세요
    - 과제 질문은 메일 또는 사전에 미리 연락하고 연구실 방문 가능
- **Tutor 연락처**
  - 정주헌
  - Email: 201802015@o.cnu.ac.kr

공지사항

- 공지사항
- Otsu Algorithm 과제 리뷰
- 실습
- 과제 (Backward구현)

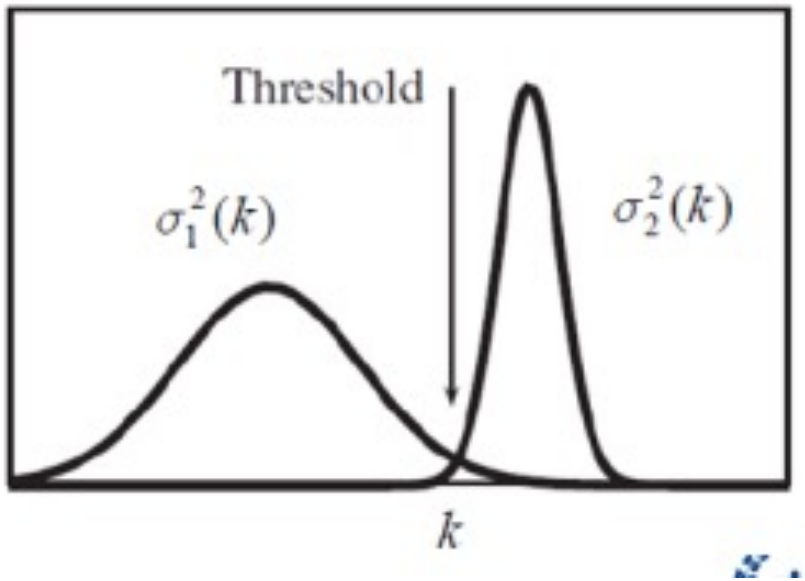
목차

- **공지사항**
  - 영상처리 02 · 03분반 과제 채점 공개
    - 이번주 중으로 Otus algorithm까지 과제 채점 진행
    - 채점은 기본적으로 각 주차 별 채점 기준표에 의거하여 각 분반 조교가 채점
    - 과제 채점에 대한 문의 사항은 각 분반 조교에게 문의
  - 영상처리 02 · 03분반 과제 채점 기준 공개
    - Otsu algorithm 채점 기준 공개
  - 과제 Copy 관련 공지
    - 과제 Copy와 관련하여 과제를 함께 진행하였을 경우 보고서에 같이 과제를 진행한 학부생 학번 기입
  - 과제 점수 문의는 메일 또는 수업시간 이후 질문시간을 이용.

Otsu algorithm 과제 리뷰

- **Otsu algorithm 과제 리뷰**
  - With in class, Between class variance 두 가지 방식을 구현

$m_1(k)$ or $m_2(k)$: mean at region 1 or region 2
$\sigma_1(k)$ or $\sigma_2(k)$: variance at region 1 or region 2



공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 기본적으로 02 · 03 분반 코드 모두 동일
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
      - get_threshold_by_inter_variance
      - get_threshold_by_within_variance
      - threshold
      - otsu_method

공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
        - 잘못 구현 시 1점 감점

```
def get_threshold_by_inter_variance(p):
    """
    :param p: 상대도수 값
    :return: k: 최적의 threshold 값
    """

    ########################################################
    # TODO
    # TODO otsu_method 완성
    # TODO  1. within-class variance를 이용한 방법
    # TODO  교수님 이론 PPT 22 page 또는 실습 PPT 참고
    ########################################################

    p += 1e-7  # q1과 q2가 0일때 나눗셈을 진행할 경우 오류를 막기 위함

    ???
```

공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
        - Moving average로 미구현시 1점 감점
        - 잘못 구현시 1점 감점

```
def get_threshold_by_within_variance(intensity, p):
    """
    :param intensity: pixel 값 0 ~ 255 범위를 갖는 행 vector 1 x 256
    :param p: 상대도수 값
    :return: k: 최적의 threshold 값
    """

    ######################################################
    # TODO
    # TODO otsu_method 완성
    # TODO  2. between-class variance를 이용한 방법
    # TODO  교수님 이론 PPT 26 page 또는 실습 PPT 참고
    ######################################################

    ???

    return k
```

공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
        - 잘못 구현 시 1점 감점

```
def get_hist(src, mask):

    """
    :param src: gray scale 이미지
    :param mask: masking를 하기 위한 값
    :return:
    """

    ########################################################################
    # TODO mask를 적용한 히스토그램 완성
    # TODO mask 값이 0인 영역은 픽셀의 빈도수를 세지 않음
    # TODO histogram을 생성해 주는 내장함수 사용금지. np.histogram, cv2.calHi
    ########################################################################
    hist = np.zeros((256,))

    ???

    return hist
```

공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
        - P값 잘못 구현 시 1점 감점

```
hist = get_hist(src, mask)
hist = hist.astype((np.int32))
intensity = np.array([i for i in range(256)])

#######################################################
# TODO 상대도수 p 구하기
# TODO 교수님 이론 PPT 17 page -> p_{i}에 해당
#######################################################
p = ???

#######################################################
# TODO otsu_method 완성
# TODO  1. within-class variance를 이용한 방법
# TODO      (get_threshold_by_within_variance 함수 사용)
# TODO  2. between-class variance를 이용한 방법
# TODO      (get_threshold_by_inter_variance 함수 사용)
#######################################################

k1 = get_threshold_by_within_variance(intensity, p)
k2 = get_threshold_by_inter_variance(p)
```

공지사항

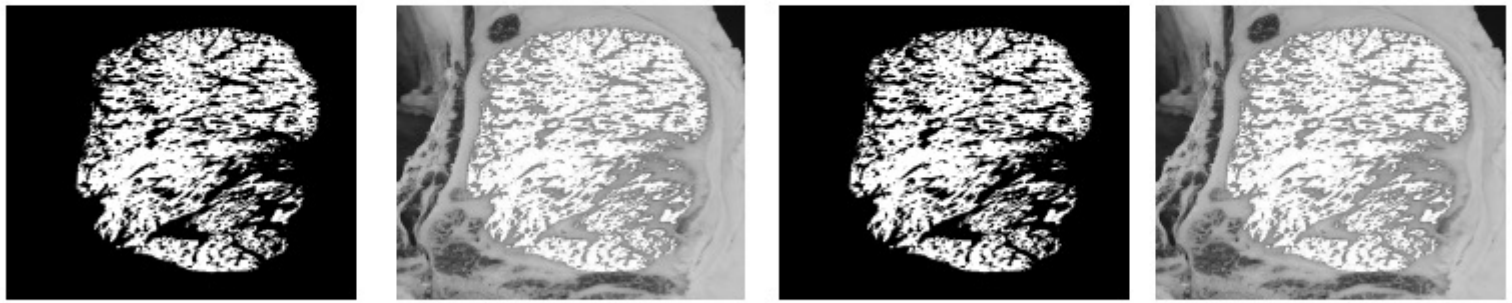
- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Otsu algorithm 채점 기준 공개
    - 코드 기준
      - 총 5가지 TODO 항목에 대하여 평가
        - 그래프 값 누락 시 2점

```
#########################################################
# TODO Bimodal histogram 완성
# TODO 2개의 peak에 해당하는 픽셀 값에 점 찍기
#########################################################
# Bi-modal Distribution
plt.plot(intensity, hist)
plt.plot(???, ???, color='red', marker='o', markersize=6)
plt.plot(???, ???, color='red', marker='o', markersize=6)
plt.xlabel('Pixel value')
plt.ylabel('Frequency')
plt.title('Interest region histogram')
plt.show()
```

공지사항

- **영상처리 02 · 03분반 과제 채점 기준 공개**
  - Canny Edge Detection 채점 기준 공개
    - 보고서 기준
    - 제출 이미지 누락 시 1점 감점
    - 코드 누락 시 1점 감점

- **결과 이미지**
  – 2가지 방식 모두 같은 결과 이미지

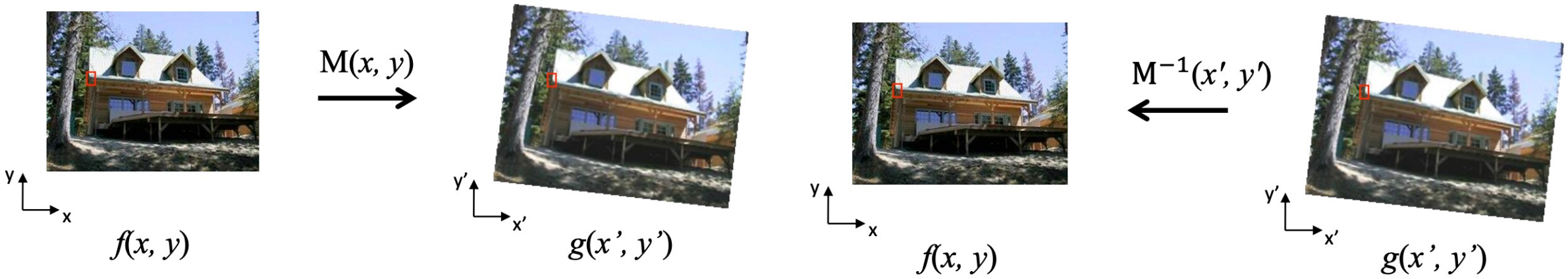


Within-class variance 방식

Between-class variance 방식

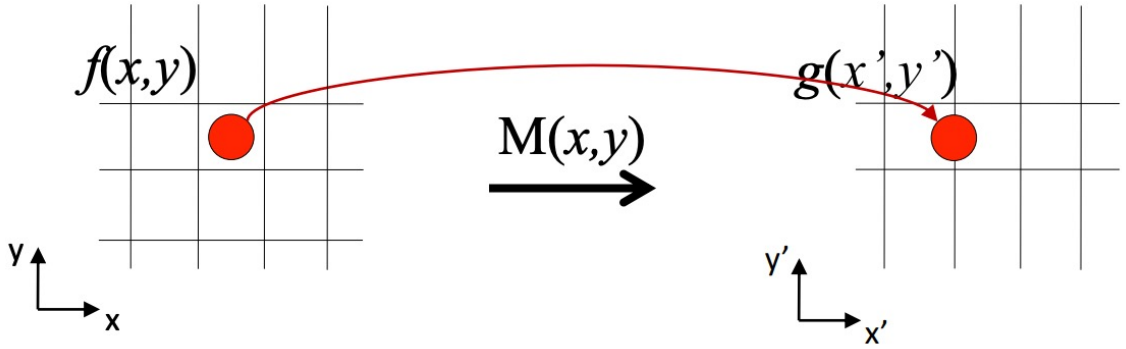
실습

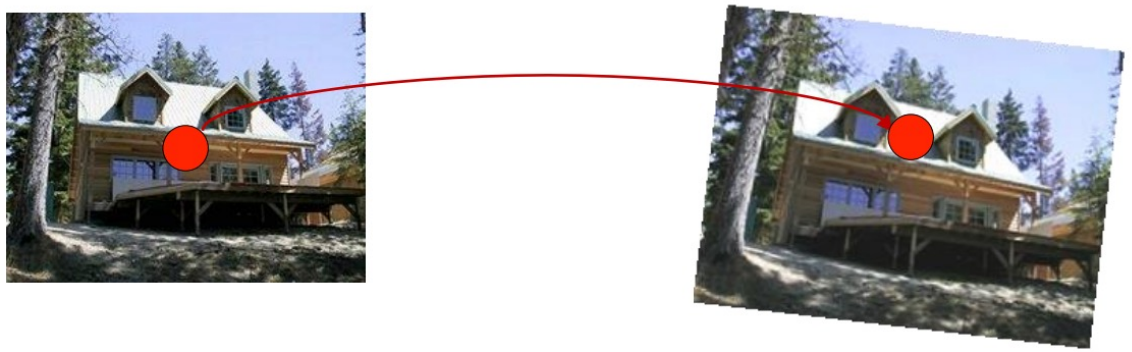
- **Forward vs Backward warping**
  - Question
    - 두 이미지 간의 좌표를 변환해주는 행렬을 알고 있다고 가정했을 때, 변환된 좌표에서의 값을 어떻게 결정할 것인가?



실습

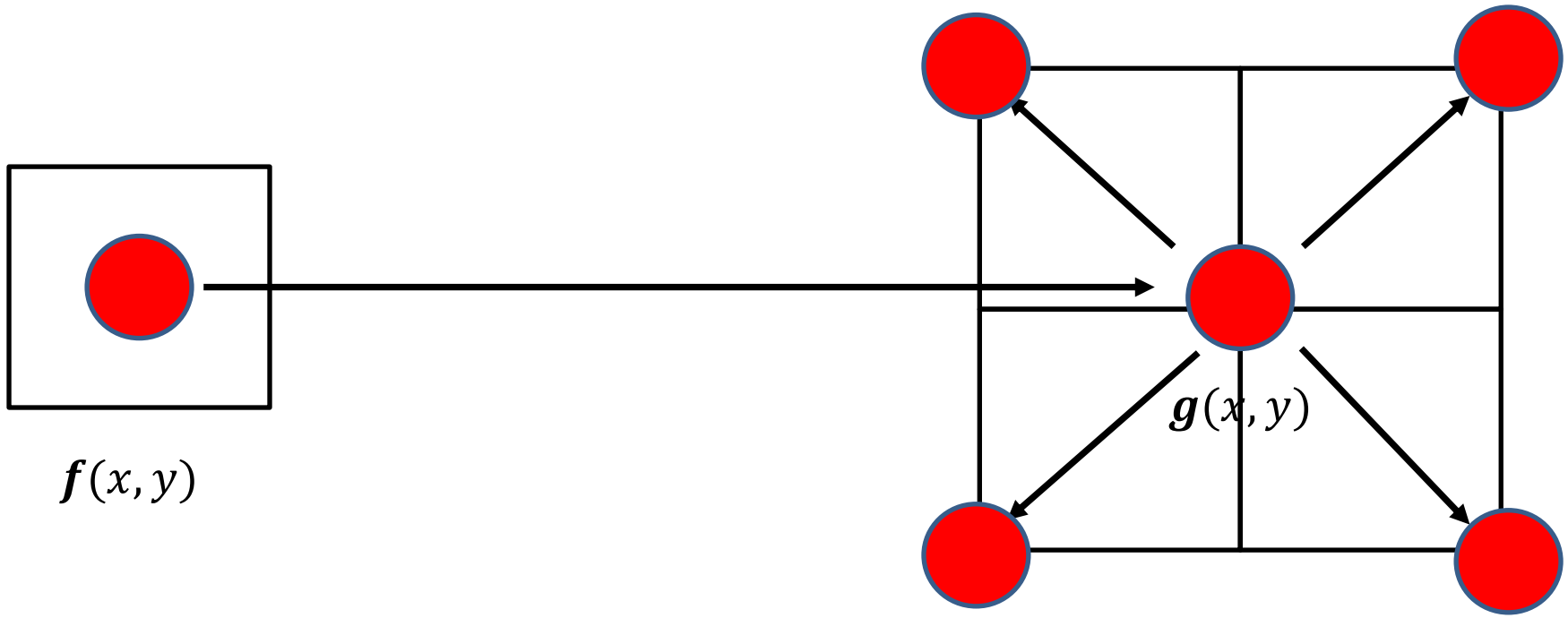
- **Forward warping**
  - **원본 이미지의 좌표(x, y)** 에서 **목표 이미지의 좌표(x', y')**로 매핑시키는 변환 행렬을 $M(x, y)$이라고 할 때, 원본 이미지에서의 **좌표 값 $f(x, y)$** 을 목표 이미지에서 변환된 좌표에서의 값$g(x', y')$으로 설정





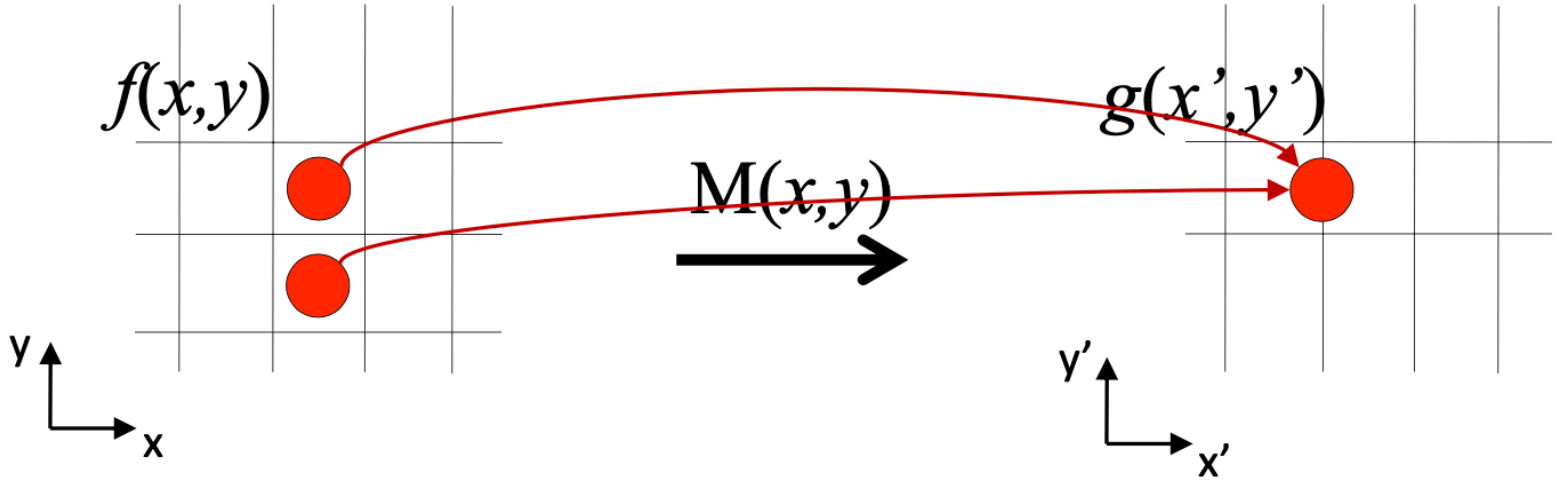
실습

- **Forward warping**
  - 이때 변환된 좌표의 위치가 소수일 경우, 소수 위치를 중심으로 주변 정수 위치 좌표에 모두 같은 값을 설정



실습

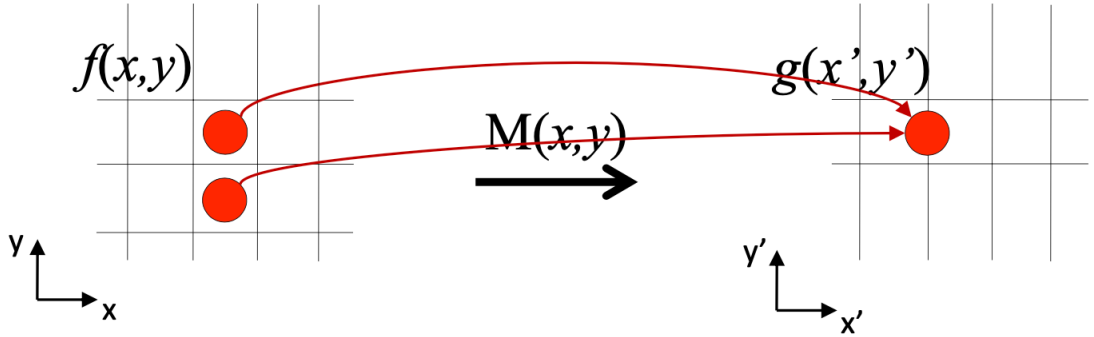
- **Forward warping**
  - 문제점 1
    - One to Many mapping 문제: 변환된 좌표에서의 값이 2개 이상 존재
    - 좌표가 변환될때 마다, 그 변환된 좌표의 수를 기록 한 후 픽셀 값들을 평균화 한다.



$$g(x', y') = \frac{1}{n}\sum_{k=1}^{n} f(x_k, y_k)$$

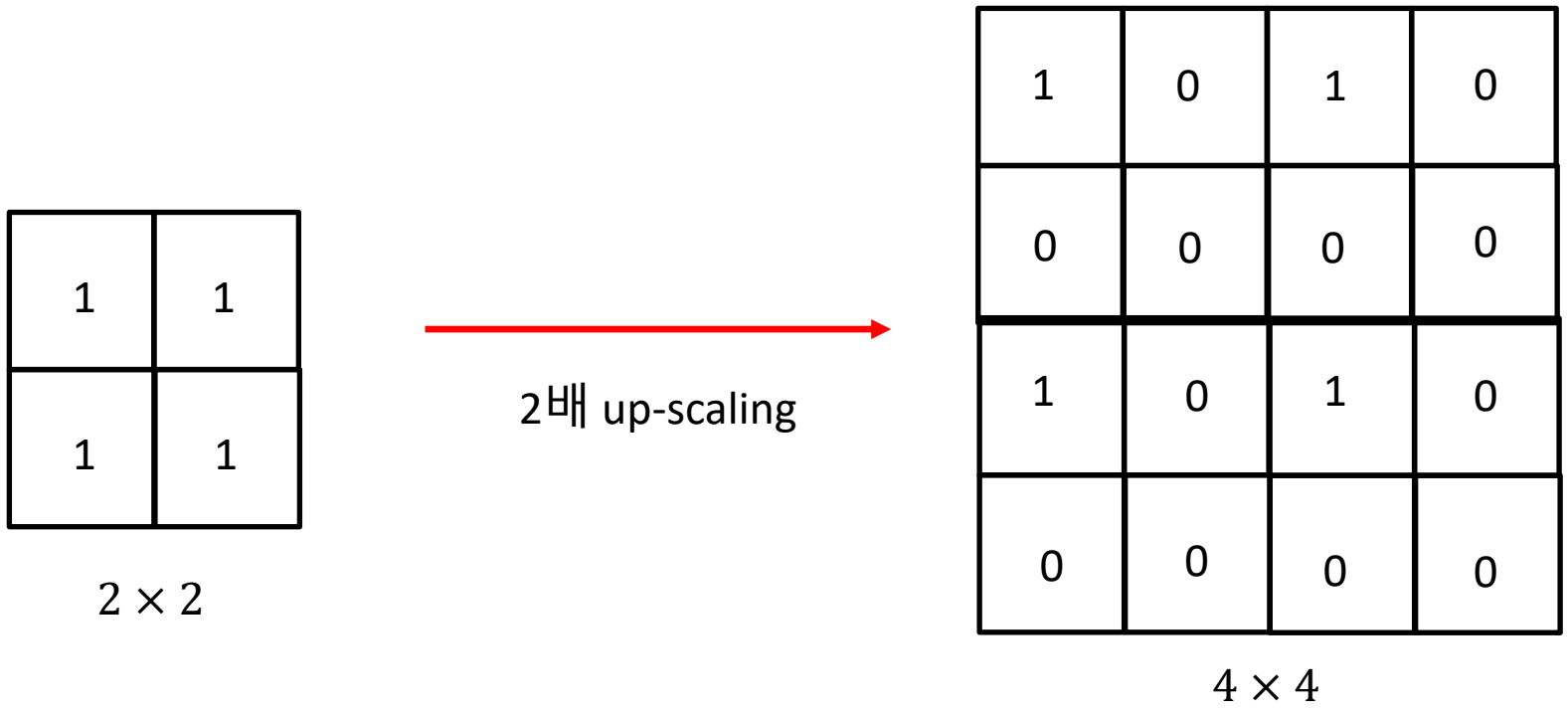
실습

- **Forward warping**
  - 문제점 2
    - **Hole 문제** : 다수의 점이 한점으로 매핑될 수 있기 때문에 변환된 이미지에서 매핑이 발생하지 않는 좌표가 나올 수 있음 -> 따라서 잘 사용하지 않음



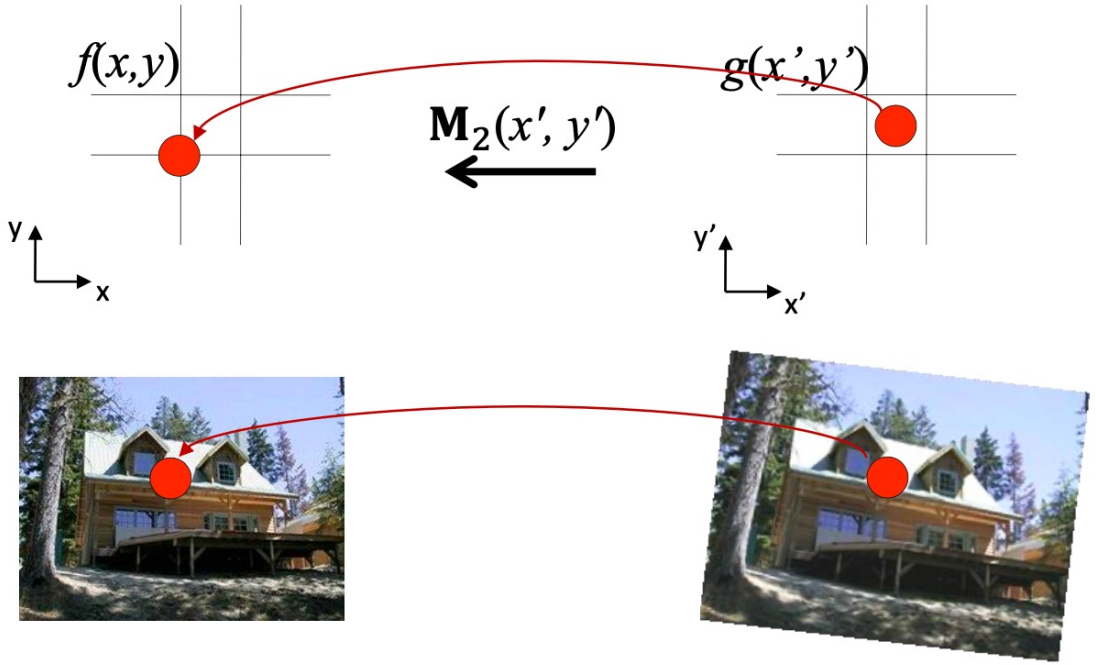
실습

- **Forward warping**
  - **Hole 문제 원리**
    - 좌표 변환 시 그 변환된 좌표들이 모두 정수일 경우 발생
    - 참고) 좌표 변환 시 그 변환된 위치가 소수일 경우 이웃한 정수 위치의 값들에 모두 값을 채움
      -> 따라서 Hole 문제가 발생하지 않음



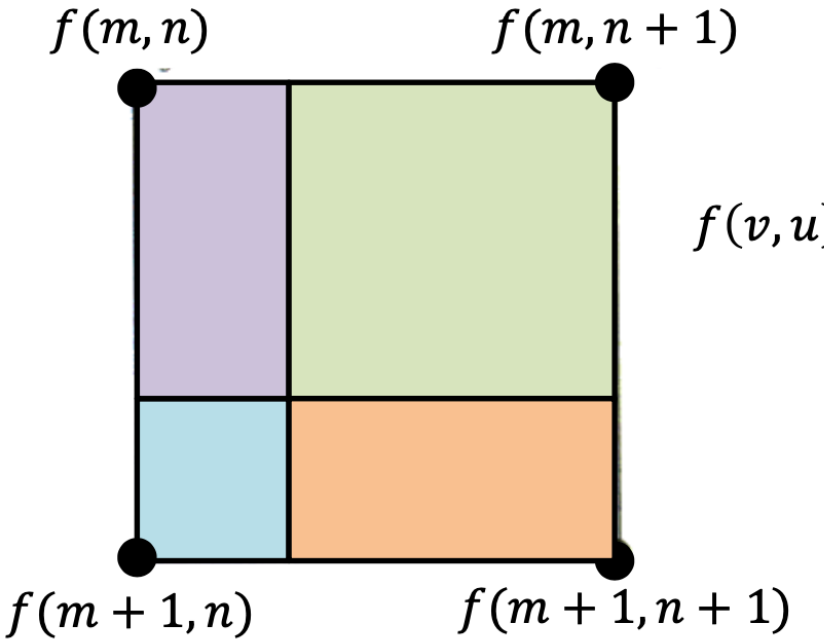
실습

- **Backward warping**
  - **역으로 목표 이미지의 변환된 좌표(x', y')** 에서 **원본 이미지의 좌표(x, y)** 로 매핑시키는 변환 행렬을 $\boldsymbol{M_2(x', y')}$이라고 할 때, 목표 이미지에서의 **변환된 좌표 값 $g(x', y')$**을 원본 이미지에서 해당 좌표 값 $f(x, y)$ 에서 가져옴
  - **Hole 문제가 없음**



실습

- **Backward warping**
  - 이때 변환된 좌표의 위치가 소수일 경우, Bilinear interpolation을 사용



$$f(v,u) = (1-s)(1-t)\cdot f(m,n) + s(1-t)\cdot f(m,n+1) + (1-s)t\cdot f(m+1,n) + st\cdot f(m+1,n+1)$$

실습

- **2D Transformation**
  - **Translation**
    - 각 축으로 주어진 값 만큼 좌표 값을 이동
    - Homogeneous coordinates

$$x' = x + t_x \qquad y' = y + t_y$$

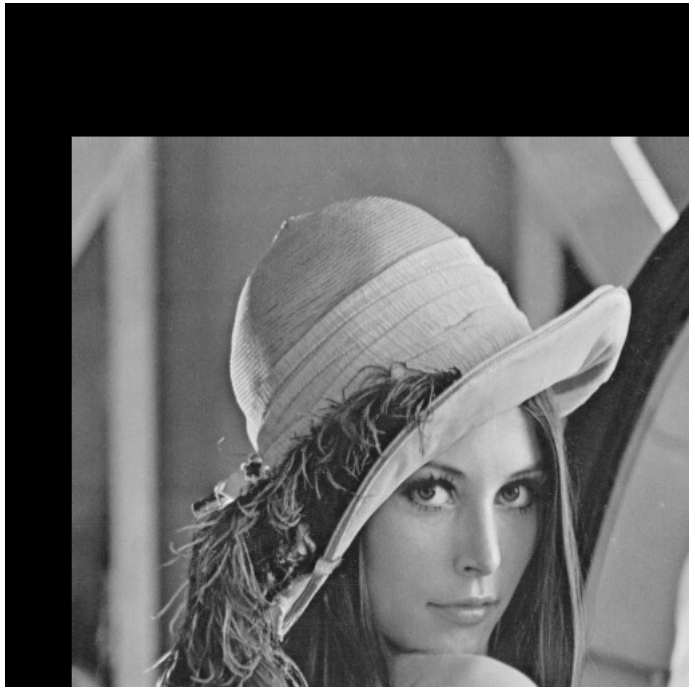
$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad M = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

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

실습

- **2D Transformation**
  - **Translation**
    - 예제 $x + 50, y + 100$ 만큼 이동

Original



Forward

Backward

실습

- **2D Transformation**
  - **Rotation**
    - x축을 기준으로 회전
    - Homogeneous coordinates

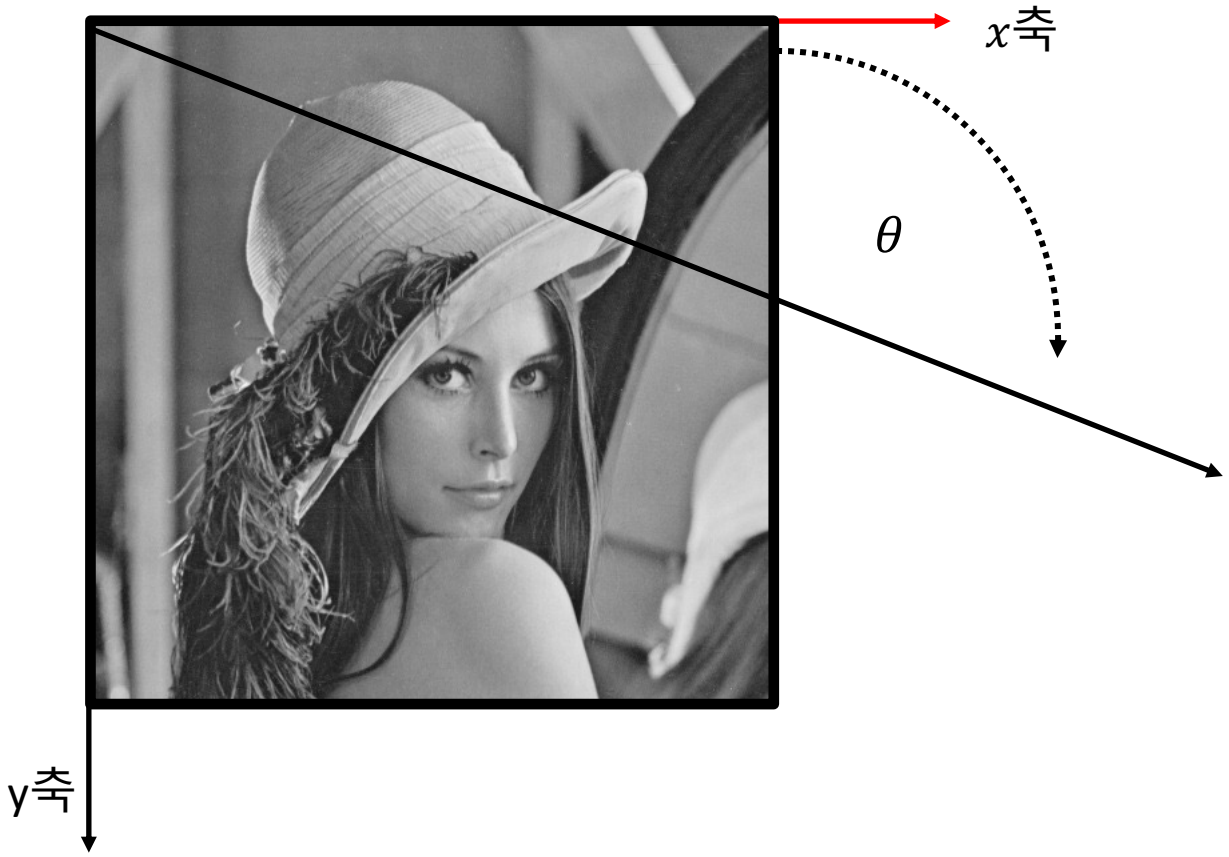
$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad M = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{pmatrix}$$

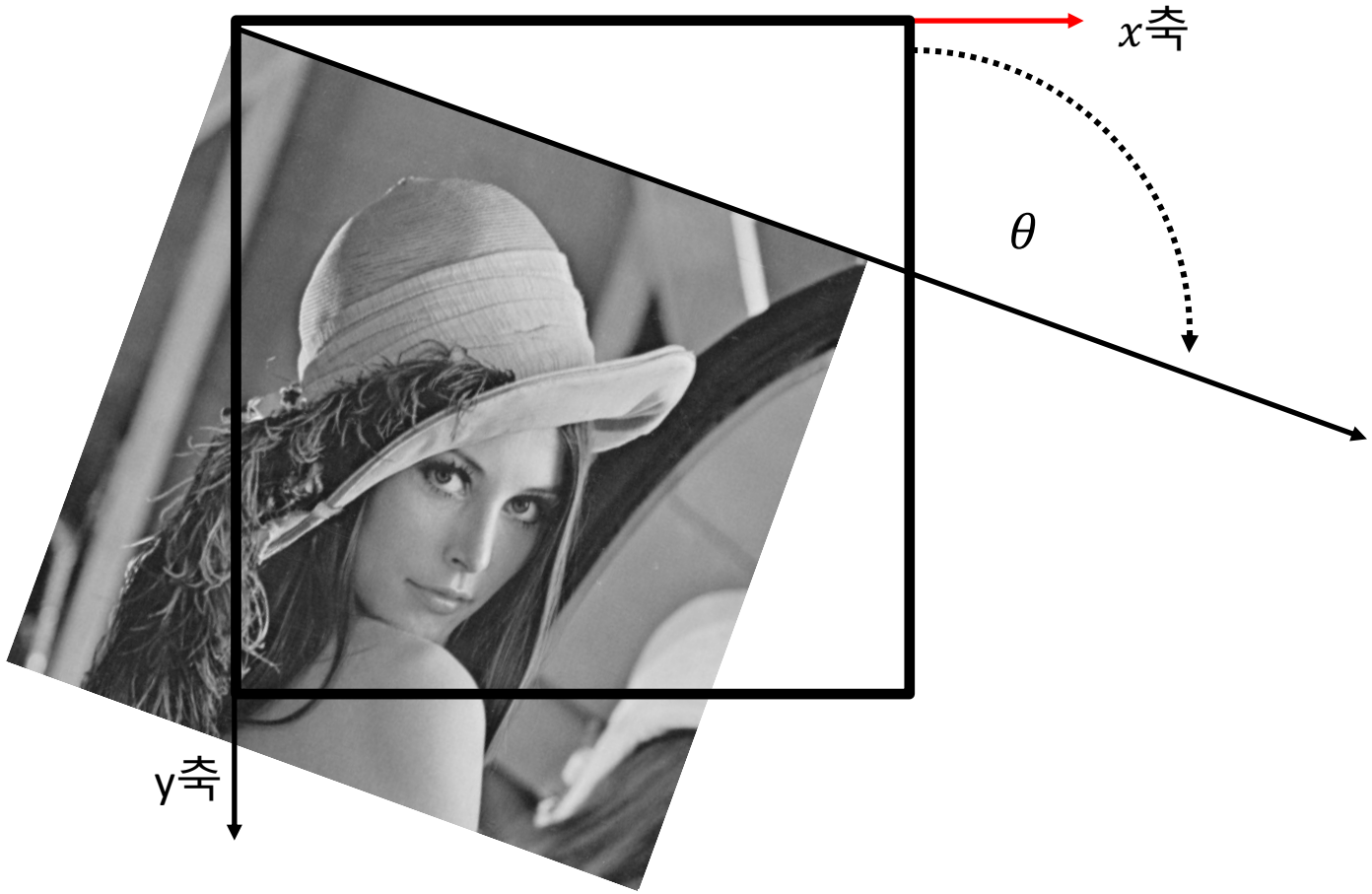
실습

- **2D Transformation**
  - **Rotation**
    - x축을 기준으로 회전
    - 예시 - x축의 양의 방향 + θ



실습

- **2D Transformation**
  - **Rotation**
    - x축을 기준으로 회전
    - 예시 - x축의 양의 방향 + θ



실습

- **2D Transformation**
  - **Rotation**
    - x축을 기준으로 회전
    - 예시 - x축의 양의 방향 + θ

실습

- **2D Transformation**
  - **Rotation**
    - x축을 기준으로 회전
    - 예시 - x축의 양의 방향 + θ

실습

- **2D Transformation**
  - **Rotation**
    - 예제- x축의 양의 방향 $+ 20^{\circ}$

Original

Forward

Backward

실습

- **2D Transformation**
  - **Scaling**
    - x축, y축으로 축소 또는 확대
    - Homogeneous coordinates

$$x' = ax$$
$$y' = by$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$M = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax \\ bx \\ 1 \end{pmatrix}$$

실습

- **2D Transformation**
  - **Scaling**
    - 예제- x축 2배, y축 2배
    - Forward warping시 Hole 현상

Original

Forward

Backward

실습

- **2D Transformation**
  - **Scaling**
    - 예제- x축 1.8배, y축 1.8배
    - Forward warping시 Hole 현상 없음

Original

Forward

Backward

실습

- **2D Transformation**
  - **Shearing**
    - $2D$ 도형의 모양을 바꿈
    - Homogeneous coordinates

$$x' = x + ay$$
$$y' = bx + y$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + ay \\ bx + y \\ 1 \end{pmatrix}$$

실습

- **2D Transformation**
  - **Shearing**
    - 예제- x축 방향 변화, $M = \begin{pmatrix} 1 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + 0.2y \\ y \\ 1 \end{pmatrix}$

Original

Forward

Backward

실습

- **2D Transformation**
  - **Shearing**
    - 예제- x축 방향 변화, $M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x+y \\ y \\ 1 \end{pmatrix}$

Original

Forward

Backward

실습

- **2D Transformation**
  - **Shearing**
    - 예제- y축 방향 변화, $M = \begin{pmatrix} 1 & 0 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ 0.2x + y \\ 1 \end{pmatrix}$

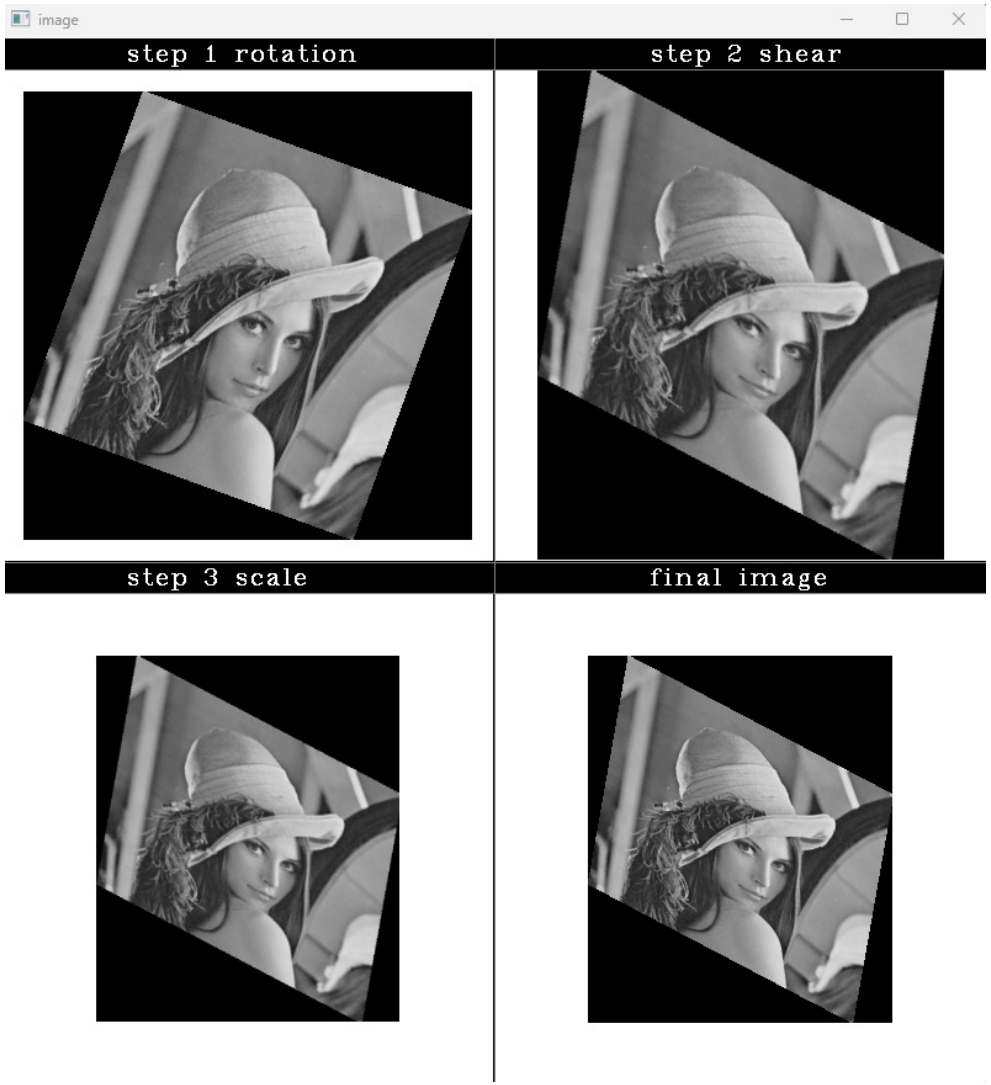
Original

Forward

Backward

과제

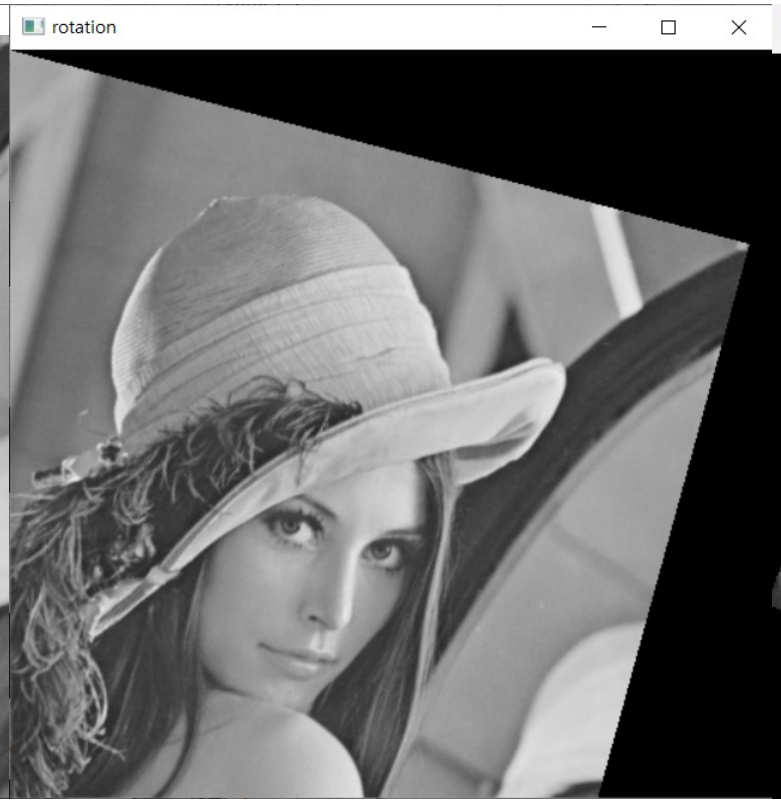
- **Backward warping 구현**



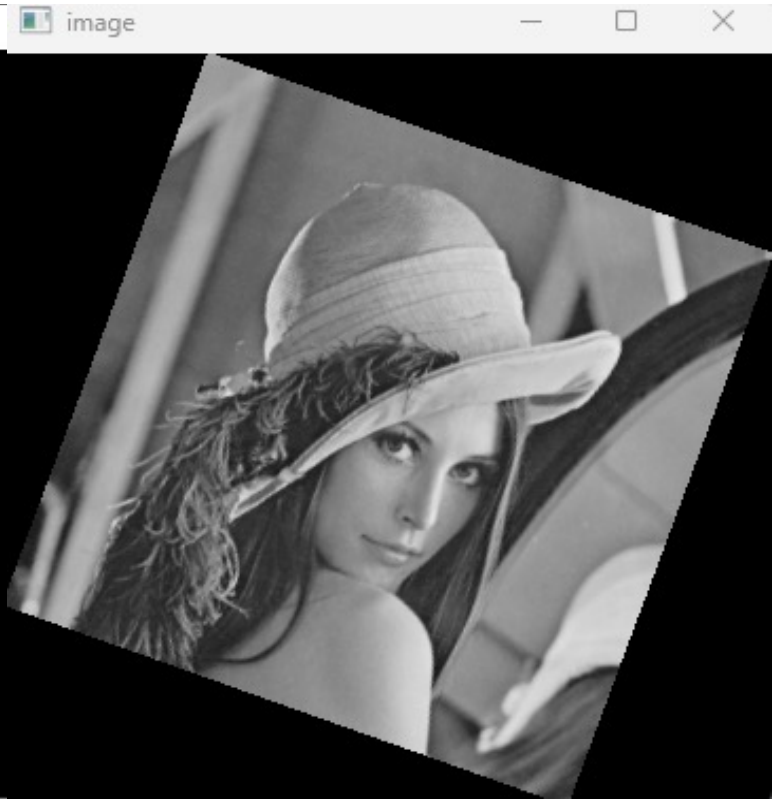
과제

- **Backward warping**
  - Warping 과정을 진행하면 이미지 잘림 현상이 나타남.
  - 따라서 결과 이미지의 크기를 구해서 보정해야 함

Input Image



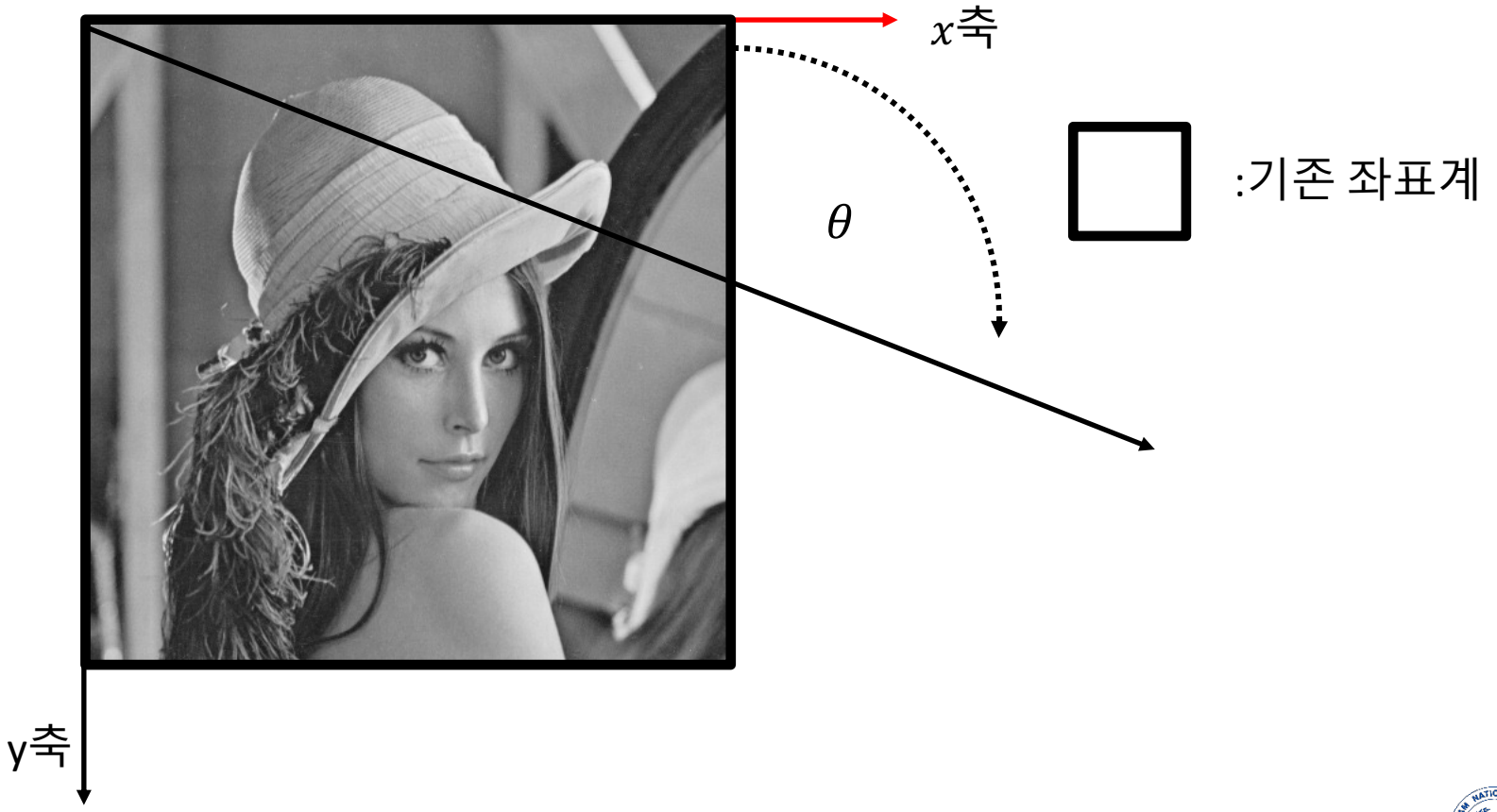
보정 X



보정 O

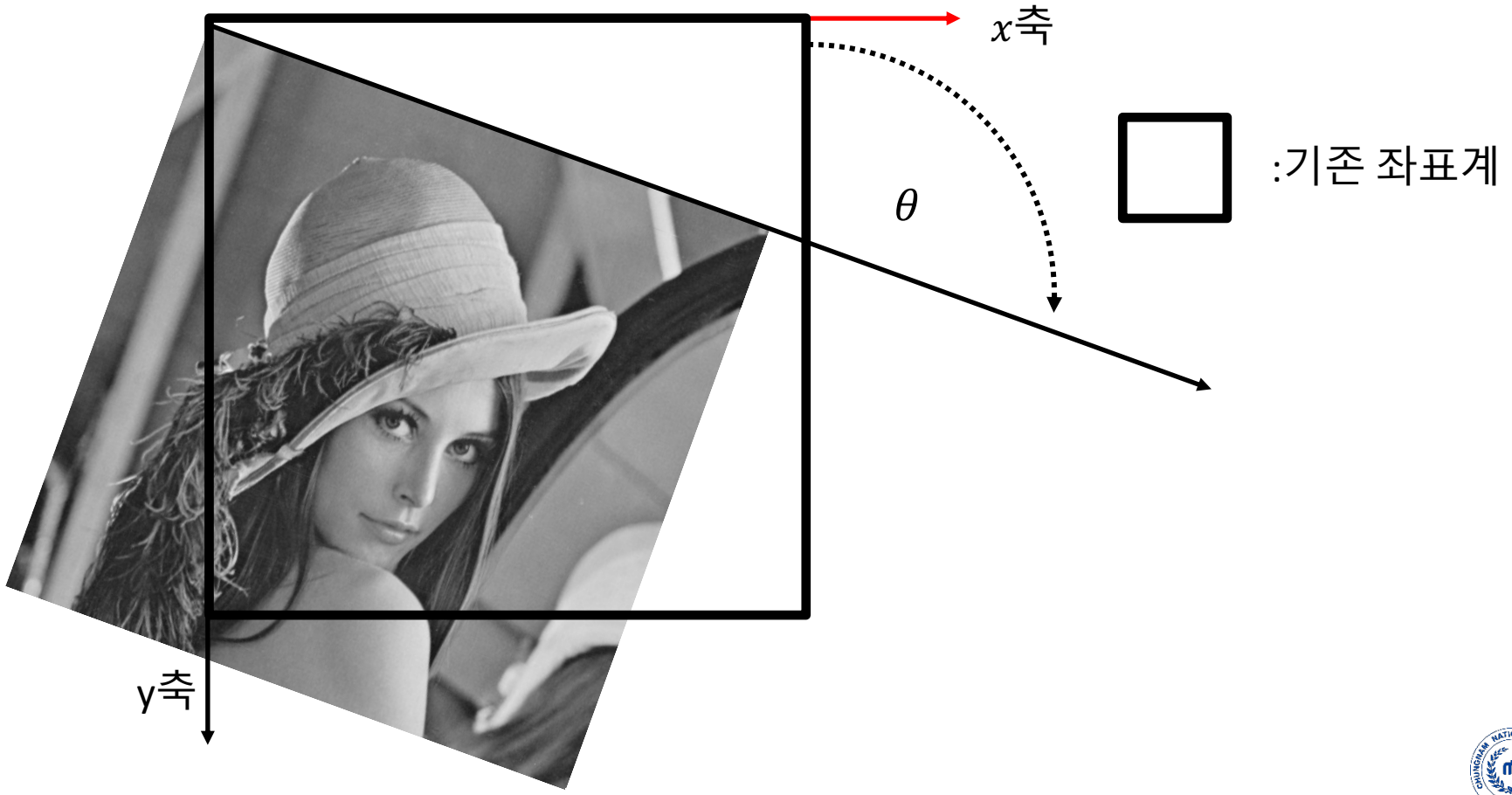
과제

- **Backward warping**
  - 이미지의 변환 시 잘림 현상 분석
    - Recap – Rotation x축의 양의 방향 + θ



과제

- **Backward warping**
  - 이미지의 변환 시 잘림 현상 분석
    - **예시 – Rotation** x축의 양의 방향 + θ



과제

- **Backward warping**
  - 이미지의 변환 시 잘림 현상 분석
    - 이미지를 변환 시킨 다는 것은 기본적으로 좌표계를 변환 시킨 다는 것
    - 기존 좌표계에서 본다면 변환된 좌표 값들은 기존 좌표계의 영역 밖일 수 있음

과제

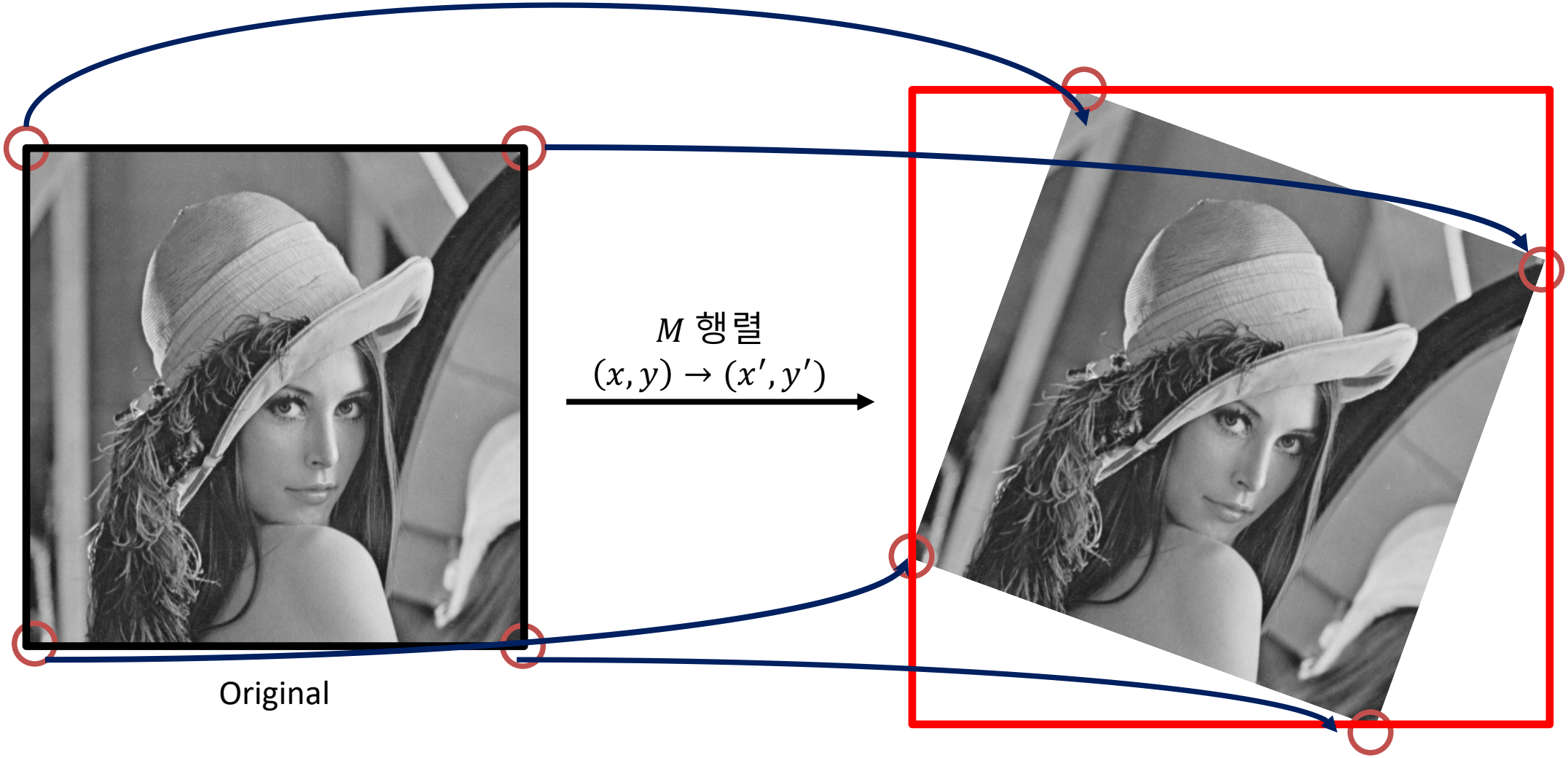
- **Backward warping**
  - 이미지의 변환 시 잘림 현상 분석
    - 이미지에서의 좌표는 모두 양의 정수이나 변환된 좌표 값들은 음수이거나 기존 좌표계의 최댓값을 초과할 수 있음
    - 그에 따라서 자연스럽게 변환된 이미지의 크기도 달라짐

:기존 좌표계

:변환 좌표계

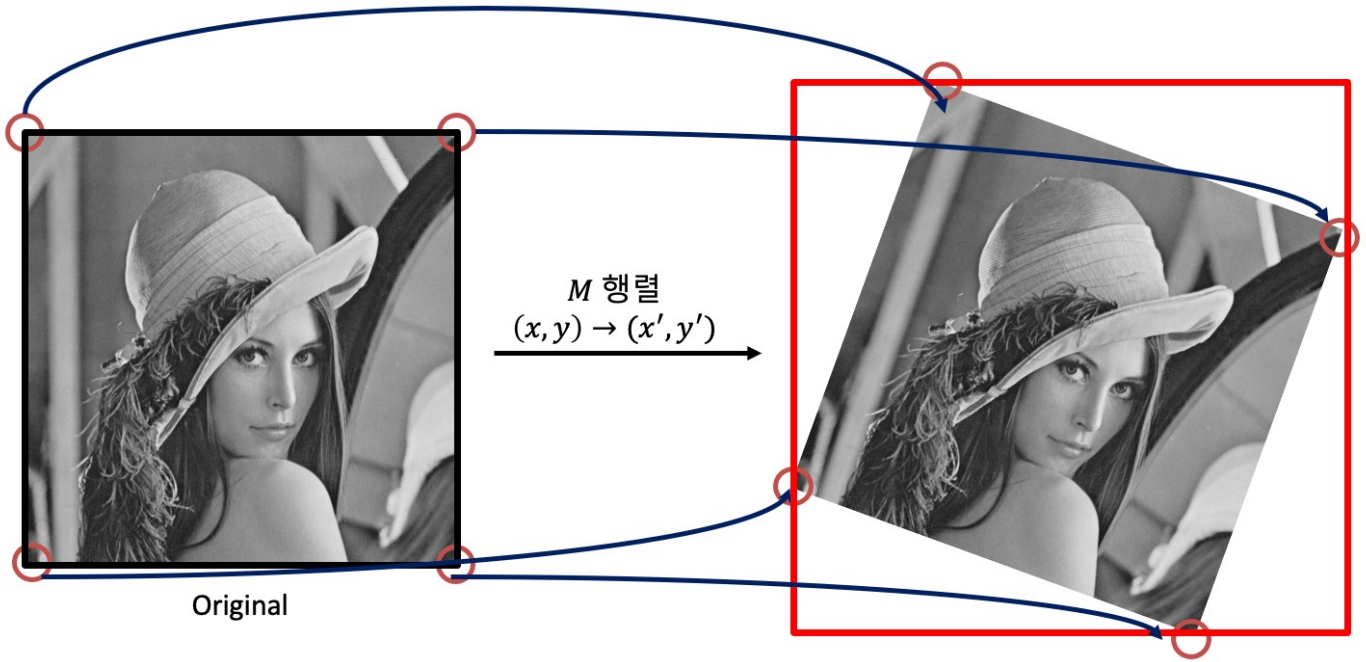
과제

- **Backward warping**
  - 보정과정은 다음과 같이 진행



과제

- **Backward warping**
  - 4개의 점의 좌표를 통해 변환된 좌표의 최대 최소 값을 알 수 있으므로 이미지 크기를 추정할 수 있음



1. 이미지1의 4개의 끝점(또는 ROI(Region of Interest)의 좌표) $P(x_k, y_k)$에서 변환 행렬 M에 의해 그에 해당하는 변환 좌표를 $P'(x'_k, y'_k)$라고 한다면,

2. 변환된 좌표계에서의 보정된 크기는 다음과 같이 구할 수 있다.

$$H' = int\left(\max(y'_k) - \min(y'_k)\right), \mathrm{k} = 1, \ldots., 4$$

$$W' = int\left(\max(x'_k) - \min(x'_k)\right), \mathrm{k} = 1, \ldots., 4$$

과제

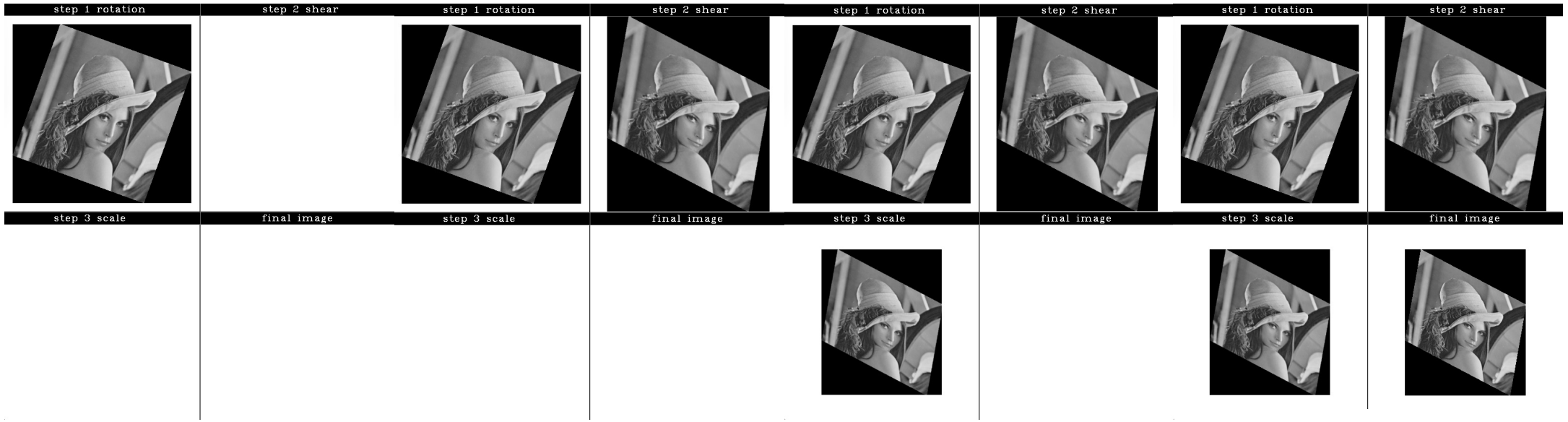
- **Backward warping**
  - 1. Backward를 step by step 형태로 진행
    매 step 마다 ROI 좌표와 그 좌표들을 활용하여 결과 이미지의 크기를 구해야함

    ex) rotate -> shear -> scaling

    2. 3가지 과정을 한번에 행렬로 나타내어 Backward를 진행.

과제

- 결과 이미지



Step 1 Step 2 Step 3 Result

과제

- **Backward warping**
  - 1. 결과 이미지 첨부 각 스텝별 이미지 (4장)
  - 2. 코드 첨부 및 코드 설명 추가.
  - 과제 진행 2023.5월 26일 ~ 2023년 6월 9일 23:59

Q & A