기하학 처리

기하학 처리

- 기하학 처리
  - ✓ 기하학(geometry)은 점, 선, 면, 도형 등의 기하학적인 대상을 다루는 학문인데 대상 의 길이, 넓이, 각도 등을 측정하거나 공간상의 특성을 연구하는 수학의 한 분야
  - ✓ 기하학의 영어 단어 geometry는 토지를 뜻하는 geo 와 측량을 뜻하는 metry 라는 단어가 합해져서 만들어진 용어
  - ✓ 영상처리에서 기하학 처리는 영상 내에 있는 기하학적인 대상의 공간적 배치를 변경하는 과정인데 이것을 픽셀의 입장에서 보면 영상을 구성하는 픽셀들의 공간적 위치를 재배치하는 과정이라 할 수 있음
  - ✓ 변환에는 크게 회전, 크기 변경, 평행이동 등이 있으며 영상처리 관련 논문에서는 이 세 가지 변환을 일컬어 RST 변환이라고 하는데 Rotation, S는 Scaling, T는 Translation의 첫 글자

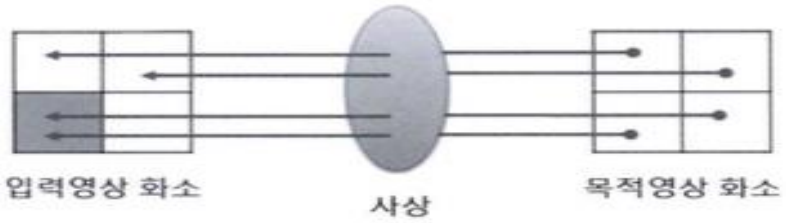
기하학 처리

- 사상
  - ✓ 기하학적 처리의 기본은 픽셀들의 배치를 변경하는 것
  - ✓ 사상은 픽셀들의 배치를 변경할 때 입력 영상의 좌표가 새롭게 배치될 해당 목적 영상의 좌표를 찾아서 픽셀 값을 옮기는 과정
  - ✓ 순방향 사상(forward mapping)과 역방향 사상(reverse mapping)
    - ❑ 순방향 사상은 입력 영상의 좌표를 중심으로 목적 영상의 좌표를 계산하여 픽셀의 위치를 변환하는 방식으로 이 방식은 일반적으로 입력 영상과 목적 영상이 크기가 같을 때 유용하게 사용되는 반면 두 영상의 크기가 달라지면 홀(hole)이나 오버랩(overlap)의 문제가 발생할 수 있음

    - ❑ 홀은 입력 영상의 좌표들로 목적 영상의 좌표를 만드는 과정에서 사상되지 않은 픽셀을 의미하는데 보통 영상을 확대하거나 회전할 때에 발생하는 반면 오버랩은 영상을 축소할 때 주로 발생하는데 이것은 입력 영상의 여러 픽셀들이 목적 영상의 한 픽셀로 사상되는 것을 의미

기하학 처리

- 사상
  - ✓ 순방향 사상(forward mapping)과 역방향 사상(reverse mapping)
    - ❑ 역방향 사상은 목적 영상의 좌표를 중심으로 역변환을 계산하여 해당하는 원본 영상의 좌표를 찾아서 픽셀 값을 가져오는 방식



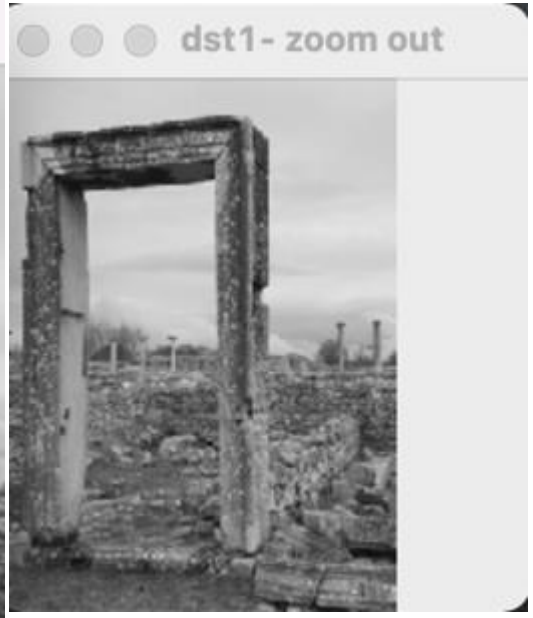
    - ❑ 입력 영상에서 하단 왼쪽 한 개의 픽셀이 목적 영상의 두 개 픽셀로 각각 사상
    - ❑ 역방향 사상의 방식은 홀이나 오버랩은 발생하지 않지만, 입력 영상의 한 픽셀을 목적 영상의 여러 픽셀에서 사용하게 되면 결과 영상의 품질이 떨어질 수 있음

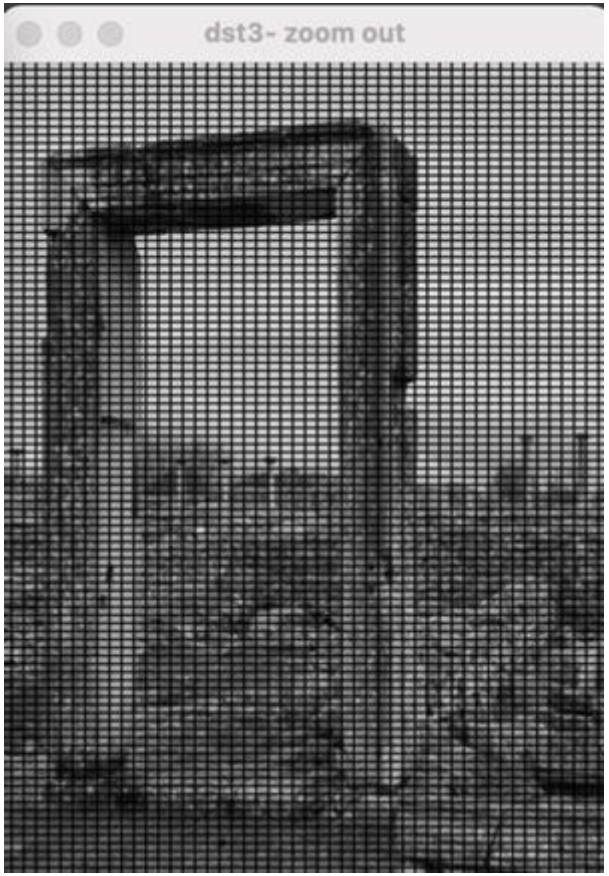
기하학 처리

- 크기 변경
  - ✓ 크기 변경(scaling)은 입력 영상의 가로와 세로로 크기를 변경해서 목적 영상을 만드는 방법
  - ✓ 비율을 이용해서 수행할 수 있는데 변경하려는 가로와 세로의 비율을 지정하여 입력 영상의 좌표에 곱하면 목적 영상의 좌표를 계산할 수 있음
  - ✓ 목적 영상의 크기를 지정해서 변경할 수도 있는데 이것은 입력 영상과 목적 영상의 크기로 비율을 계산하고 계산된 비율을 이용해서 목적 영상의 좌표를 계산
  - ✓ 순방향 사상을 이용해서 크기를 확대하면 채워지지 않은 홀이 다수 발생해 영상의 화질이 상당히 좋지 못함

기하학 처리

❖ 크기 변경





기하학 처리

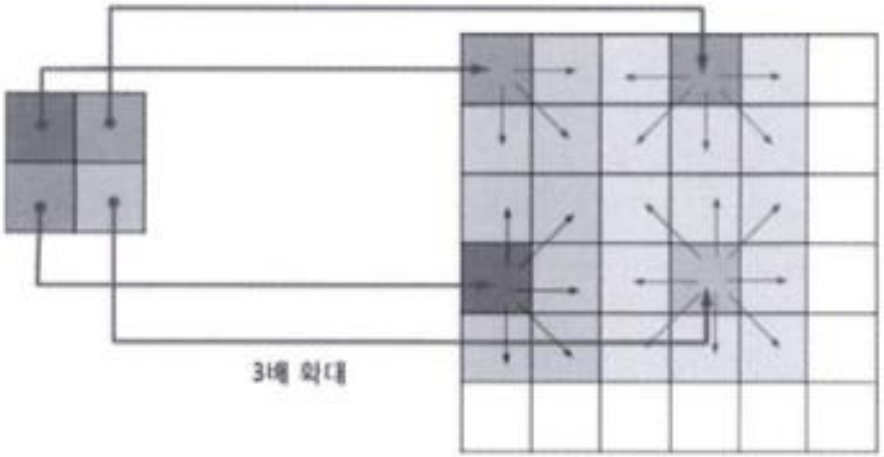
- 보간
  - ✓ 목적 영상에서 홀의 픽셀들을 채우며 오버랩되지 않게 픽셀들을 배치하여 목적 영상을 만드는 기법이 보간법(interpolation)
  - ✓ 보간법의 종류
    - ❑ 최근접 이웃 보간법(nearest neighbor interpolation)
    - ❑ 양선형 보간법(bilinear interpolation)
    - ❑ 3차 회선 보간법(cubic convolution interpolation)

기하학 처리

❖ 보간

✓ 최근접 이웃 보간법(nearest neighbor interpolation)

❑ 목적 영상을 만드는 과정에서 홀이 되어 픽셀 값을 할당 받지 못한 위치에 값을 찾을 때 그 위치에 가장 가깝게 이웃한 입력 영상의 픽셀 값을 가져오는 방법

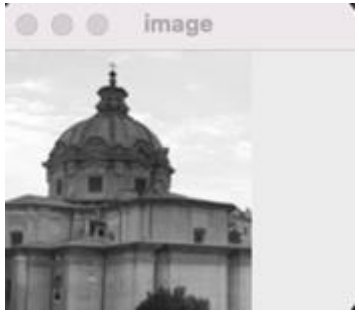


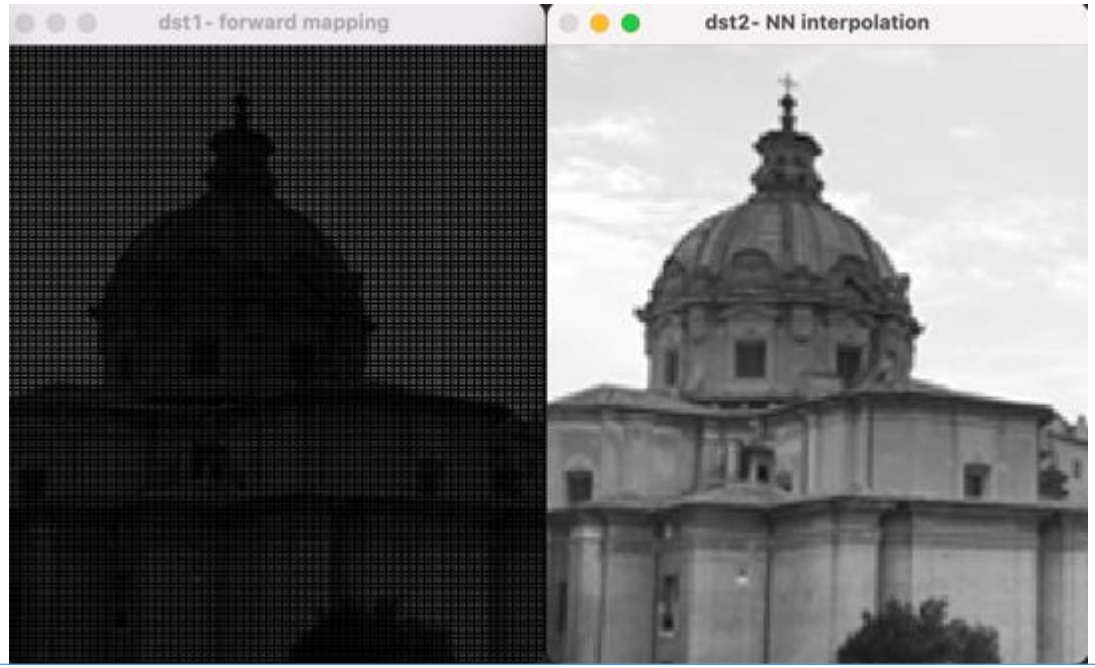
❑ 목적 픽셀의 좌표를 반올림하는 간단한 알고리즘으로 비어있는 홀들을 채울 수 있어 쉽고 빠르게 목적 영상의 품질을 높일 수 있지만 확대의 비율이 커지면 영상 내 에서 경계선이나 모서리 부분에서 계단 현상이 나타날 수 있음

기하학 처리

❖ 보간

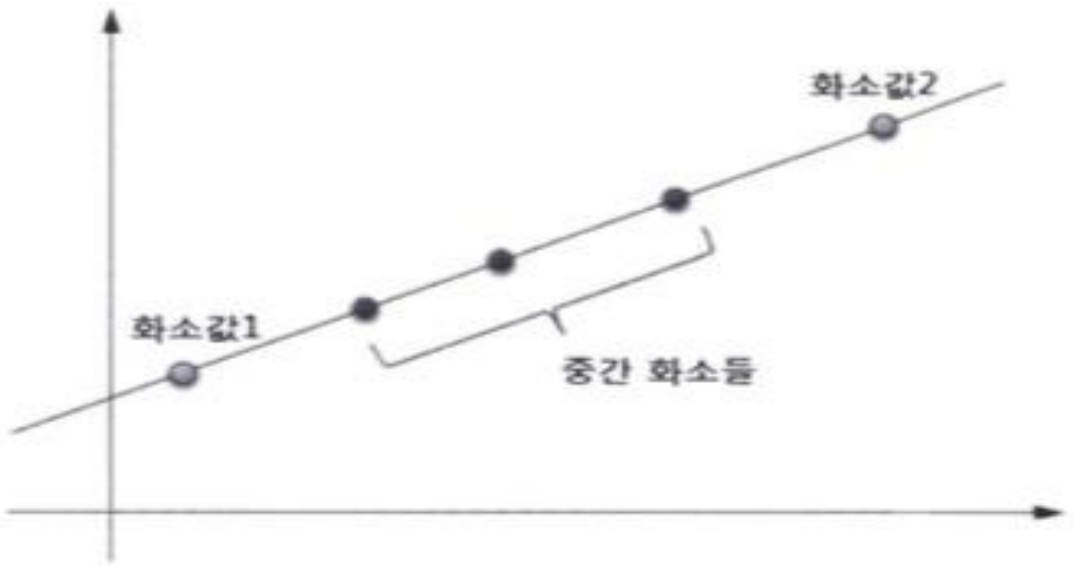
✓ 최근접 이웃 보간법(nearest neighbor interpolation)





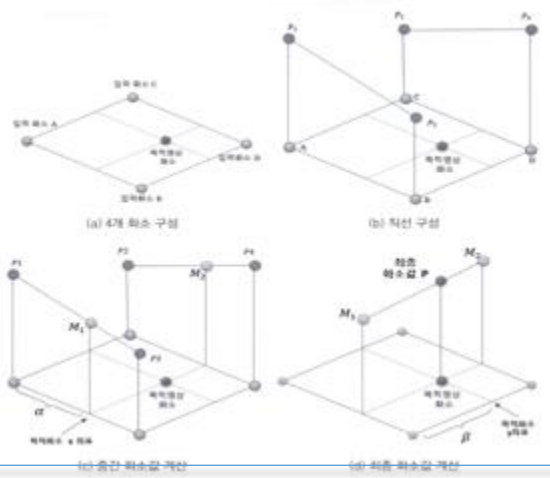
기하학 처리

- ❖ 보간
  - ✓ 양선형 보간법(bilinear interpolation)
    - ❑ 영상을 확대할 때 확대비율이 커지면, 최근접 이웃 보간법은 모자이크 현상 혹은 경계 부분에서 계단 현상이 나타나게 되는데 이러한 문제를 보완하는 방법이 양선형 보간법 (bilinear interpolation)
    - ❑ 선형(linear)은 그림과 같이 두 개 픽셀의 값을 알고 있을 때 그 값으로 직선을 그린 후 직선 위에 위치한 픽셀 들의 값은 직선의 수식을 따른다는 것



기하학 처리

- ❖ 보간
  - ✓ 양선형 보간법(bilinear interpolation)
    - ❑ 선형 보간을 두 번에 걸쳐서 수행하는 것이 양선형 보간
    - ❑ 과정
      - 목적 영상의 픽셀(P)을 역변환으로 계산하여 가장 가까운 위치에 있는 입력 영상의 4개 픽셀(A, B, C, D)을 가져옴
      - 가져온 4개 픽셀을 두 개씩(AB, CD) 묶어서 픽셀 값(P_1, P_2, P_3, P_4)으로 두 픽셀을 잇는 직선을 구성
      - 직선의 선상에서 목적 영상 픽셀의 좌표로 중간 위치를 찾고 그 위치의 픽셀 값(M_1, M_2)을 계산하는데 중간 위치의 픽셀 값은 기준 픽셀 값(P1, P2, P3, P4)과 거리 비율(a, 1-a)을 바탕으로 직선의 수식을 이용해서 계산



기하학 처리

❖ 보간

✓ 양선형 보간법(bilinear interpolation)

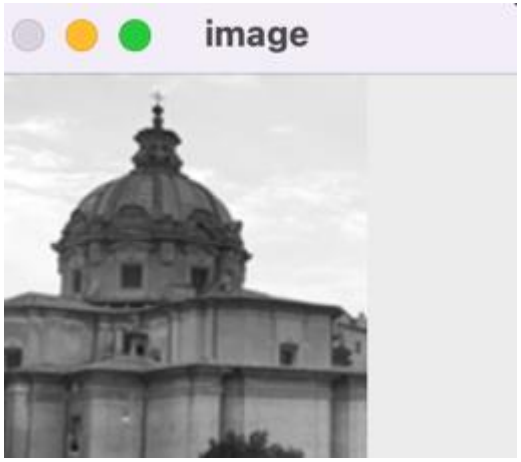
❑ OpenCV에서 제공하는 보간이 필요한 함수들은 다양한 보간 방법을 상수로 지원

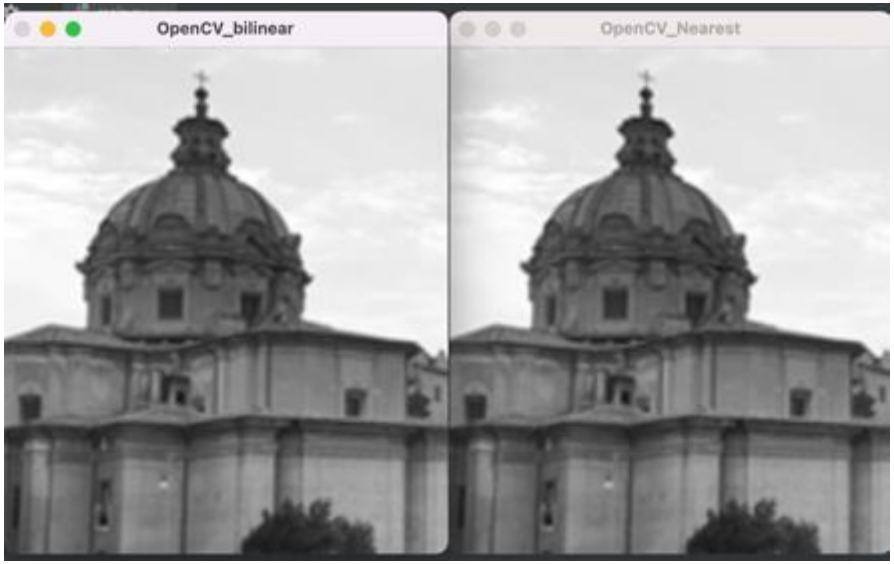
옵션 상수	값	설명
cv2.INTER_NEAREST	0	최근접 이웃 보간
cv2.INTER_LINEAR	1	양선형 보간 (기본값)
cv2.INTER_CUBIC	2	바이큐빅 보간 – 4×4 이웃 화소 이용
cv2.INTER_AREA	3	픽셀 영역의 관계로 리샘플링
cv2.INTER_LANCZOS4	4	Lanczos 보간 – 8×8 이웃 화소 이용

기하학 처리

❖ 보간

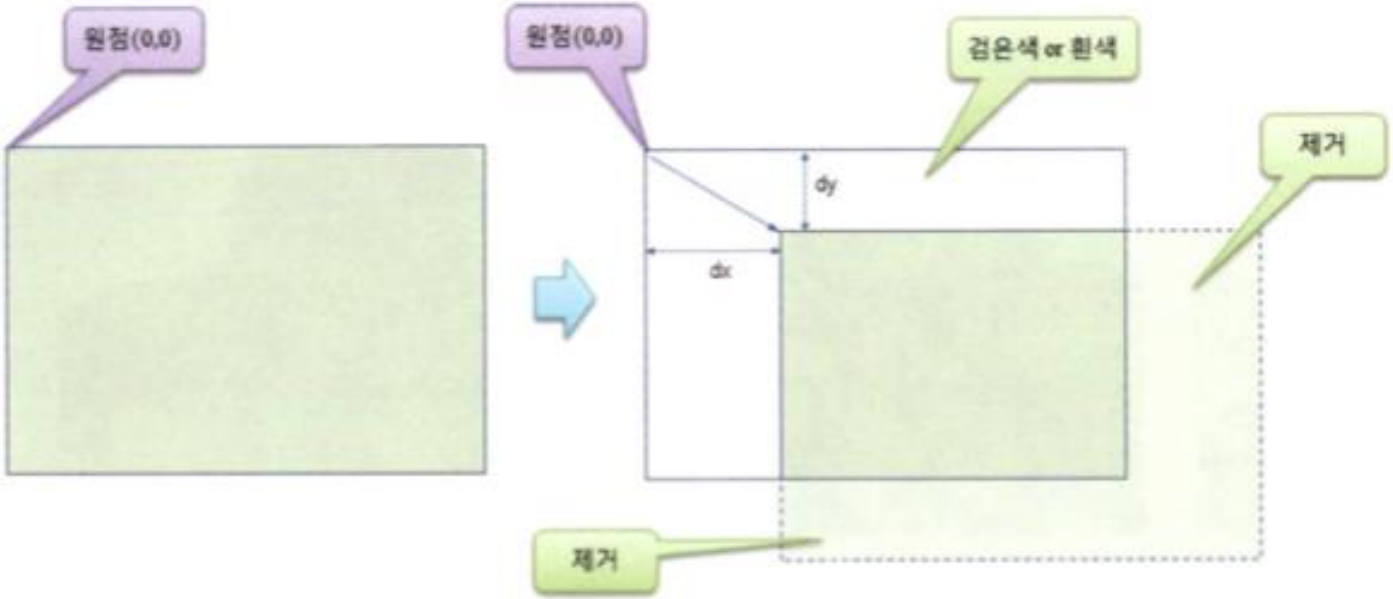
✓ 양선형 보간법(bilinear interpolation)





기하학 처리

- 평행 이동
  - ✓ 그래프에 좌표를 표시할 때와는 다르게 영상에서 원점 좌표는 기본적으로 최상단 왼쪽
  - ✓ 평행 이동(translation)은 영상의 원점을 기준으로 모든 픽셀을 동일하게 가로 방향과 세로 방향으로 옮기는 것



기하학 처리

❖ 평행 이동

기하학 처리

- ❖ 회전
  - ✓ 회전은 입력 영상의 모든 픽셀을 영상의 원점을 기준으로 원하는 각도만큼 모든 픽셀에 대해서 회전 변환을 시키는 것
  - ✓ 2차원 평면에서 회전 변환을 나타내는 행렬을 통해서 수식으로 표현
  - ✓ 회전 변환을 수행하는 행렬을 수식으로 나타낸 것으로 회전 변환의 역행렬이 sin() 함수의 부호만 다르기 때문에 순방향 사상과 역방향 사상도 단지 sin() 함수의 부호 만 차이가 남

순방향사상

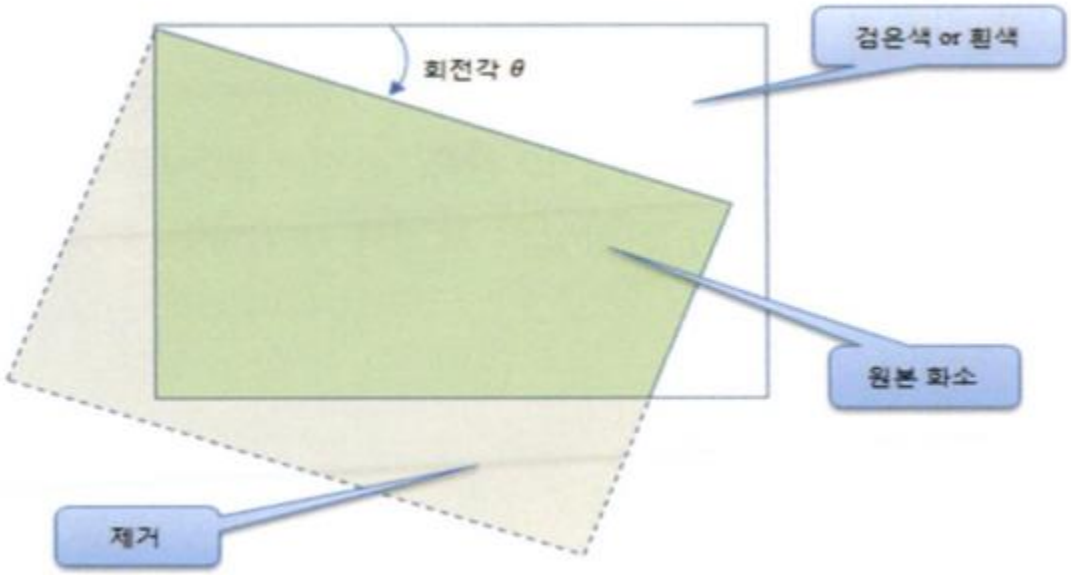
$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$
$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$

역방향사상

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

기하학 처리

- 회전
  - ✓ 목적 영상의 모든 픽셀(x′, y′)에 대해서 역방향 사상의 수식을 적용하여 입력 픽셀을 계산하면 그림과 같이 원점으로부터 시계 방향으로 정해진 각도만큼 회전된 영상이 생성
  - ✓ 직교 좌표계에서 회전 변환은 반시계 방향으로 적용되지만 영상 좌표계에서는 y 좌표가 하단으로 내려갈수록 증가하기 때문에 시계 방향 회전으로 표현됨에 유의
  - ✓ 평행 이동과 마찬가지로 목적 영상의 범위를 벗어나는 입력 픽셀은 제거되며 입력 영상에서 찾지 못하는 목적 픽셀은 검은색이나 흰색으로 지정



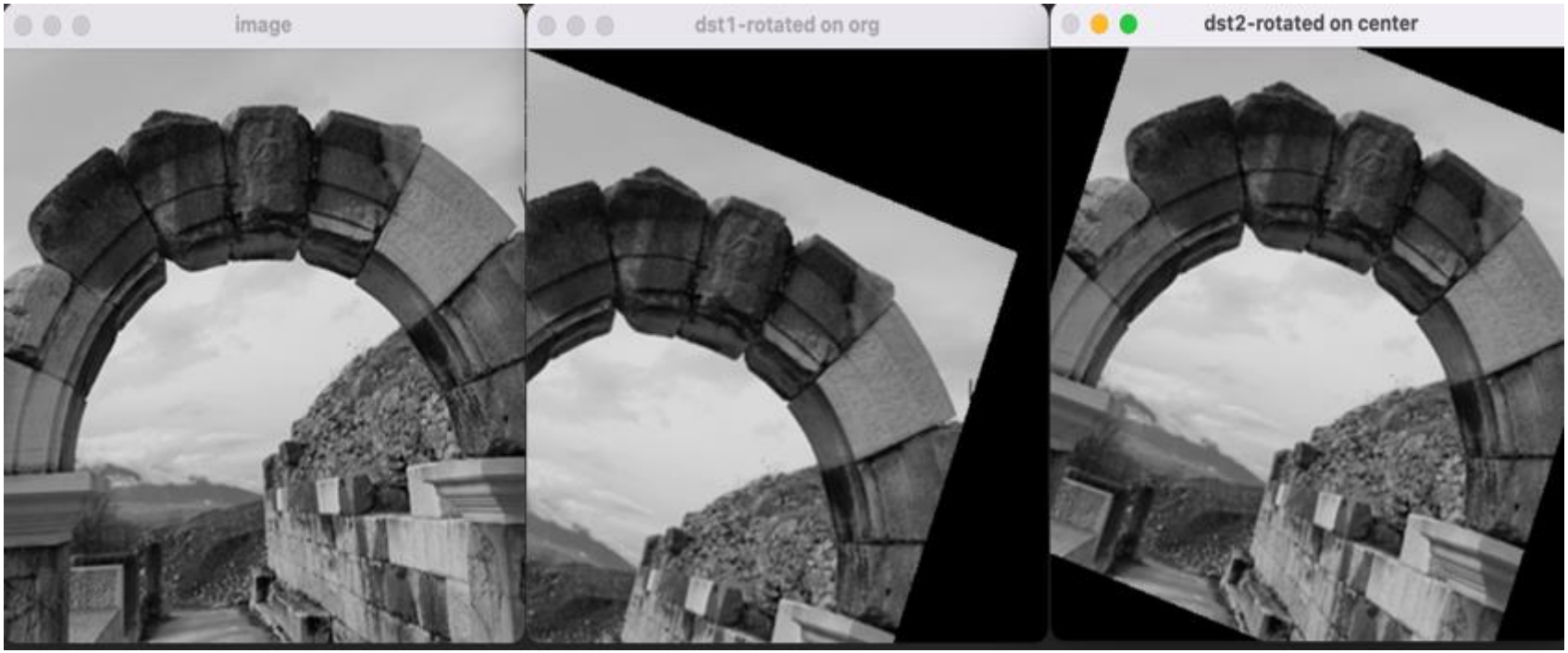
기하학 처리

- 회전
  - ✓ 일반적으로 영상을 회전시킬 때에는 회전의 기준을 영상의 기준 원점인 좌상단이 아닌 물체의 중심(center X, center Y)으로 하는 경우가 많은데 이 경우에는 다음과 같이 평행 이동의 수식을 포함하여 회전 변환을 수행
  - ✓ 영상을 원점으로 이동시킨 후 회전을 수행하고 다시 기준점 좌표로 이동

$$x = (x' - center\ X) \cdot \cos\theta + (y' - center\ Y) \cdot \sin\theta + center\ X$$
$$y = -(x' - center\ X) \cdot \sin\theta + (y' - center\ Y) \cdot \cos\theta + center\ Y$$

기하학 처리

❖ 회전



기하학 처리

- 어파인 변환
  - ✓ 기하학 변환들의 수식은 행렬식으로 표현이 가능
  - ✓ 기하학 변환 수식은 행렬의 곱으로 표현 가능
    - ❑ 회전

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

    - ❑ 크기 변경

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

    - ❑ 평행 이동

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

기하학 처리

- 어파인 변환
  - ✓ 회전과 크기 변경은 2 X 2 행렬로 표현이 가능하지만 덧셈이 포함된 평행 이동까지 나타내려면 2 X 3 행렬이 필요
  - ✓ 다음 수식과 같이 2 X 3 행렬로 변환 행렬을 구성하는 것을 어파인 변환(affine transform) 이라 함

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  - ✓ 어파인 변환은 변환 전과 변환 후의 두 어파인 공간 사이의 공선점을 보존하는 변환
  - ✓ 변환 전에 직선은 변환 후에도 그대로 직선이며 그 거리의 비율도 유지되며 변환 전에 평행선도 변환 후에 평행선이 됨

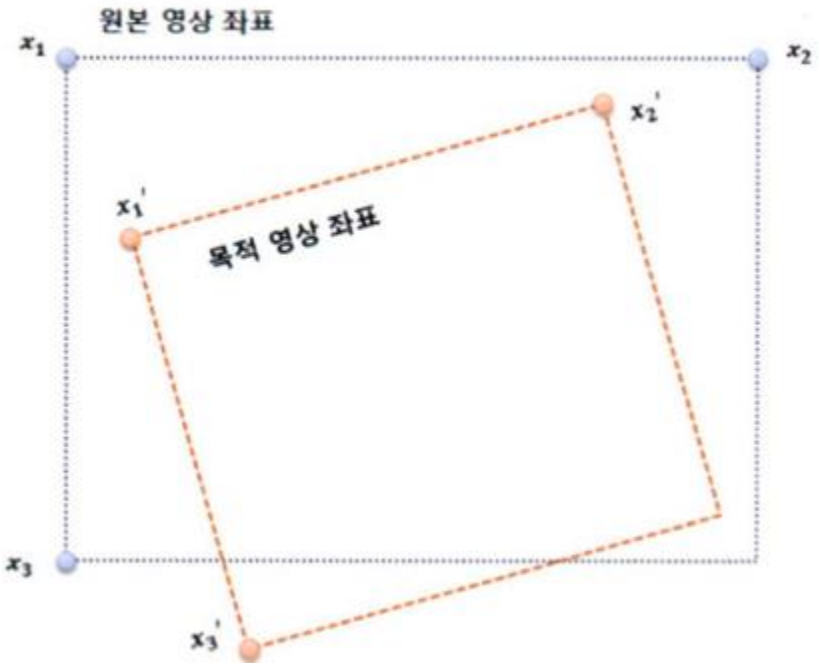
기하학 처리

- ❖ 어파인 변환
  - ✓ 어파인 변환을 수행하는 방법
    - ❑ 회전 각도, 크기 변경 비율, 평행 이동의 정도를 지정해서 각각 변환 행렬을 구성하고 각 변환 행렬을 행렬 곱으로 구성하면 하나의 변환 행렬을 만들 수 있는데 각 행렬을 곱하는 순서는 변환하고자는 방식에 따라서 달라질 수 있으며 이때 2 X 3 크기의 어파인 행렬로 구성하면 행렬의 곱을 계산할 수 없기 때문에 다음 수식과 같이 3 X 3 크기의 행렬로 구성하여 행렬 곱을 수행

$$\text{어파인 변환행렬} = \begin{bmatrix} \cos\theta & -\sin'\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

기하학 처리

- ❖ 어파인 변환
  - ✓ 어파인 변환을 수행하는 방법
    - ❑ 그림과 같이 변환 전인 입력 영상의 좌표 3개(x_1, x_2, x_3) 와 변환이 완료된 목적 영상에서 상응하는 좌표 3개(x_1', x_2', x3')를 알면 두 좌표(x -> x') 사이를 변환해주는 어파인 변환 행렬을 구할 수 있는데 이렇게 행렬의 곱으로 기하학 변환을 적용하면 단순하면서도 쉽게 입력 영상에 대한 변환이 가능



기하학 처리

- 어파인 변환
  - ✓ OpenCV에서도 어파인 변환을 수행할 수 있는 cv2.warpAffine() 함수를 제공하는데 이 함수는 지정된 어파인 변환 행렬을 적용하면 입력 영상에 어파인 변환을 수행한 목적 영상을 리턴
  - ✓ cv2.getAffineTransform() 은 변환 전의 좌표 3개와 변환 후의 좌표 3개를 지정하면 해당 변환을 수행해 줄 수 있는 어파인 행렬을 반환
  - ✓ cv2.getRotationMatrix2D()는 회전 변환과 크기 변경을 수행하는 어파인 행렬을 리턴하며 여기서 회전의 방향은 양수일 때 반시계 방향으로 회전하는 행렬을 반환하는데 이것은 영상 좌표에서 직교 좌표계에서의 회전과 같은 방향으로 표현하기 위함

기하학 처리

❖ 어파인 변환

함수 및 인수 구조		
cv2.warpAffine (src, M, dsize [, dst [, flags [, borderMode [, borderValue]]]]) → dst ■ 설명: 입력 영상에 어파인 변환을 수행해서 반환한다.		
인수 설명	■ src	입력 영상
	■ dst	반환 영상
	■ M	어파인 변환 행렬
	■ dsize	반환 영상의 크기
	■ flags	보간 방법
	■ borderMode	경계지정 방법
cv2.getAffineTransform(src, dst) → retval ■ 설명: 3개의 좌표쌍을 입력하면 어파인 변환 행렬을 반환한다.		
인수 설명	■ src	입력 영상 좌표 3개 (행렬로 구성)
	■ dst	목적 영상 좌표 3개 (행렬로 구성)
cv2.getRotationMatrix2D(center, angle, scale) → retval ■ 설명: 회전 변환과 크기 변경을 수행할 수 있는 어파인 행렬을 반환한다.		
인수 설명	■ center	회전의 중심점
	■ angle	회전각도, 양수 각도가 반시계 방향 회전 수행
	■ scale	변경할 크기
cv2.invertAffineTransform(M [, iM]) → iM ■ 설명: 어파인 변환 행렬의 역 행렬을 반환한다.		
인수 설명	■ M	어파인 변환 행렬
	■ iM	어파인 역변환 행렬

기하학 처리

❖ 어파인 변환