컴퓨터 그래픽스
3. 2차원 그래픽스의 변환

2025년 2학기

학습 내용

- **2차원 그래픽스 변환**
  - 기본 변환: 이동, 회전, 신축
  - 그 외, 반사, 밀림

기본 기하 변환

- **기본 기하 변환**
  - 이동 (Translation)
  - 회전 (Rotation)
  - 신축 (Scale, 크기 변환)

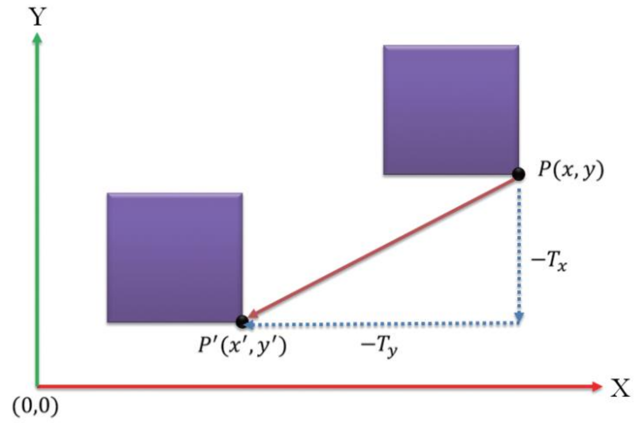
  - 동차좌표계를 이용하여 변환 적용 (2차원 → 3x3 행렬 사용)

기본 기하 변환: 이동

- Translation (이동)
  - 좌표계의 한 곳에서 다른 곳으로 직선 경로를 따라 객체의 위치를 바꾼다.
  - 객체의 크기나 모양, 방향 등은 바뀌지 않는다.
  - P(x, y) → P'(x', y')

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

    (t_x, t_y): 이동 벡터 (translation vector)



행렬을 사용하면

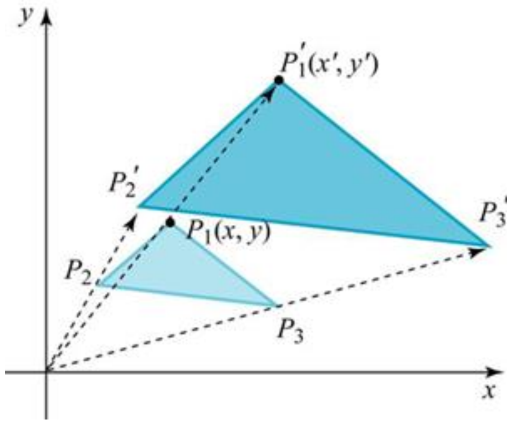
P' =

기본 기하 변환: 신축

- Scaling (신축, 확대/축소)
  - 객체의 크기를 확대/축소 시킨다.
  - 객체의 크기뿐 아니라 기준점 (원점)으로부터의 위치도 배율에 따라 변한다.
  - P(x, y) → P'(x', y')

$x' = s_x \cdot x$ $y' = s_y \cdot y$

(s_x, s_y): 신축률 (scaling factor)



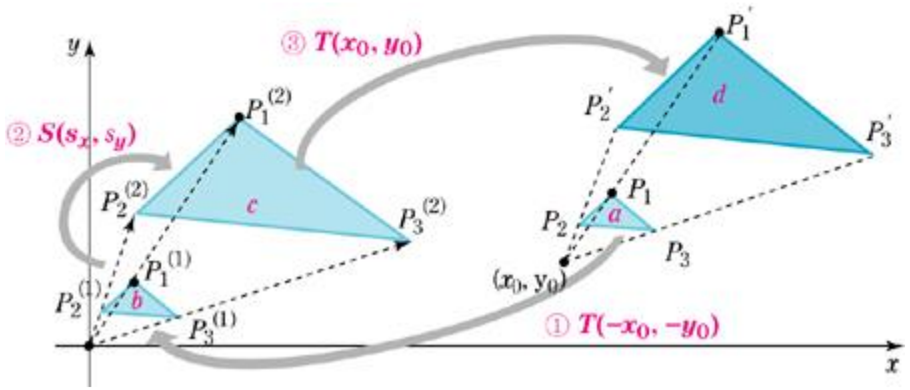
행렬을 사용하면

P' =

- s > 1:
- s = 1:
- 0 < s < 1:
- s < 0:

기본 기하 변환: 신축

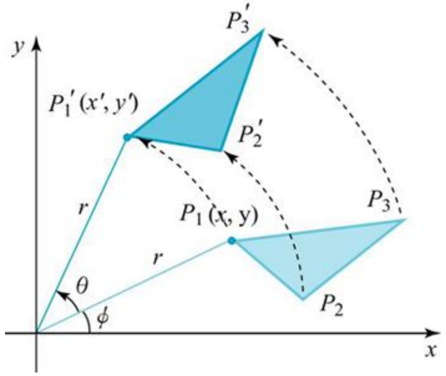
- 임의의 점 (x_0, y_0)에 대하여 신축률 (s_x, s_y)만큼 신축
  - 신축 기준점을 원점이 되도록 객체를 이동: $T(-x_0, -y_0)$
  - 원점에 대하여 신축: $S(s_x, s_y)$
  - 제자리로 이동: $T(x_0, y_0)$



기본 기하 변환: 회전

- Rotation (회전)
  - xy평면에서 원 경로를 따라 객체를 재배치
  - 객체의 모양 변화는 없이 객체가 놓여있는 방향이 변한다.
    - x = rcos Φ, y = rsin Φ
  - P(x, y) → P'(x', y')
    - x' = rcos(Φ+θ) = <u>rcosΦ</u>cosθ – <u>rsinΦ</u>sinθ = xcosθ - ysinθ
    - y' = rsin(Φ+θ) = <u>rcosΦ</u>sinθ + <u>rsinΦ</u>cosθ = xsinθ + ycosθ

회전각 : θ, 회전점 (Pivot Point): (x_r, y_r)

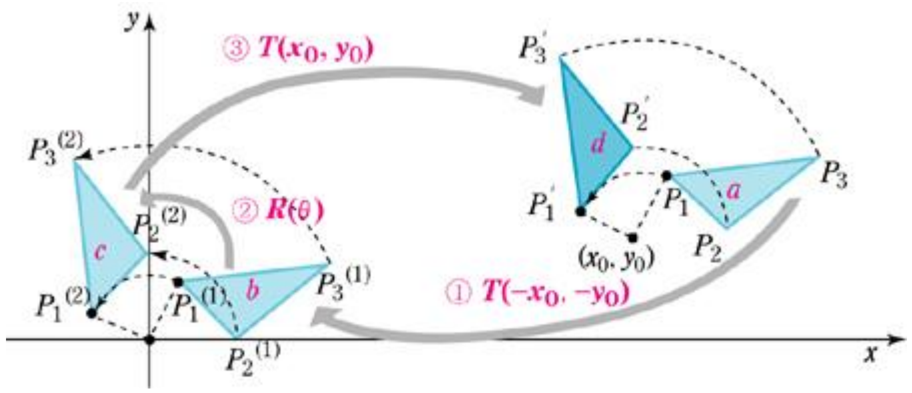


  - 행렬을 사용하면

P' =

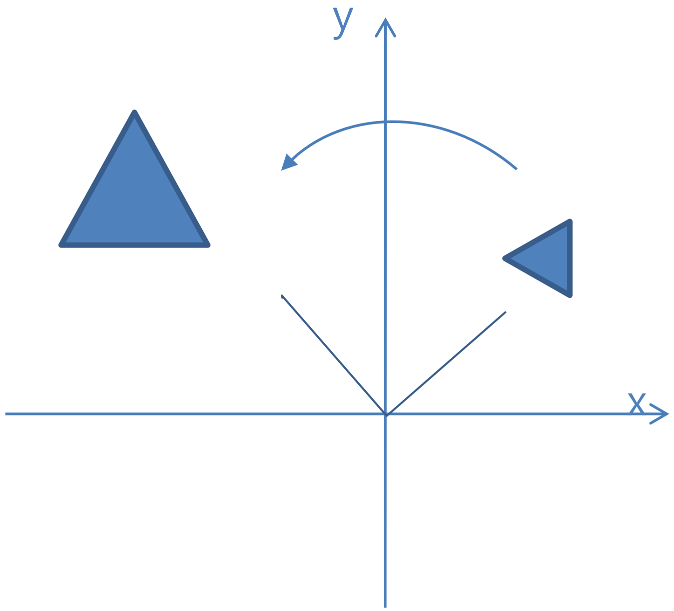
기본 기하 변환: 회전

- 임의의 점 (x_0, y_0)에 대하여 θ만큼 회전
  - 회전 중심점이 원점이 되도록 객체를 이동: $T(-x_0, -y_0)$
  - 원점을 중심으로 θ만큼 회전: $R(\theta)$
  - 반대 방향으로 이동: $T(x_0, y_0)$



기본 기하 변환

- **기하 변환 예)**



- (0, 1)에 대하여 시계반대방향으로 90도 회전 후,
- 2배 스케일
- 삼각형 좌표값 (5, 7) (10, 5) (8, 10)
  → 변환 좌표값은?

동차 좌표계 (Homogeneous Coordinate System)

- **동차 좌표계**
  - n차원의 공간을 n+1의 좌표로 표현하는 좌표계
  - 여러 단계의 변환 행렬을 하나로 결합하여 표현할 수 있게 한다.
  - 순차적인 기하 변환을 처리할 때 각 단계별 좌표 값을 구하지 않고 바로 계산 하려면 행렬의 합(A)을 제거해야 한다.
    - $P_2 = M \cdot P_1 + A_1$
    - $P_3 = M_2P_2 + A_2 = M_2(M_1P_1 + A_1) + A_2 \quad = M_2M_1P_1 + M_2A_1 + A_2$
  - 동차 좌표계를 이용하여 기본 변환을 행렬 곱으로만 표현한다 → 변환을 간단히 처리, 계산량을 줄일 수 있다. 즉,

$$
\begin{aligned}
P_n &= M_{n-1} \cdot P_{n-1} = M_{n-1} \cdot M_{n-2} \cdot P_{n-2} = \dots \\
&= M_{n-1} \cdot M_{n-2} \cdot \dots \cdot M_1 \cdot P_1 \\
&= M \cdot P_1
\end{aligned}
$$

  - 2차원의 데이터 v = (v1, v2)가 있을 때, 이 데이터의 동차 좌표는 V’ = (v1, v2, p)
    - p = 0 -> v’는 벡터
    - p $\neq$ 0 -> v’는 점
    - 동차 좌표계에서 다음의 점들은 모두 같은 점이다.
      - (5, 1, 1) = (10, 2, 2) = (15, 3, 3) = (20, 4, 4)⋯

동차 좌표계 (Homogeneous Coordinate System)

- **행렬식**
  - 2차원의 점 P(x, y)를 동차 좌표계로 표현하면 차원이 하나 증가된다.
    - 즉, P(h·x, h·y, h) (h $\neq$0), h는 임의의 값
    - 한 점은 동차 좌표계에서 h의 값에 따라 여러 개의 좌표로 표현될 수 있다.
    - 동차 좌표계에서 한 점의 좌표 P(X, Y, h)로 주어지면 → 2차원 기하 평면에서 P($\frac{X}{h}$, $\frac{Y}{h}$)로 대응된다.
    - P(x_1, y_1, h_1), P(x_2, y_2, h_2)로 주어질 때, ($\frac{x1}{h1}$, $\frac{y1}{h1}$) = ($\frac{x2}{h2}$, $\frac{y2}{h2}$) 이면 2차원 기하 평면에서 동일한 점이 된다.
    - h = 1 이면 (x, y) → (x, y, 1)

  - 이동의 3차원 행렬: T(t_x, t_y) = $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

  - 회전의 3차원 행렬: R(θ) = $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

  - 신축의 3차원 행렬: S(s_x, s_y) = $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

동차 좌표계 (Homogeneous Coordinate System)

- 연속된 기본 변환을 동차 좌표계를 사용하여 하나의 행렬로 나타낼 수 있다.
  - 하나의 변환 행렬로 표현한 합성 변환에서는 한번의 행렬 곱셈만 필요하다

  - 이동: 연속적으로 2번 이동하는 경우

    $P' = T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) \cdot P = T(t_{x2}+t_{x1}, t_{y2}+t_{y1}) \cdot P$
  - 신축: 연속적으로 2번 신축 하는 경우

    $P' = S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) \cdot P = S(s_{x2} \cdot s_{x1}, s_{y2} \cdot s_{y1}) \cdot P$
  - 회전 : 연속적으로 2번 회전하는 경우

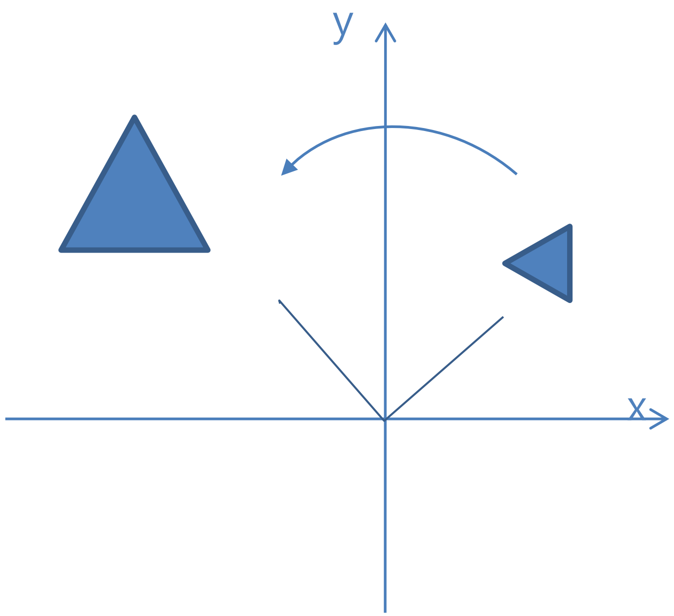
    $P' = R(\theta_2) \cdot R(\theta_1) \cdot P = R(\theta_2 + \theta_1) \cdot P$

  - 임의의 점 (x_0, y_0)에 대하여 신축 (s_x, s_y)하는 경우
    - $P' = T(x_0, y_0) \cdot S(s_x, s_y) \cdot T(-x_0, -y_0) \cdot P$

  - 행렬의 곱셈에서 결합법칙은 성립한다.
    - $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
  - 교환 법칙은 성립하지 않는다.
    - $A \cdot B \neq B \cdot A$
  - 단, 같은 종류의 기하변환에서 교환 법칙은 성립한다.

기본 기하 변환

- **기하 변환 예) 동차좌표계 적용하여**

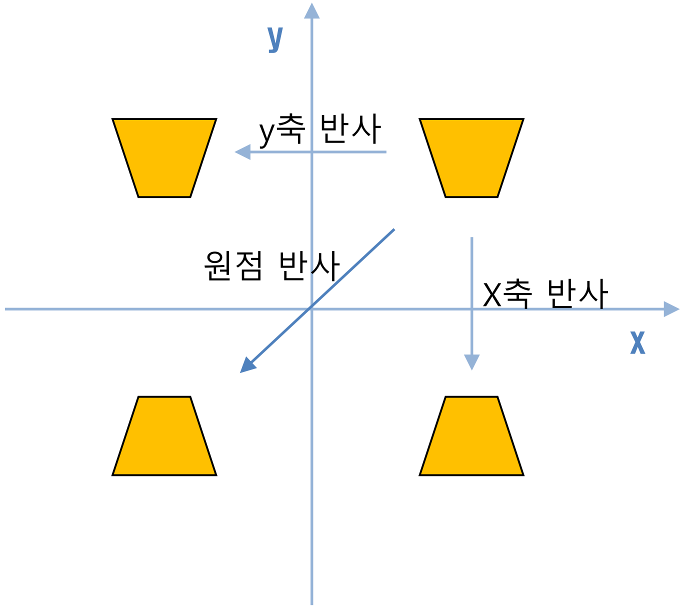


- (0, 1)에 대하여 시계반대방향으로 90도 회전 후,
- 2배 스케일
- 삼각형 좌표값 (5, 7) (10, 5) (8, 10)

  → 변환 좌표값은?

기타 기하 변환: 반사

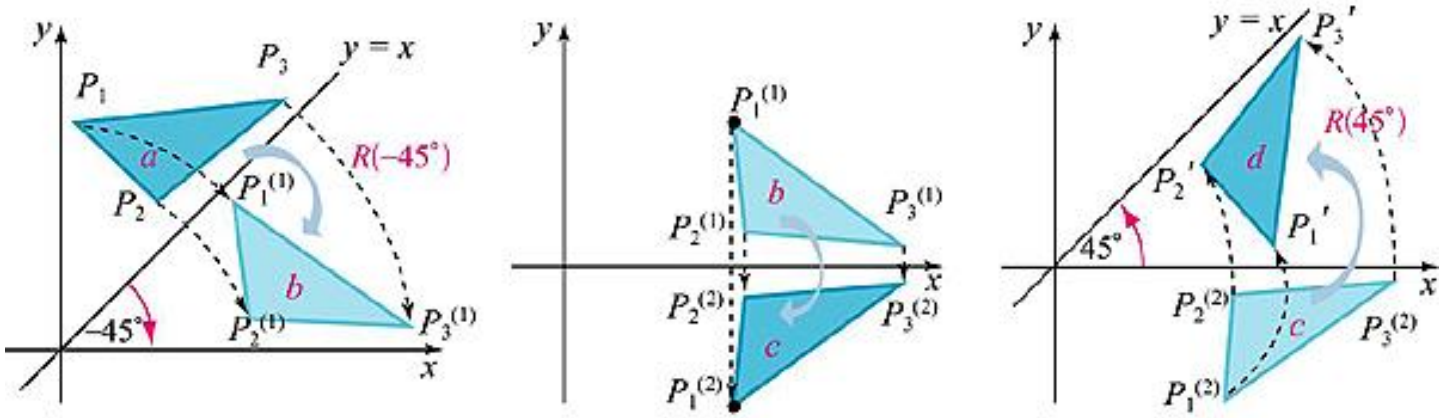
- Reflection (반사): 거울 영상
  - 고정점에 대하여 객체가 반대방향으로 변환되는 것
  - y=0 (x축)에 대하여 반사
    - y 좌표값 부호 변경
  - x=0 (y축)에 대하여 반사
    - x 좌표값 부호 변경
  - 원점 (0,0)에 대하여 반사
    - 모두 변경



기타 기하 변환: 반사

- y = x에 대한 반사
  - 시계방향으로 45’ 회전
  - X축에 대하여 반사
  - 시계 반대 방향으로 45’ 회전

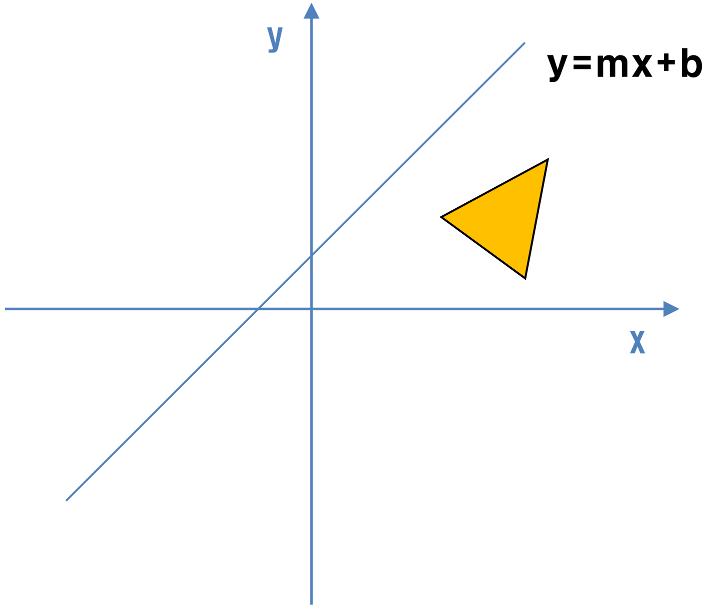
- y = -x에 대한 반사



(a) 다각형과 반사축을 회전 (b) 반사축을 기준으로 반사 (c) 반대방향으로 회전

기타 기하 변환: 반사

- $y = mx + b$에 대하여 반사



- 예) $y = x + 2$ 직선에 대하여 선분 (0, 5) (8, 2) 반사

기타 기하 변환: 밀림

- **밀림 (Shearing)**
  - 2차원 평면상에서 객체의 한 부분을 고정시키고 다른 부분을 밀어서 생기는 변환
    - 고정된 지점에서 멀수록 밀리는 거리가 커진다. (고정된 지점과의 거리에 비례하여 밀리는 경우가 결정된다)
    - x축에 대한 밀림:

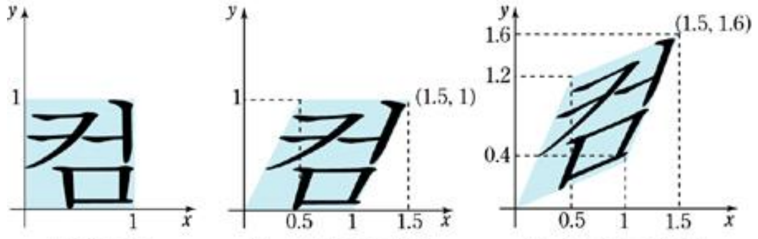
      $x' = x + h_x \cdot y$ (h_x: 밀림 비율)

      $y' = y$
    - y축에 대한 밀림:

      $x' = x$

      $y' = y + h_y \cdot x$ (h_y :밀림 비율)
  - 행렬을 사용하면,

    $P' =$

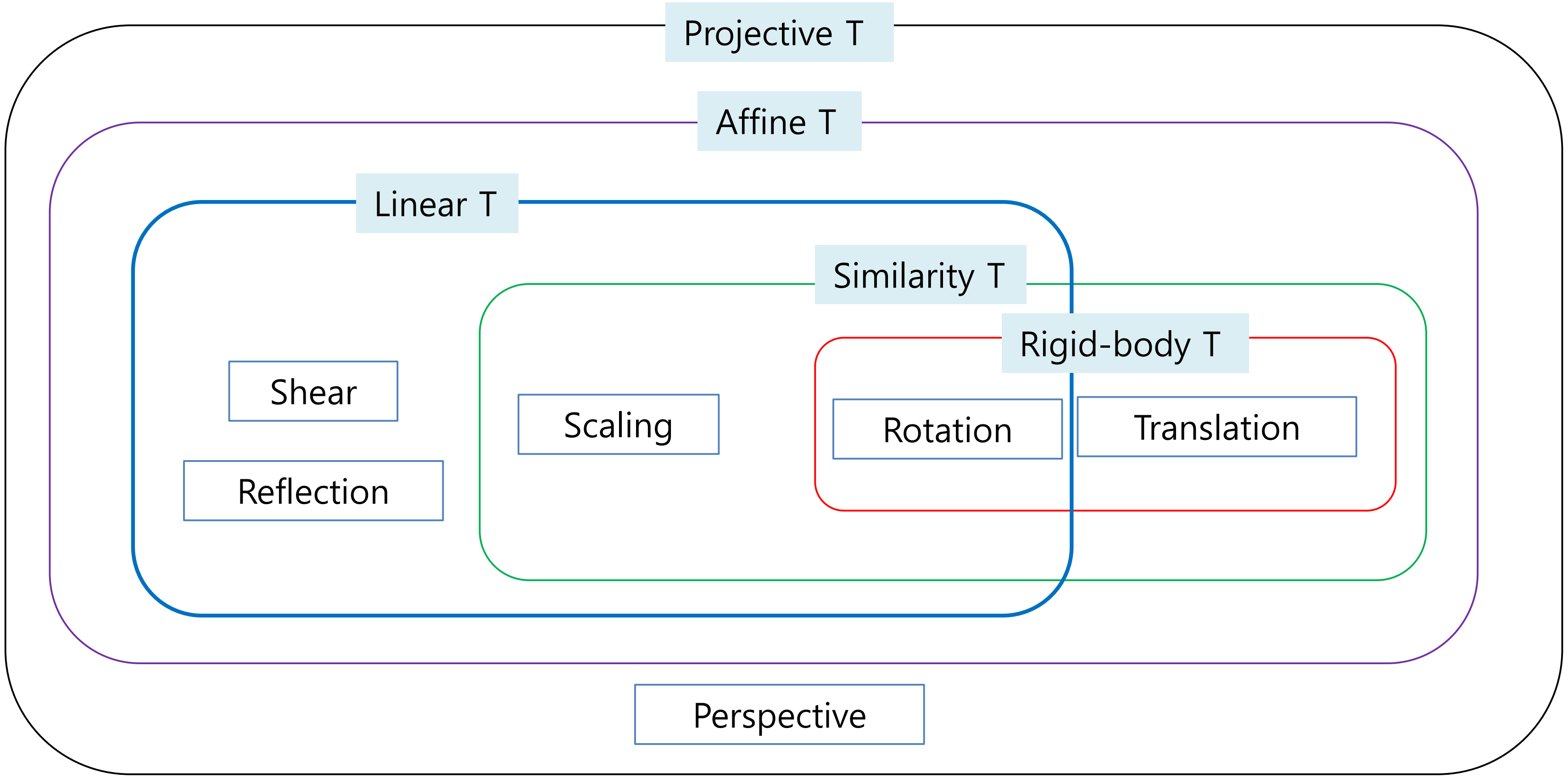


(a) 원래 글자 (b) *x*방향으로 0.5 밀림 (c) *y*방향으로 0.4 밀림

변환 유형

- **강체 변환 (Rigid-body Transformation): 크기와 각도 보전**
  - 이동, 회전
- **유사 변환 (Similarity Transformation): 모양이 유사한 변환**
  - 이동, 회전, 신축
- **선형 변환 (Linear Transformation): 선형성을 보존하는 변환**
  - 선형성: 벡터 공간에서 다음의 연산성질을 보존하며 사상하는 변환
    - $T(u+v) = T(u) + T(v)$
    - $T(cu) = cT(u)$
  - 회전, 신축, 반사, 밀림
  - 이동 변환은 선형 변환이 아님
- **어파인 변환(Affine Transformation): 직선, 길이의 비, 평행성을 보존하는 변환**
  - 이동, 회전, 신축, 반사, 밀림
- **원근 변환 (Projective Transformation): 원근감을 표시하는 변환**
  - 투영 변환

변환 유형


Projective T
Affine T
Linear T
Similarity T
Rigid-body T
Shear
Reflection
Scaling
Rotation
Translation
Perspective

이번주에는

- **2차원 그래픽스 변환**
  - 기본 변환: 이동, 회전, 신축
  - 기타 변환: 반사, 밀림

- **다음 시간에는**
  - 윈도우와 뷰포트
  - 클리핑