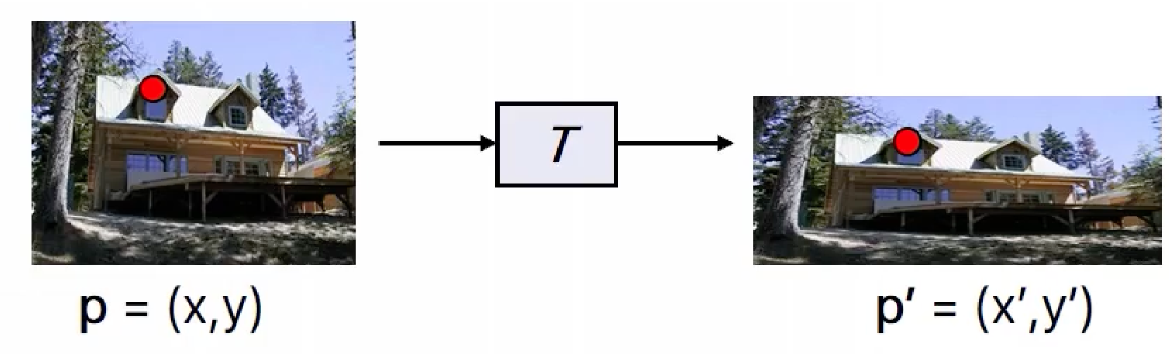
# 2D transformations

## 학습목표

2D transformation(Image transformation)에 대해 학습한다

# 2D Transform(Global Transform, Parametric Transform)




Transformation T is a coordinate-changing machine: $p\prime = T(p)$

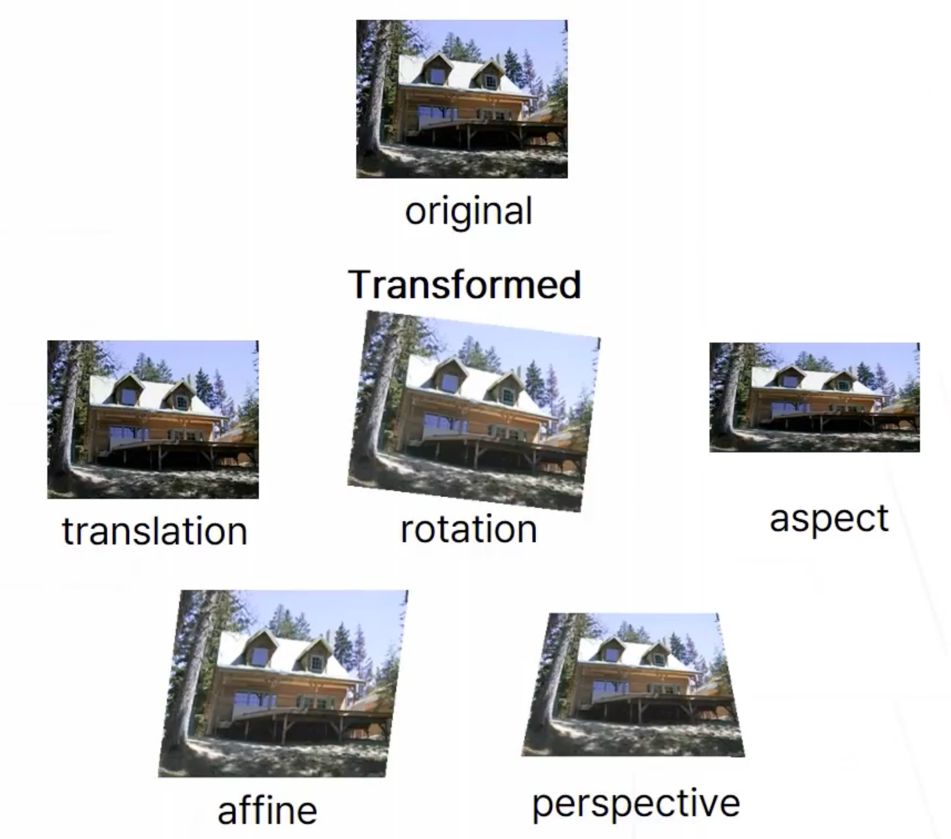
what does it mean that T is global?

- Is the same for any point p
- can be describe by just a few numbers(parameters)

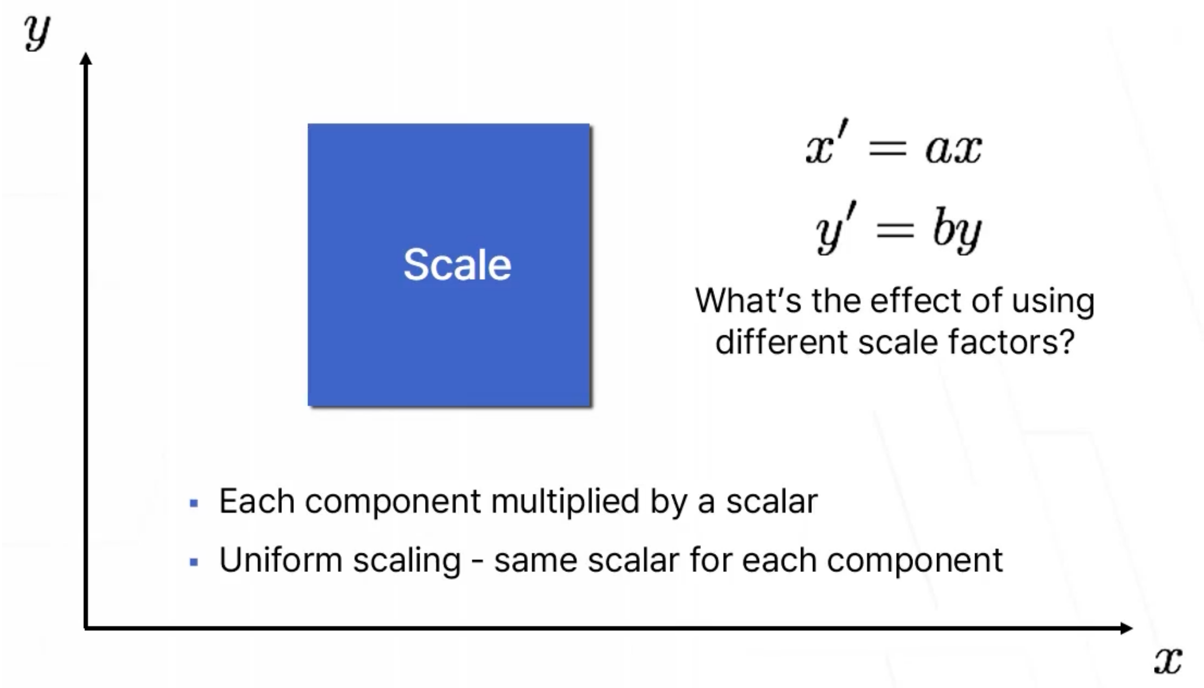
For linear trasformations, we can represent T as a matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix}$$

심플한 Transformation 기법



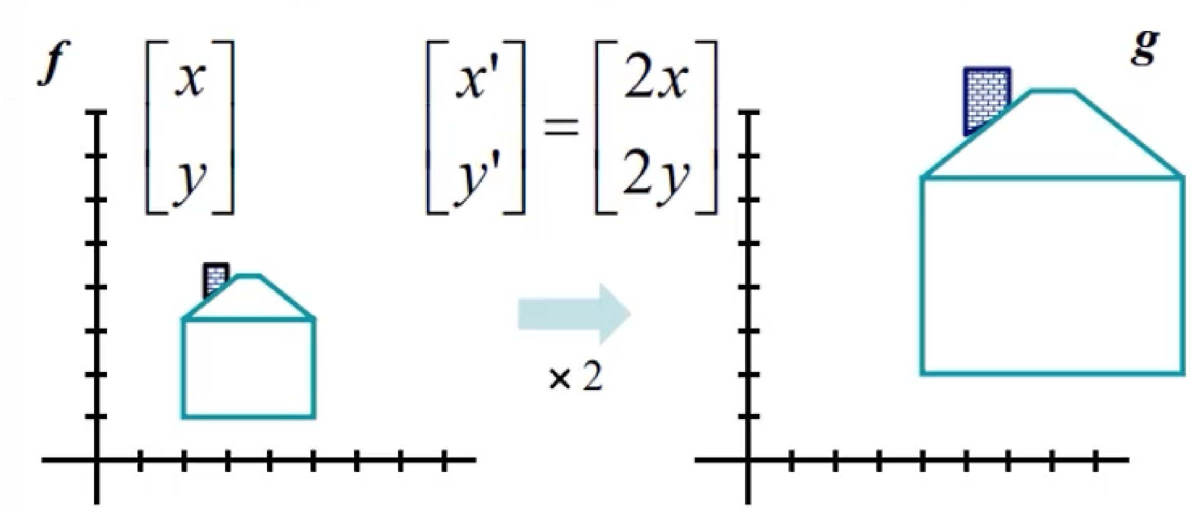

# 2D planar trasformations




키우고자 하는 비율, 크기로 a, b를 x, y좌표에 곱해줌 이걸 행렬과 백터의 배열로 나타낼 수 있음

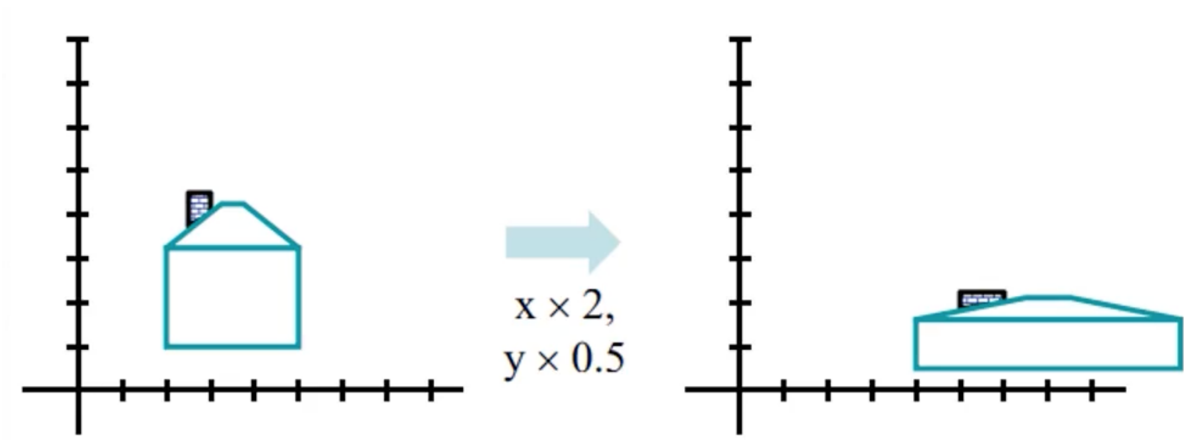
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix S}} \begin{bmatrix} x \\ y \end{bmatrix}$$

### Uniform scaling



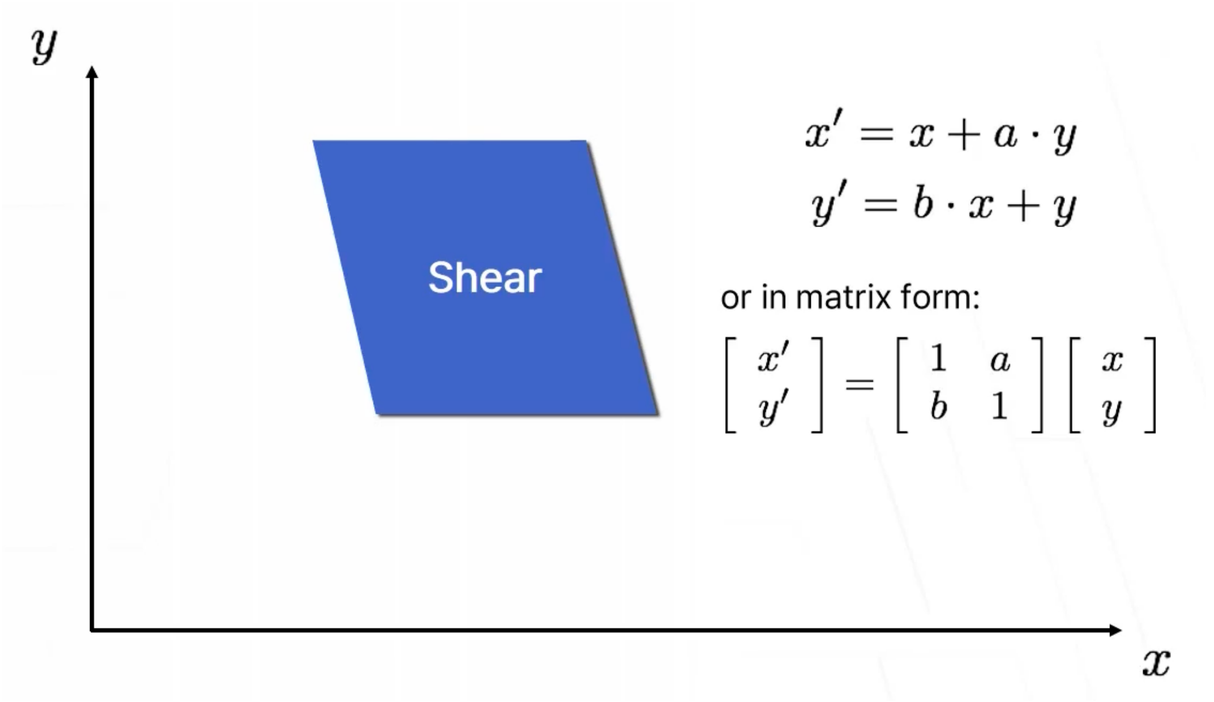

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

**Non-uniform scaling: differnt scalars per compoent**

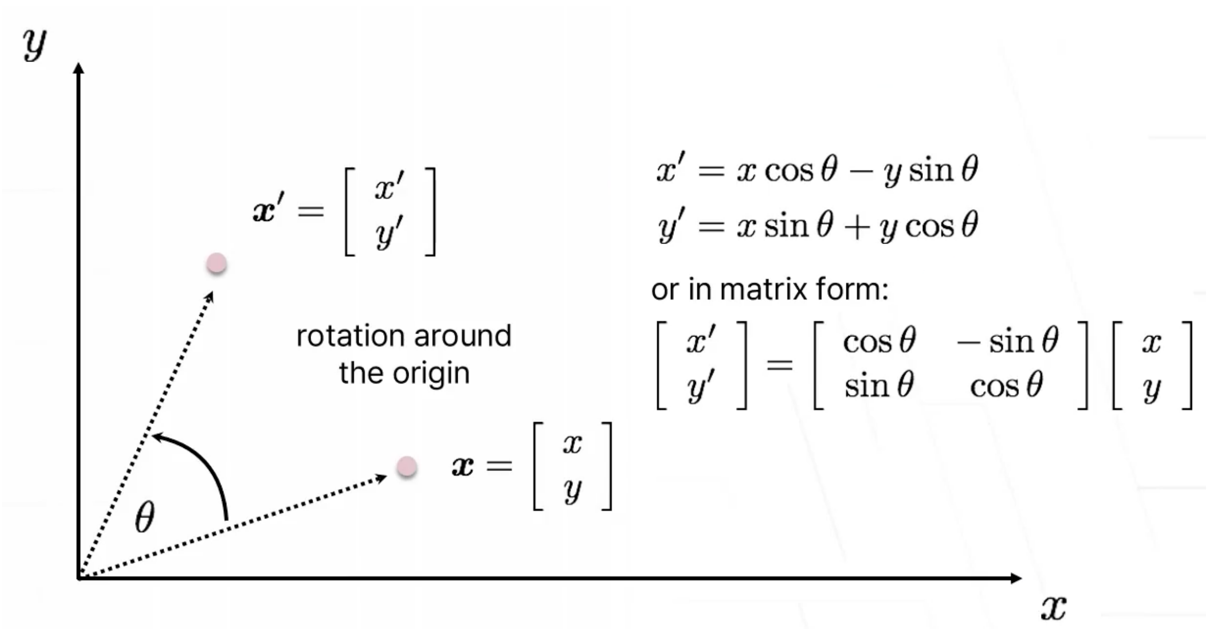



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ 0.5y \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

**Shear**




## Rotation




몇가지를 알아본 결과

$$x' = f(x; p)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = M \begin{bmatrix} x \\ y \end{bmatrix}$$

parameters $p$　　　point $x$

우리는 이 M을 수정해서 수정하면 됨

- Scale

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

- Flip across y

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Rotate

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

- Flip across origin

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Shear

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

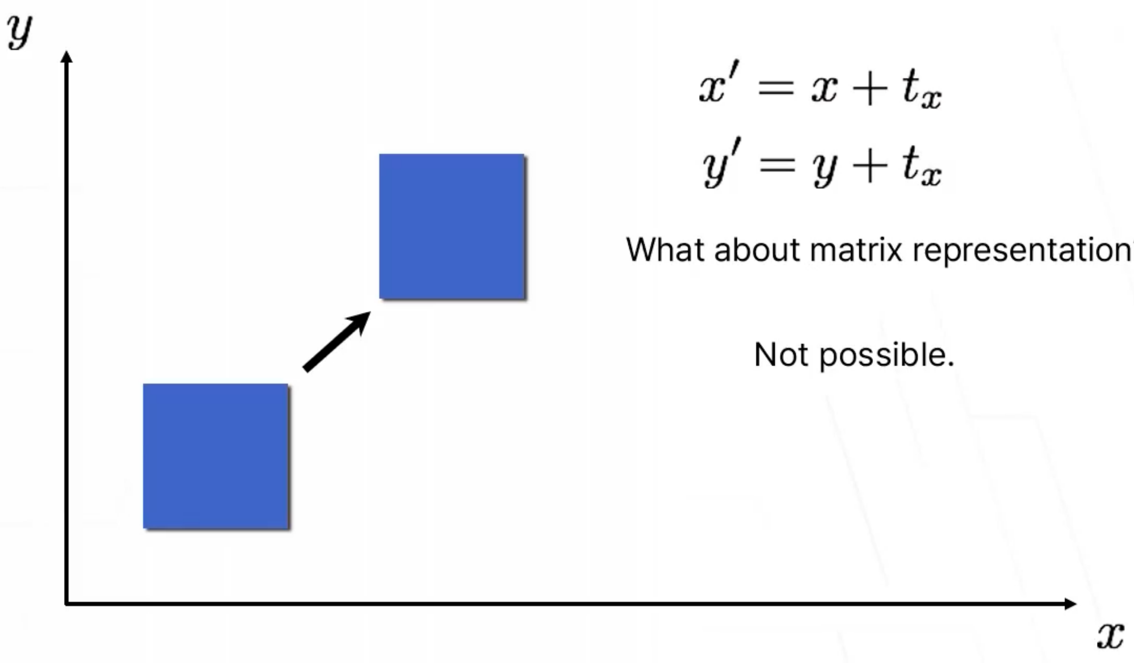
- Identity

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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix} \quad \text{linear}$$

다 선형 조합으로 표현이 됨

2D translation은 선형적으로 표현이 안됨

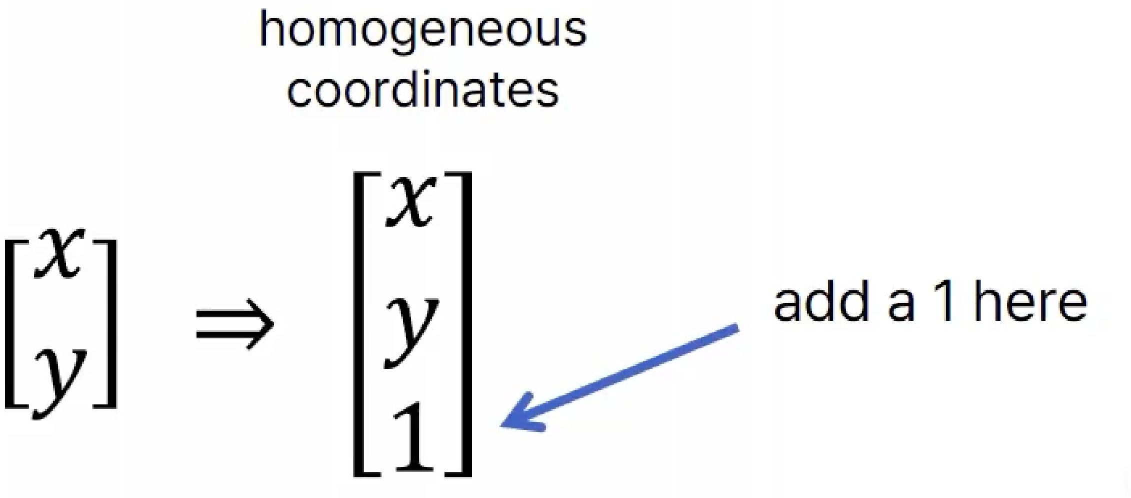



$$x' = x + t_x$$
$$y' = y + t_x$$

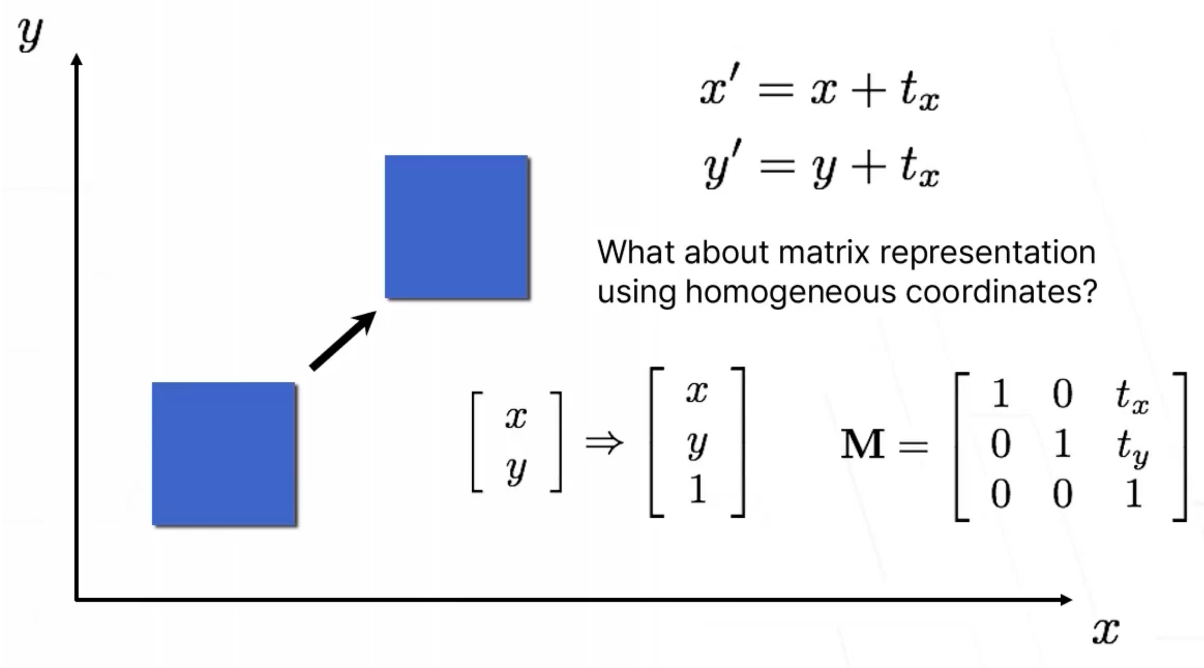
What about matrix representation?

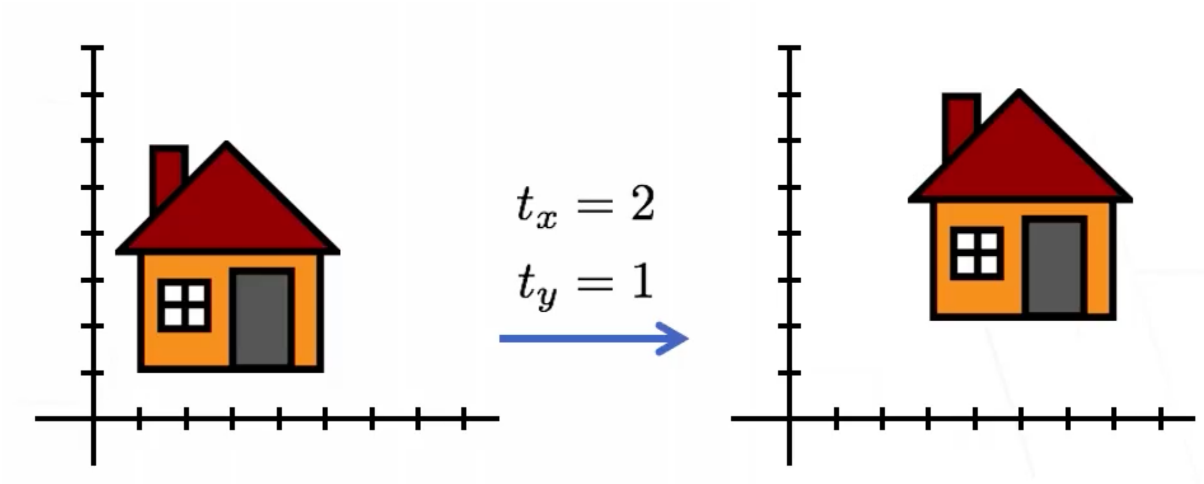
Not possible.

Homogeneous corrdinates로 표현 가능




Represent 2D point with a 3D vector






translation 혼자 차원이 다르니깐 translation을 기준으로 차원을 맞춰서 표현 하면 됨

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

translation

$$\begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_x & 0 & 0 \\ 0 & \mathbf{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \beta_x & 0 \\ \beta_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

shearing

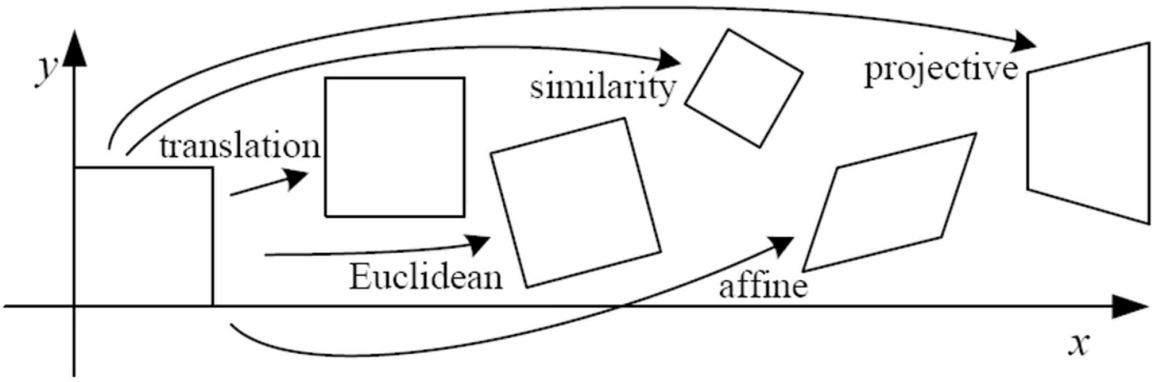
이렇게 동시에 표현이 가능하겠지

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left( \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$\mathbf{p}'$ = translation($t_x$,$t_y$) rotation($\theta$) scale(s,s) $\mathbf{p}$

matrix에서 $A \times B \neq B \times A$ 니깐 순서에 따라 결과가 달라짐

그리고 속도 계산 측면에서 $M_1, M_2, M_3$ (translation, rotation, scale)를 미리 행렬곱 한 $M^*$으로 곱해주는게 더 빠를거임




이렇게 여러개의 매트릭스를 걸쳐 다양한 변환을 할 수 있음

Euclidean (rigid):
rotation + translation

$$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

Euclidean이라는 힌트를 몰랐으면 이렇게 파라미터가 6개로 알았어야 했는데

Euclidean (rigid):
rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

알게 되면 이렇게 파라미터 3개로 표현 가능함(각도, x축 이동, y축 이동)

이때 $r_3, r_6$을 바로 x, y축으로 이동할 값이라고 생각하면 안됨 - 행렬곱이니

similarity는 rotation 부분에 하나의 선형적 부분 추가해줘서 파라미터 4개

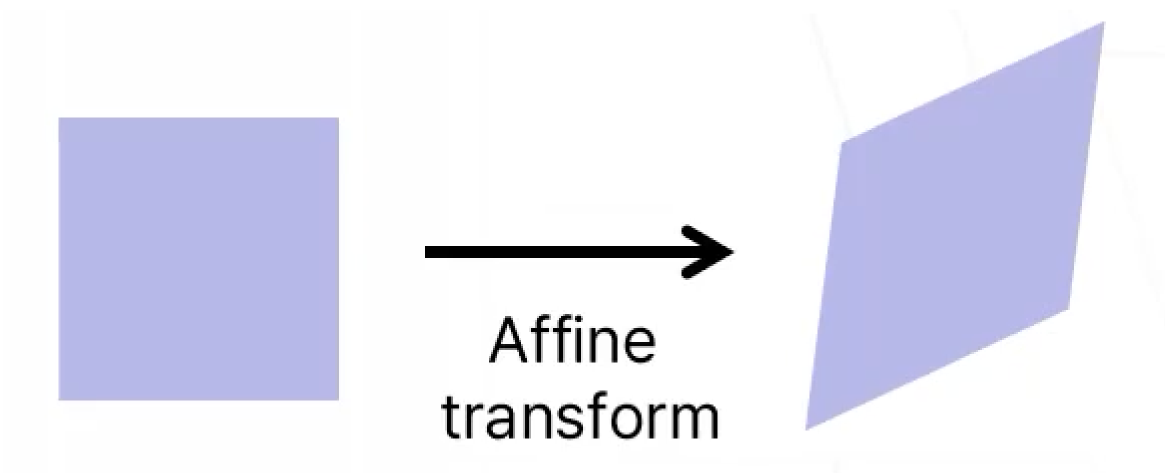
affine는 6개(간소화 할수가 없어서) 이런식으로..

Affine transform:
uniform scaling + shearing
+ rotation + translation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

저걸 다써야 Affine이 아니라 rotation 하나만 해도 affine이라고도 말할 수 있음 (가장 큰 범위)

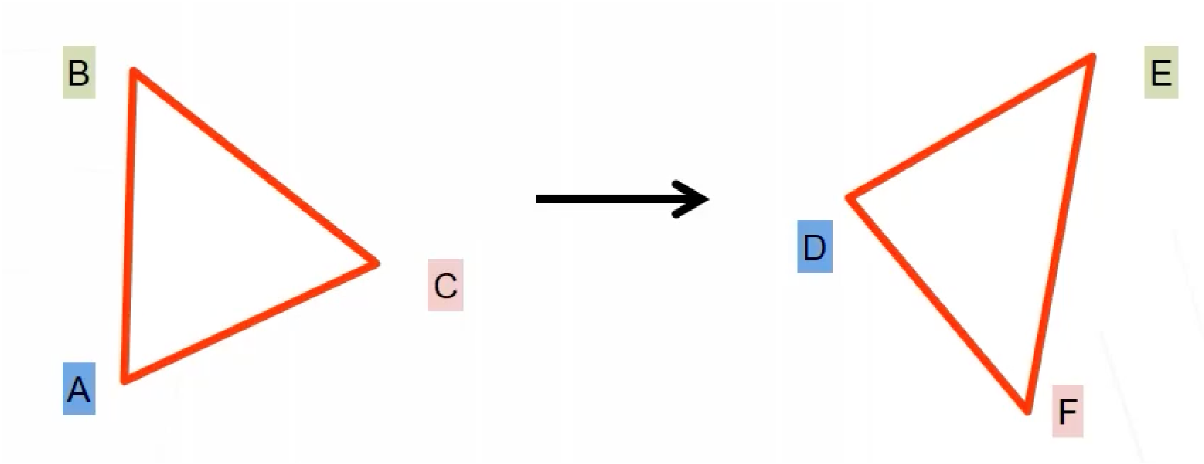



## Linear transformations(scale, rotation, shear) and Translations

$$
\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
$$

이제 이 값들(M)을 어떻게 채울까..

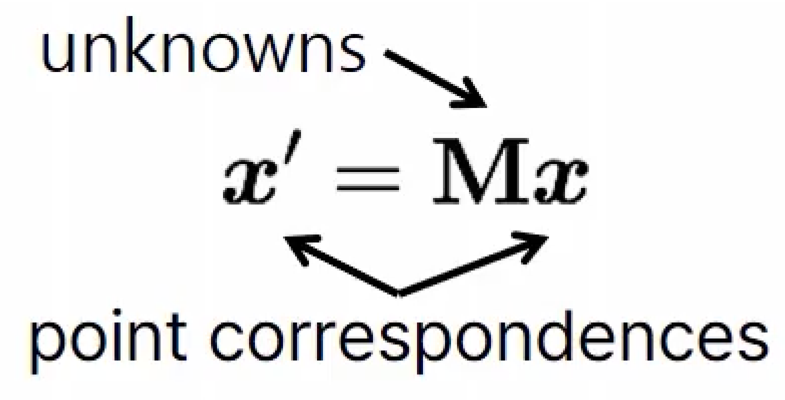
## Determining unknown transformations



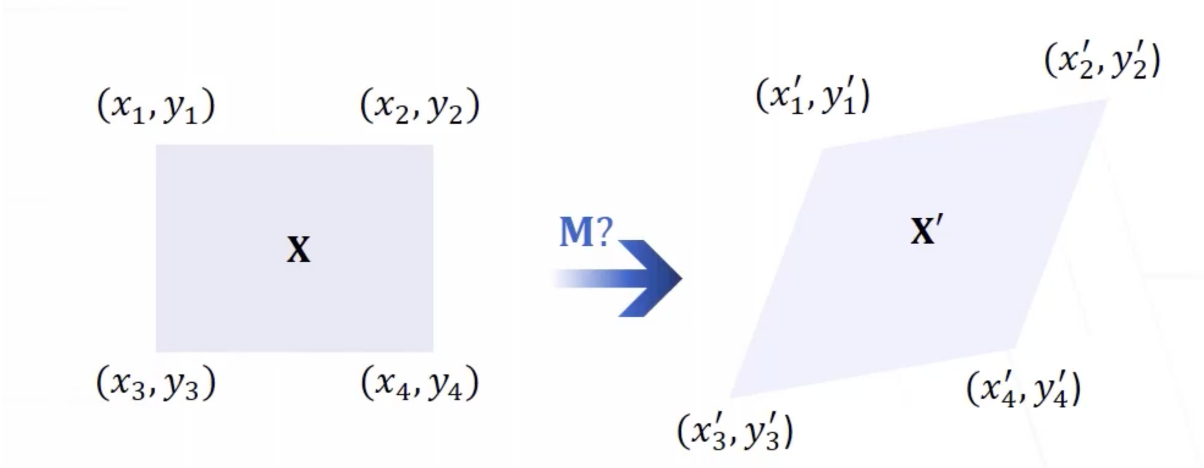

- what type of transformation will map A to D, B to E, and C to F?
- How do we determine the unknown parameters?

Affine transform:
uniform scaling + shearing
+ rotation + translation

$$
\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix}
$$




서로 ma




$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

4개의 point가 주어졌음

$(x_1, y_1), (x\prime_1, y\prime_1)$을 대입해보자

$$x_1' = ax_1 + by_1 + c$$
$$y_1' = dx_1 + ey_1 + f$$

$(x_2, y_2), (x\prime_2, y\prime_2)$,

$(x_3, y_3), (x\prime_3, y\prime_3)$,

$(x_4, y_4), (x\prime_4, y\prime_4)$

까지 다 대입

$$\begin{aligned} x_1' &= ax_1 + by_1 + c & \quad x_2' &= ax_2 + by_2 + c \\ y_1' &= dx_1 + ey_1 + f & \quad y_2' &= dx_2 + ey_2 + f \end{aligned}$$

$$\begin{aligned} x_3' &= ax_3 + by_3 + c & \quad x_4' &= ax_4 + by_4 + c \\ y_3' &= dx_3 + ey_3 + f & \quad y_4' &= dx_4 + ey_4 + f \end{aligned}$$

8개의 식이 나옴

이 식들을 행렬로 표현 할 수 있는데

$$\overset{\mathbf{A}}{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ & & & \vdots & & \\ & & & & & \\ & & & & & \end{bmatrix}} \overset{\mathbf{x}}{\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}} = \overset{\mathbf{b}}{\begin{bmatrix} x_1' \\ y_1' \\ \vdots \\ \\ \\ \end{bmatrix}}$$

**Ax = b** 형태로 나타 낼 수 있음. 이때 A와 B는 Known(포인트가 주어졌으니깐), x는 unknown(파라미터들)

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$$

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$Ax = b$ 인데

$x = A^{-1}b$는 안됨 A가 정방행렬이 아니니 역행렬이 존재한다고 증명할수 없음. 따라서

$A^\intercal Ax = A^\intercal b.$ $A^\intercal$를 곱해줘 정방행렬로 만들어 줌

$(A^\intercal A)^{-1}A^\intercal Ax = (A^\intercal A)^{-1}A^\intercal Ab$

$\therefore x = (A^\intercal A)^{-1}A^\intercal Ab$

위 식으로 인해 행렬 M을 구할 수 있음

Known 파라미터가 6개였으니 식이 최소 6개 있어야 함. 즉위 affine transform은 mapping point가 최소 3개 있어야지 구할 수 있음.