Translation - it's impossible!

Proof: consider the origin $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Solution #1:

$\underset{\leftarrow}{T}$ is now a matrix plus a vector.

$$T = (A_1, \vec{t})$$

$$T(\vec{x}) = A\vec{x} + \vec{t}$$

Annoyance: $T_1 \circ T_2$??? $A_1(A_2\vec{x} + \vec{t}_2) + \vec{t}_1$

Solution #2: Clever math hack

Want:

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11}x + a_{12}y + t_x \\ a_{21}x + a_{21}y + t_y \\ 1 \end{bmatrix}}_{} \leftarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Homogeneous Coordinates:

Tack on an extra dimension;

- points get 3rd coord of 1
- matrices get 3rd row $[0 \quad 0 \quad 1]$
- third column contains $\vec{t}$

These are: **Affine Transformations**

affine: $\underset{\uparrow}{a}x + \underset{\uparrow}{b}$

linear — translation

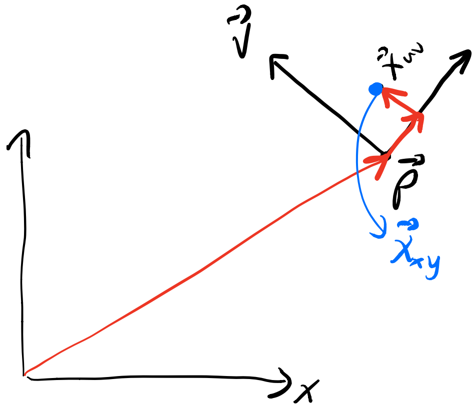
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

do not touch!

Point-moving-machine view: 2x2 top left of A

1. Apply linear part $(M\vec{x})$
2. Translate $(+\vec{t})$

Change-of-~~basis~~ frame view



$\vec{x}_{uv} = (1, 1)$

$$\boxed{T = \begin{bmatrix} \vec{u} & \vec{v} & \vec{p} \\ 0 & 0 & 1 \end{bmatrix}}$$

From $\vec{u}, \vec{v}, \vec{p}$ frame

to canonical frame.

Special case: if $\vec{u}, \vec{v}$ are orthonormal, this affine transformation

is called a rigid transformation

- translation
- rotation
- flipping

- basis vectors pairwise orthogonal
  unit length

Bonus: composition Just Works via matrix mult

HW3: prove this!

Example:

Description?

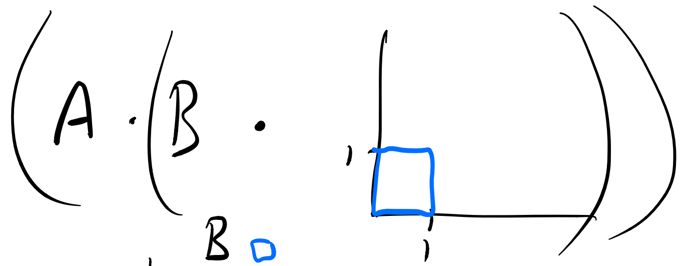
$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

flip over y

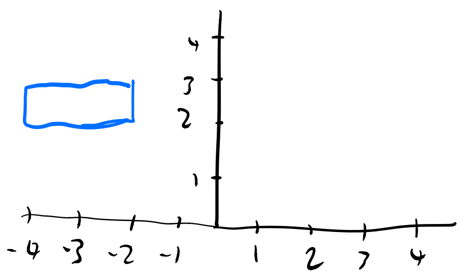
$$B = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

scale x by 2

translate b $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$



Final: AB



B