Computer Graphics

3 - Transformations

Yoonsang Lee
Hanyang University

Spring 2025

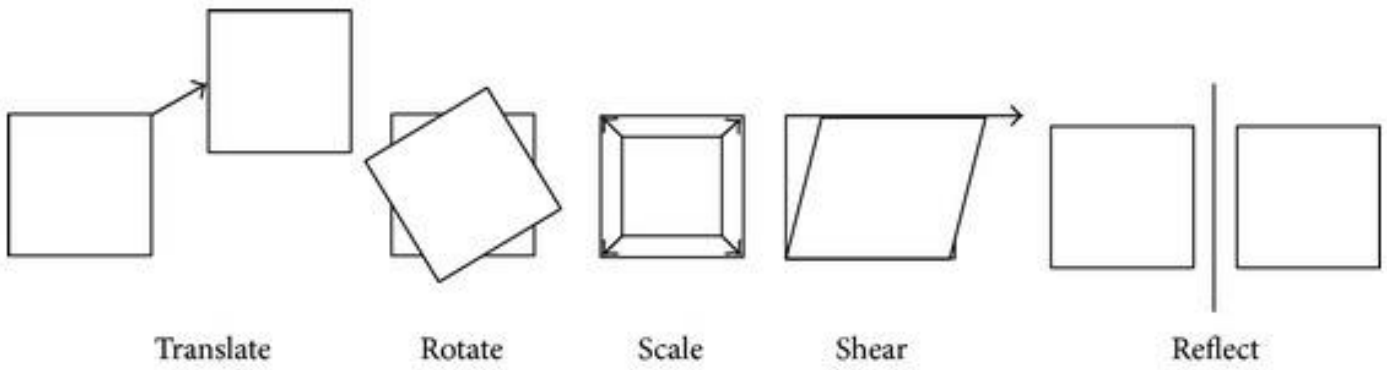
Outline

- 2D Transformations
  - Scaling, Rotation, Shearing, Reflection
  - Translation
- Classes of Transformations
- Composition of Transformations & Homogeneous Coordinates
- Two Types of 3D Cartesian Coordinate System
- 3D Affine Transformations

2D Transformations

What is Transformation?

- Geometric **Transformation**
  - The process of changing the position, orientation, size, or shape of a geometric object using mathematical operations.
    → "Moving a set of points"
  - Essential in computer graphics as it allows the creation of complex scenes and animations.
- Examples:

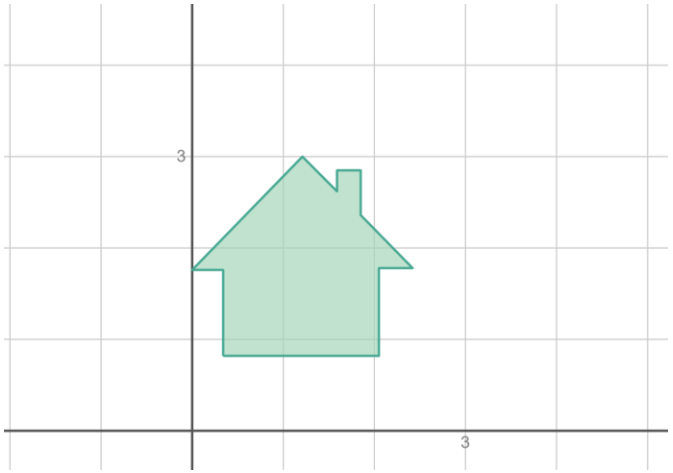


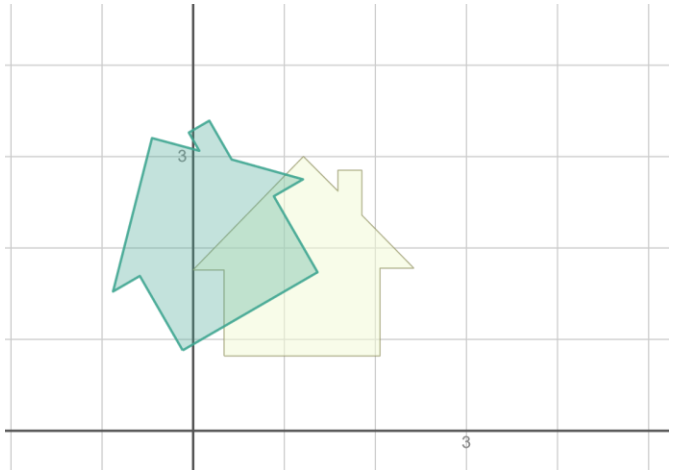
* This image is from the slides of Prof. Roger D. Eastman (University of Maryland): https://www.cs.umd.edu/~reastman/slides/L05P1Transformations.pdf

Transformation

- "Moving a set of points"
  - Transformation T maps any input vector v in the vector space S to T(v).

$$S \rightarrow \{T(\mathbf{v}) \mid \mathbf{v} \in S\}$$





Linear Transformation

- One way to define a transformation is by matrix multiplication:

$$T(\mathbf{v}) = M\mathbf{v}$$

- This is called a **linear transformation** because a matrix multiplication represents a linear mapping.
  - A linear transformation must satisfy:
    $T(\mathbf{v}_1 + \mathbf{v}_2) = T(\mathbf{v}_1) + T(\mathbf{v}_2), \;\; T(c\mathbf{v}) = cT(\mathbf{v})$
  - A matrix multiplication satisfies them:
    $M(\mathbf{v}_1 + \mathbf{v}_2) = M\mathbf{v}_1 + M\mathbf{v}_2 \;, \;\; M(c\mathbf{v}) = c(M\mathbf{v})$

2D Linear Transformations

- 2x2 matrices represent 2D linear transformations such as:
  - uniform scaling
  - non-uniform scaling
  - rotation
  - shearing
  - reflection

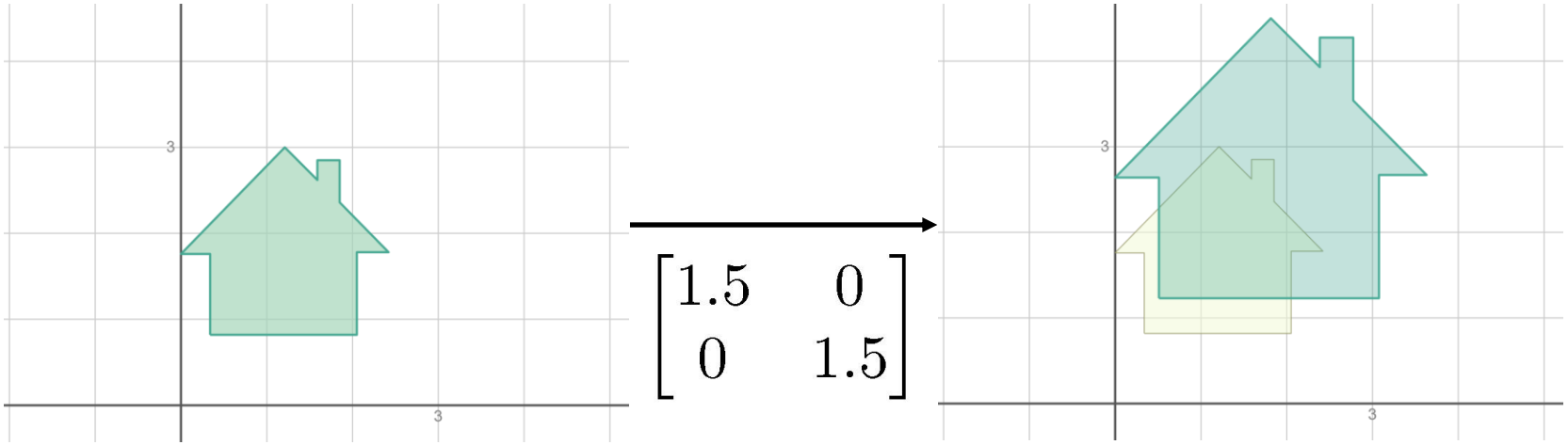
2D Linear Trans. – Uniform Scaling

- Uniformly shrinks or enlarges both in x and y directions.

$$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} sx \\ sy \end{bmatrix}$$

2x2 scale matrix

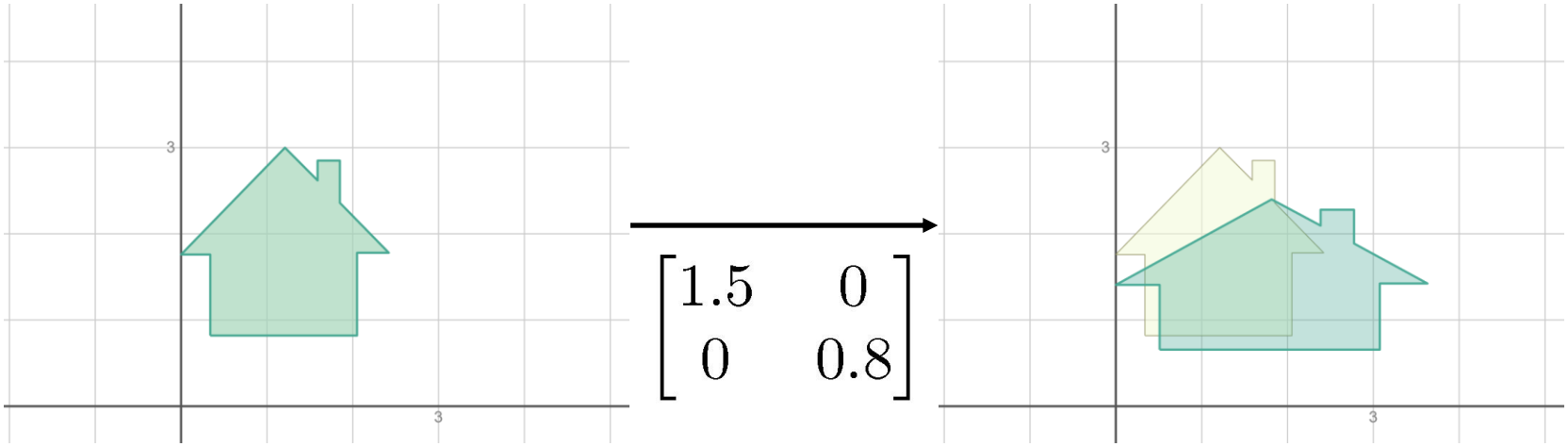
S p = p'



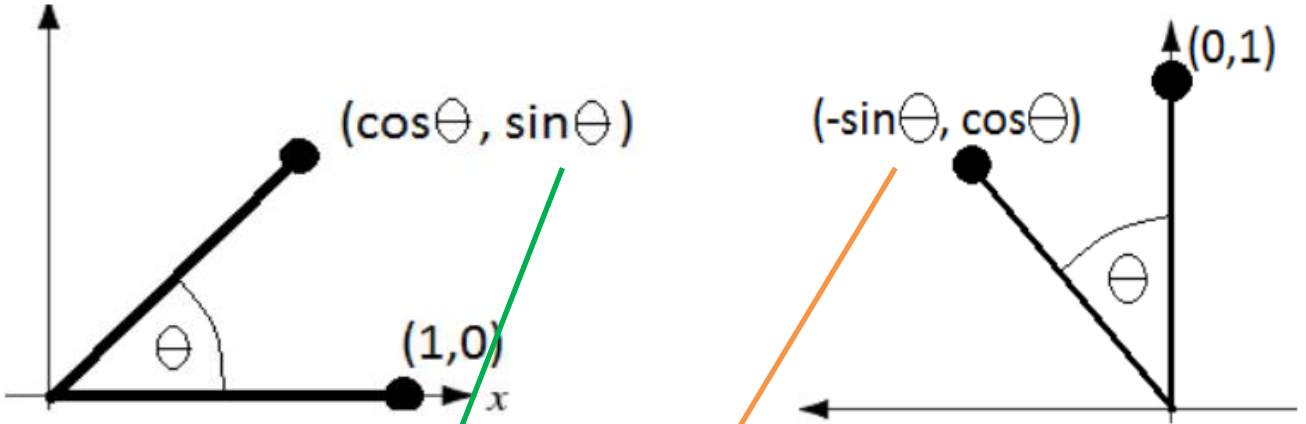
2D Linear Trans. – Nonuniform Scaling

- Non-uniformly shrinks or enlarges in x and y directions.

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$



2D Linear Trans. – Rotation



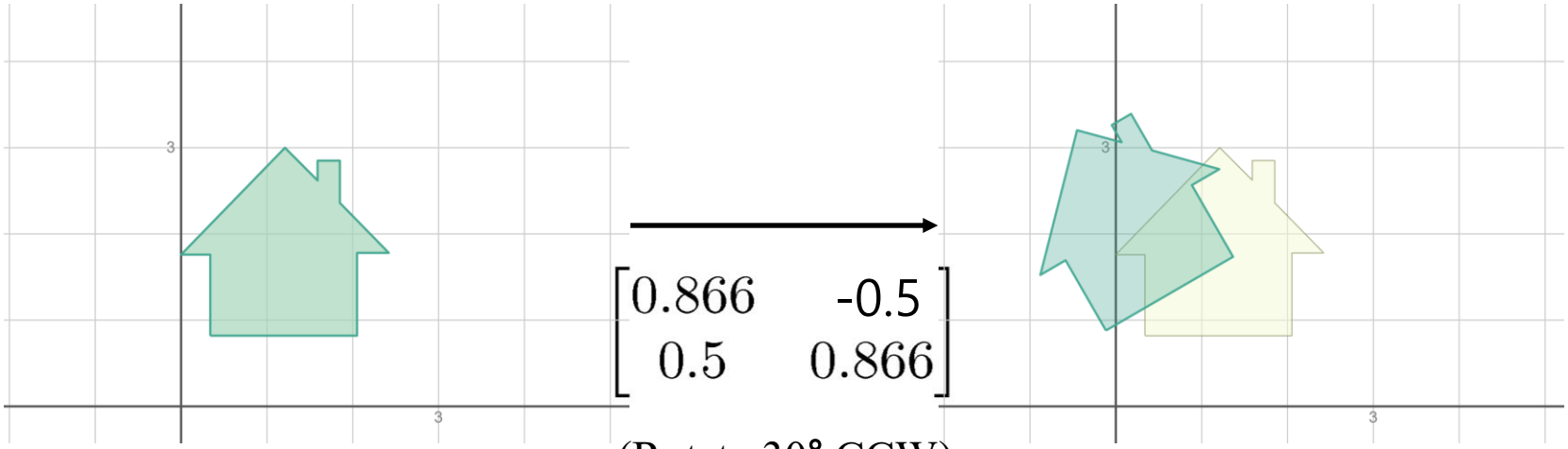
$$\Rightarrow R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

: Rotation matrix

2D Linear Trans. – Rotation

- Rotation can be written in matrix multiplication, so it's also a linear transformation.
  - Note that positive angle means CCW rotation.

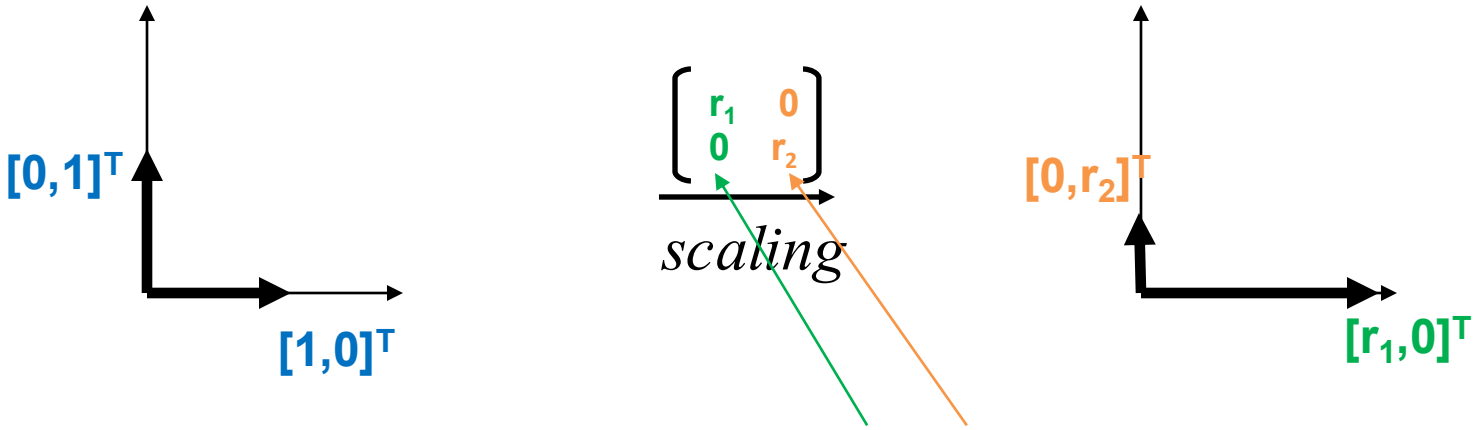
$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$



(Rotate 30° CCW)

Numbers in Matrices: Scaling, Rotation

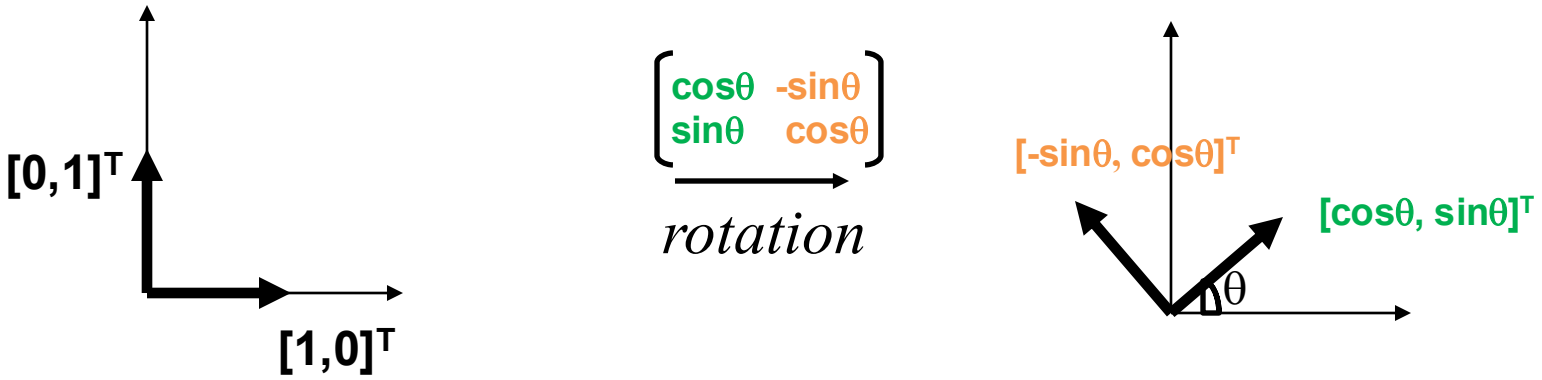
- Let's think about what the numbers in the matrix means.



Canonical basis vectors: unit vectors pointing in the direction of the axes of a Cartesian coordinate system.

1st & 2nd basis vector of the transformed coordinates

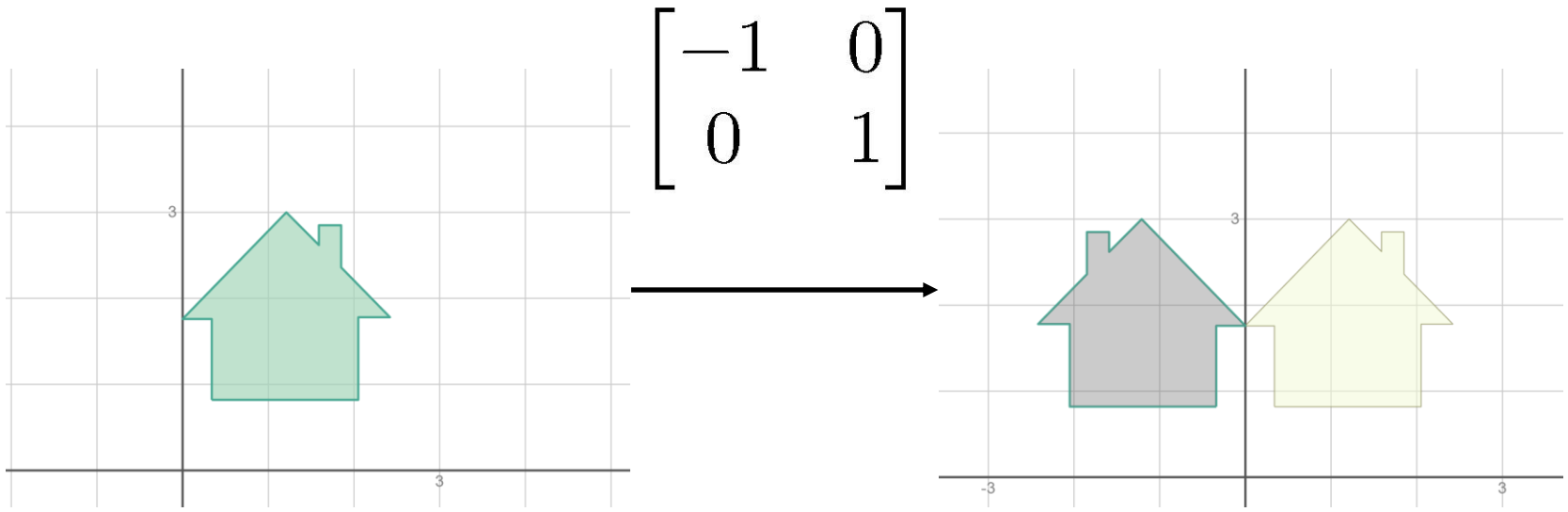
Numbers in Matrices: Scaling, Rotation



- *Column vectors* of a matrix is the *basis vectors of the column space* of the matrix.
  - *Column space*: The span (a set of all possible linear combinations) of the column vectors.

2D Linear Trans. – Reflection

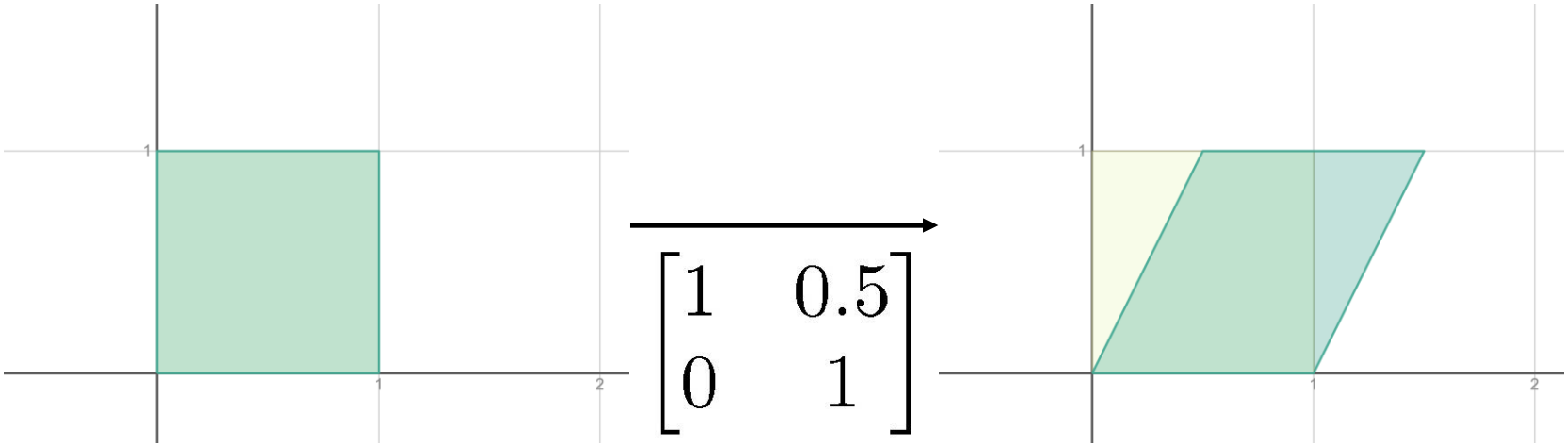
- Reflection can be considered as a special case of non-uniform scale.



2D Linear Trans. – Shearing

- "Push things sideways"

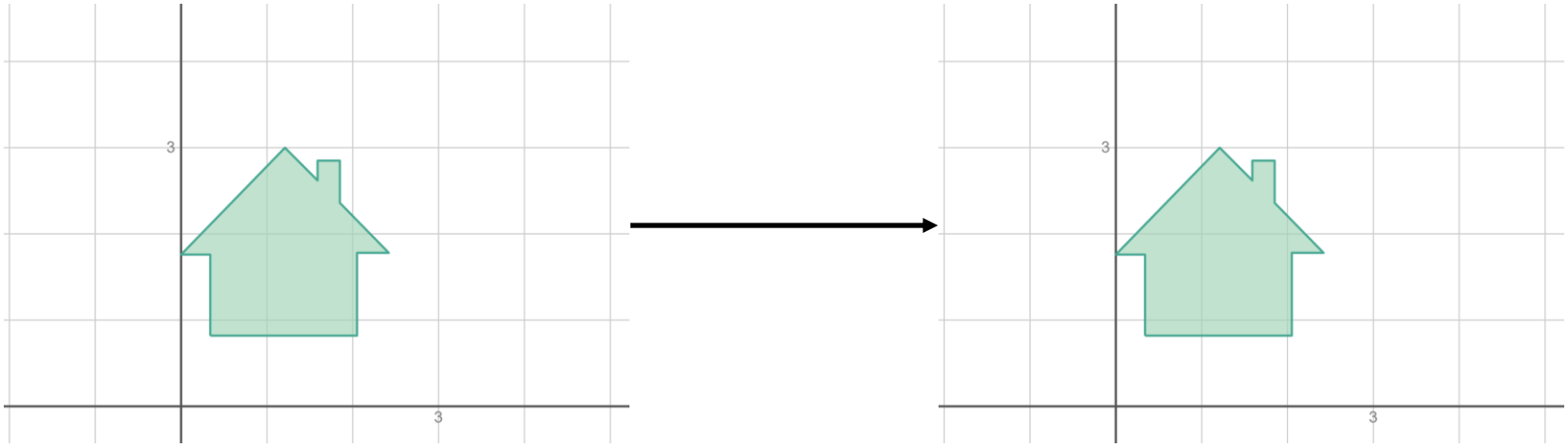
$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ay \\ y \end{bmatrix}$$



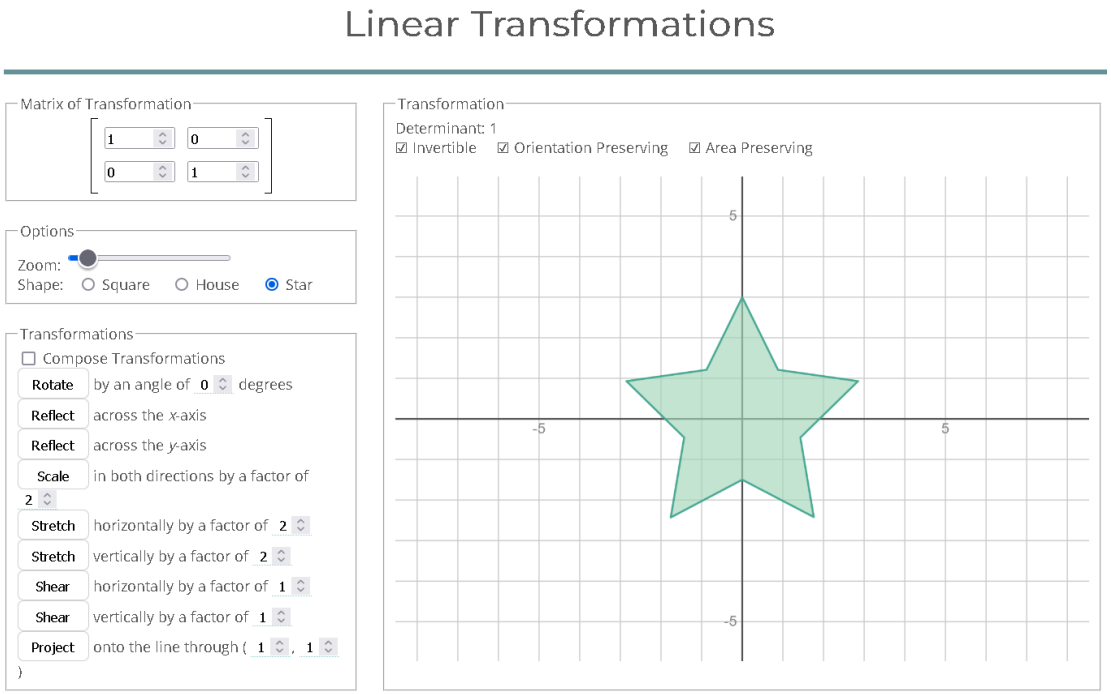
Identity Matrix

- "Doing nothing"

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



[Demo] 2D Linear Transformations



https://www.integral-domain.org/lwilliams/Applets/algebra/linearTransformations.php

- Try changing the values of matrix elements.
- Try pressing the transformation buttons.

Quiz 1

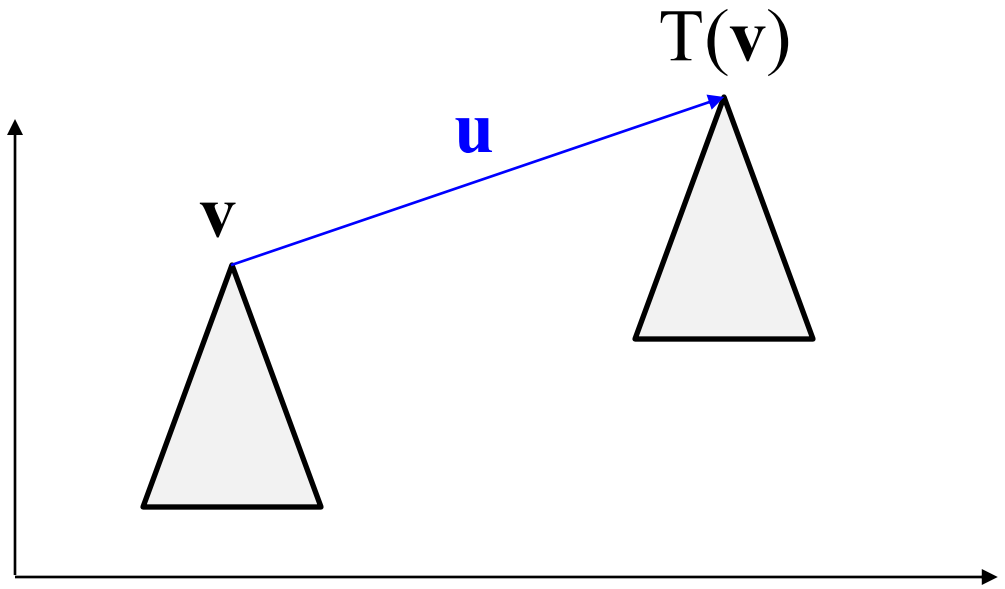
- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to be counted as attendance.

2D Translation

- Translation is the simplest transformation:
  $T(\mathbf{v}) = \mathbf{v} + \mathbf{u}$
- Inverse:
  $T^{-1}(\mathbf{v}) = \mathbf{v} - \mathbf{u}$



Is translation linear transformation?

- No. $T(c\mathbf{v}) = c\mathbf{v} + \mathbf{b}$ $\neq$
  $cT(\mathbf{v}) = c(\mathbf{v} + \mathbf{b}) = c\mathbf{v} + c\mathbf{b}$

- We can express translation using vector addition:
  $T(\mathbf{v}) = \mathbf{v} + \mathbf{u}$

- Combining with linear transformation:
  $T(\mathbf{v}) = M\mathbf{v} + \mathbf{u}$

- **→ Affine transformation**

Let's check again

- Linear transformation
  - Scaling, rotation, reflection, shearing
  - Represented as matrix multiplications

$$T(\mathbf{v}) = M\mathbf{v}$$

- Translation
  - Not a linear transformation
  - Can be expressed using vector addition

$$T(\mathbf{v}) = \mathbf{v} + \mathbf{u}$$

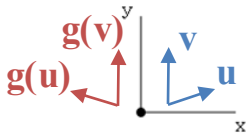
- Affine transformation
  - Combination of linear transformation and translation

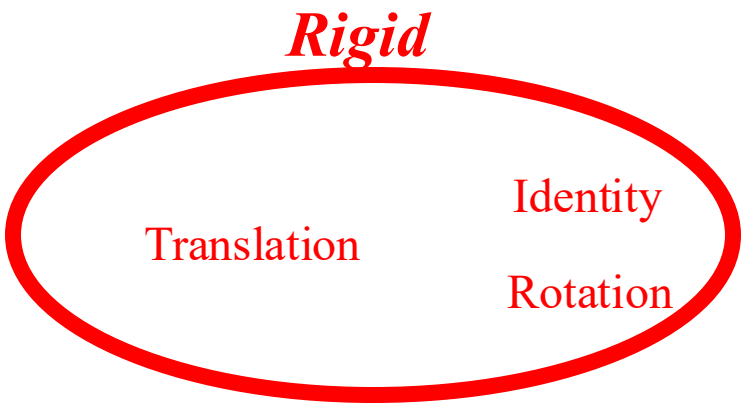
$$T(\mathbf{v}) = M\mathbf{v} + \mathbf{u}$$

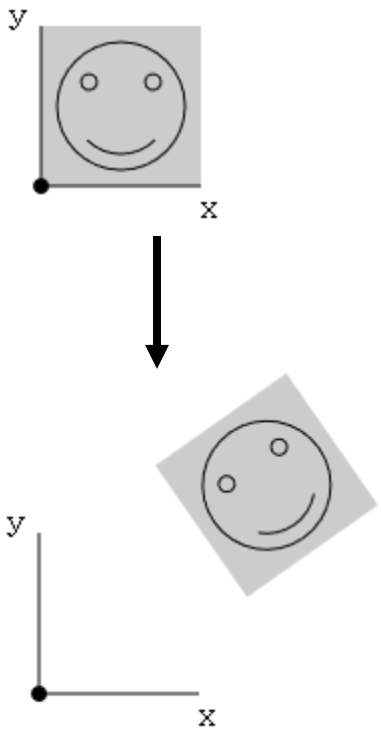
Classes of Transformations

Rigid Transformations

- Preserve distances between all points.
  - $\|g(\mathbf{u}) - g(\mathbf{v})\| = \|\mathbf{u} - \mathbf{v}\|, \forall \mathbf{u}, \mathbf{v} \in R^3$ (g: rigid transform map)
- Preserve "handness".
  - $g(\mathbf{u}) \times g(\mathbf{v}) = g(\mathbf{u} \times \mathbf{v}), \forall \mathbf{u}, \mathbf{v} \in R^3$
  - Reflections do not satisfy this property.





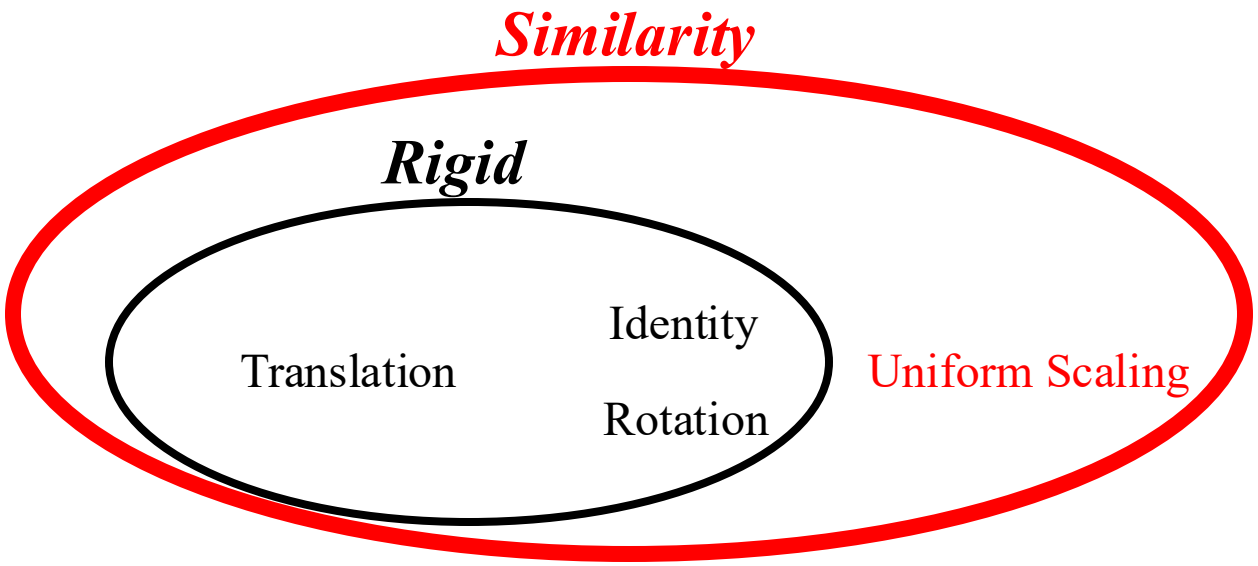


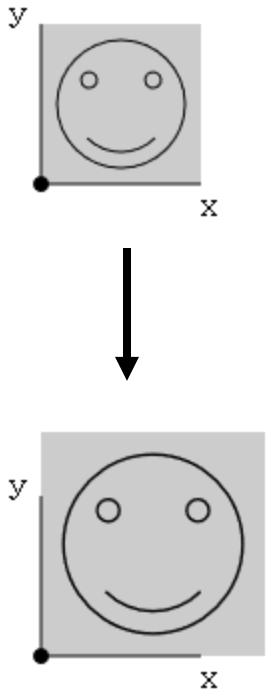
- Some people categorize reflection as a rigid transformation. We don't do that in this lecture.

* The diagram is from the slides of Prof. Frédo Durand and Prof. Barbara Cutler (MIT): https://dspace.mit.edu/bitstream/handle/1721.1/86191/6-837-fall-2003/contents/lecture-notes/index.htm

Similarity Transformations

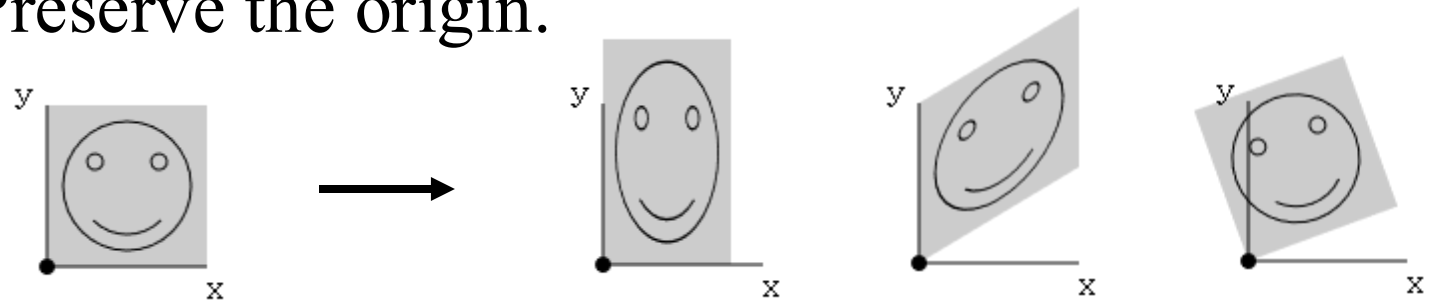
- Preserve angles.
- (This diagram indicates rigid transforms also preserve angles.)

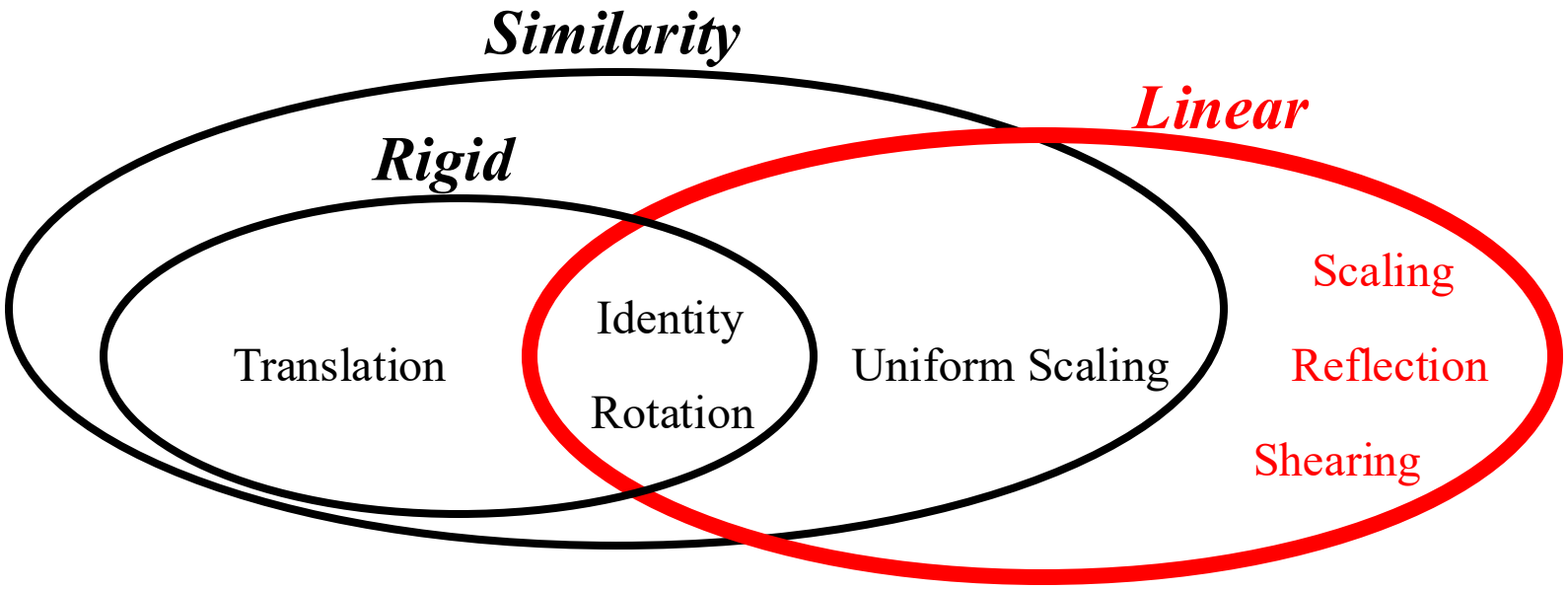




Linear Transformations

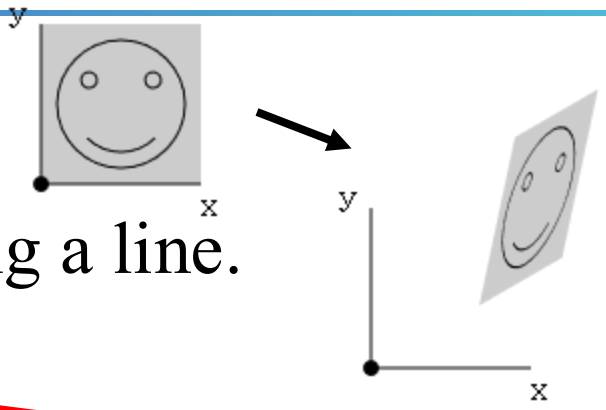
- Preserve the origin.

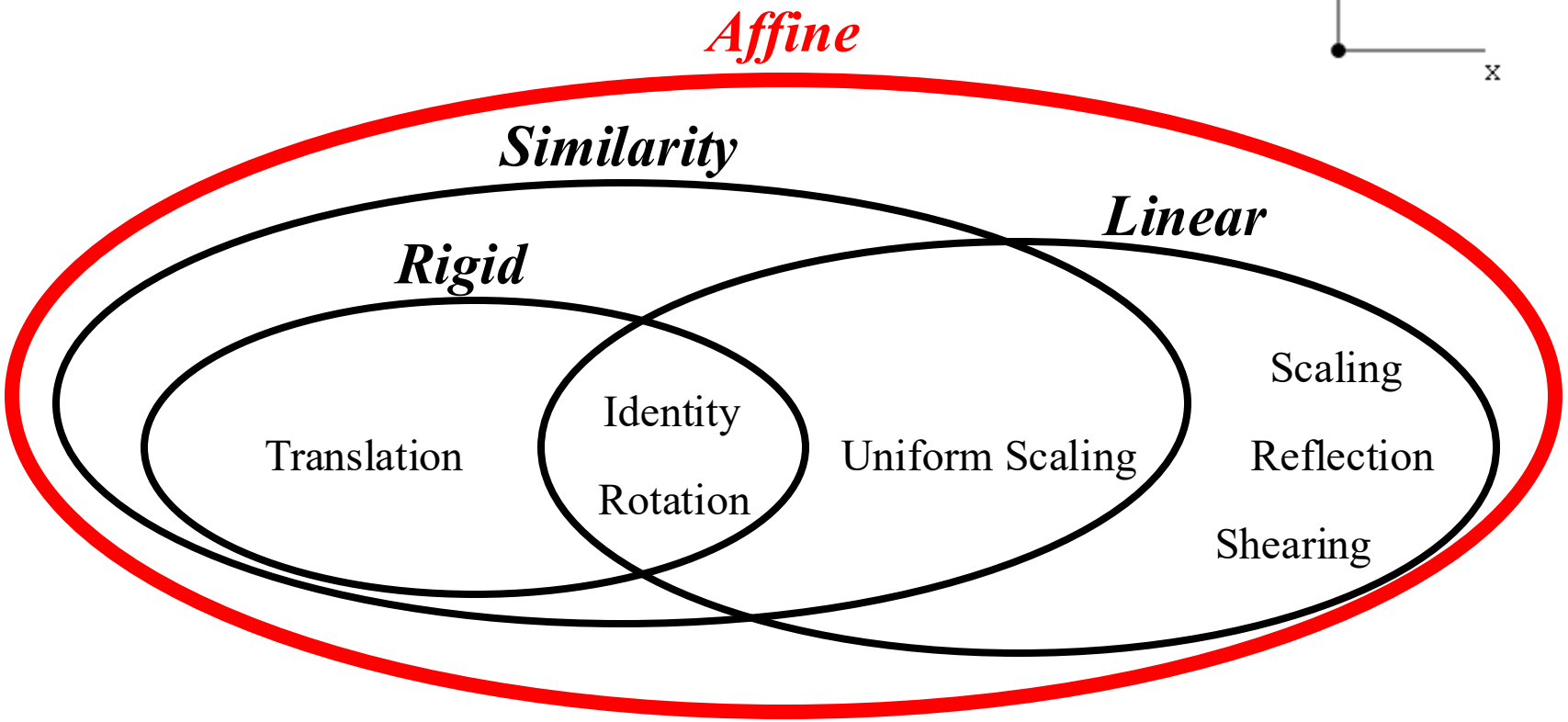




Affine Transformations

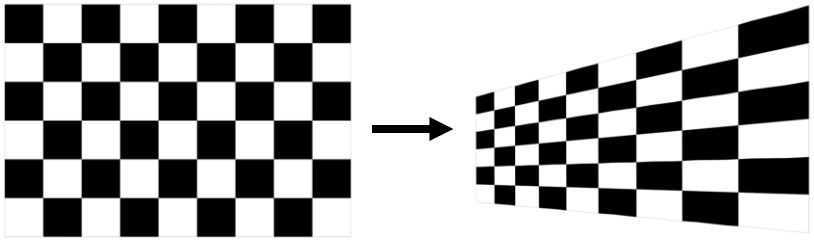
- Preserve parallel lines.
- Preserve ratios of distance along a line.

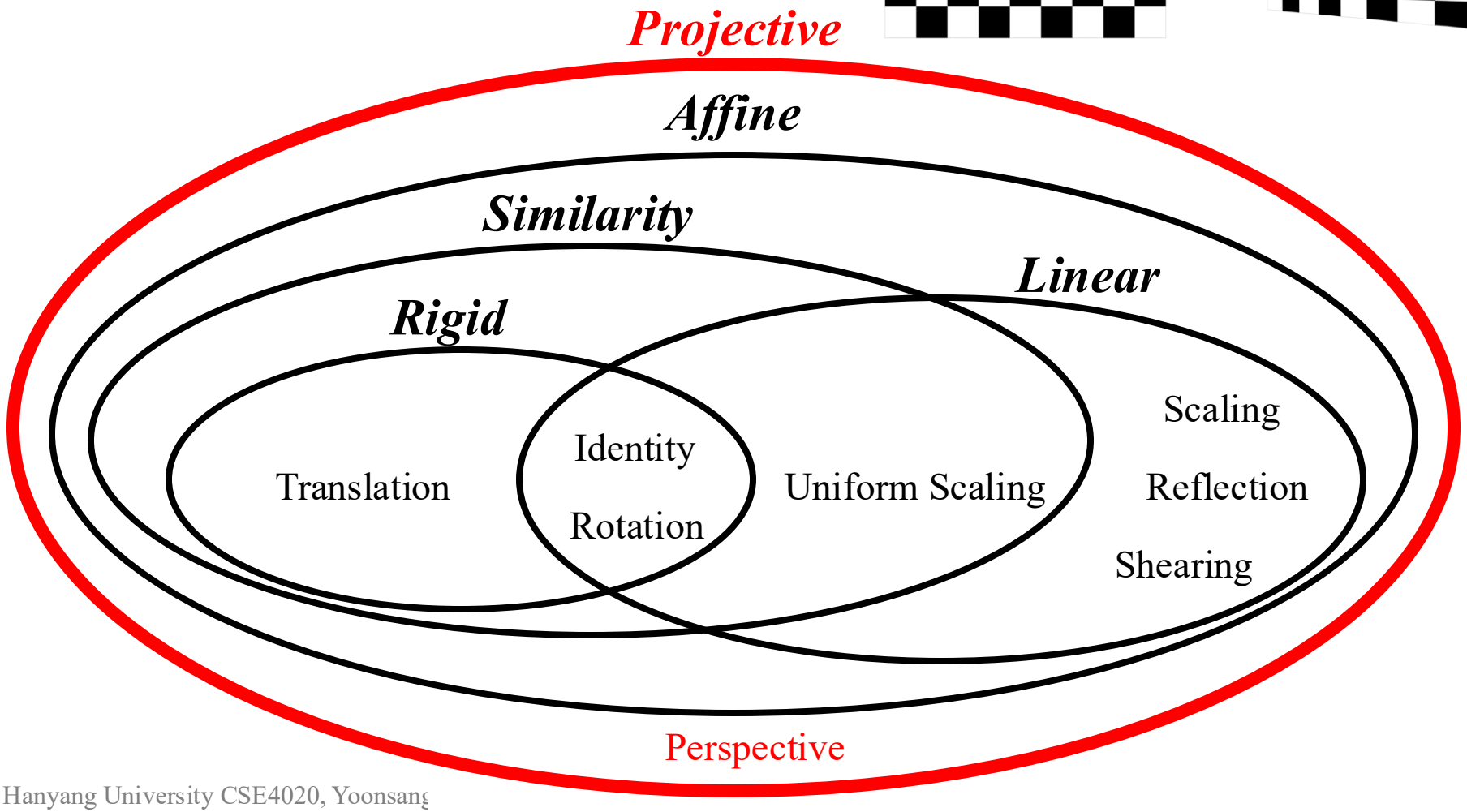




Projective Transformations

- Preserve lines.





Composition of Transformations & Homogeneous Coordinates

Composition of Transformations

- Move an object by T, then move it more by S:

$$\mathbf{p} \rightarrow T(\mathbf{p}) \rightarrow S(T(\mathbf{p})) = (S \circ T)(\mathbf{p})$$

- **Composing 2D linear transformations** just works by **2x2 matrix multiplication:**

$$T(\mathbf{p}) = M_T\mathbf{p}; S(\mathbf{p}) = M_S\mathbf{p}$$

$$(S \circ T)(\mathbf{p}) = M_S M_T \mathbf{p} = (M_S M_T)\mathbf{p} = M_S(M_T\mathbf{p})$$

* The equation images are from the slides of Prof. Steve Marschner (Cornell University): https://www.cs.cornell.edu/courses/cs4620/2018fa/

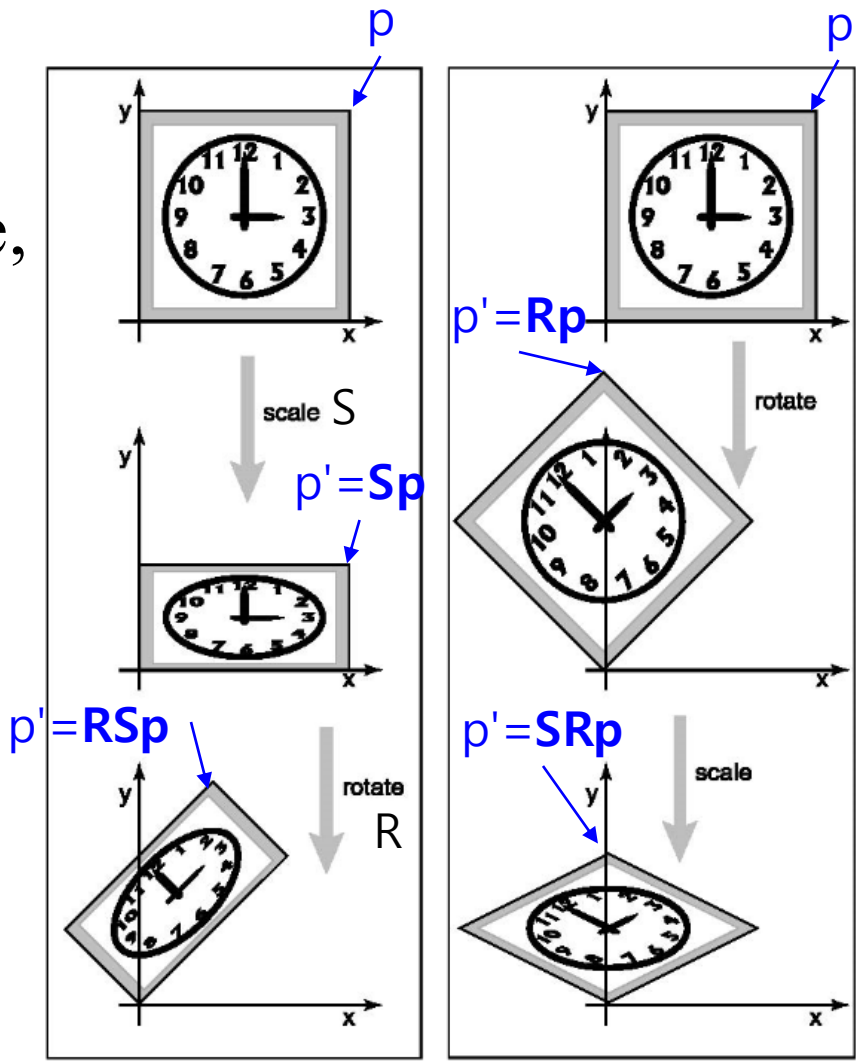
Order Matters!

- Note that matrix multiplication is associative, but **not commutative**.

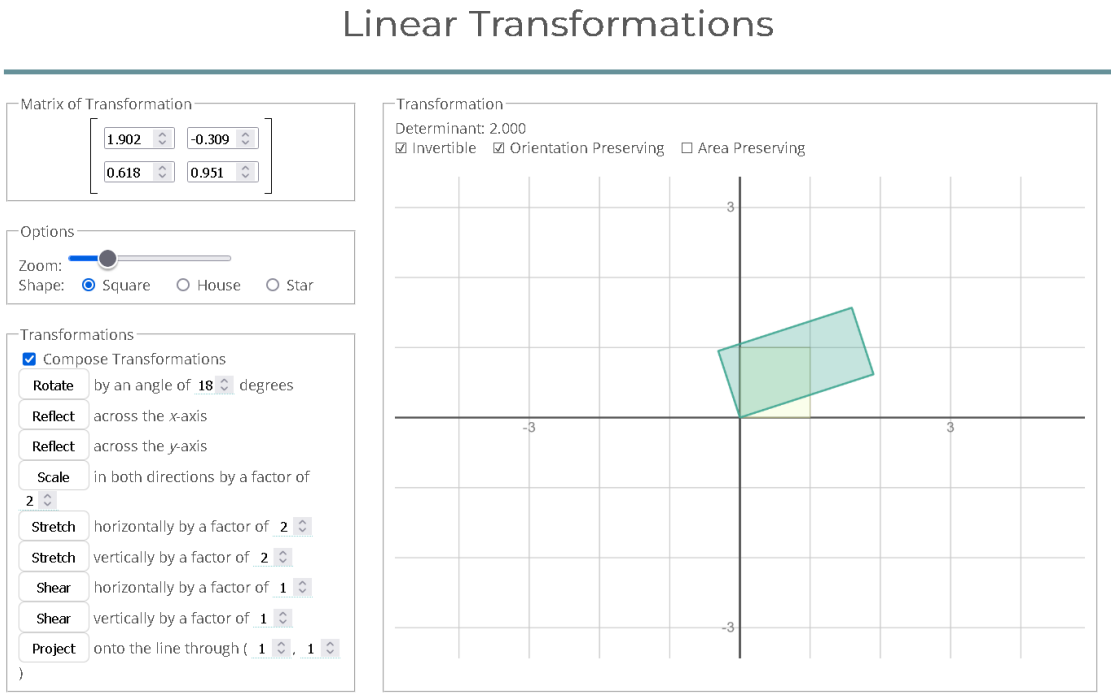
$$(AB)C = A(BC)$$

$$AB \neq BA$$

- As a result, the **order of transforms is very important.**



[Demo] Composition of Linear Transformations



https://www.integral-domain.org/lwilliams/Applets/algebra/linearTransformations.php

- Reset the matrix to the identity matrix (by entering 1 0 0 1).
- Check 'Compose Transformations' button.
- Composites two transforms in different order.

Problems when handling Translation as Vector Addition

- Cannot treat linear transformation (rotation, scale,…) and translation in a consistent manner.

- Composing affine transformations is complicated:

$$T(\mathbf{p}) = M_T\mathbf{p} + \mathbf{u}_T \qquad (S \circ T)(\mathbf{p}) = M_S(M_T\mathbf{p} + \mathbf{u}_T) + \mathbf{u}_S$$
$$S(\mathbf{p}) = M_S\mathbf{p} + \mathbf{u}_S \qquad = (M_SM_T)\mathbf{p} + (M_S\mathbf{u}_T + \mathbf{u}_S)$$

- We need a cleaner way!
- → **Homogeneous coordinates**

Homogeneous Coordinates

- Key idea: Represent 2D points in 3D coordinate space.
- Extra component *w* for vectors, extra row/column for matrices.
  - For points, always $w = 1$
  - 2D point $[x, y]^T \rightarrow [x, y, 1]^T$.
- 2D linear transformations are represented as:

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- 2D translations are represented as:

$$\begin{bmatrix} 1 & 0 & t \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+t \\ y+s \\ 1 \end{bmatrix}$$

- 2D affine transformations are represented as:

linear part

$$\begin{bmatrix} m_{11} & m_{12} & u_x \\ m_{21} & m_{22} & u_y \\ 0 & 0 & 1 \end{bmatrix}$$

translational part

Homogeneous Coordinates

- **Composing affine transformations** just works by **3x3 matrix multiplication.**

$$T(\mathbf{p}) = M_T\mathbf{p} + \mathbf{u}_T$$
$$S(\mathbf{p}) = M_S\mathbf{p} + \mathbf{u}_S$$

$$T(\mathbf{p}) = \begin{bmatrix} \overset{2x2}{M_T} & \overset{2x1}{\mathbf{u}_T} \\ 0 & 1 \end{bmatrix} \qquad S(\mathbf{p}) = \begin{bmatrix} \overset{2x2}{M_S} & \overset{2x1}{\mathbf{u}_S} \\ 0 & 1 \end{bmatrix}$$

(in block matrix representation)

Homogeneous Coordinates

- **Composing affine transformations** just works by **3x3 matrix multiplication.**

$$(S \circ T)(\mathbf{p}) = \begin{bmatrix} \overset{2x2}{M_S} & \overset{2x1}{\mathbf{u}_S} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \overset{2x2}{M_T} & \overset{2x1}{\mathbf{u}_T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \overset{2x1}{\mathbf{p}} \\ 1 \end{bmatrix}$$

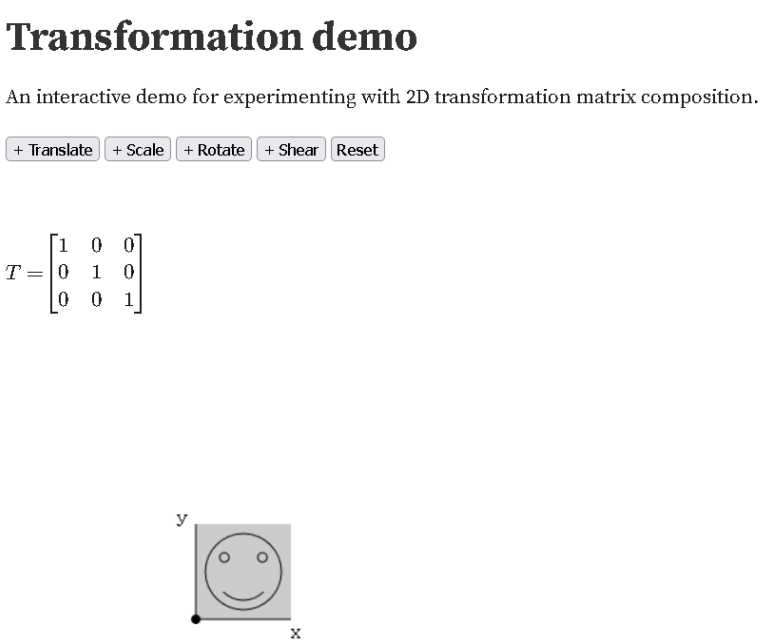
$$= \begin{bmatrix} (M_S M_T)\mathbf{p} + (M_S \mathbf{u}_T + \mathbf{u}_S) \\ 1 \end{bmatrix}$$

- The result is the same, but much cleaner.
  - cf. $(S \circ T)(\mathbf{p}) = M_S(M_T\mathbf{p} + \mathbf{u}_T) + \mathbf{u}_S$

$$= (M_S M_T)\mathbf{p} + (M_S \mathbf{u}_T + \mathbf{u}_S)$$

* The equation images are from the slides of Prof. Steve Marschner (Cornell University): https://www.cs.cornell.edu/courses/cs4620/2018fa/

[Demo] Composition of Affine Transformations in Homogeneous Coordinates



https://observablehq.com/@esperanc/transformation-demo

- Add translation and linear transforms in various orders with '+' buttons.
- Drag the slider to see the matrix value change and the shape transform.
- **Note that the last transform added is the first applied transform.**

Summary: Homogeneous Coordinates in 2D

- Use $\mathbf{(x,y,1)^T}$ instead of $(x,y)^T$ for **2D points**
- Use **3x3 matrices** instead of 2x2 matrices **for 2D linear transformations**
- Use **3x3 matrices** instead of vector additions **for 2D translations**

- → We can treat linear transformations and translations **in a consistent manner!**

Quiz 2

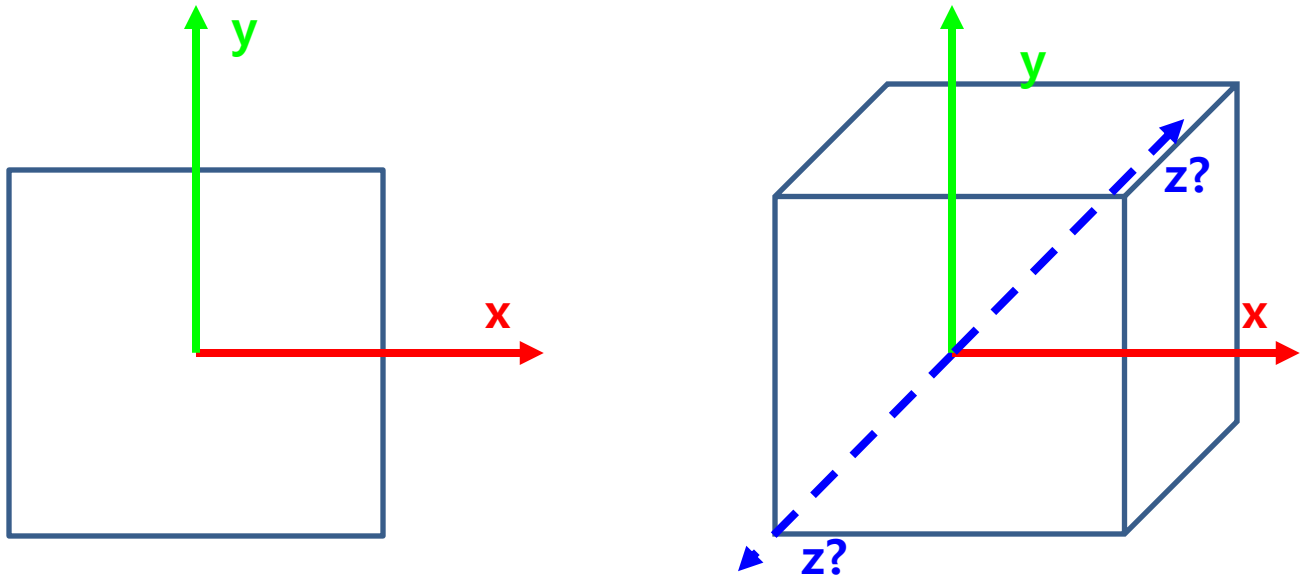
- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to be counted as attendance.

Two Types of 3D Cartesian Coordinate System

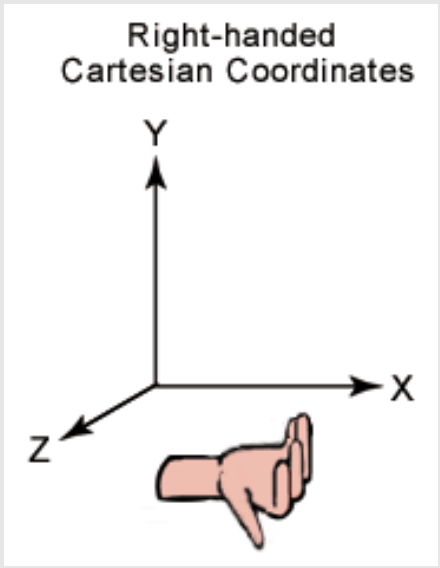
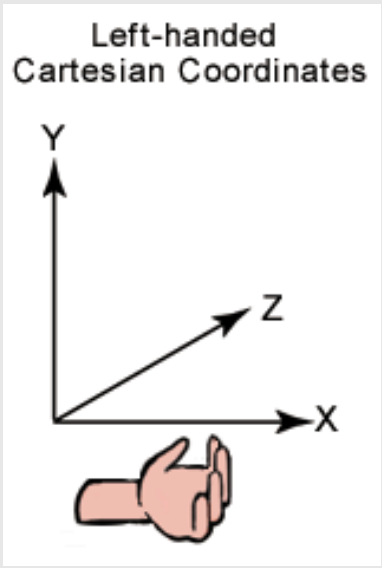
Now, Let's go to the 3D world!



- Coordinate system (좌표계)
  - Cartesian coordinate system (직교좌표계)

Right-Handed and Left-Handed Coordinate Systems

What we're using

	Right-handed Cartesian Coordinates	Left-handed Cartesian Coordinates
		
Positive rotation direction	**counterclockwise** about the axis of rotation 	**clockwise** about the axis of rotation 
Used in…	**OpenGL**, Maya, Houdini, AutoCAD, ... Standard for Physics & Math	DirectX, Unity, Unreal, …

3D Affine Transformations

Point Representations in Cartesian & Homogeneous Coordinate System

	Cartesian coordinate system	Homogeneous coordinate system
A **2D point** is represented as…	$\begin{bmatrix} p_x \\ p_y \end{bmatrix}$	$\begin{bmatrix} p_x \\ p_y \\ \boxed{1} \end{bmatrix}$
A **3D point** is represented as…	$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$	$\begin{bmatrix} p_x \\ p_y \\ p_z \\ \boxed{1} \end{bmatrix}$

Review of Linear Transformations in 2D

- Linear transformations in **2D** can be represented as matrix multiplication of …

2x2 matrix
(in Cartesian coordinates)

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

or

3x3 matrix
(in homogeneous coordinates)

$$\begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

Linear Transformations in 3D

- Linear transformations in **3D** can be represented as matrix multiplication of …

3x3 matrix
(in Cartesian coordinates)

or

4x4 matrix
(in homogeneous coordinates)

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Linear Transformations in 3D

Scale:

3D

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 \\ 0 & \mathbf{S}_y & 0 \\ 0 & 0 & \mathbf{S}_z \end{bmatrix}$$

3D-H

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 & 0 \\ 0 & \mathbf{S}_y & 0 & 0 \\ 0 & 0 & \mathbf{S}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear (in x, based on y,z position):

$$\mathbf{H}_{x,\mathbf{d}} = \begin{bmatrix} 1 & \mathbf{d}_y & \mathbf{d}_z \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

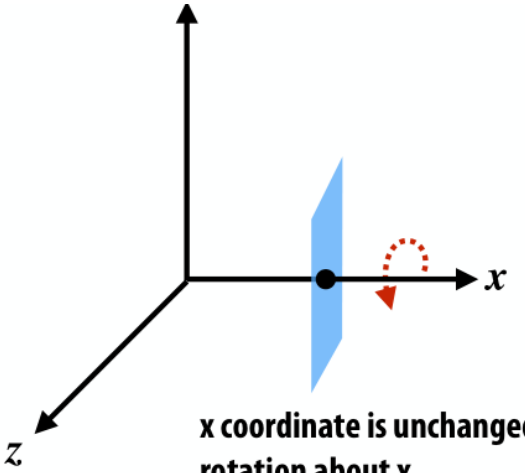
$$\mathbf{H}_{x,\mathbf{d}} = \begin{bmatrix} 1 & \mathbf{d}_y & \mathbf{d}_z & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* This slide is from the slides of Prof. Kayvon Fatahalian and Prof. Keenan Crane (CMU): http://15462.courses.cs.cmu.edu/fall2015/

Linear Transformations in 3D

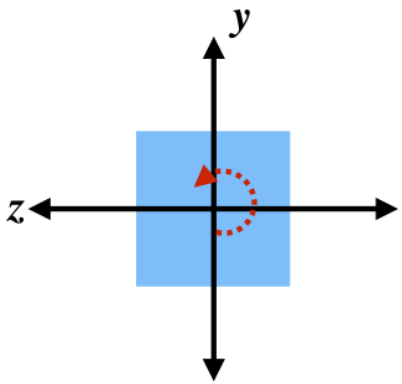
Rotation about x axis:

$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$



x coordinate is unchanged by rotation about x

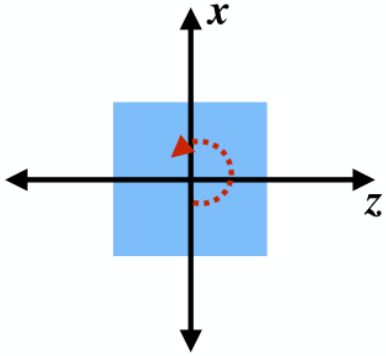
View looking down -x axis:



Rotation about y axis:

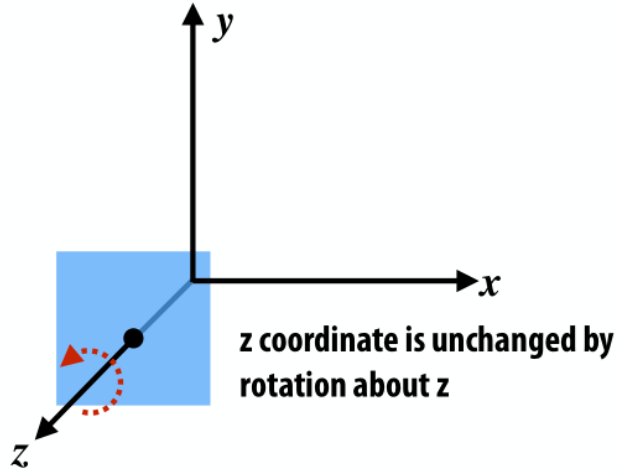
$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

View looking down -y axis:



Rotation about z axis:

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



z coordinate is unchanged by rotation about z

* This slide is from the slides of Prof. Kayvon Fatahalian and Prof. Keenan Crane (CMU): http://15462.courses.cs.cmu.edu/fall2015/

Review of Translations in 2D

- Translations in **2D** can be represented as …

Vector addition
(in Cartesian coordinates)

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

Matrix multiplication of
3x3 matrix
(in homogeneous coordinates)

$$\begin{bmatrix} 1 & 0 & u_x \\ 0 & 1 & u_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

Translations in 3D

- Translations in **3D** can be represented as …

Vector addition
(in Cartesian coordinates)

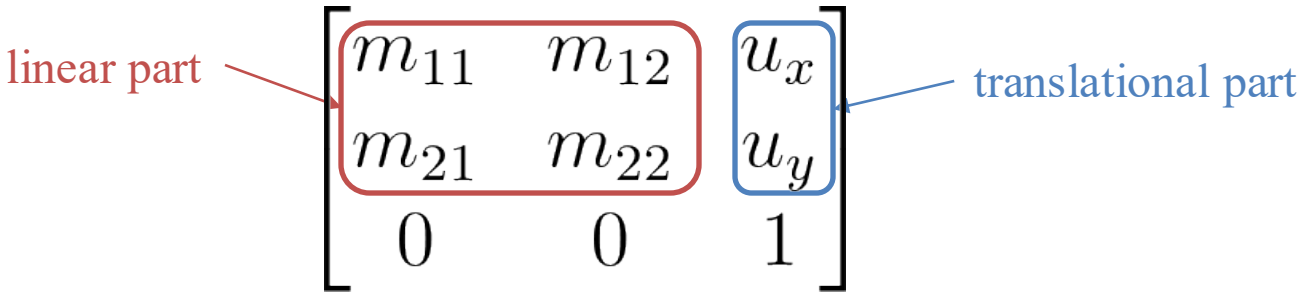
$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

Matrix multiplication of
4x4 matrix
(in homogeneous coordinates)

$$\begin{bmatrix} 1 & 0 & 0 & u_x \\ 0 & 1 & 0 & u_y \\ 0 & 0 & 1 & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

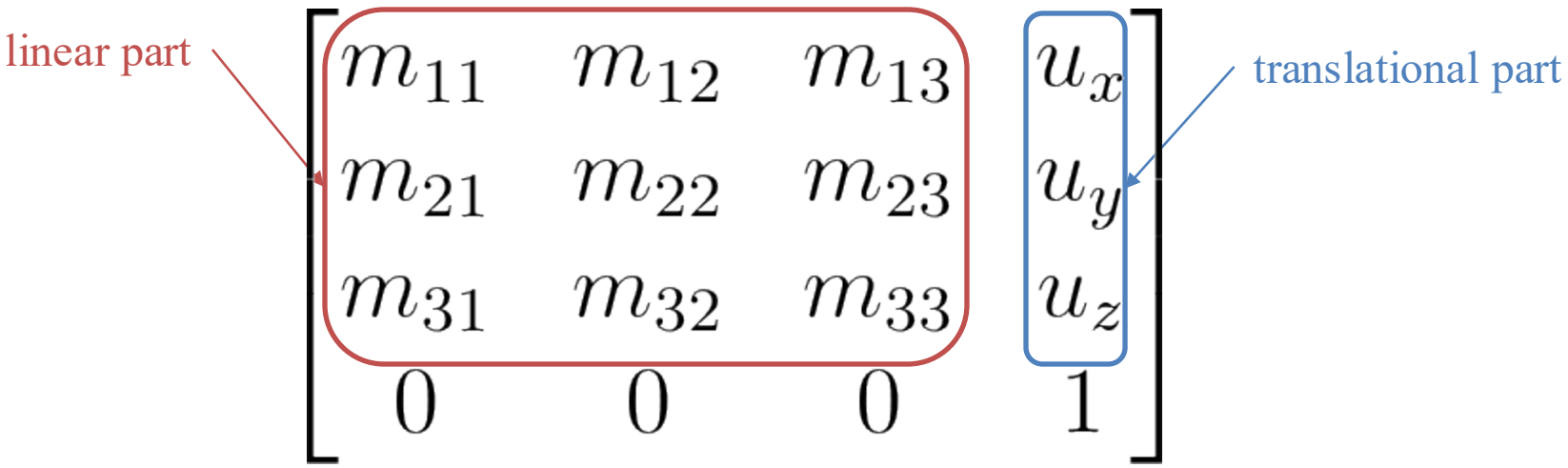
Review of Affine Transformations in 2D

- In homogeneous coordinates, **2D** affine transformations can be represented as multiplication of **3x3 matrix**:

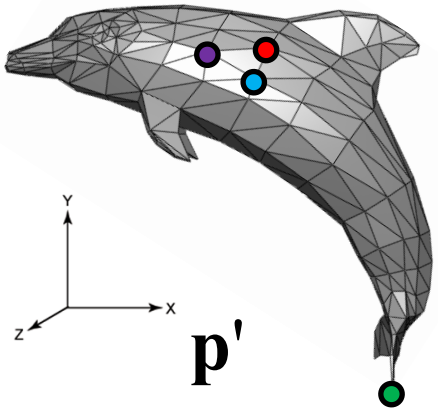


Affine Transformations in 3D

- In homogeneous coordinates, **3D** affine transformations can be represented as multiplication of **4x4 matrix**:



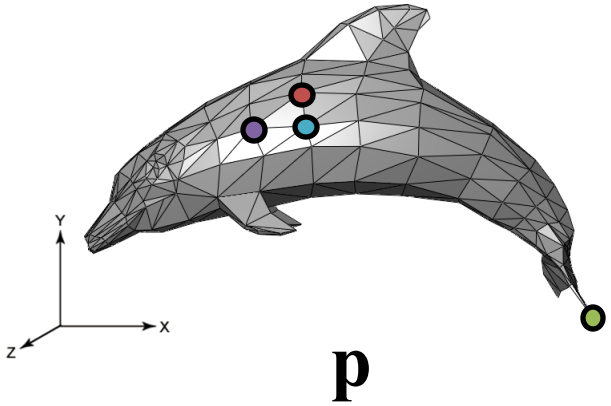
Summary: Affine Transformation



p'

Affine transformation

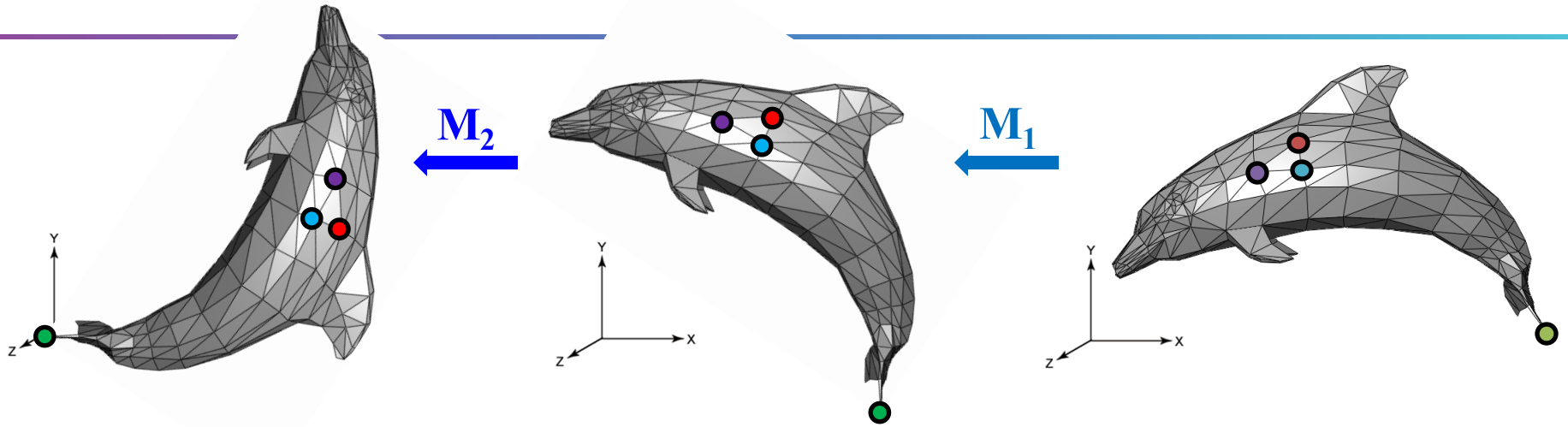
$$\mathbf{M}= \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_1 \\ m_{21} & m_{22} & m_{23} & u_2 \\ m_{31} & m_{32} & m_{33} & u_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



p

$$\begin{aligned} \mathbf{p}_1' &\leftarrow \mathbf{M}\,\mathbf{p}_1 \\ \mathbf{p}_2' &\leftarrow \mathbf{M}\,\mathbf{p}_2 \\ \mathbf{p}_3' &\leftarrow \mathbf{M}\,\mathbf{p}_3 \\ &\vdots \\ \mathbf{p}_N' &\leftarrow \mathbf{M}\,\mathbf{p}_N \end{aligned}$$

Summary: Composition of Affine Transformations



$$\mathbf{p}_1' \leftarrow \mathbf{M}_2\,\mathbf{M}_1\,\mathbf{p}_1$$
$$\mathbf{p}_2' \leftarrow \mathbf{M}_2\,\mathbf{M}_1\,\mathbf{p}_2$$
$$\mathbf{p}_3' \leftarrow \mathbf{M}_2\,\mathbf{M}_1\,\mathbf{p}_3$$
$$\vdots$$
$$\mathbf{p}_N' \leftarrow \mathbf{M}_2\,\mathbf{M}_1\,\mathbf{p}_N$$

Lab Session

- Now, let's start the lab today.