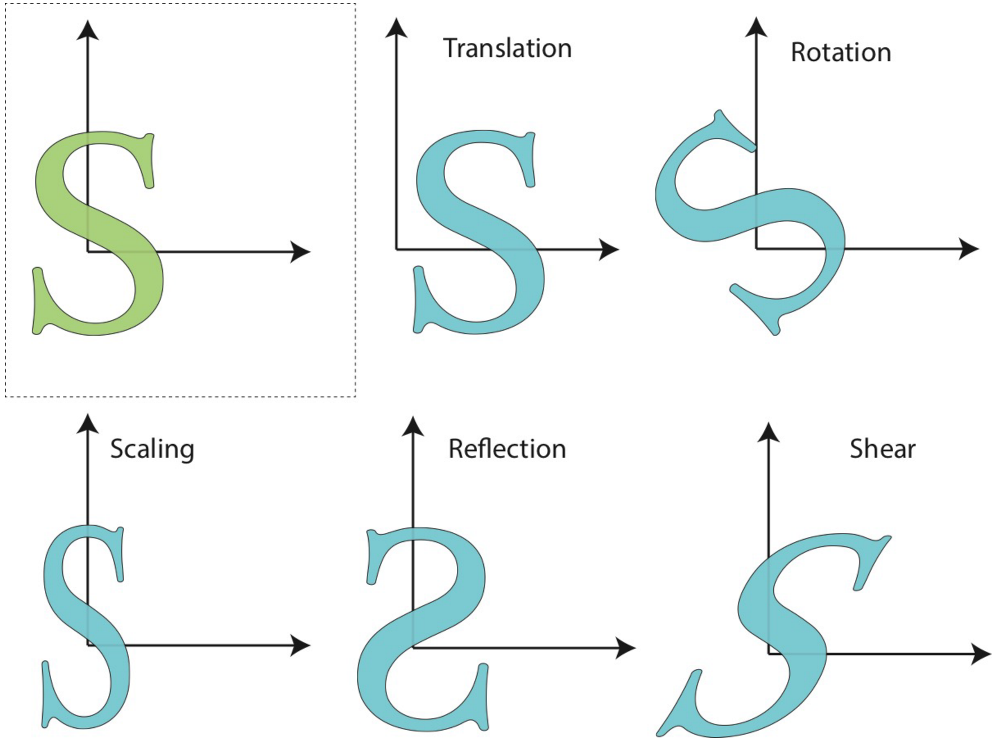
CPSC 427
Video Game Programming

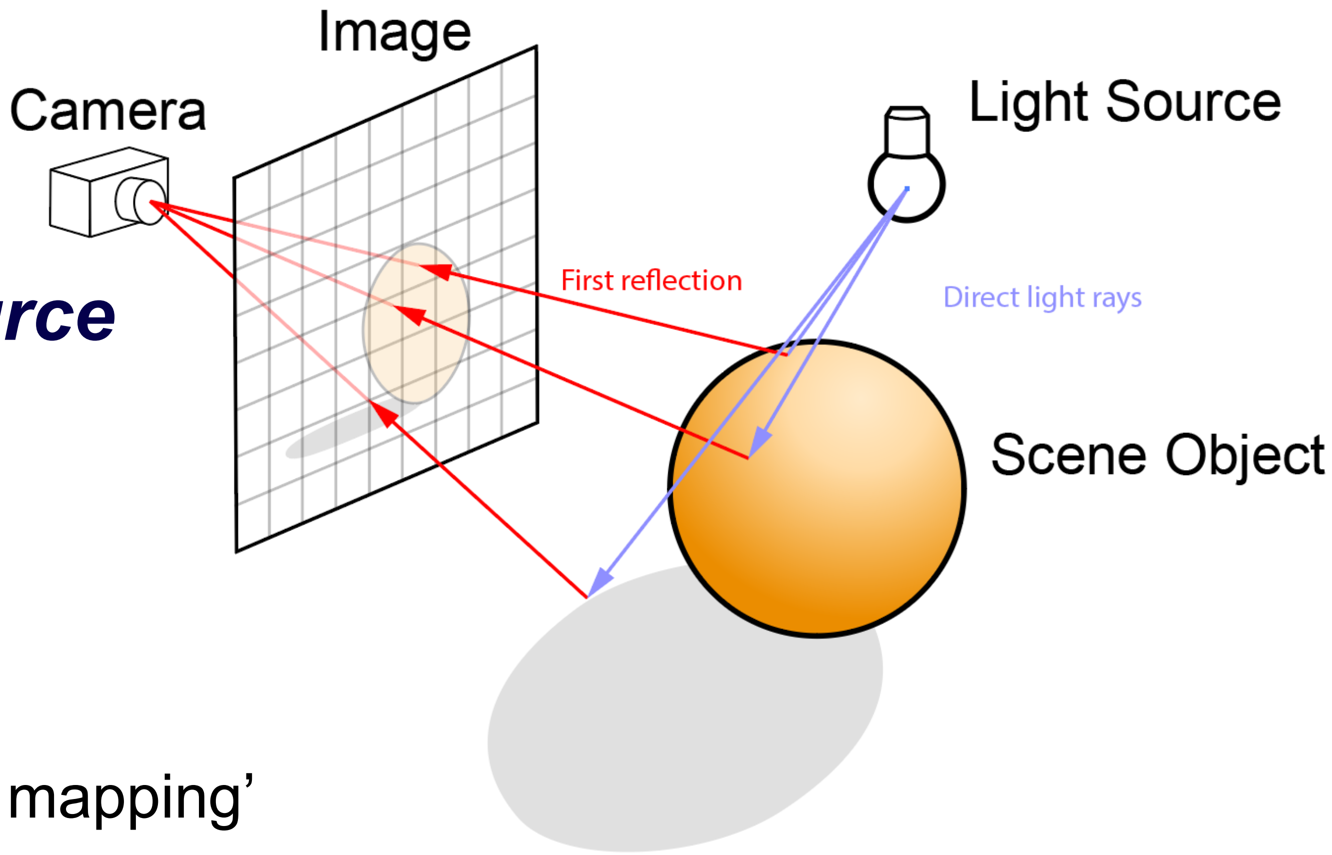
Transformations



Helge Rhodin

Rendering – Photon Tracing

- ***simulate physical light transport from a source to the camera***
  - *the paths of photons*



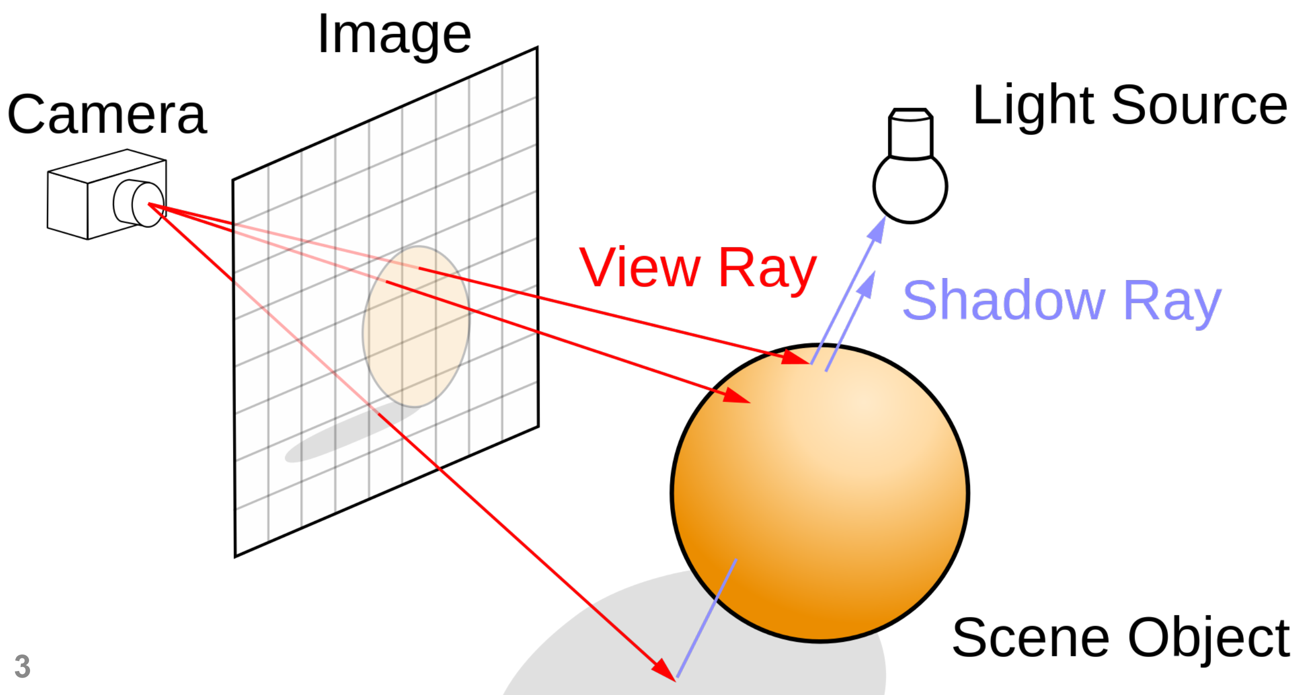
- ***shoot rays from the light source***
  - *random direction*
- ***compute first intersection***
  - *continue towards the camera*

- used for indirect illumination: ‘photon mapping’

Rendering – Ray Tracing

Start rays from the camera (opposes physics, an optimization)

- *View rays: trace from every pixel to the first occlude*
- *Shadow ray: test light visibility*



Nvidia RTX does ray tracing

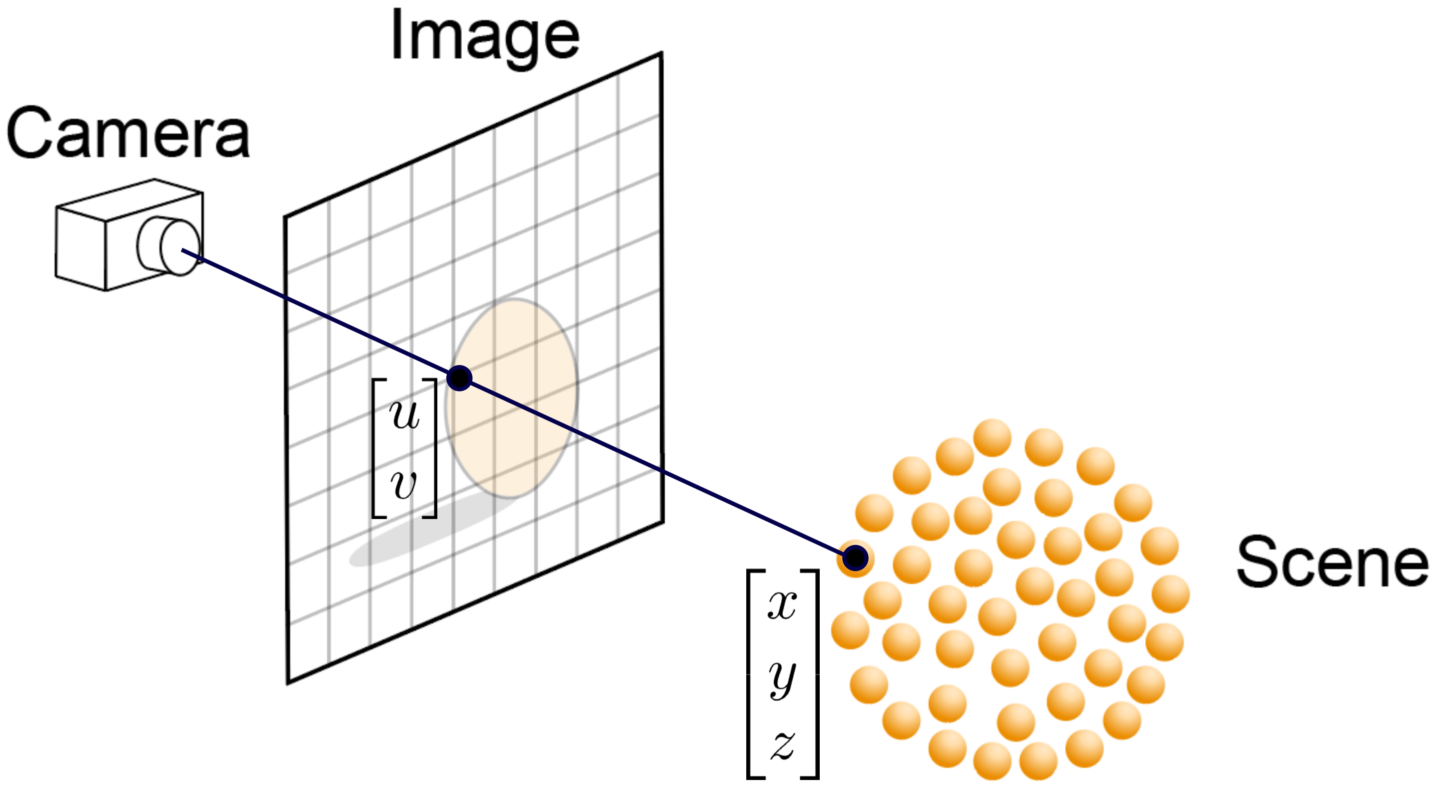
Rendering – Splatting

Approximate scene with spheres

- *sort spheres back-to front*
- *project each sphere*
  - simple equation

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix}$$

  - O(n) for n spheres



Many spheres needed!

Shadows?

Rendering – Rasterization

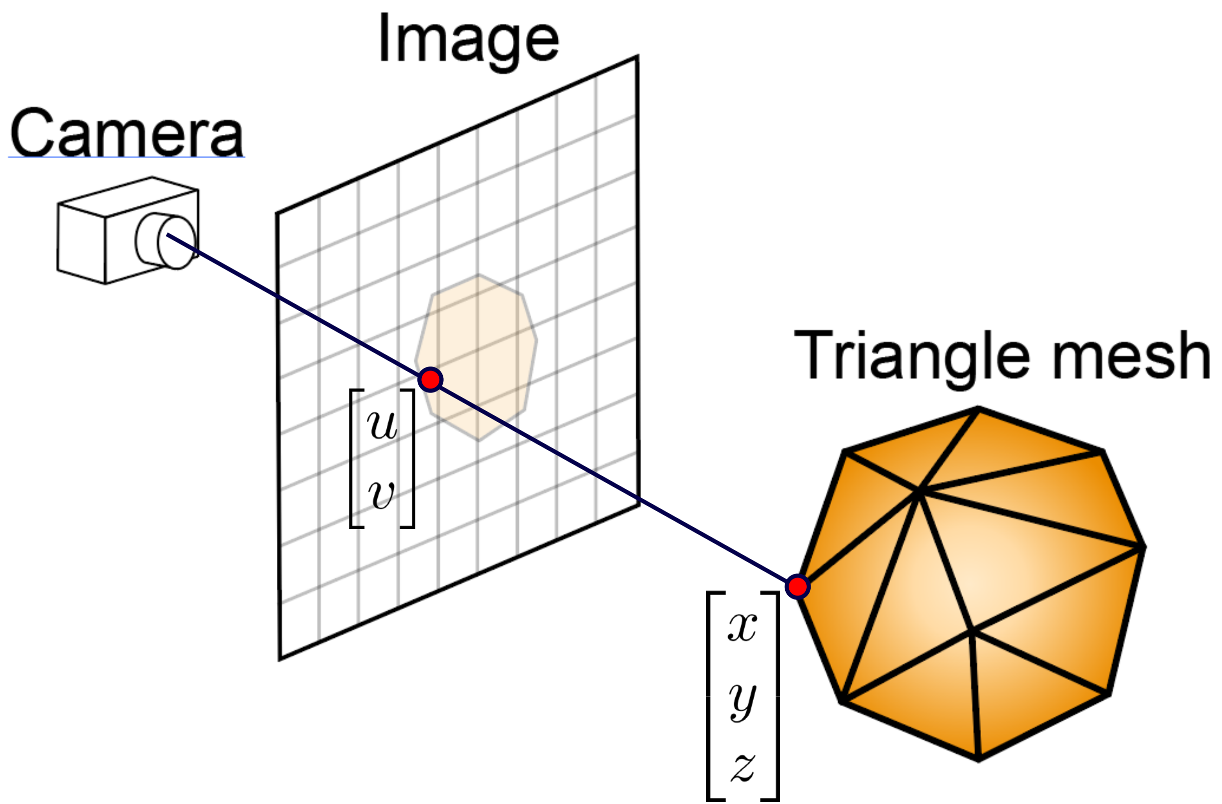
Approximate objects with triangles

1. *Project each corner/vertex*

- projection of triangle stays a triangle

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix}$$

- O(n) for n vertices

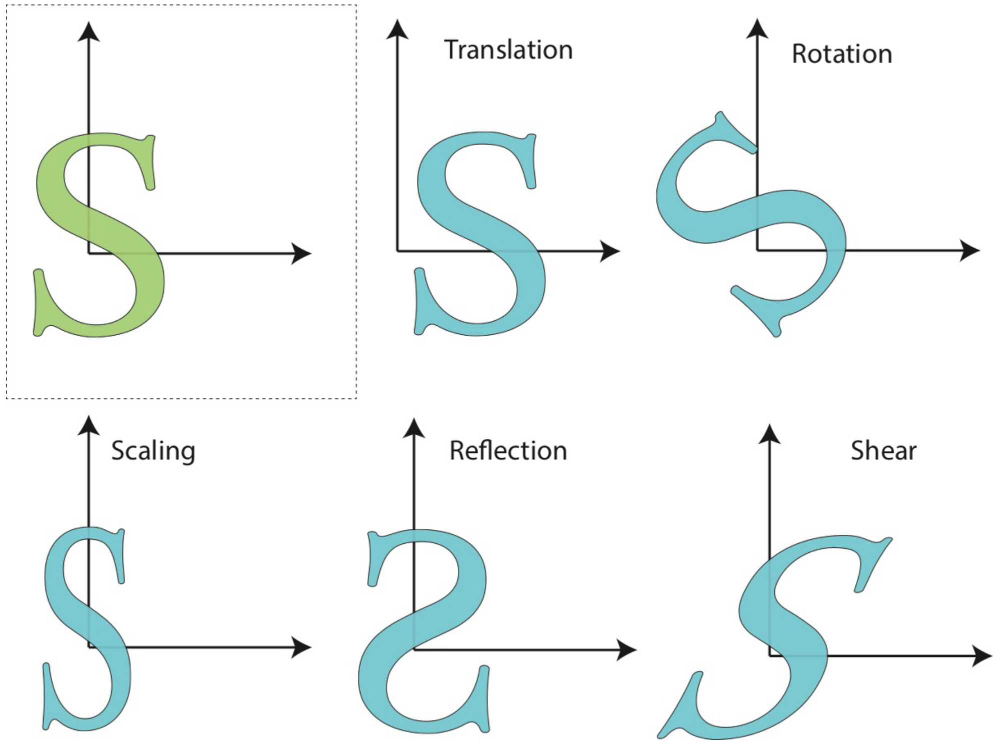


2. *Fill pixels enclosed by triangle*

- e.g., scan-line algorithm

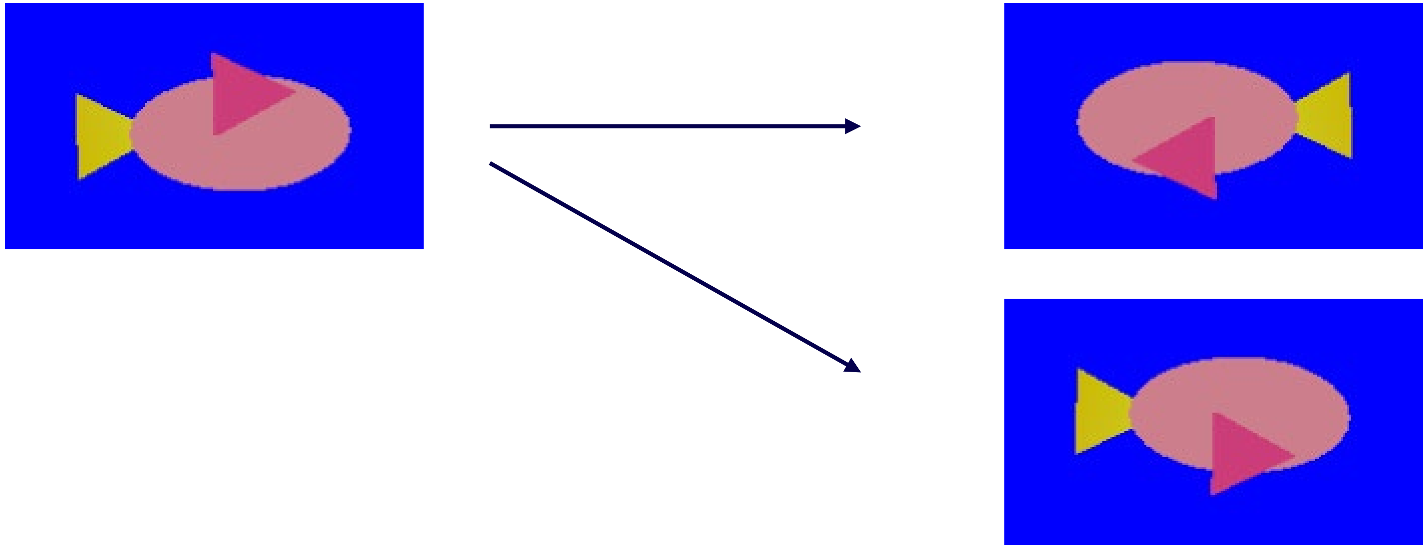
CPSC 427
Video Game Programming

Transformations



Helge Rhodin

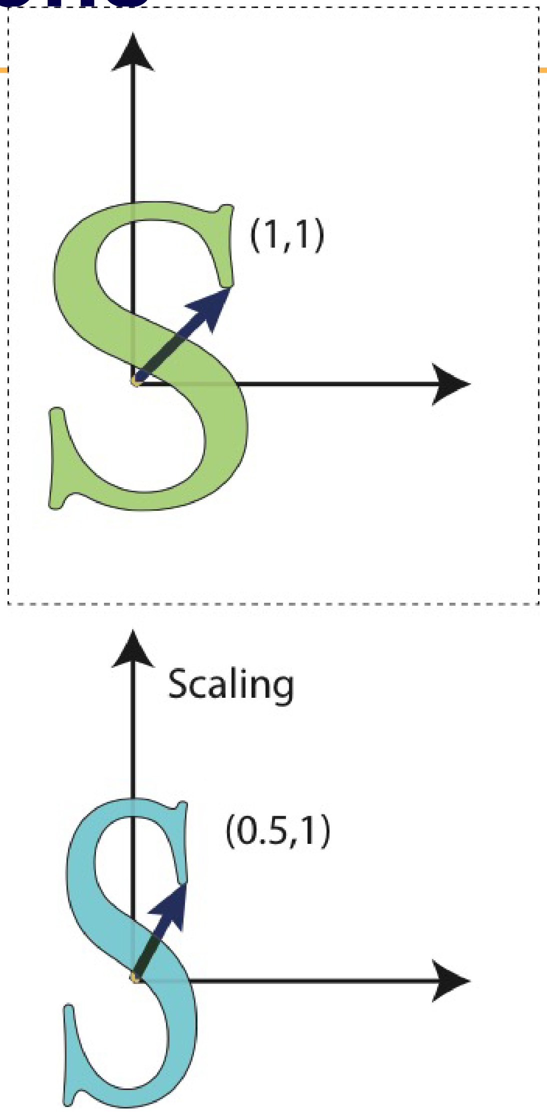
How to turn the fish?



Both versions are fine for Assignment 1 (A1)!

Matrix representations

Scale:



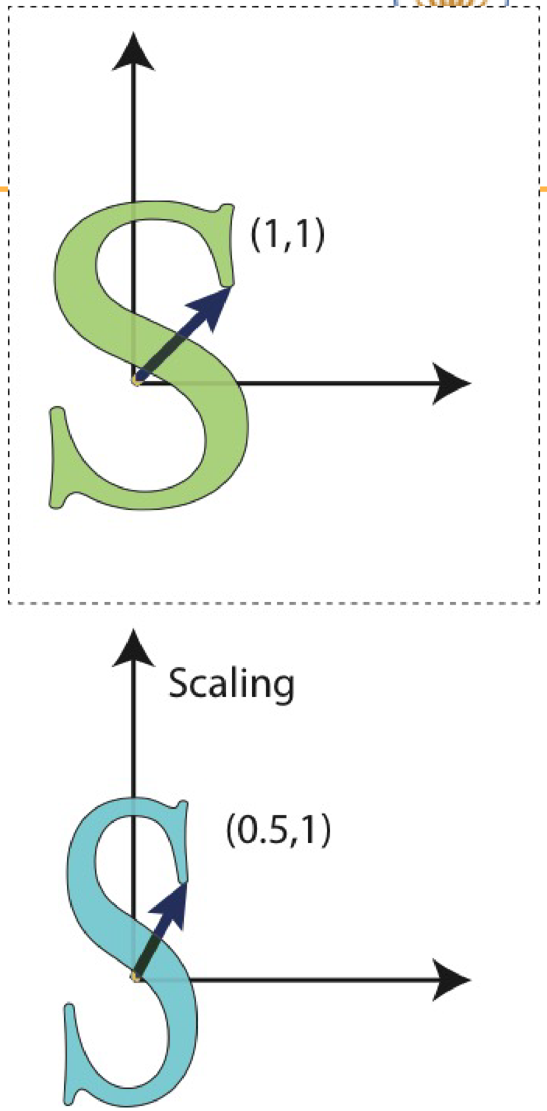
Matrix representations

Scale:

$$M = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$$

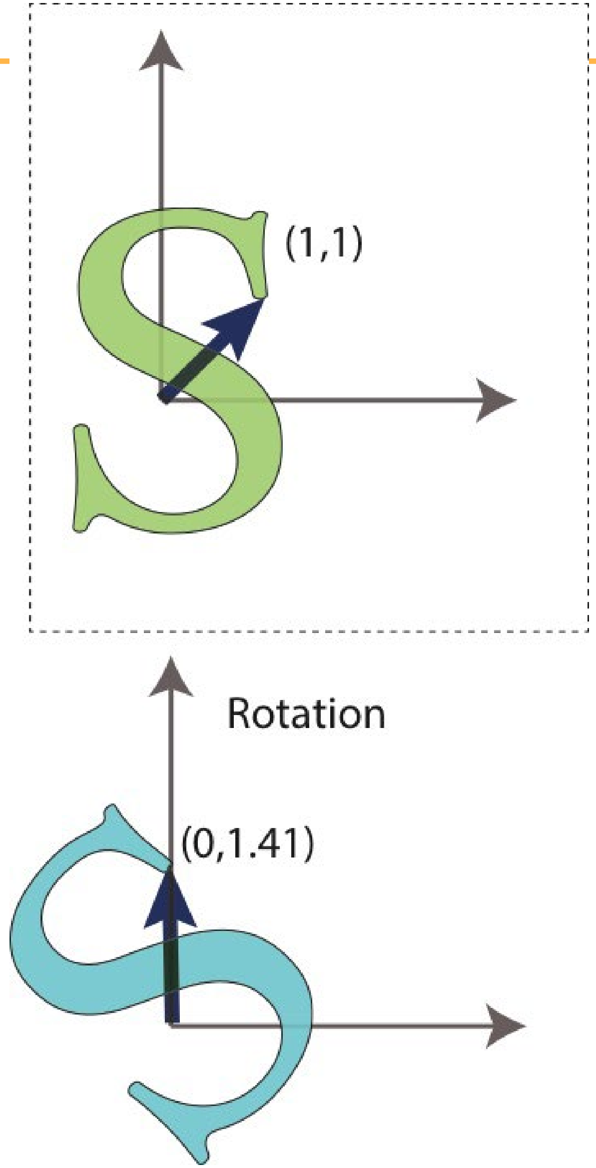
Example:

$$\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \alpha \\ 2\beta \end{pmatrix}$$



Matrix representations

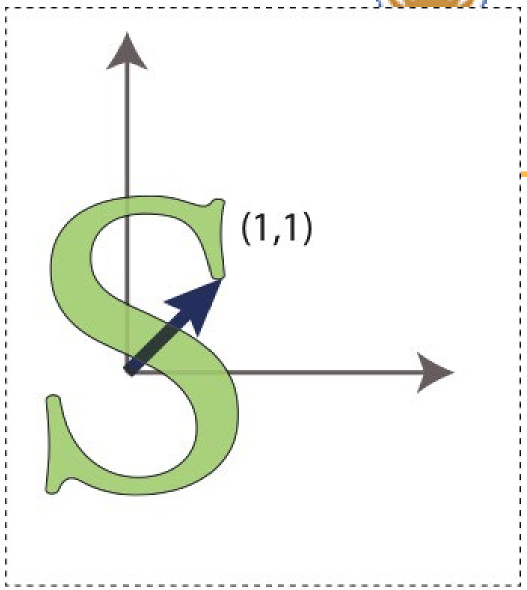
Rotation



Matrix representations

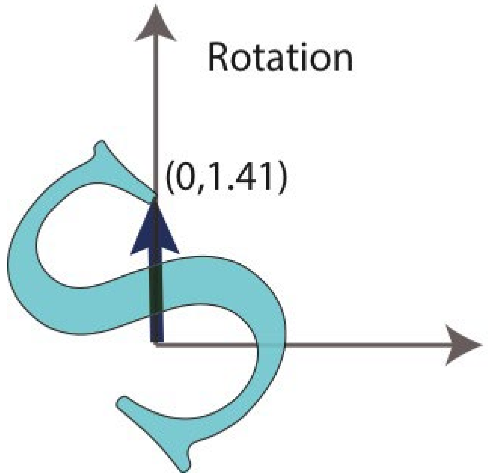
Rotation

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$



Example:

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) - \sin(\alpha) \\ \cos(\alpha) + \sin(\alpha) \end{pmatrix}$$



What does this 2D transformation do?

A. Rotates by 90 deg
B. Scales by a factor of 2
C. Rotates by -90 deg
D. Nothing

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

What does this 2D transformation do?

A. Rotates by 90 deg
B. Reflects the object
C. Rotates by -90 deg
D. Scales the object

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

TRANSLATION

There's a minor glitch.

- Translation: can't be represented as 2x2 matrix multiplication

general transformations

We need to represent all the linear transformations + translation.

Ideas?

$$T(\boldsymbol{v}) = M\boldsymbol{v} + \boldsymbol{b}$$

AUGMENTED MATRIX

$$M_{2x2}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{bmatrix} M_{2x2} & & \begin{matrix} 0 \\ 0 \end{matrix} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Haven't changed much, have we?

AUGMENTED MATRIX

$$\begin{bmatrix} M_{2x2} & \begin{matrix} b_x \\ b_y \end{matrix} \\ 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' + b_x \\ y' + b_y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} + \boldsymbol{b}$$

Translation

Affine transformations

- Linear (rotation, scaling, shear, reflections) + TRANSLATION

Affine transformations

- Linear (rotation, scaling, shear, reflections) + TRANSLATION

- How to convert a linear transformation matrix into affine matrix?

AFFINE Transformations

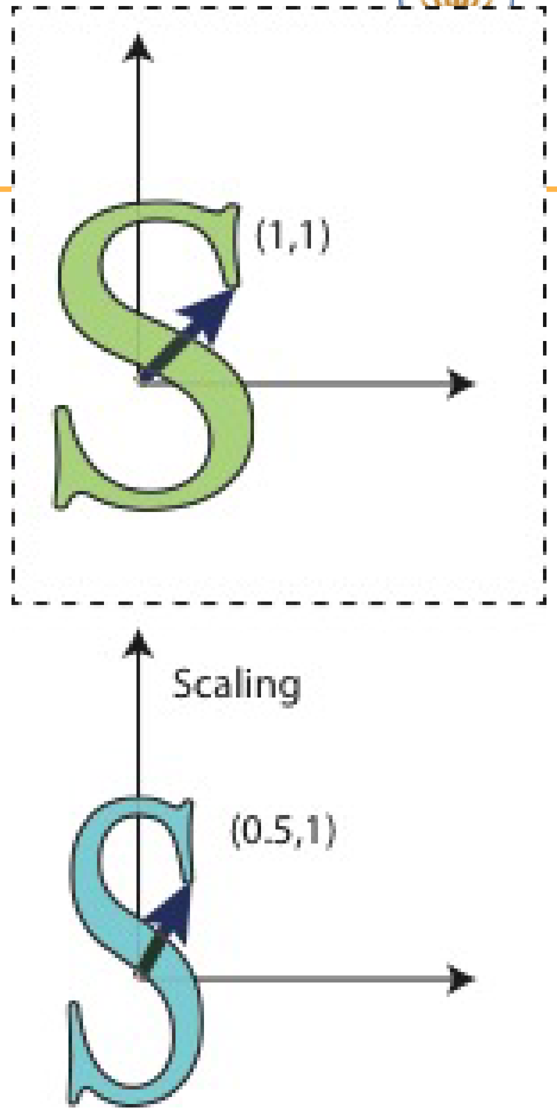
Scale:

$$M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

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

Example:

$$\begin{pmatrix} a \cdot 1 \\ b \cdot 2 \\ 1 \end{pmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$



AFFINE Transformations

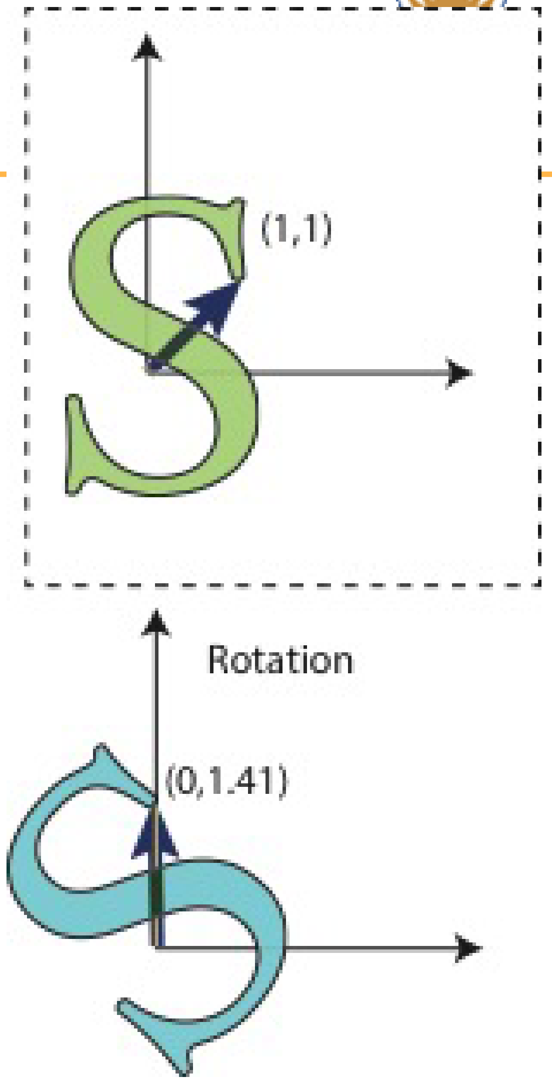
Rotation

$$M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

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

Example:

$$\begin{pmatrix} a \cdot 1 \\ b \cdot 2 \\ 1 \end{pmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$



AFFINE Transformations

Translation

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

Example:

$$\begin{bmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + C_x \\ y + C_y \\ 1 \end{pmatrix}$$

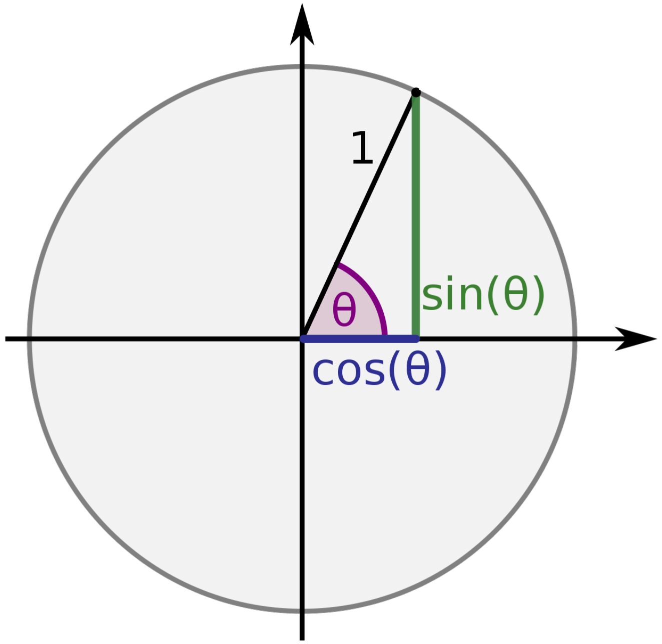
Linear transformations

- Rotations, scaling, shearing
- Can be expressed as 2x2 matrix (for 2D points)
- E.g.

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

- or a rotation

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$



Rotation angle θ, cos, and sin

https://en.wikipedia.org/wiki/Trigonometric_functions

Affine transformations

- Linear transformations + translations
- Can be expressed as 2x2 matrix + 2 vector
- E.g. scale + translation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Modeling Transformation

Adding a third coordinate

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \Rightarrow \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 0 & t_x \\ 0 & 2 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Affine transformations are now linear

- one 3x3 matrix can express: 2D rotation, scale, shear, and translation

Combination of Transformations?

- ***How can we combine***
  - translation
  - rotation
  - scaling

… into one matrix?

Self study: Homogeneous coordinates

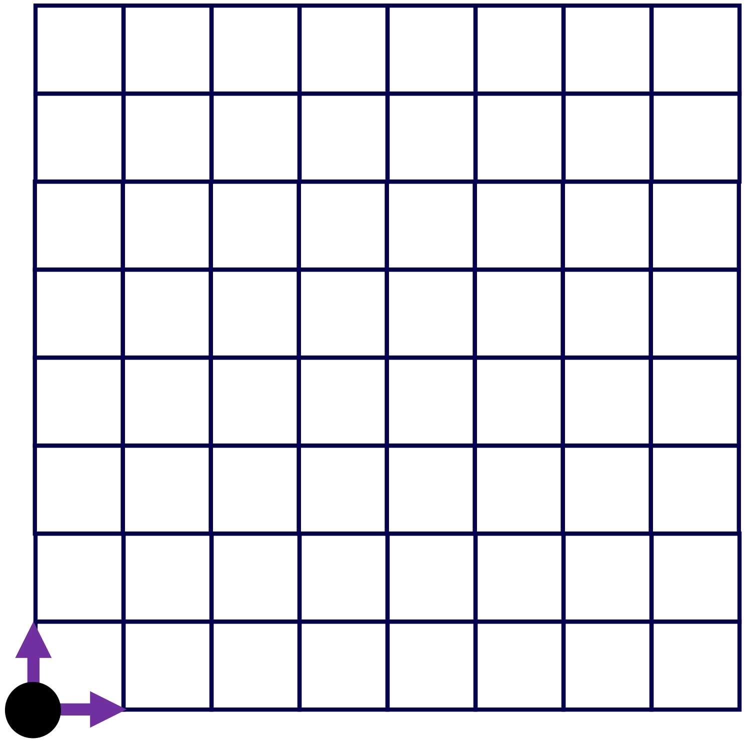
- Homogeneous coordinates are defined as vectors, with equivalence

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix} = \begin{pmatrix} x\lambda \\ y\lambda \\ z\lambda \end{pmatrix}$$

- Can also represent projective equations

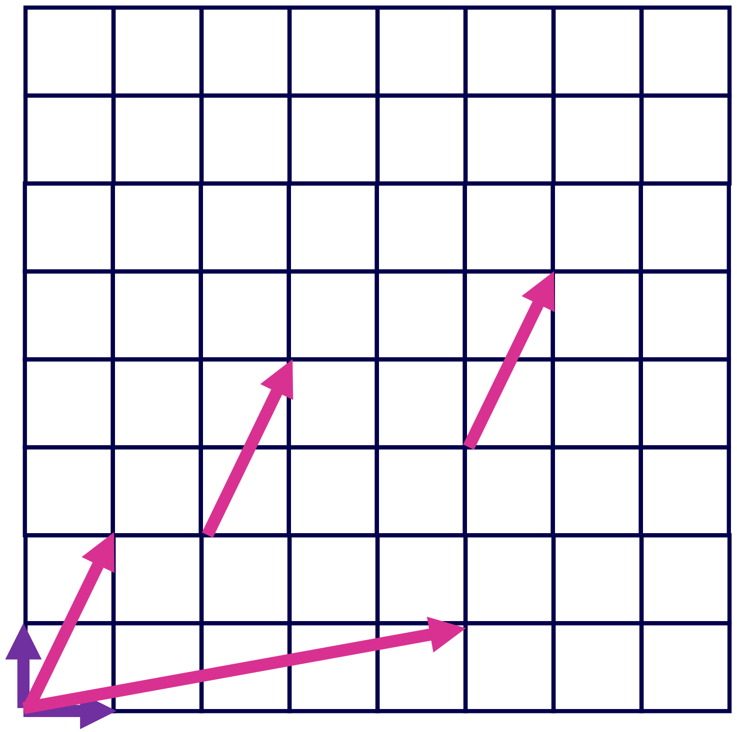
COORDINATE SYSTEMS

Coordinate system = Origin + Basis Vectors

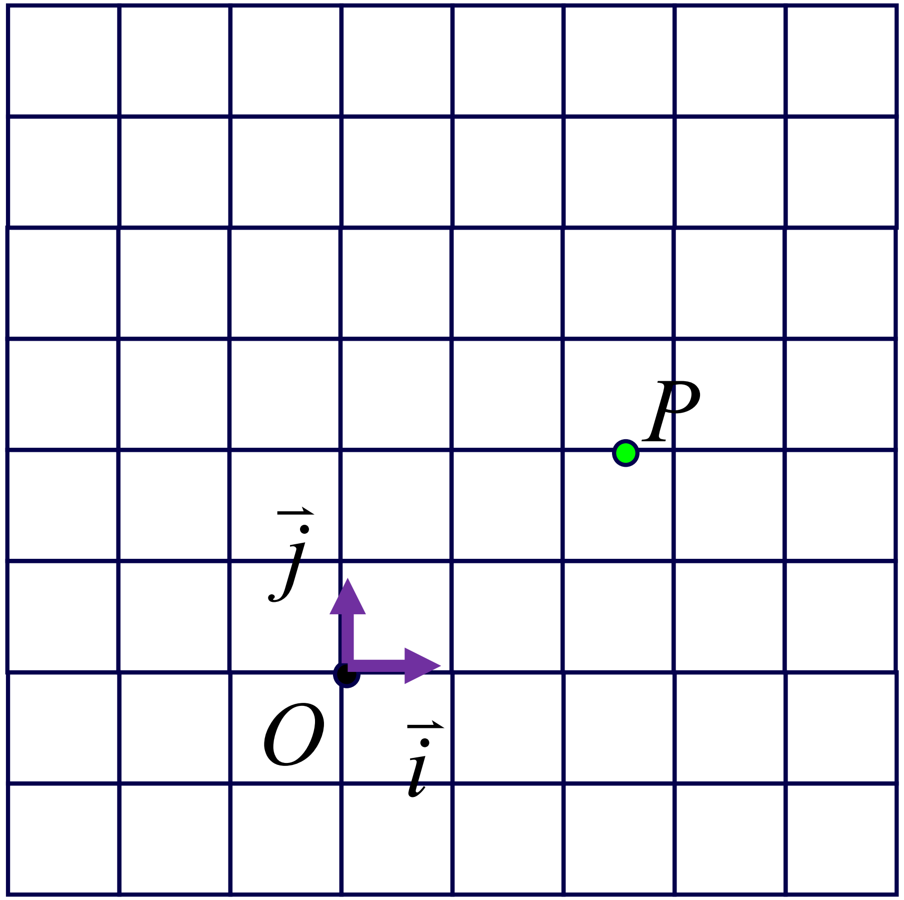


COORDINATE SYSTEMS

Coordinate system = Origin + Basis Vectors



COORDINATE SYSTEMS

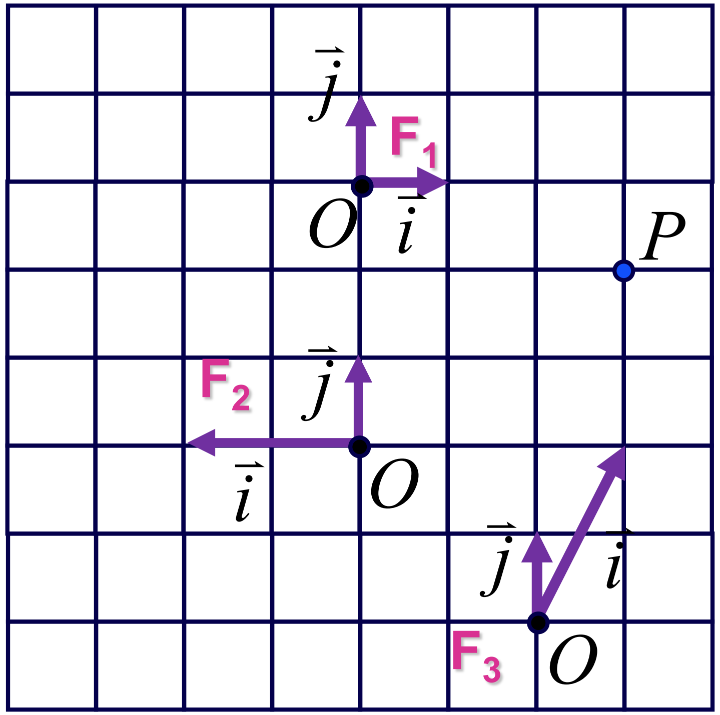


$$P = O + x\vec{i} + y\vec{j}$$

equivalent: $P = (x, y)$

COORDINATE SYSTEMS

What is the position of P in each of the coordinate frames?



$$P = O + x\vec{i} + y\vec{j}$$

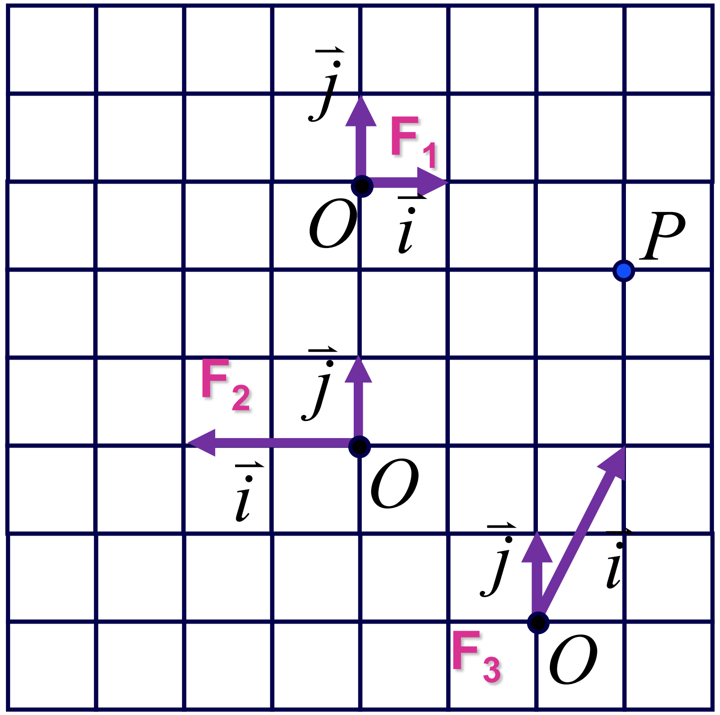
F_1

F_2

F_3

COORDINATE SYSTEMS

What is the position of P in each of the coordinate frames?



$$P = O + x\vec{i} + y\vec{j}$$

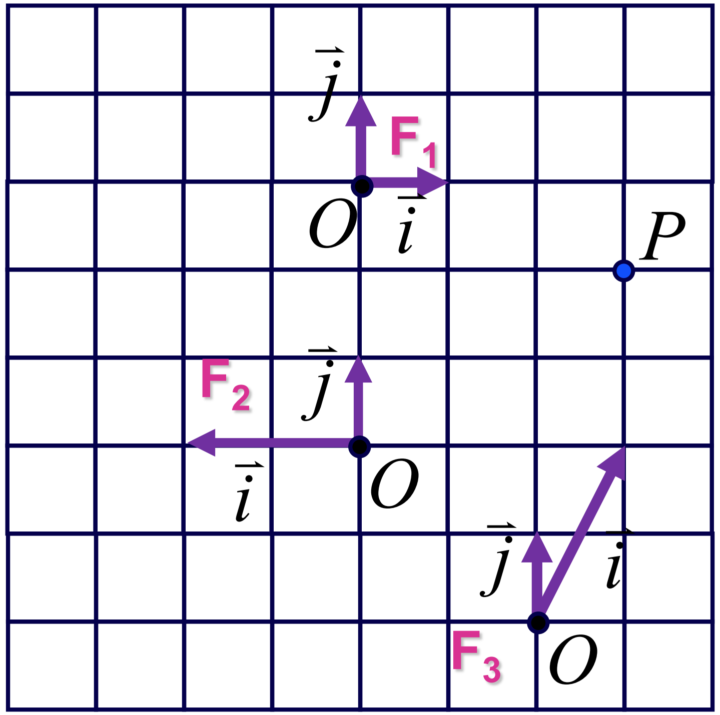
F_1 **P(3,-1)**

F_2

F_3

COORDINATE SYSTEMS

What is the position of P in each of the coordinate frames?



$$P = O + x\vec{i} + y\vec{j}$$

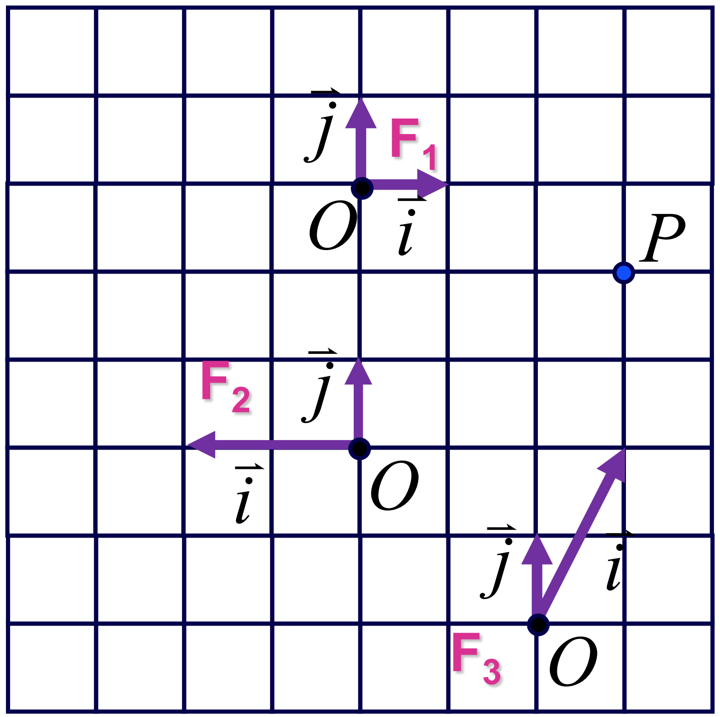
F_1 **P(3,-1)**

F_2 **P(-1.5,2)**

F_3

COORDINATE SYSTEMS

What is the position of P in each of the coordinate frames?



$$P = O + x\vec{i} + y\vec{j}$$

F_1 **P(3,-1)**

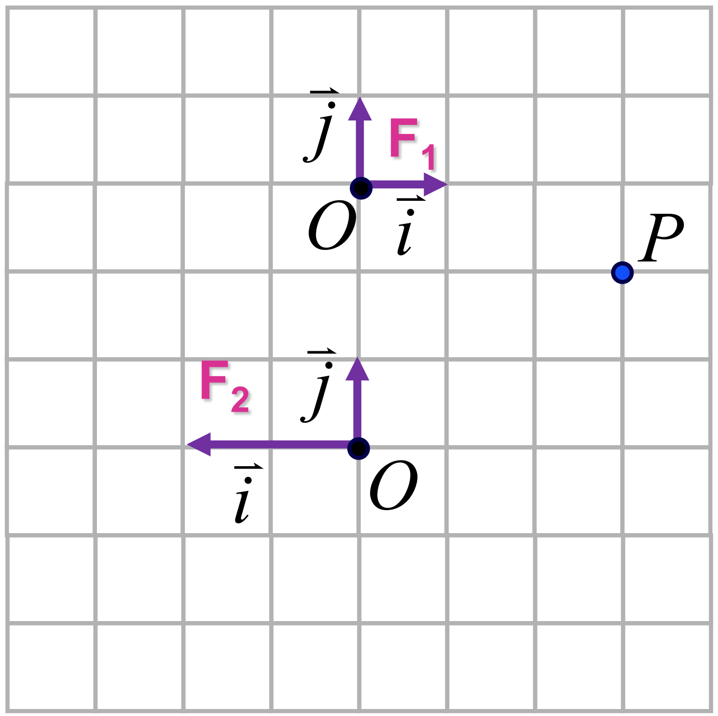
F_2 **P(-1.5,2)**

F_3 **P(1,2), other solutions possible?**

Transformations

Transformations as a change of frame

$$P = O + x\vec{i} + y\vec{j}$$



$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_1 + x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix}_1 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_1$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}_2 + x_1 \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}_2 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_2$$

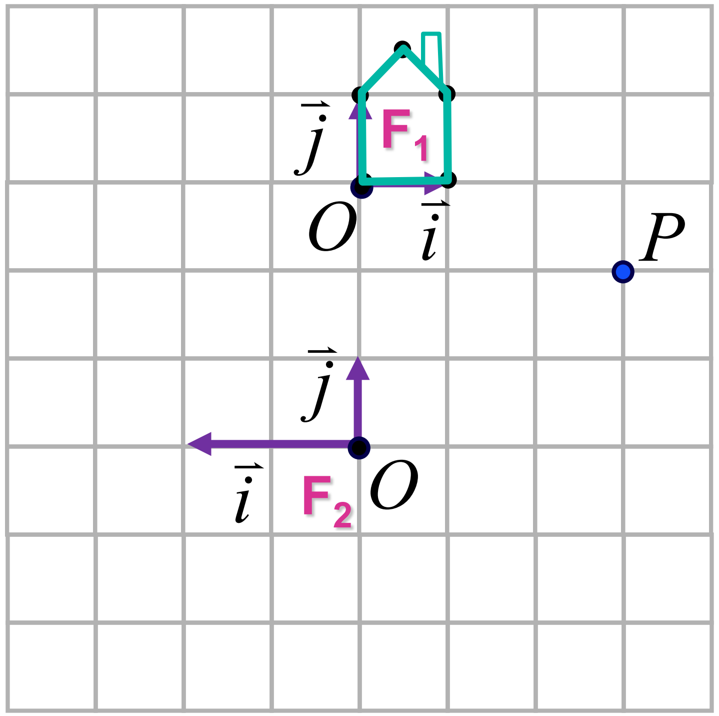
$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

check: $P_1(3,-1)$ becomes $P_2(-1.5,2)$

$$P_2 = MP_1$$

TRANSFORMATIONS

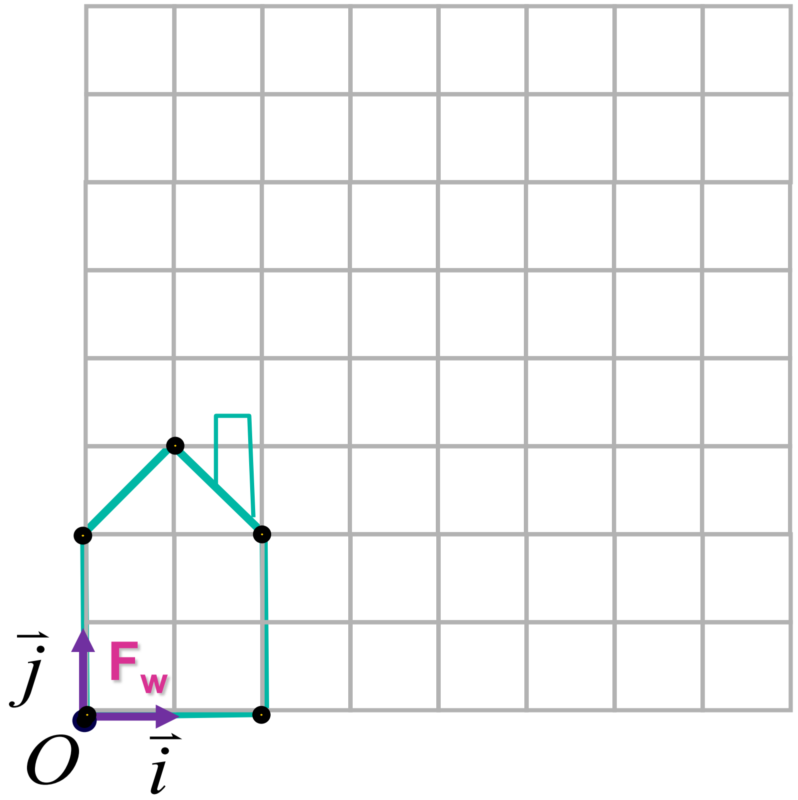
change of basis expressed using a matrix



$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

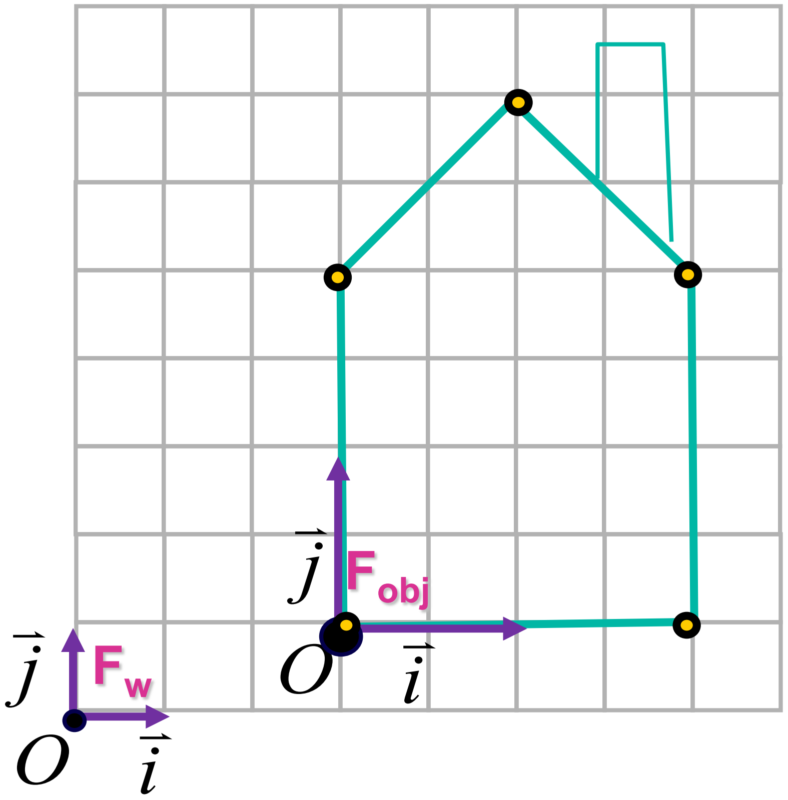
Usage of Transformations



set up the modeling matrix M

for each vertex v
  v' = Mv

Usage of Transformations



$$P = O + x\vec{i} + y\vec{j}$$

$$\begin{bmatrix} x_{obj} \\ y_{obj} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{obj} + x_{obj} \begin{bmatrix} 1 \\ 0 \end{bmatrix}_{obj} + y_{obj} \begin{bmatrix} 0 \\ 1 \end{bmatrix}_{obj}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_w = \begin{bmatrix} 3 \\ 1 \end{bmatrix}_w + x_{obj} \begin{bmatrix} 2 \\ 0 \end{bmatrix}_w + y_{obj} \begin{bmatrix} 0 \\ 2 \end{bmatrix}_w$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

Using Transformations

$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

2D

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

3D

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$