CSC 4356
Interactive Computer Graphics
Lecture 4: Geometric Transformations (3D)

Jinwei Ye

http://www.csc.lsu.edu/~jye/CSC4356/

Tue & Thu: 10:30 - 11:50am
218 Tureaud Hall

Update

- TA: Sirazum M Tisha
  - Email: stisha1@lsu.edu
  - Office hour: 2:00 – 4:00pm, Wednesday
  - Location: Lobby of 3272 Zone, PFT

Lecture 4: Geometric Transformations (3D)

- 3-Dimensional Transformations
  - Translation, Scaling, Rotation …
  - Inverse transformation
  - Composite transformation
- Coordinate Transformation
- Classes of Transformations
- Reading:
  - Textbook Chap 9

3D Transformation

- Recap: 2D transformations apply to 2D primitives and image coordinates
- Why we need 3D transformations?
- Computer graphics deals with 3D models
  - Put different models together
  - Arrange models
  - See a model from different angles

Homogeneous Coordinate

- Add an extra dimension
  - In 2D, we have $[x, y, 1]^T$
  - In 3D, we have $[x, y, z, 1]^T$
- Transformations are represented by matrix multiplication

Rotation, scaling, shear, reflection (upper-left 3×3 block)

Translation (fourth column, rows 1–3)

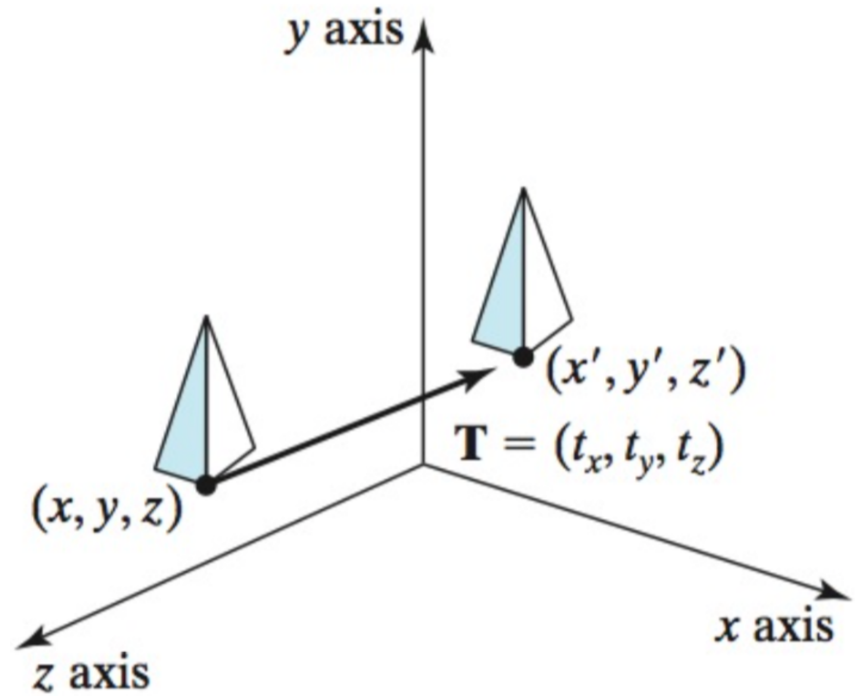
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Translation

- Change the location of object

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

$$P' = TP$$

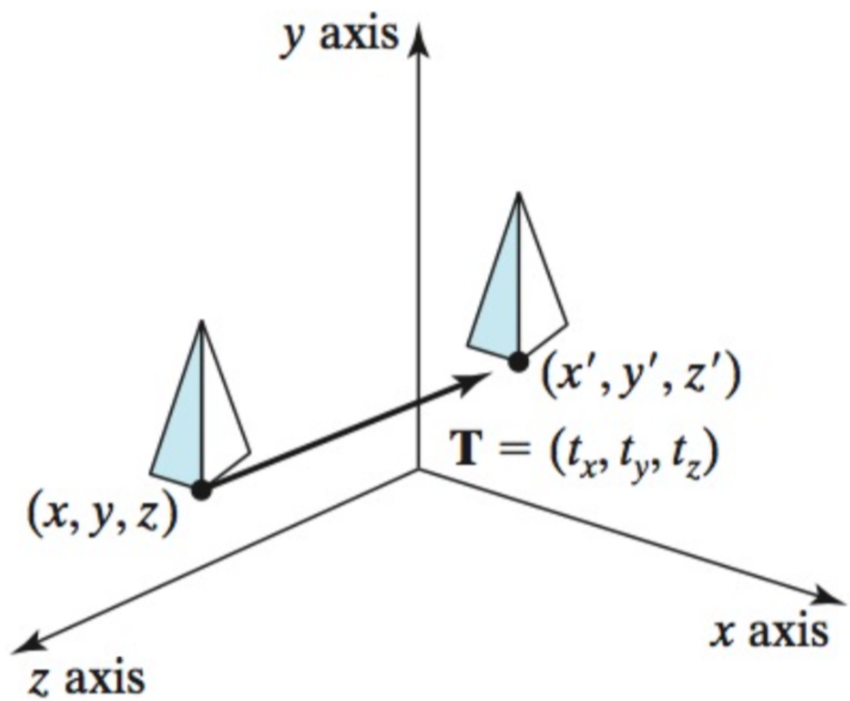


3D Translation

- Inverse transformation

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

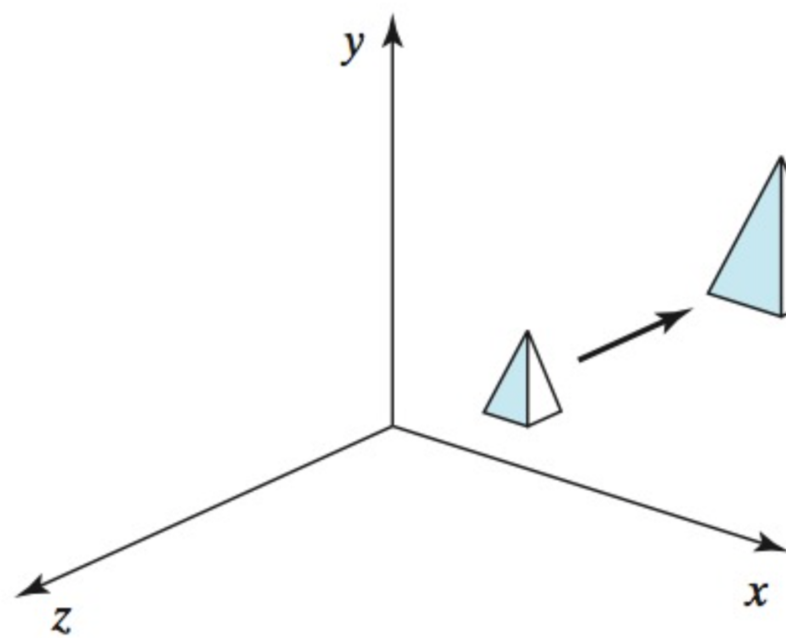


3D Scaling

- Change the size of object
  - Also reposition the object!

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = SP$$

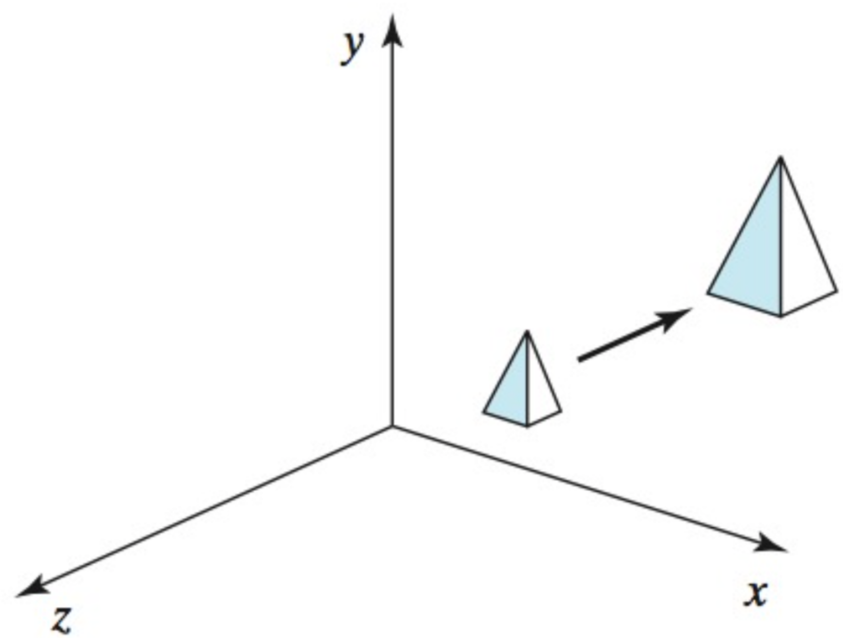


3D Scaling

- Inverse transformation

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

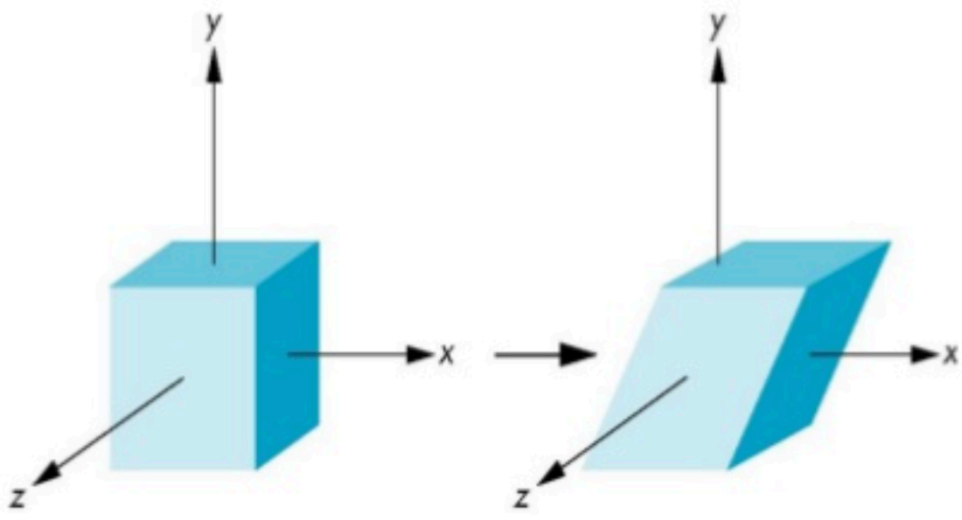
$$S^{-1} = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



3D Shear

- Distort the shape of object proportional to some distance
  - x-direction, y-direction, z-direction
  - reference point

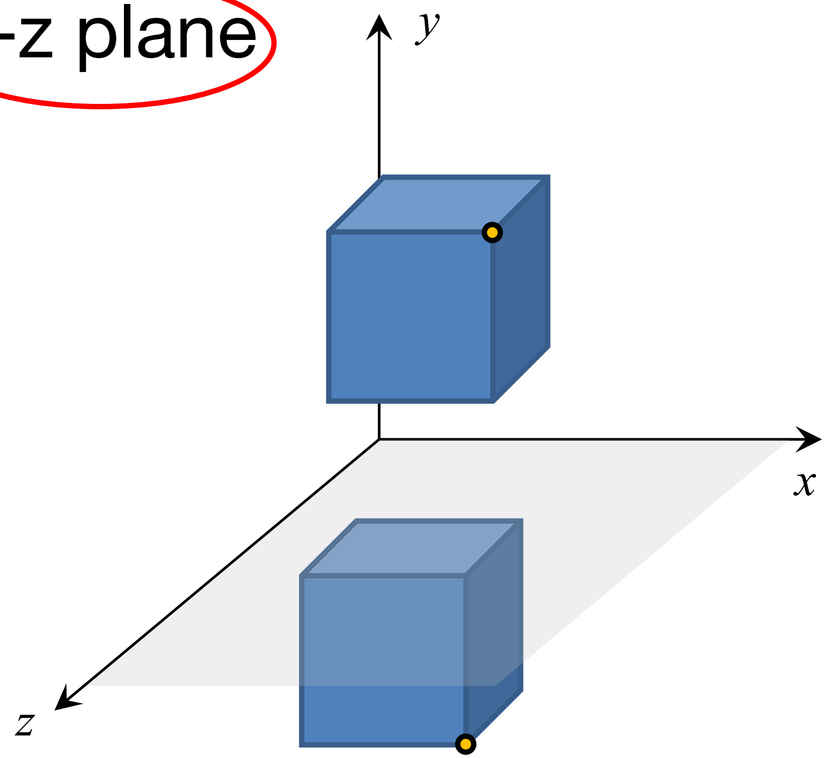
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



3D Reflection

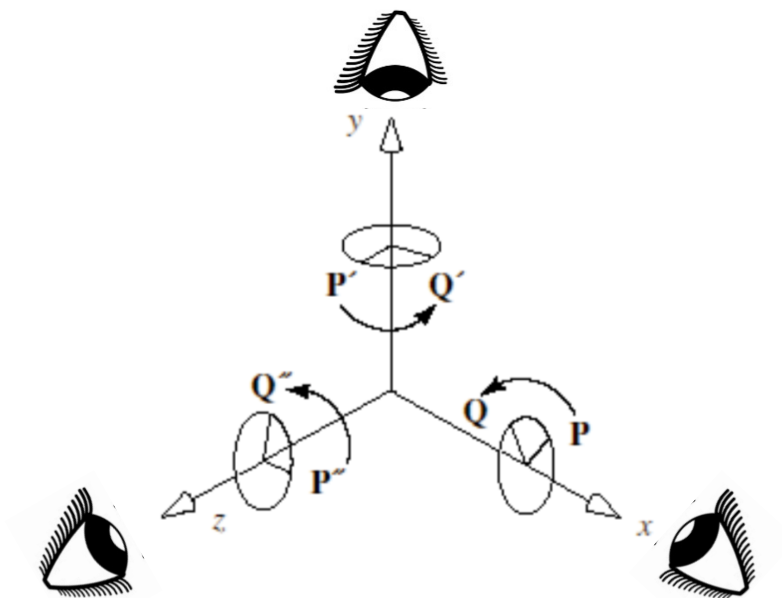
- Flip about a reference plane
  - x-y plane, y-z plane, x-z plane
  - Arbitrary plane?

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



3D Rotation

- 3D rotation is about a rotation axis,
  - Basic rotation matrix rotates about the three coordinate axes
  - Positive rotations are counter-clockwise when viewing towards the rotation axis

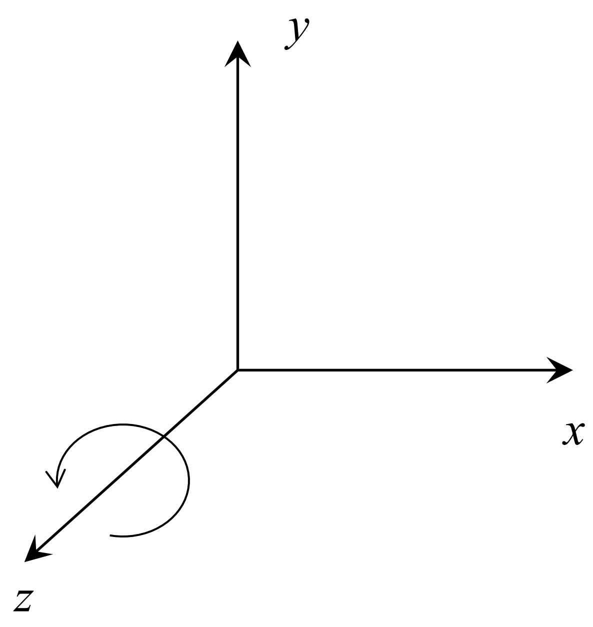


3D Rotation (z-axis)

- Rotate about the z-axis

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

$$P' = R_z(\theta) P$$

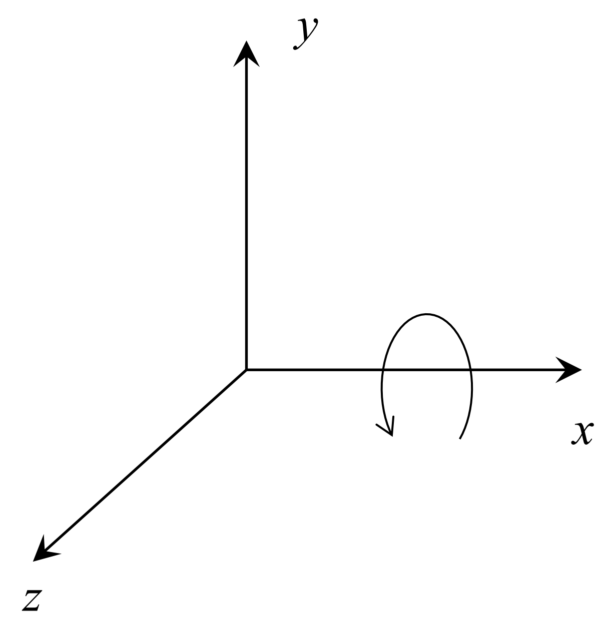


3D Rotation (x-axis)

- Rotation about x-axis

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

$$P' = R_x(\theta) P$$

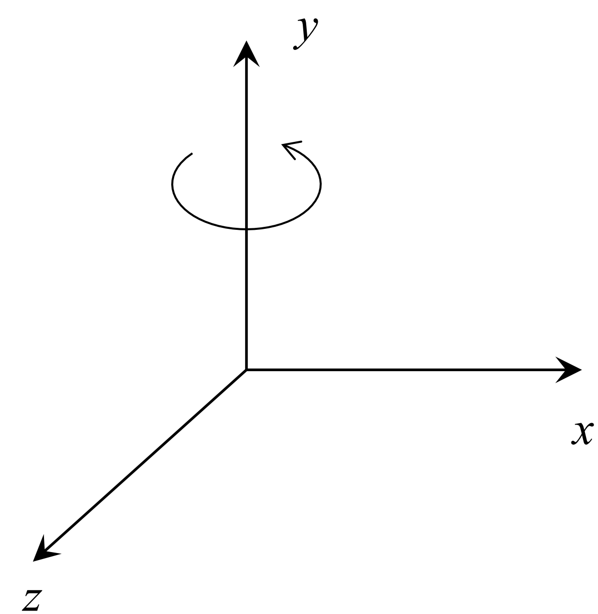


3D Rotation (y-axis)

- Rotation about the y-axis

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

$$P' = R_y(\theta) P$$



More About Rotation

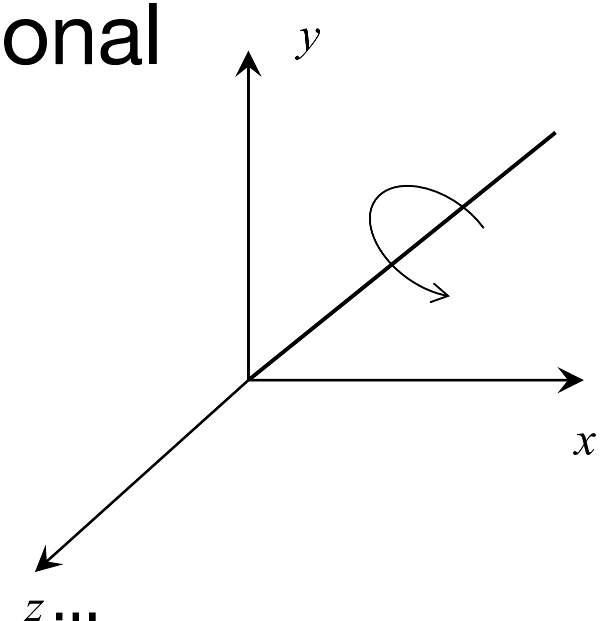
- Rotation matrices are orthogonal

$$R_z^{-1}(\theta) = R_z^{T}(\theta) = R_z(-\theta)$$

- Inverse Rotation

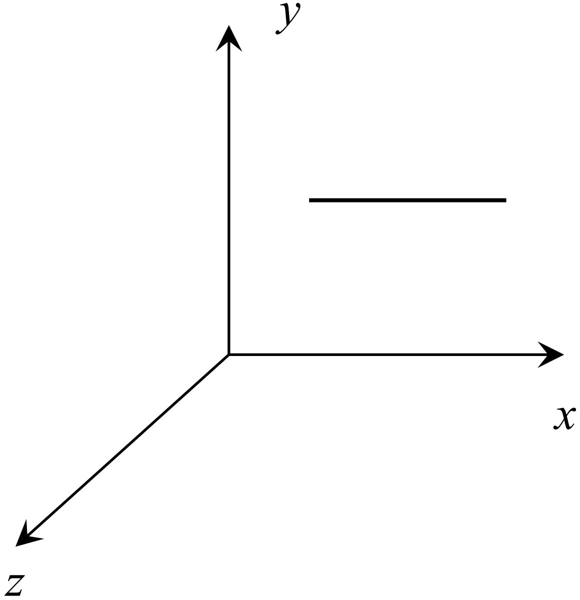
$$P = R_z^{-1}(\theta)P' = R_z^{T}(\theta)P' = R_z(-\theta)P'$$

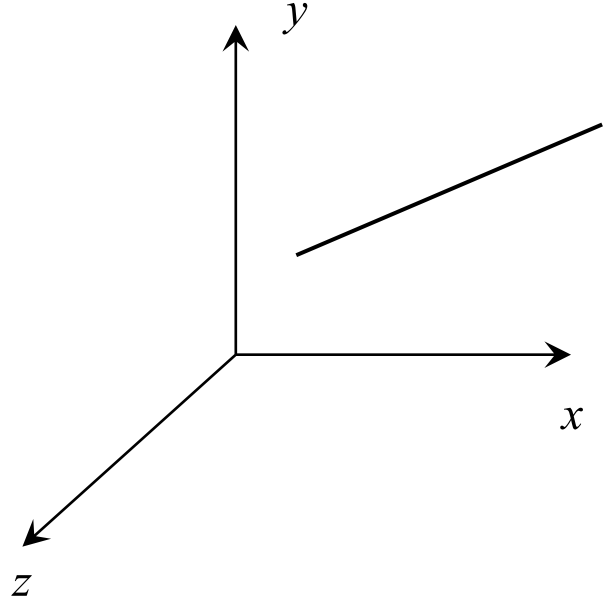
- Composite Rotation
  - Combination of R_x, R_y and R_z will perform any rotation about an axis passing through the origin.



Arbitrary Rotation?

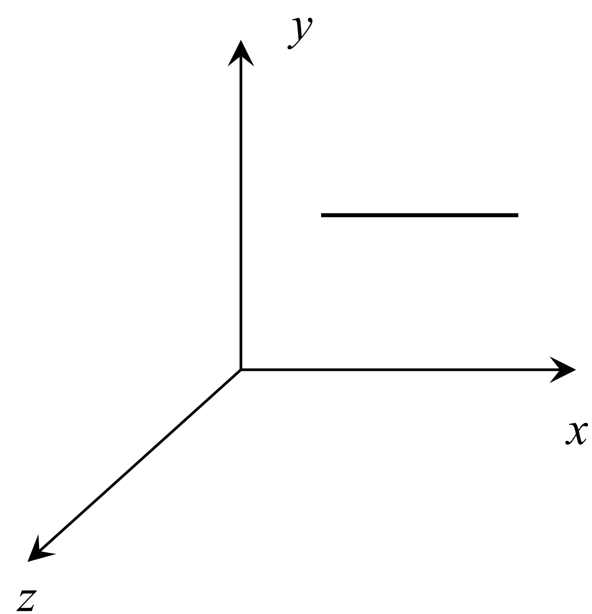
- Rotation axis parallel to one of the coordinate axis
- Rotation axis non-parallel to all coordinate axes





Arbitrary Rotation

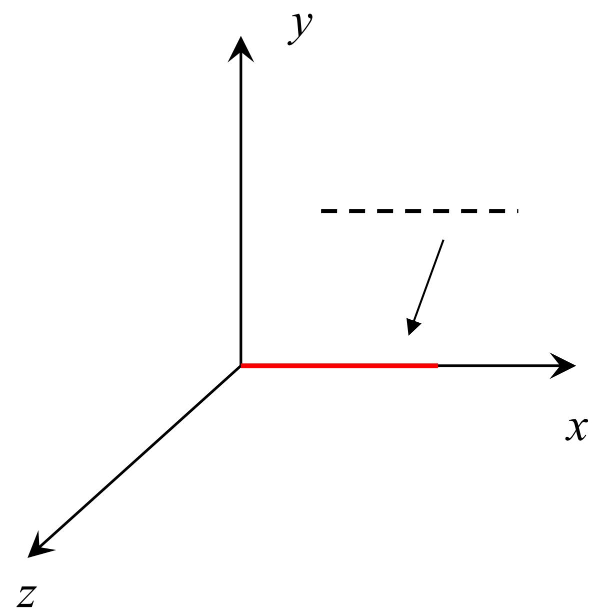
- Rotation axis parallel to one of the coordinate axis



Arbitrary Rotation

- Rotation axis parallel to one of the coordinate axis

Step 1: translate rotation axis to pass through origin

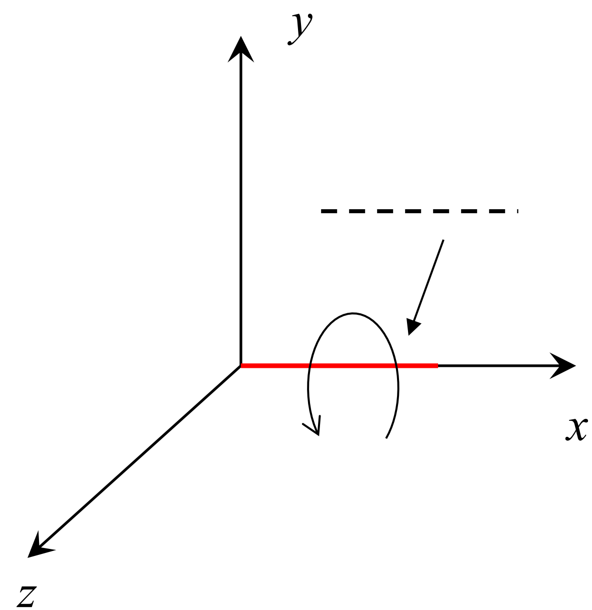


Arbitrary Rotation

- Rotation axis parallel to one of the coordinate axis

  Step 1: translate rotation axis to pass through origin

  Step 2: rotate



Arbitrary Rotation

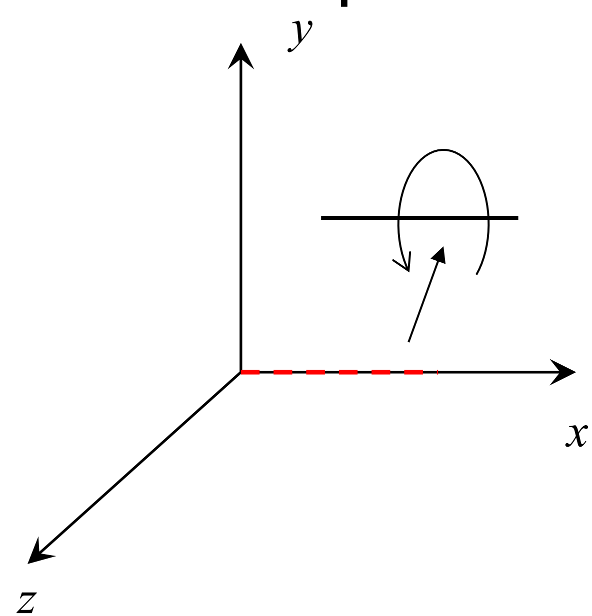
- Rotation axis parallel to one of the coordinate axis

  Step 1: translate rotation axis to pass through origin

  Step 2: rotate

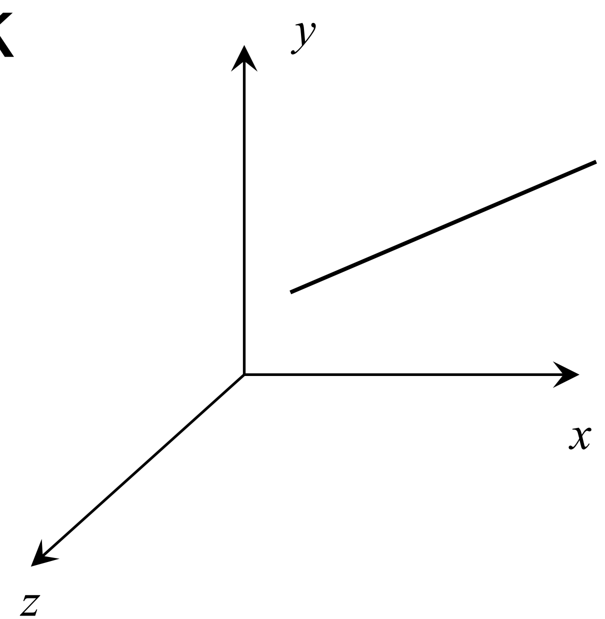
  Step 3: translate back

$$P' = T^{-1} R_x(\theta) T P$$



Arbitrary Rotation

- Rotation axis non-parallel to all coordinate axes
- Idea: make the rotation axis coincide with one of the coordinate axes, rotate, and then transform back



Arbitrary Rotation

- Rotation axis non-parallel to all coordinate axes

  Step 1: translate to origin

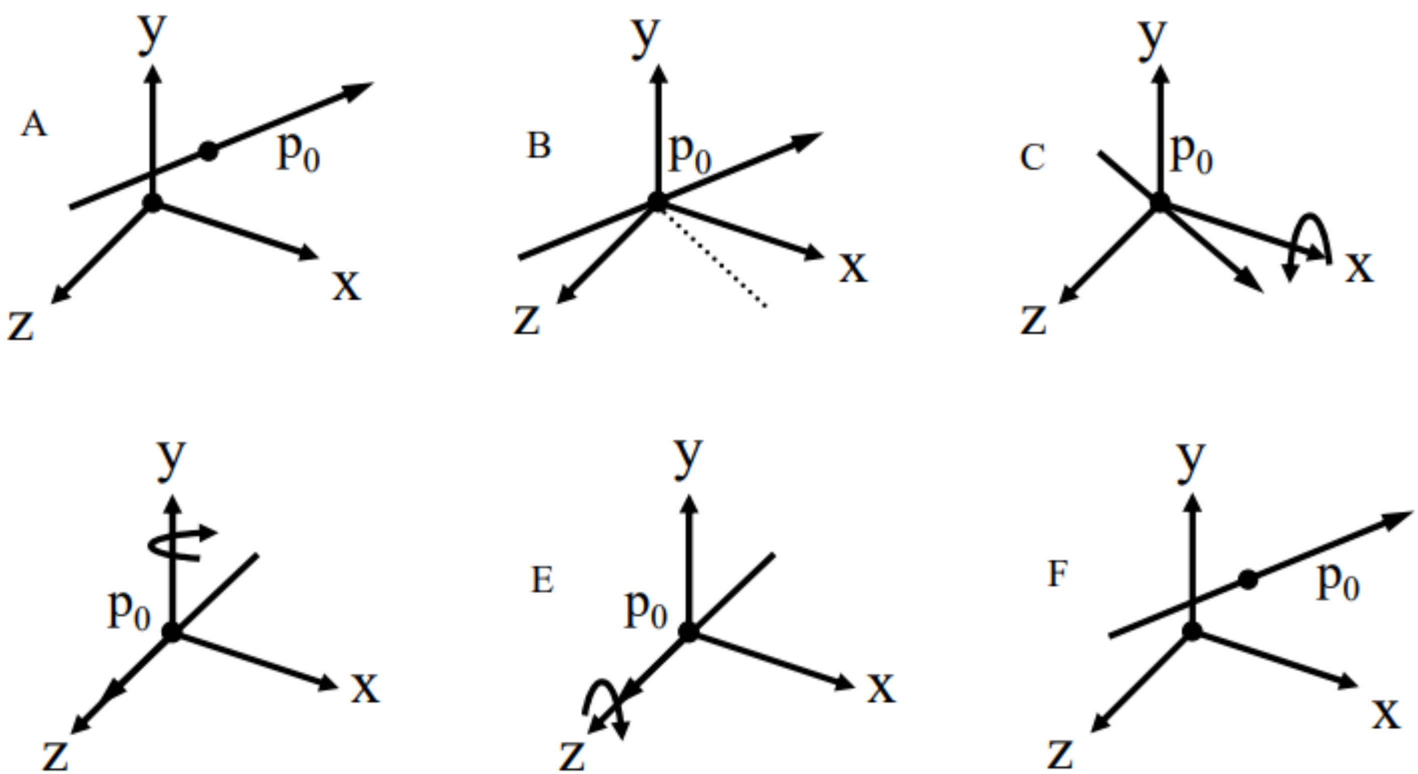
  Step 2: rotate axis to align with one coordinate axis (e.g., z-axis)

  Step 3: rotate the object

  Step 4: rotate axis back to its original orientation

  Step 5: translate to its original position

Arbitrary Rotation



$$P' = T^{-1} R_x(-\theta_x) R_y(-\theta_y) R_z(\theta) R_y(\theta_y) R_x(\theta_x) T P$$

3D Rotation

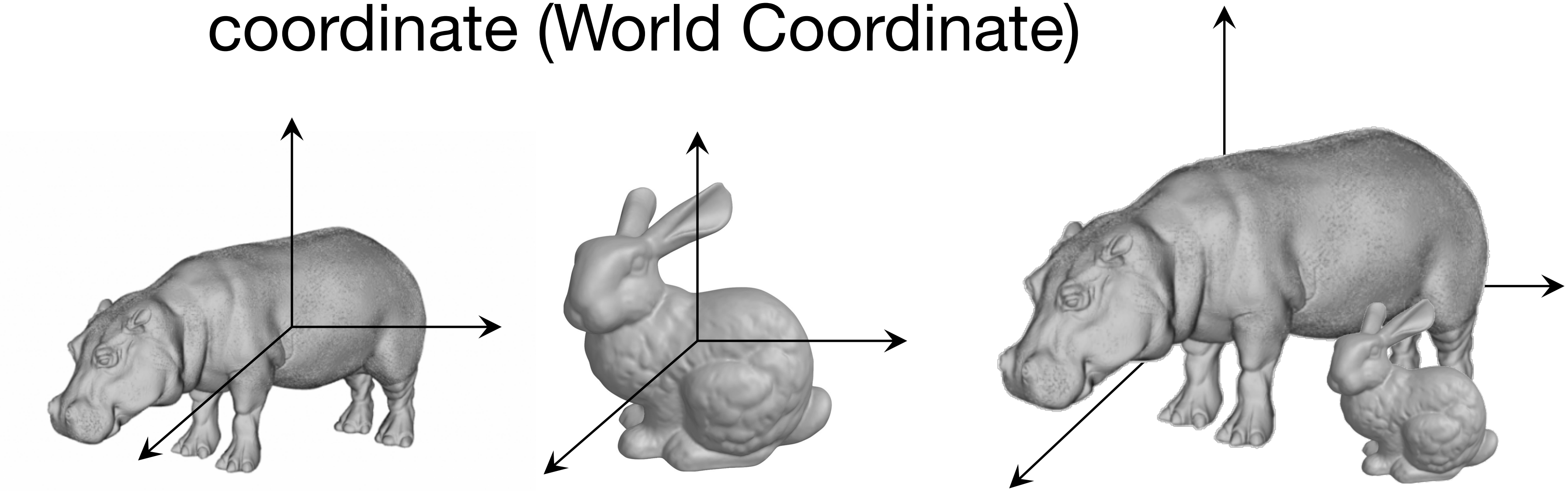
- 3D rotations are defined by rotation axis and rotation angle
  - A rotation axis has 6 DoF: Position (3DoF) & Direction (3DoF)
- Rotations matrix are orthonormal

$$P' = T^{-1} R_x(-\theta_x) R_y(-\theta_y) R_z(\theta) R_y(\theta_y) R_x(\theta_x) T P \Rightarrow P' = RP$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R^{-1} = R^T$$

Coordinate Transformation

- Why we need to transform coordinates?
  - 3D models are often in its own coordinate space
  - When constructing a 3D scene, we need to transform different objects into a common coordinate (World Coordinate)



3D Transformation in CG

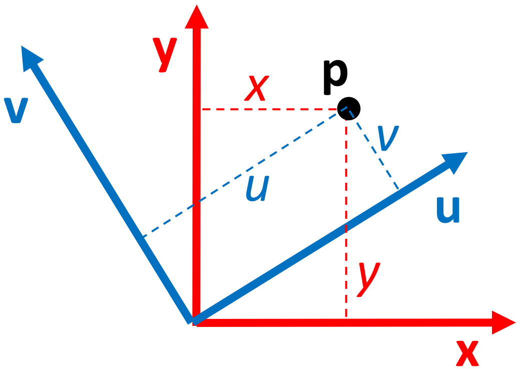
- Model Transformation
  - Transform a model within its own coordinate system
- World Transformation
  - Transformations between different coordinate systems

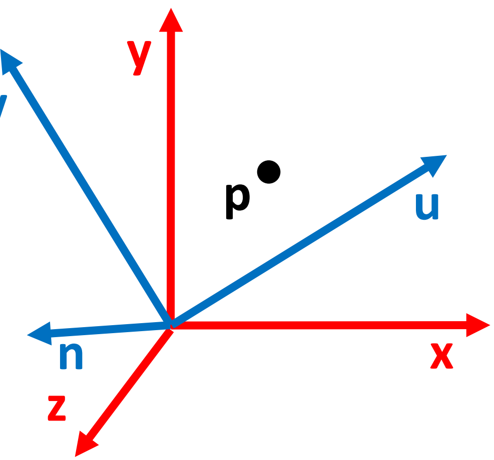
Coordinate Transformation

- Given:
  - coordinate frames

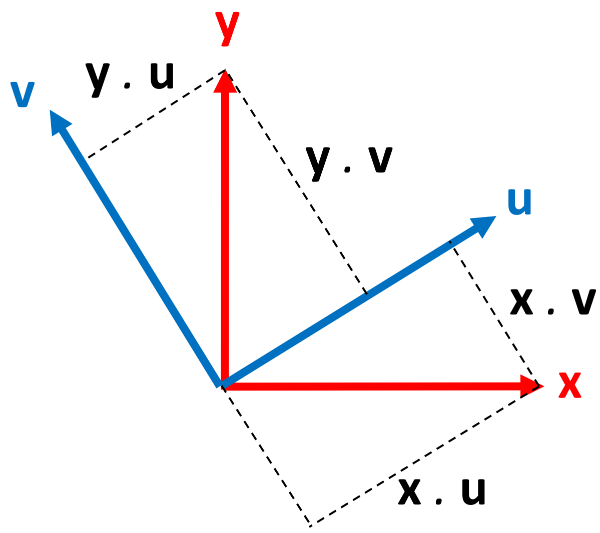
    xyz and **uvn**
  - point **p**=(x,y,z)

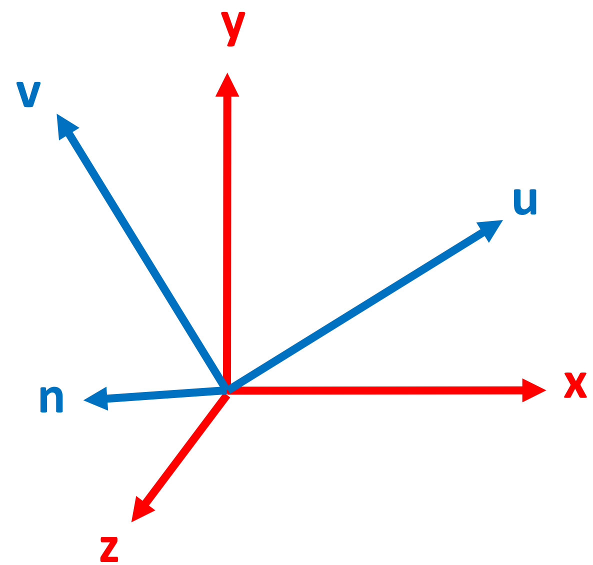
- Find:
  - **p**=(u,v,n)





Change of Orthonormal Basis





$$\mathbf{x} = (\mathbf{x} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{x} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{x} \cdot \mathbf{n})\,\mathbf{n}$$

$$\mathbf{y} = (\mathbf{y} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{y} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{y} \cdot \mathbf{n})\,\mathbf{n}$$

$$\mathbf{z} = (\mathbf{z} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{z} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{z} \cdot \mathbf{n})\,\mathbf{n}$$

- Coordinates are orthogonal basis (unit vectors)
- Find projection by dot product

Find P's Projection

$$\mathbf{x} = (\mathbf{x} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{x} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{x} \cdot \mathbf{n})\,\mathbf{n}$$
$$\mathbf{y} = (\mathbf{y} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{y} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{y} \cdot \mathbf{n})\,\mathbf{n}$$
$$\mathbf{z} = (\mathbf{z} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{z} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{z} \cdot \mathbf{n})\,\mathbf{n}$$

Substitute into equation for *p*:

$$\mathbf{p} = (x,y,z) = x\,\mathbf{x} + y\,\mathbf{y} + z\,\mathbf{z}$$

$$\begin{aligned}\mathbf{p} = \; & x\,[\,(\mathbf{x} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{x} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{x} \cdot \mathbf{n})\,\mathbf{n}\,] \;+ \\ & y\,[\,(\mathbf{y} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{y} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{y} \cdot \mathbf{n})\,\mathbf{n}\,] \;+ \\ & z\,[\,(\mathbf{z} \cdot \mathbf{u})\,\mathbf{u} + (\mathbf{z} \cdot \mathbf{v})\,\mathbf{v} + (\mathbf{z} \cdot \mathbf{n})\,\mathbf{n}\,]\end{aligned}$$

Find P's Projection

$$\mathbf{p} = x\,[\,(\mathbf{x}\cdot\mathbf{u})\,\mathbf{u} + (\mathbf{x}\cdot\mathbf{v})\,\mathbf{v} + (\mathbf{x}\cdot\mathbf{n})\,\mathbf{n}\,] + y\,[\,(\mathbf{y}\cdot\mathbf{u})\,\mathbf{u} + (\mathbf{y}\cdot\mathbf{v})\,\mathbf{v} + (\mathbf{y}\cdot\mathbf{n})\,\mathbf{n}\,] + z\,[\,(\mathbf{z}\cdot\mathbf{u})\,\mathbf{u} + (\mathbf{z}\cdot\mathbf{v})\,\mathbf{v} + (\mathbf{z}\cdot\mathbf{n})\,\mathbf{n}\,]$$

Rewrite:

$$\mathbf{p} = [\,x\,(\mathbf{x}\cdot\mathbf{u}) + y\,(\mathbf{y}\cdot\mathbf{u}) + z\,(\mathbf{z}\cdot\mathbf{u})\,]\,\mathbf{u} + [\,x\,(\mathbf{x}\cdot\mathbf{v}) + y\,(\mathbf{y}\cdot\mathbf{v}) + z\,(\mathbf{z}\cdot\mathbf{v})\,]\,\mathbf{v} + [\,x\,(\mathbf{x}\cdot\mathbf{n}) + y\,(\mathbf{y}\cdot\mathbf{n}) + z\,(\mathbf{z}\cdot\mathbf{n})\,]\,\mathbf{n}$$

Coordinate Transformation

$$\mathbf{p} = [\; x\,(\mathbf{x}\,.\,\mathbf{u}) \;+\; y\,(\mathbf{y}\,.\,\mathbf{u}) \;+\; z\,(\mathbf{z}\,.\,\mathbf{u}) \;]\,\mathbf{u} \;+$$
$$[\; x\,(\mathbf{x}\,.\,\mathbf{v}) \;+\; y\,(\mathbf{y}\,.\,\mathbf{v}) \;+\; z\,(\mathbf{z}\,.\,\mathbf{v}) \;]\,\mathbf{v} \;+$$
$$[\; x\,(\mathbf{x}\,.\,\mathbf{n}) \;+\; y\,(\mathbf{y}\,.\,\mathbf{n}) \;+\; z\,(\mathbf{z}\,.\,\mathbf{n}) \;]\,\mathbf{n}$$

$$\mathbf{p} = (u,v,n) = u\,\mathbf{u} + v\,\mathbf{v} + n\,\mathbf{n}$$

Expressed in **uvn** basis:

$$u = x\,(\mathbf{x}\,.\,\mathbf{u}) + y\,(\mathbf{y}\,.\,\mathbf{u}) + z\,(\mathbf{z}\,.\,\mathbf{u})$$
$$v = x\,(\mathbf{x}\,.\,\mathbf{v}) + y\,(\mathbf{y}\,.\,\mathbf{v}) + z\,(\mathbf{z}\,.\,\mathbf{v})$$
$$n = x\,(\mathbf{x}\,.\,\mathbf{n}) + y\,(\mathbf{y}\,.\,\mathbf{n}) + z\,(\mathbf{z}\,.\,\mathbf{n})$$

Coordinate Tranformation

$$u = x\,(\mathbf{x} \cdot \mathbf{u}) + y\,(\mathbf{y} \cdot \mathbf{u}) + z\,(\mathbf{z} \cdot \mathbf{u})$$
$$v = x\,(\mathbf{x} \cdot \mathbf{v}) + y\,(\mathbf{y} \cdot \mathbf{v}) + z\,(\mathbf{z} \cdot \mathbf{v})$$
$$n = x\,(\mathbf{x} \cdot \mathbf{n}) + y\,(\mathbf{y} \cdot \mathbf{n}) + z\,(\mathbf{z} \cdot \mathbf{n})$$

In matrix form:

$$\begin{pmatrix} u \\ v \\ n \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

where:

$u_x = \mathbf{x} \cdot \mathbf{u}$

$u_y = \mathbf{y} \cdot \mathbf{u}$

etc.

Inverse Transformation

$$\begin{pmatrix} u \\ v \\ n \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{M} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

What's the invers transformation of M?

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_u & x_v & x_n \\ y_u & y_v & y_n \\ z_u & z_v & z_n \end{pmatrix} \begin{pmatrix} u \\ v \\ n \end{pmatrix}$$

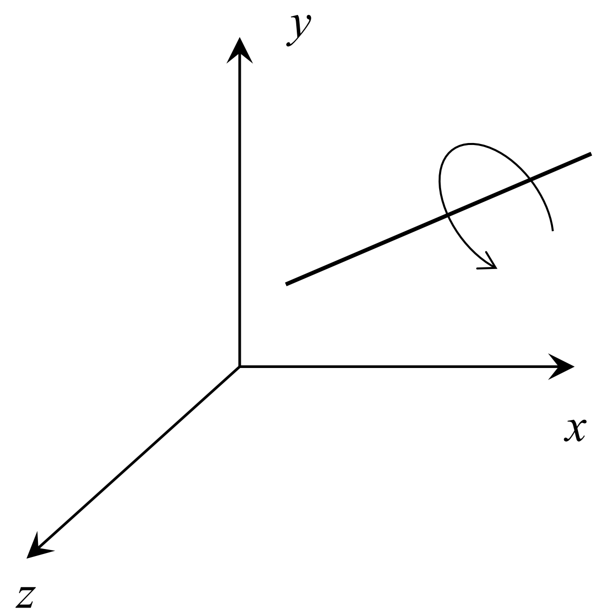
$u_x = \mathbf{x} \cdot \mathbf{u} = \mathbf{u} \cdot \mathbf{x} = x_u$

$$M^{-1} = M^T$$

Coordinate transformation matrix is orthonormal

Arbitrary Rotation

- How to perform arbitrary rotation using coordinate transformation?



Arbitrary Rotation

- How to perform arbitrary rotation using coordinate transformation?

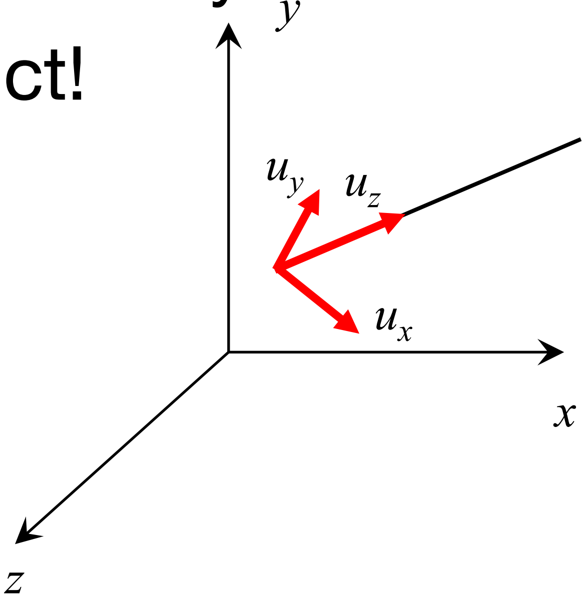
Step 1: form a local coordinate system

How? Cross product!

$$\mathbf{u}_z = \mathbf{u}$$

$$\mathbf{u}_y = \frac{\mathbf{u} \times \mathbf{x}}{\|\mathbf{u} \times \mathbf{x}\|}$$

$$\mathbf{u}_x = \mathbf{u}_y \times \mathbf{u}_z$$



Arbitrary Rotation

- How to perform arbitrary rotation using coordinate transformation?

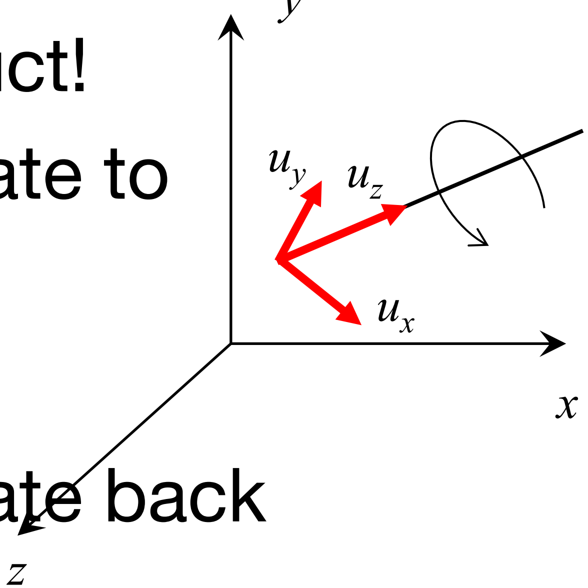
  Step 1: form a local coordinate system

  How? Cross product!

  Step 2: transform coordinate to the local system

  Step 3: rotate

  Step 4: transform coordinate back



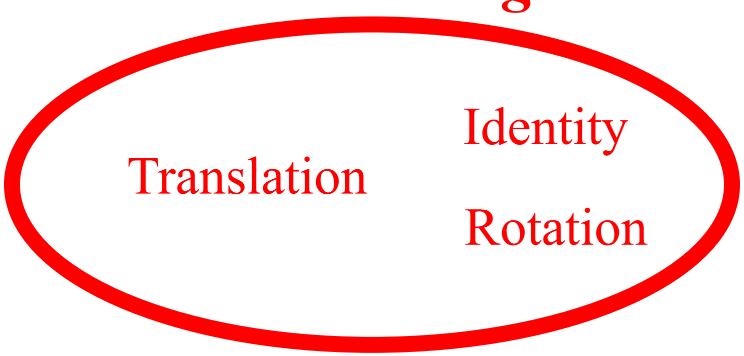
Classes of Transformations

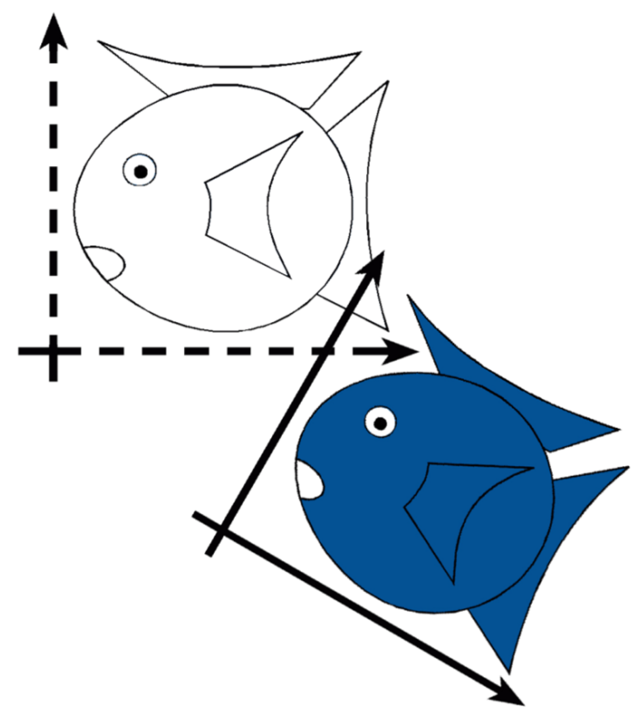
- Euclidean
- Similitude
- Linear
- Affine
- Projective

Euclidean Transformations

- Combination of rotation & translation define the Euclidean Tranform
- Properties
  - Preserve distance
  - Preserve angles

Euclidean / Rigid

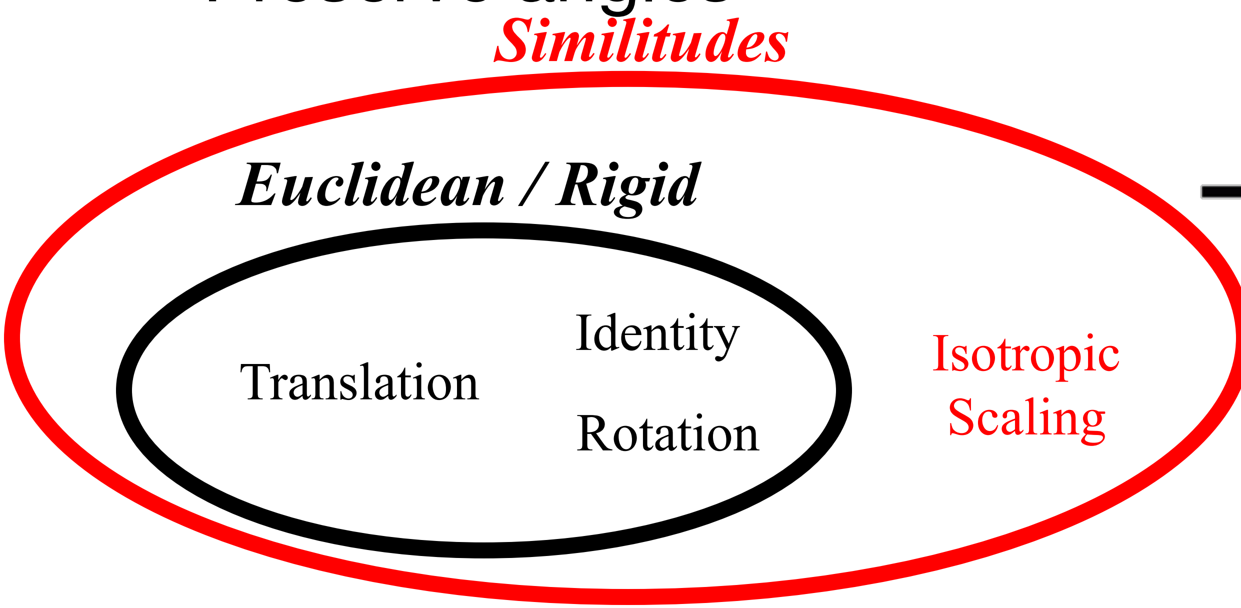


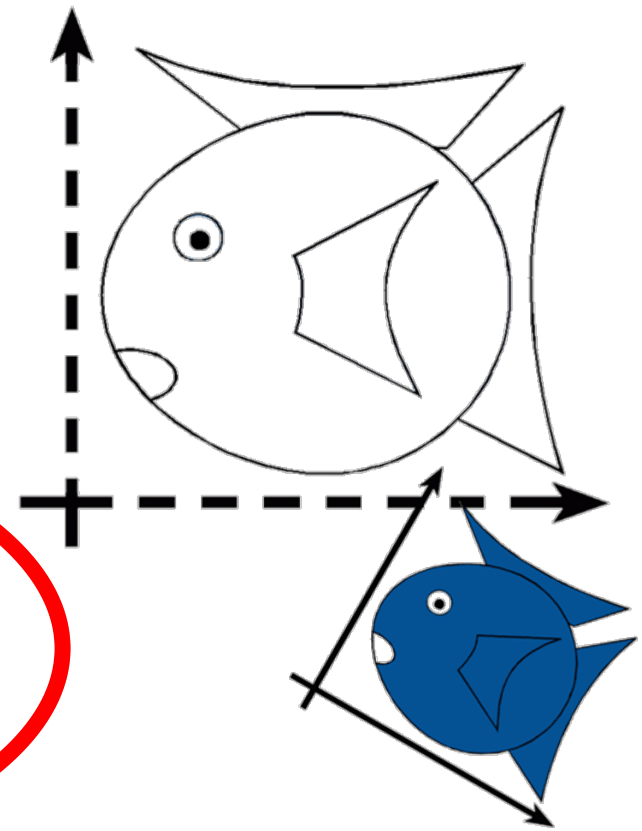


Slides adopted from MIT EECS 6.837, Durand and Cutler

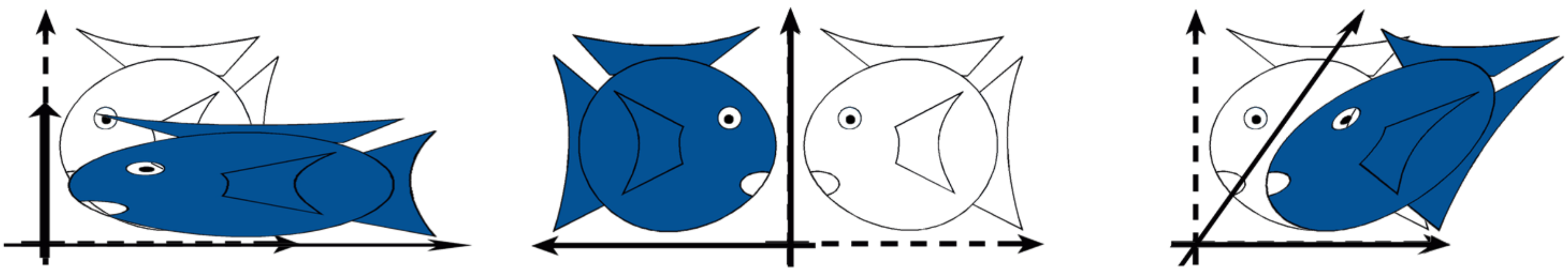
Similitude Transformations

- 4-parameter superset of Euclidean Transformation
- Properties
  - Preserve angles

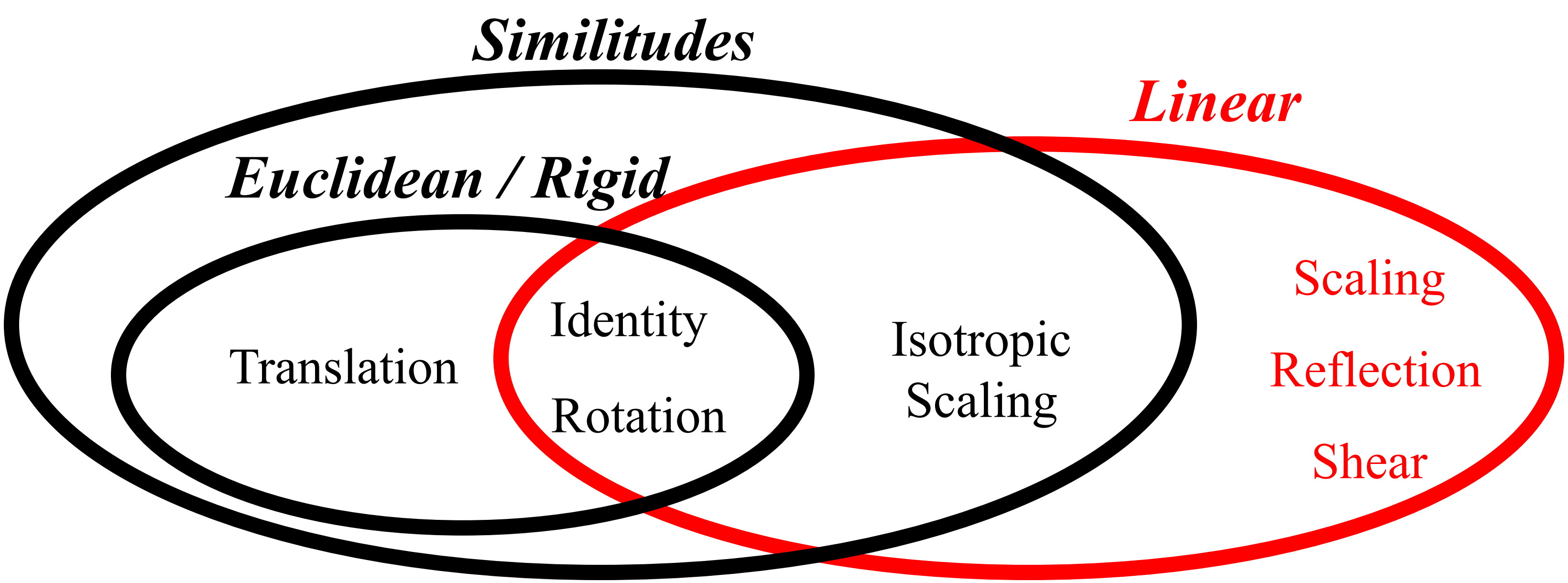




Linear Transformations

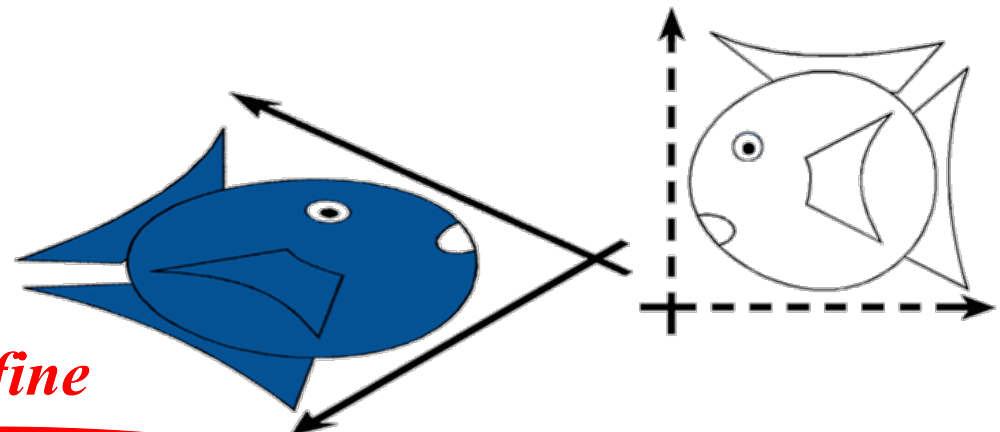


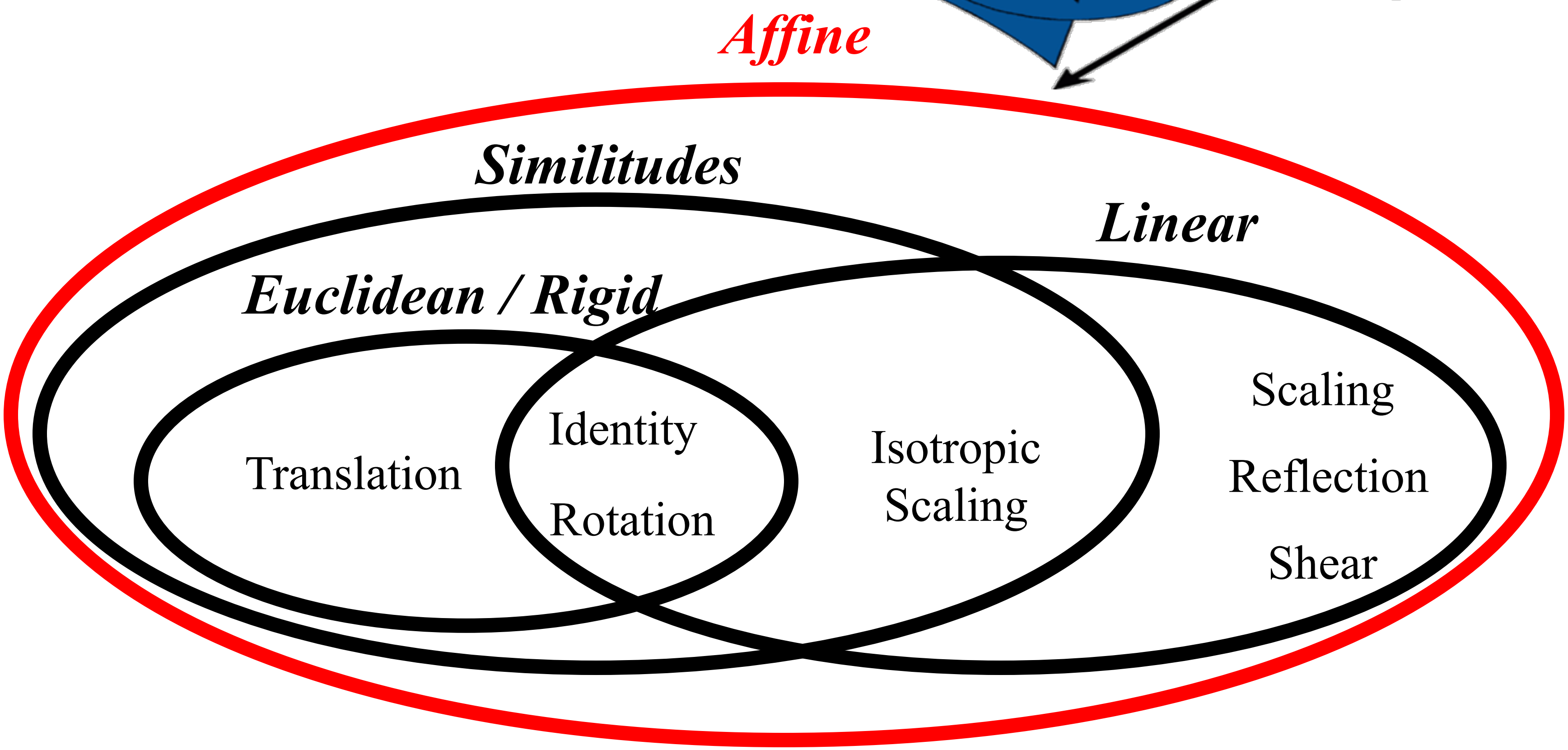
Scaling Reflection Shear



Affine Transformations

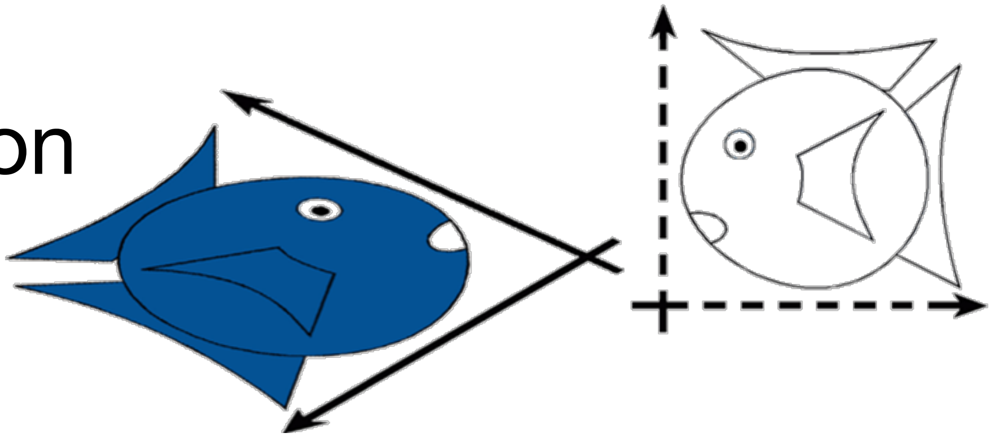
- Preserves parallel lines





Affine Transformations

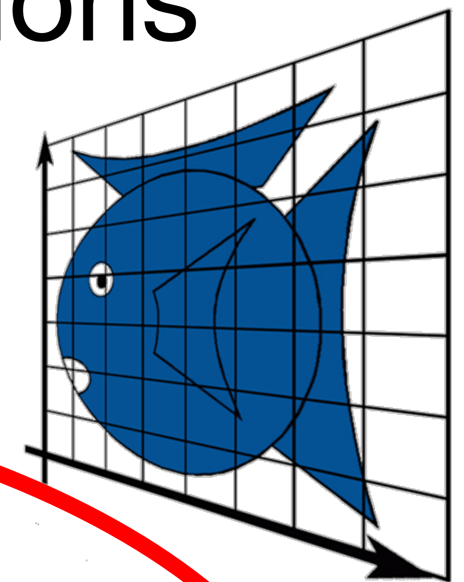
- Matrix representation
  - 12 DoF

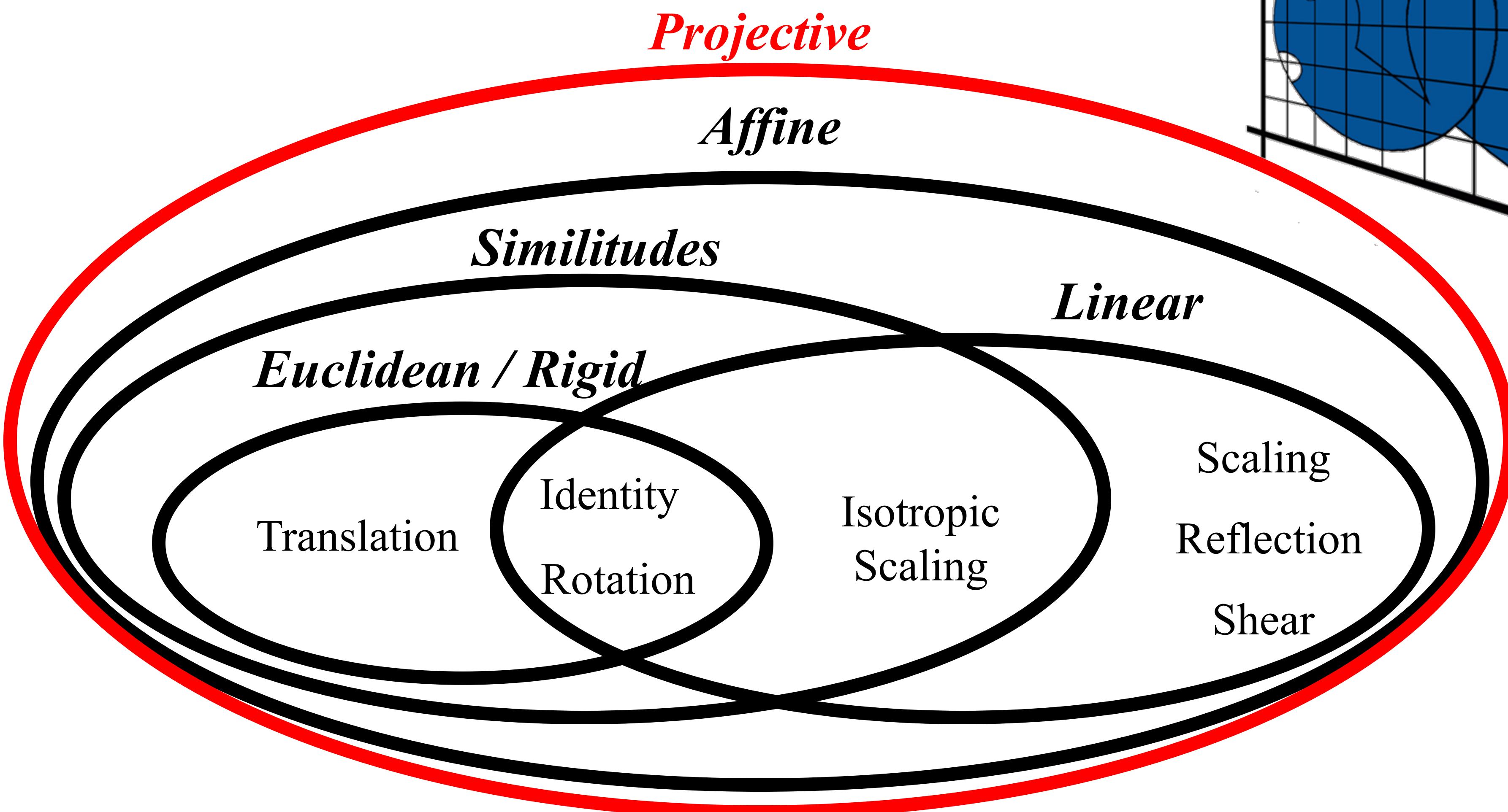


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projective Transformations

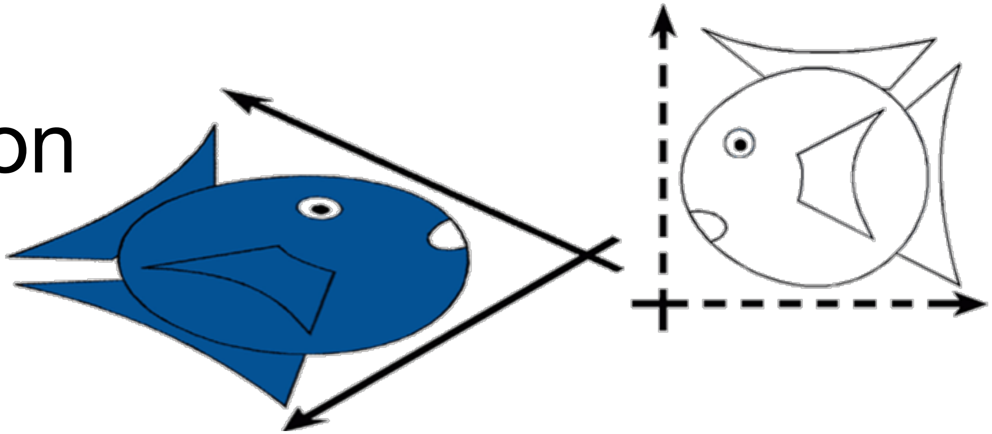
- Preserves lines





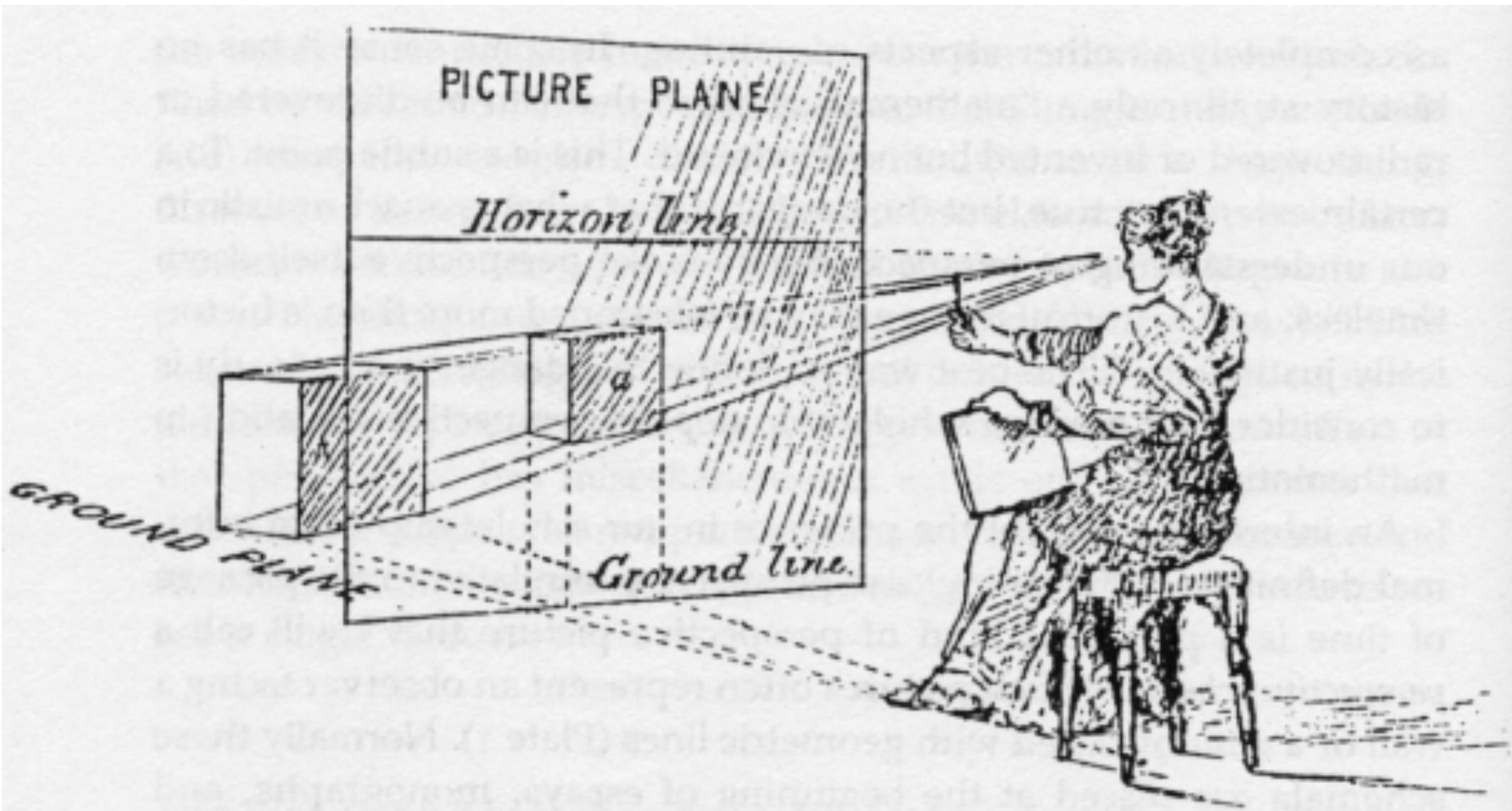
Projective Transformations

- Matrix representation
  - 15 DoF

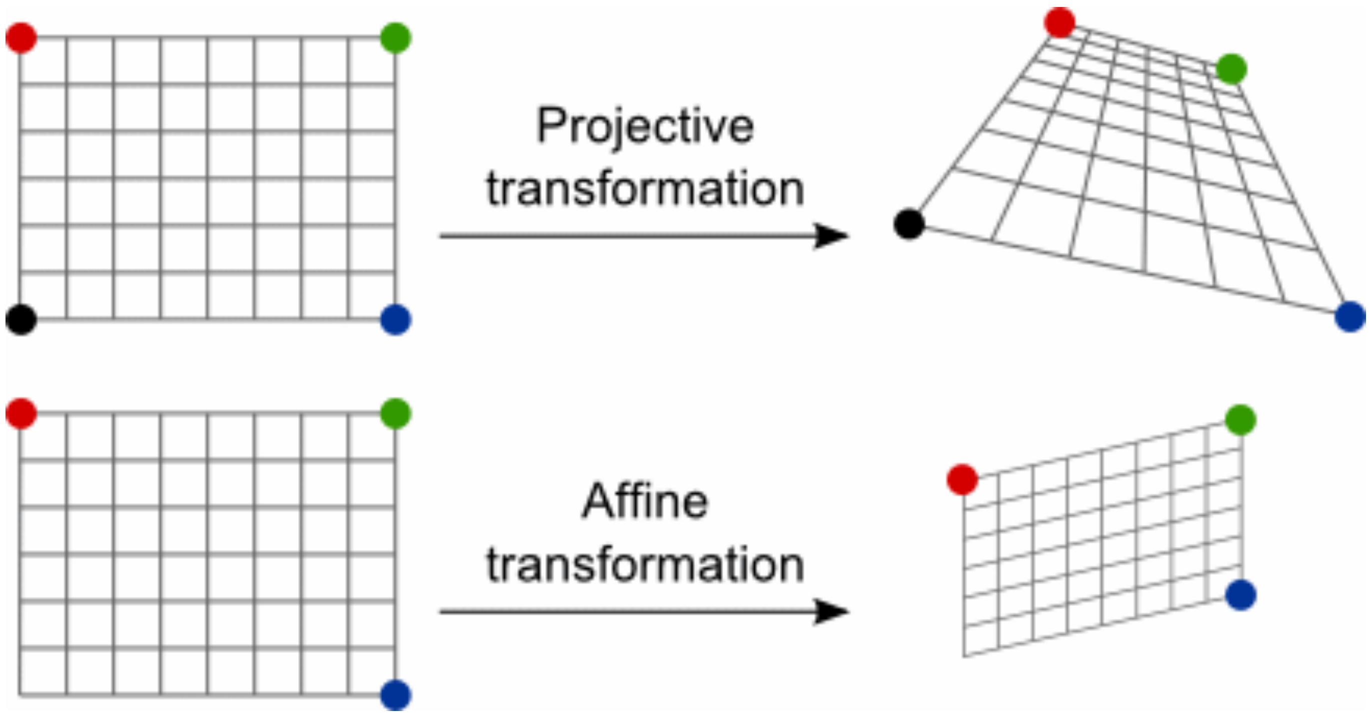


$$\begin{bmatrix} wx' \\ wy' \\ wz' \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection



Projective vs. Affine (2D)



Next time…

- Tired of math?
- OpenGL!
  - Program structure
  - Data structure
  - How to draw a primitive?