On the Midterm Exam

- **Monday, 10/17 in class**
- **Closed book and closed notes**
- **One-side and one page cheat sheet is allowed**
- **A calculator is allowed**
- **Covers the topics until the class on Wednesday, 10/12**

Today’s Agenda

Affine transformation

Homogeneous Coordinates

In general, the homogeneous coordinates for a 3D point [x y z] is given as

$$P = [x' \; y' \; z' \; w]^T = [wx \; wy \; wz \; w]^T$$

When $w \neq 0$, we return to a 3D point by $P = [x \; y \; z \; 1]$

where $x \leftarrow x'/w, y \leftarrow y'/w, z \leftarrow z'/w$

If $w=0$, the representation is that of a vector

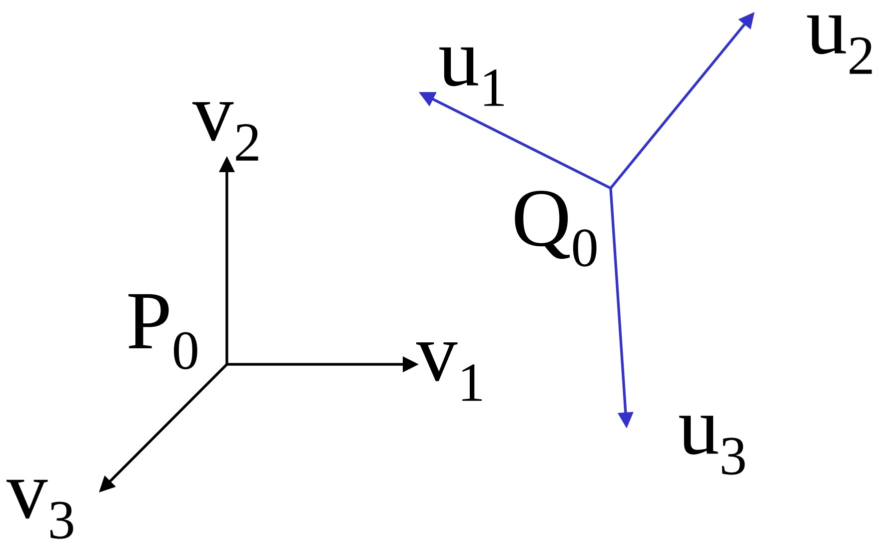
Change of Frames

We can apply a similar process in homogeneous coordinates to the representations of both points and vectors

Consider two frames:

(P_0, v_1, v_2, v_3)

(Q_0, u_1, u_2, u_3)



Any point or vector can be represented in either frame

We can represent Q_0, u_1, u_2, u_3 in terms of P_0, v_1, v_2, v_3

Representing One Frame in Terms of the Other

Extending what we did with change of bases

$$\mathbf{u}_1 = \gamma_{11}\mathbf{v}_1+\gamma_{12}\mathbf{v}_2+\gamma_{13}\mathbf{v}_3$$
$$\mathbf{u}_2 = \gamma_{21}\mathbf{v}_1+\gamma_{22}\mathbf{v}_2+\gamma_{23}\mathbf{v}_3$$
$$\mathbf{u}_3 = \gamma_{31}\mathbf{v}_1+\gamma_{32}\mathbf{v}_2+\gamma_{33}\mathbf{v}_3$$
$$\mathbf{Q}_0 = \gamma_{41}\mathbf{v}_1+\gamma_{42}\mathbf{v}_2+\gamma_{43}\mathbf{v}_3 +\mathbf{P}_0$$

defining a 4 x 4 matrix

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix} \Rightarrow [\mathbf{U} \quad Q_0] = [\mathbf{V} \quad P_0]\mathbf{M}^T$$

Changing Representations

Any point or vector has a representation in a frame

$\mathbf{a}=[\alpha_1\ \alpha_2\ \ \alpha_3\ \alpha_4\ ]$ in the first frame
$\mathbf{b}=[\beta_1\ \beta_2\ \ \beta_3\ \beta_4\ ]$ in the second frame

where $\alpha_4 = \beta_4 = 1$ for points and $\alpha_4 = \beta_4 = 0$ for vectors

We can change the representation from one frame to the other as

$$\mathbf{a}=\mathbf{M}^T\mathbf{b} \quad \text{and} \quad \mathbf{b}=(\mathbf{M}^T)^{-1}\mathbf{a}$$

The matrix $\mathbf{M}$ is 4 x 4 and specifies an affine transformation in homogeneous coordinates

Affine Transformations

Every linear transformation is equivalent to a change in frames

Every affine transformation preserves lines: a line in a frame transforms to a line in another frame

An affine transformation

- has only 12 *degrees of freedom* because 4 of the elements in the matrix are fixed and
- are a subset of all possible 4 x 4 linear transformations

Example

Suppose we have two bases v_1, v_2, v_3 and u_1, u_2, u_3 for two frames such that

$$u_1 = v_1 \quad u_2 = v_1 + v_2 \qquad\qquad u_3 = v_1 + v_2 + v_3$$

and $$Q_0 = P_0 + v_1 + 2v_2 + 3v_3$$

$$\mathbf{u}_1 = \gamma_{11}\mathbf{v}_1+\gamma_{12}\mathbf{v}_2+\gamma_{13}\mathbf{v}_3$$
$$\mathbf{u}_2 = \gamma_{21}\mathbf{v}_1+\gamma_{22}\mathbf{v}_2+\gamma_{23}\mathbf{v}_3$$
$$\mathbf{u}_3 = \gamma_{31}\mathbf{v}_1+\gamma_{32}\mathbf{v}_2+\gamma_{33}\mathbf{v}_3$$
$$\mathbf{Q}_0 = \gamma_{41}\mathbf{v}_1+\gamma_{42}\mathbf{v}_2+\gamma_{43}\mathbf{v}_3 +\mathbf{P}_0$$

What is M matrix?

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 1 \end{bmatrix} \qquad \left(\mathbf{M}^T\right)^{-1} = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

A point P in the first frame will transformed to P' in the second frame

$$P = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} \Rightarrow P' = \left(\mathbf{M}^{\boldsymbol{T}}\right)^{-\mathbf{1}} P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow \text{Origin of the second frame}$$

A vector ***w*** in the first frame will transformed to ***w'*** in the second frame

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} \Rightarrow \boldsymbol{w}' = \left(\mathbf{M}^{\boldsymbol{T}}\right)^{-\mathbf{1}} \boldsymbol{w} = \begin{bmatrix} -1 \\ -1 \\ 3 \\ 0 \end{bmatrix}$$

The World and Camera Frames

In OpenGL, the base frame that we start with is the world frame

Eventually we represent entities in the camera frame by changing the world representation using the model-view matrix

Initially these frames are the same ($\mathbf{M}=\mathbf{I}$)

Changes in frame are then defined by 4 x 4 matrices

Moving the Camera Frame

If objects are on both sides of z=0, we must move camera frame or object frame

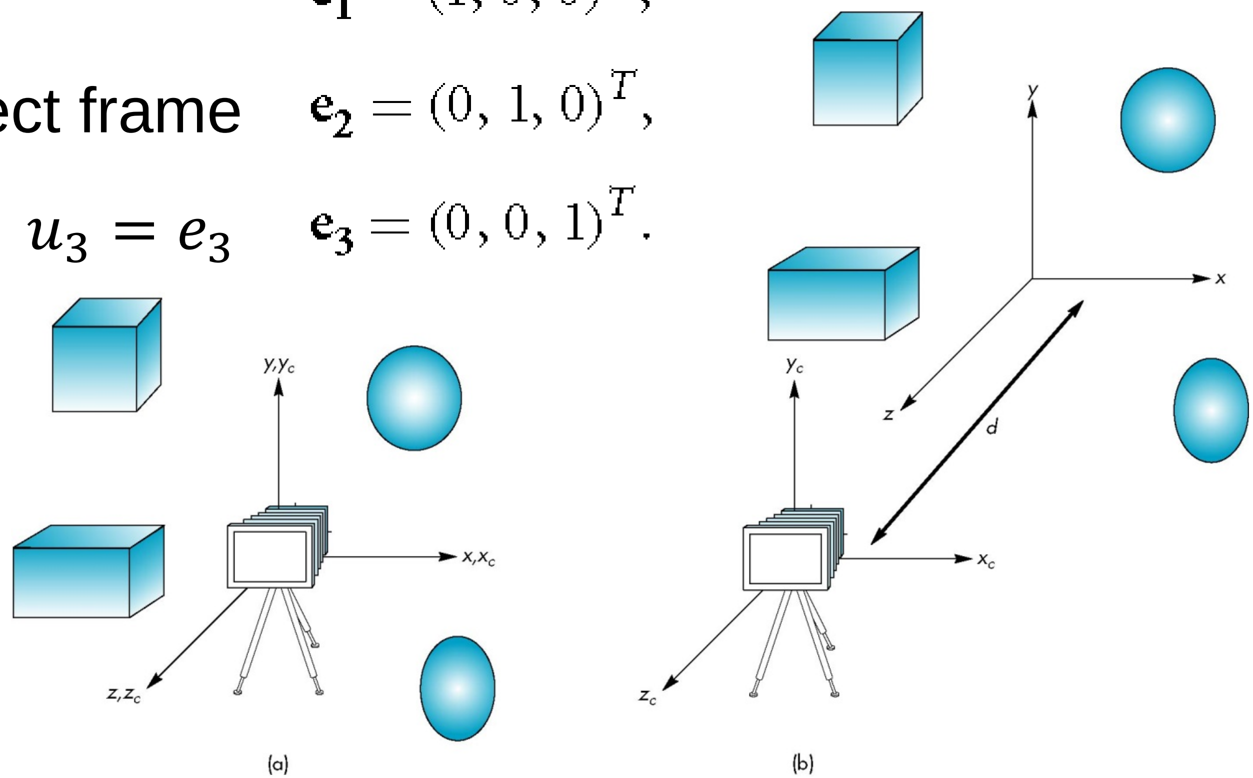
Use a 3-tuples to represent the object frame

$$\mathbf{e}_1 = (1, 0, 0)^T,$$
$$\mathbf{e}_2 = (0, 1, 0)^T,$$
$$\mathbf{e}_3 = (0, 0, 1)^T.$$

$$u_1 = e_1 \quad u_2 = e_2 \quad u_3 = e_3$$

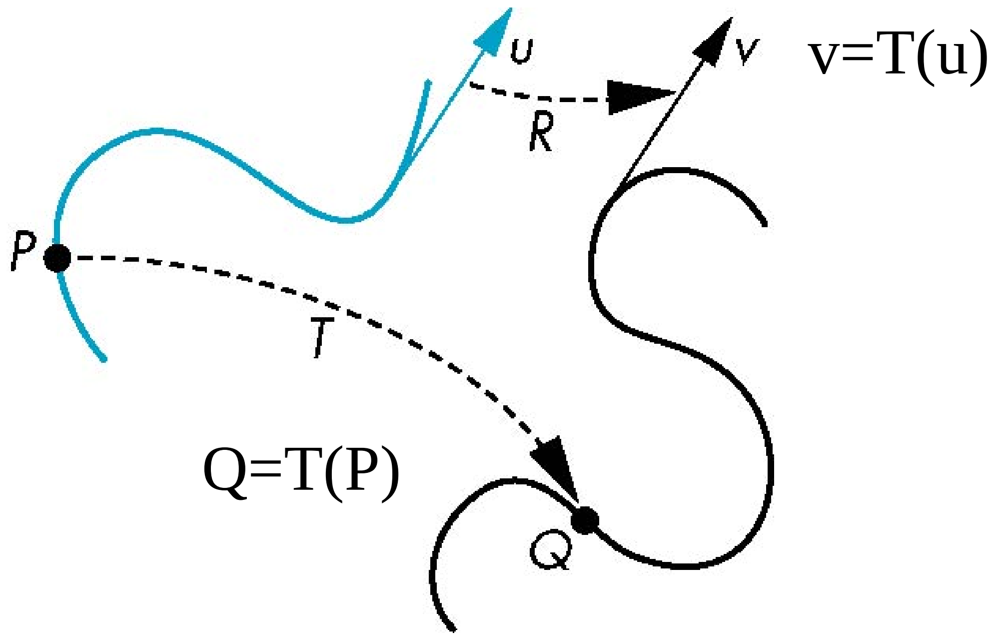
$$Q_0 = P_0 + de_3$$

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



General Transformations

A transformation maps points to other points and/or vectors to other vectors



Affine Transformations

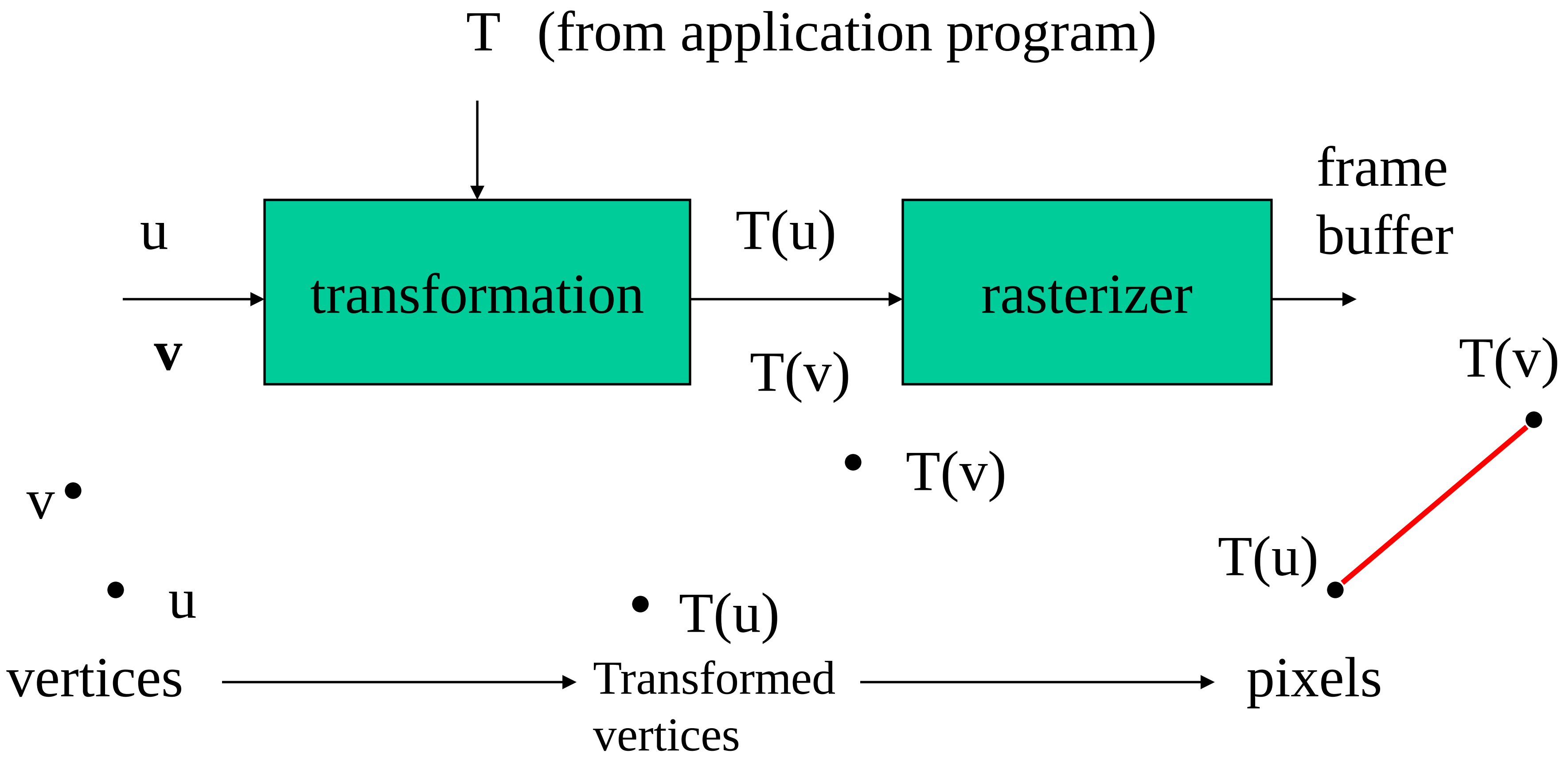
Line preserving

Characteristic of many physically important transformations

- Rigid body transformations: rotation, translation
- Scaling, shear

Note: we need only transform endpoints of line segments in graphics and the line segment between the transformed endpoints is generated during rasterization

Pipeline Implementation



Notation

We will be working with both coordinate-free representations of transformations and representations within a particular frame

P,Q, R: points in an affine space

u, v, w: vectors in an affine space

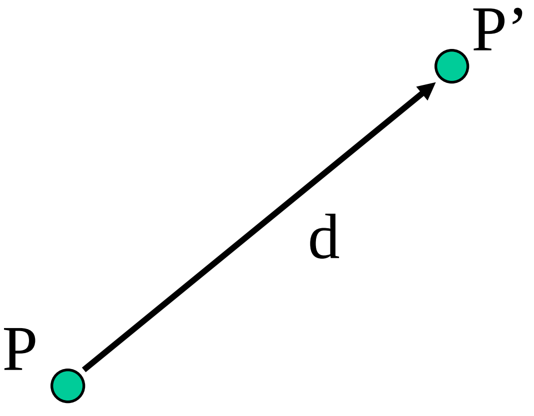
α, β, γ: scalars

d, s, l: representations of points/vectors

-vector of 4 scalars in homogeneous coordinates

Translation

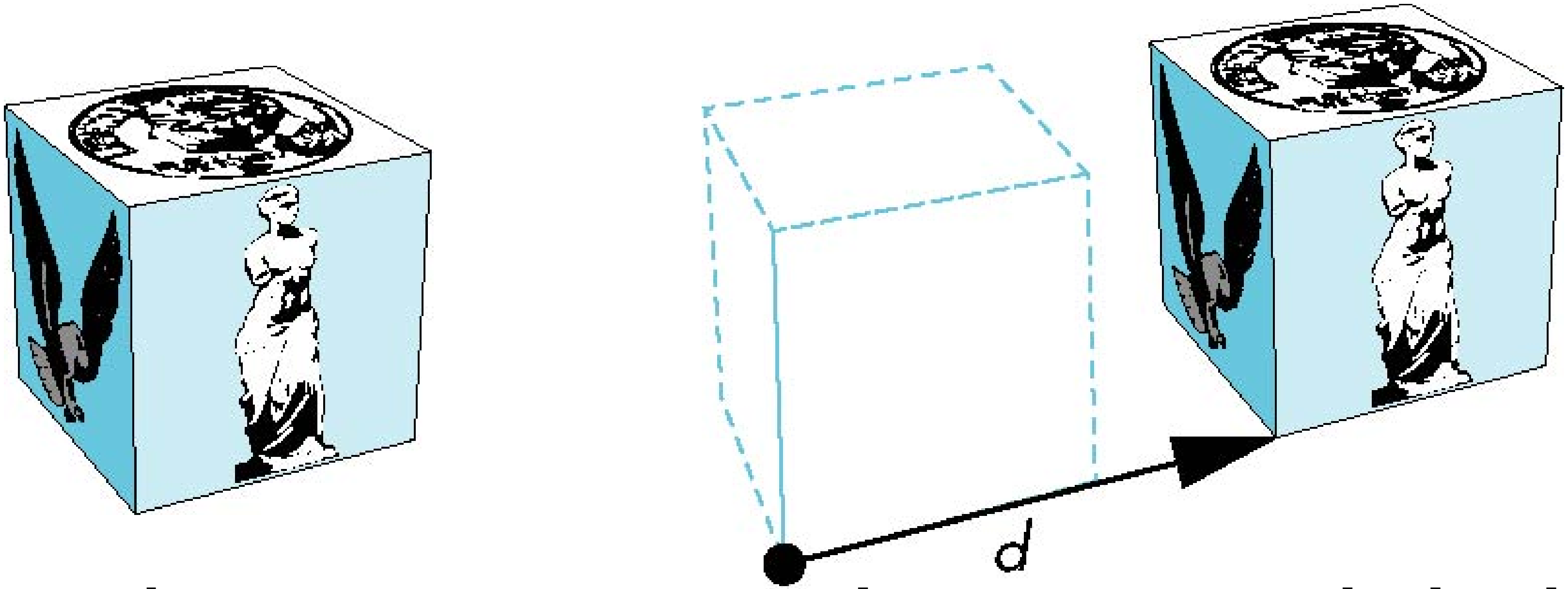
Move (translate, displace) a point to a new location



Displacement determined by a vector d

- Three degrees of freedom
- P'=P+d

Move All Points on the Object



object

translation: every point displaced by same vector

Translation Using Representations

Using the homogeneous coordinate representation in some frame

p=[x y z 1]^T

p'=[x' y' z' 1]^T

d=[dx dy dz 0]^T

Hence **p'** = **p** + **d** or

note that this expression is in four dimensions and expresses point = vector + point

$x' = x + d_x$

$y' = y + d_y$

$z' = z + d_z$

Translation Matrix

We can also express translation using a 4 x 4 matrix **T** in homogeneous coordinates

p'=Tp where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This form is better for implementation because all affine transformations can be expressed this way and multiple transformations can be concatenated together

Scaling

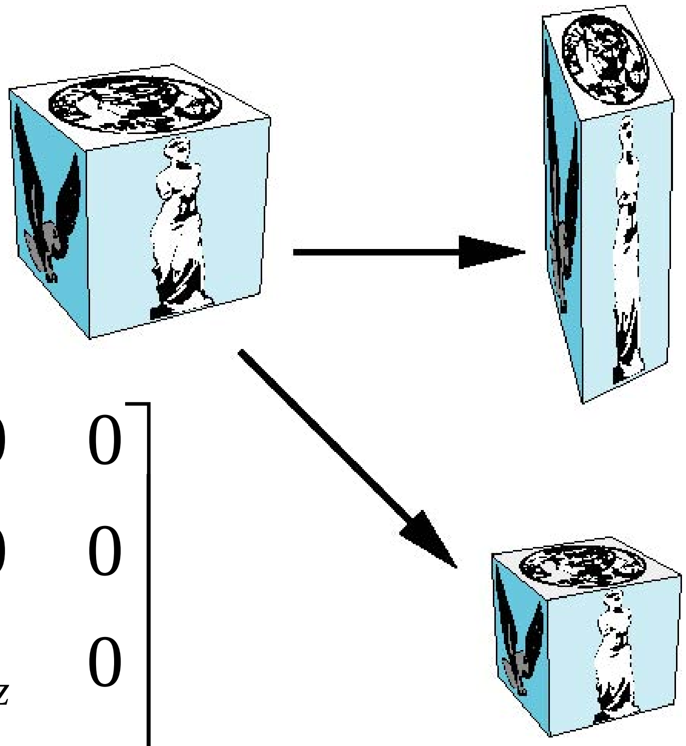
Expand or contract along each axis (fixed point of origin)

$p=[\,x\ y\ z\ 1]^T$ and $p'=[x'\ y'\ z'\ 1]^T$

$x'=s_x x$

$y'=s_y y$

$z'=s_z z$

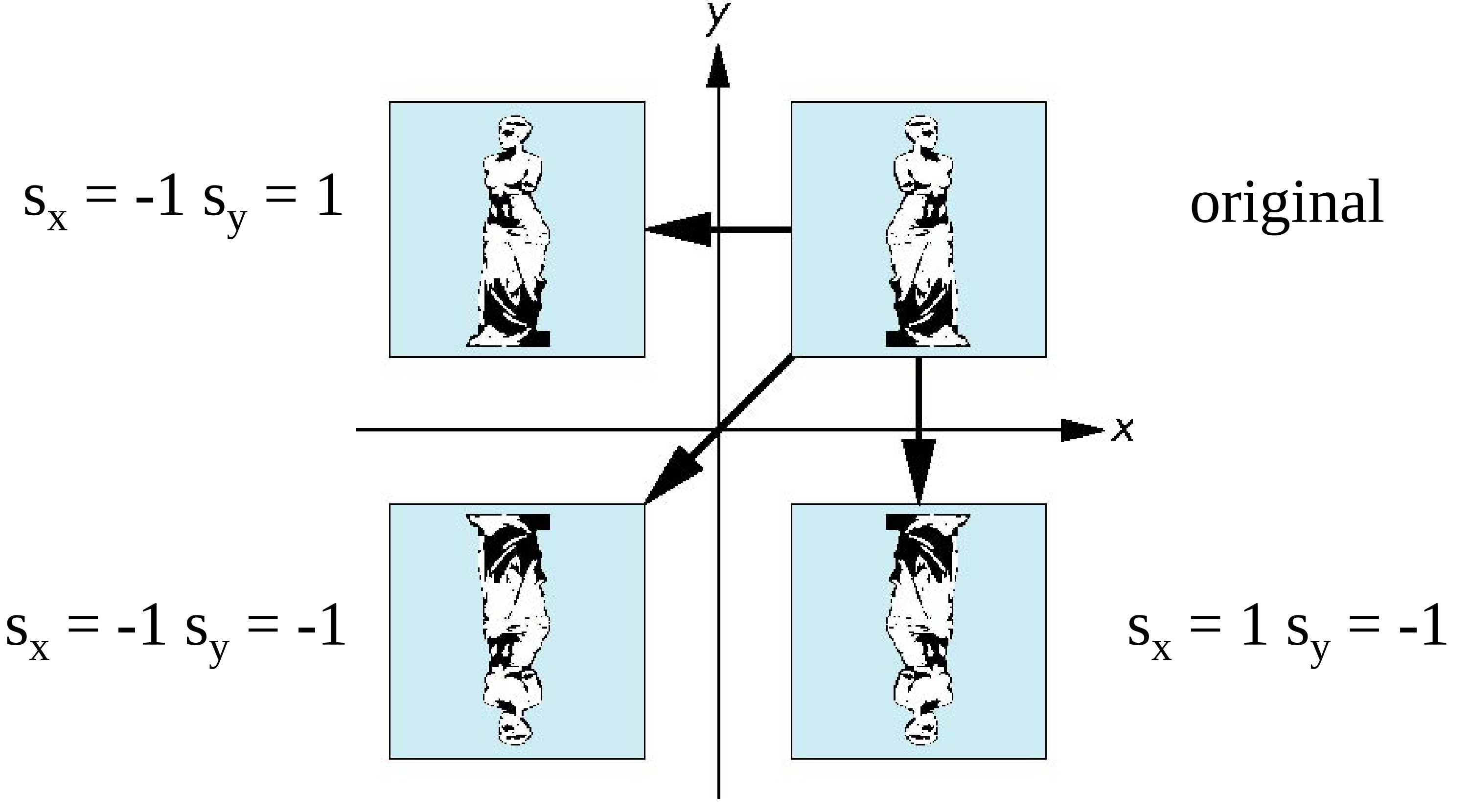


$\Rightarrow$ $\mathbf{p'=Sp}$

$$\mathbf{S} = \mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

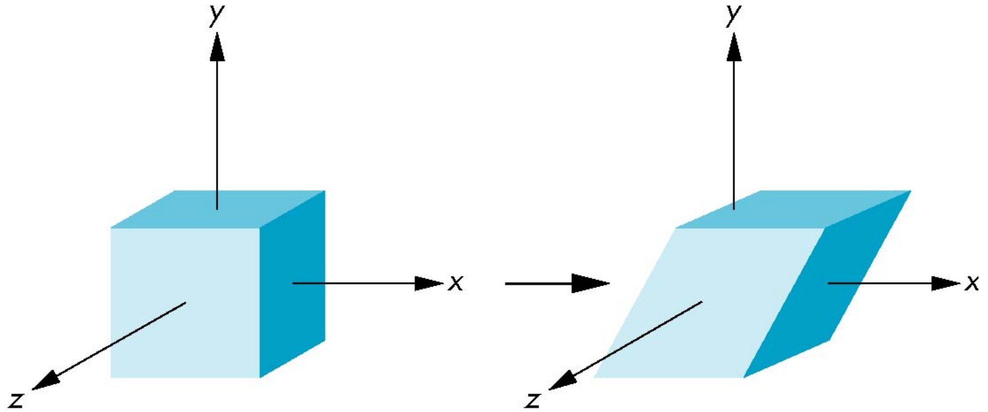
Reflection

corresponds to negative scale factors



Shear

- **Helpful to add one more basic transformation**
- **Equivalent to pulling faces in opposite directions**

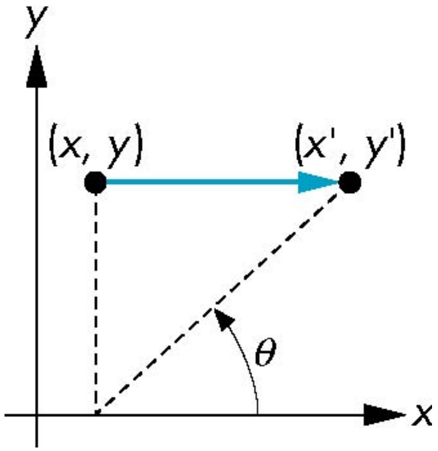


Shear Matrix

Consider simple shear along *x* axis

$$x' = x + y \cot \theta$$
$$y' = y$$
$$z' = z$$

$$\mathbf{H}(\theta) = \begin{bmatrix} 1 & \cot\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

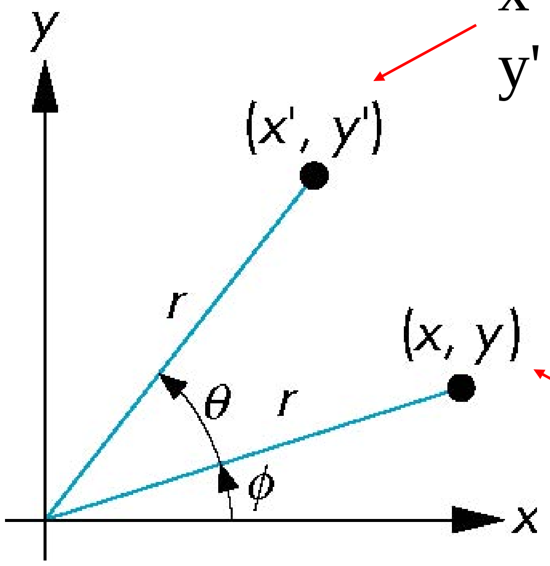


Rotation in 2D

Consider rotation about the origin by θ degrees

- radius stays the same, angle increases by θ

$$x' = r \cos(\phi + \theta)$$
$$y' = r \sin(\phi + \theta)$$



$$x' = x \cos\theta - y \sin\theta$$
$$y' = x \sin\theta + y \cos\theta$$

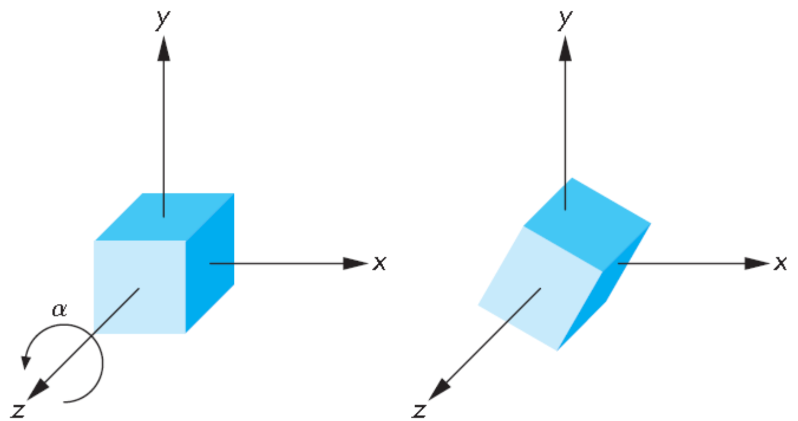
$$x = r \cos\phi$$
$$y = r \sin\phi$$

Rotation about the z axis

Rotation about z axis in three dimensions leaves all points with the same z

- Equivalent to rotation in two dimensions in planes of constant z

$$x' = x\cos\alpha - y\sin\alpha$$
$$y' = x\sin\alpha + y\cos\alpha$$
$$z' = z$$



- or in homogeneous coordinates

$$\mathbf{p}' = \mathbf{R}_z(\alpha)\mathbf{p}$$

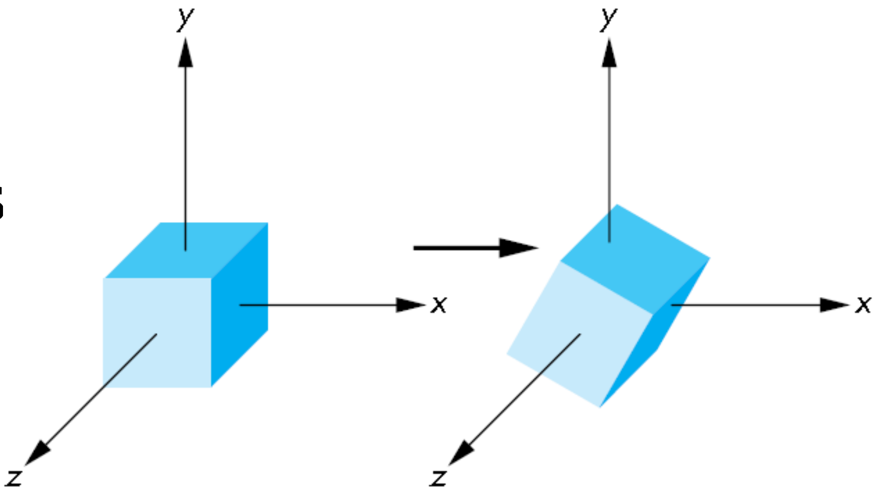
Rotation Matrix $\leftarrow \mathbf{R}_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

General Rotation About the Origin

A general rotation about the origin can be decomposed into successive of rotations about the *x*, *y*, and *z* axes

$$\mathbf{R} = \mathbf{R}_z(\alpha)\,\mathbf{R}_y(\beta)\,\mathbf{R}_x(\gamma)$$

α, β, γ are called the Euler angles



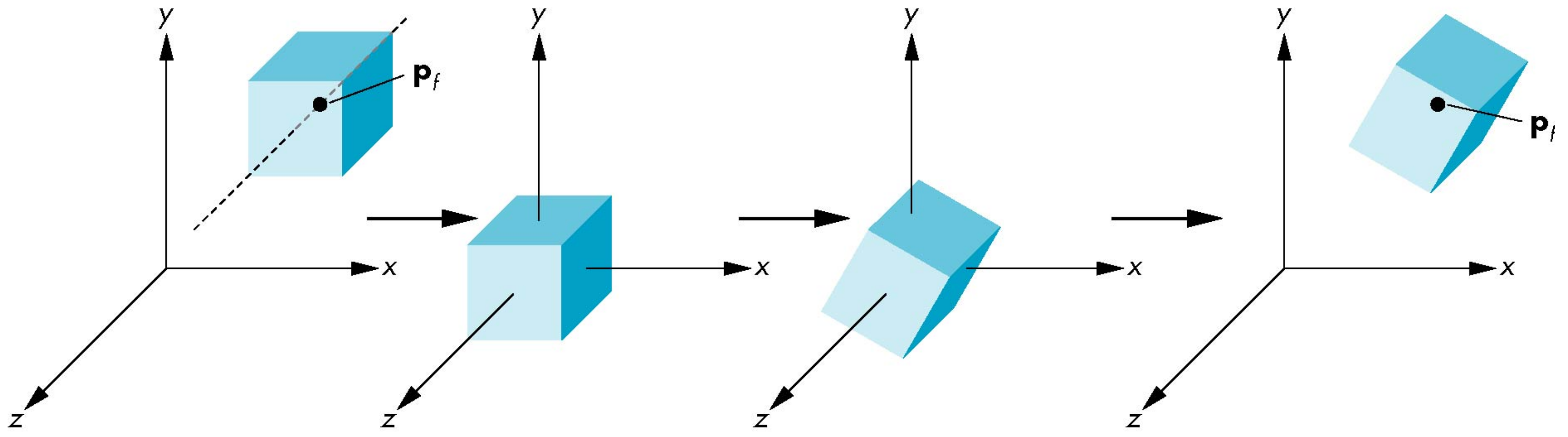
E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012

Important:

- **R** is unique
- For a given order, rotations do not commute
- We can use rotations in another order but with different angles

Rotation About a Fixed Point Other than the Origin

- Move fixed point to origin
- Rotate around the origin
- Move fixed point back



Instancing

How do we describe multiple object in a scene?

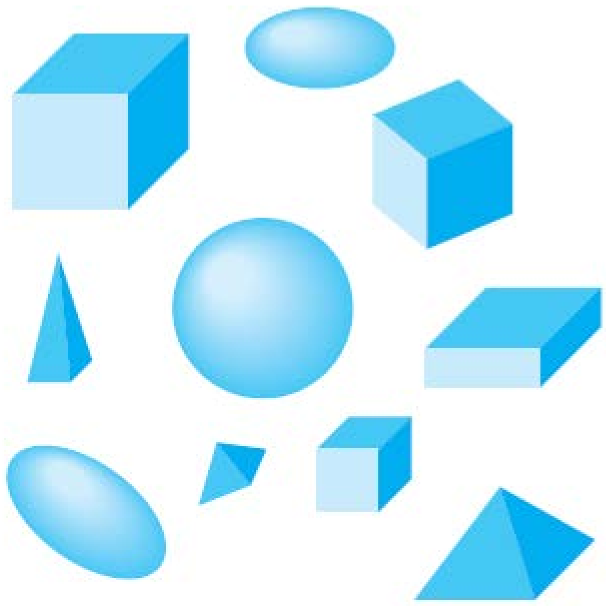
Intuitive solution:

Specify the vertices for each object

A better solution:

Specify a set of simple objects with

- a convenient size,
- a convenient location,
- a convenient orientation



E. Angel and D. Shreiner

Instancing

In modeling, we often start with a simple object centered at the origin, oriented with the axis, and at a standard size

An occurrence of this object is an **instance** of the object class

We apply an ***instance transformation*** to its vertices to