15-462: Computer Graphics

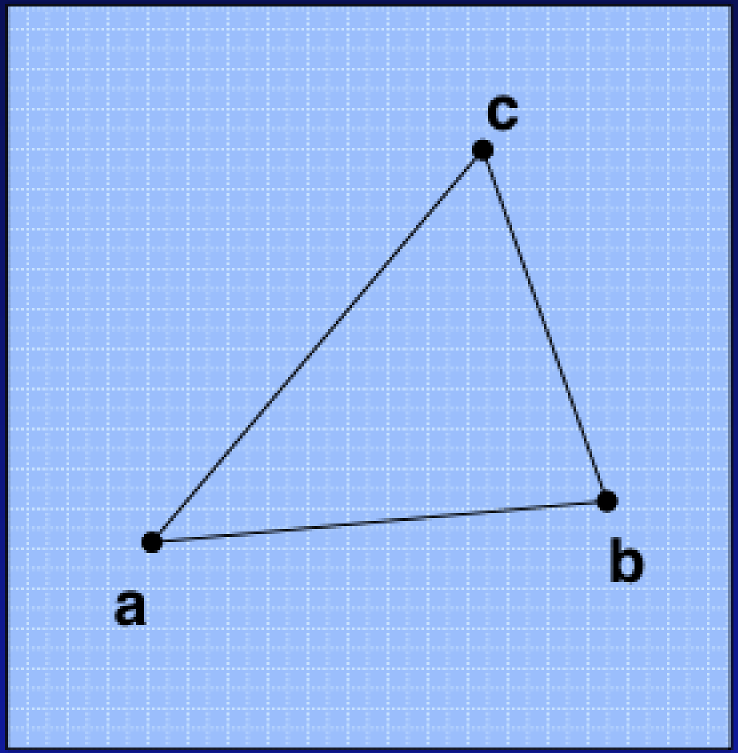
Math for Computer Graphics

Topics for Today

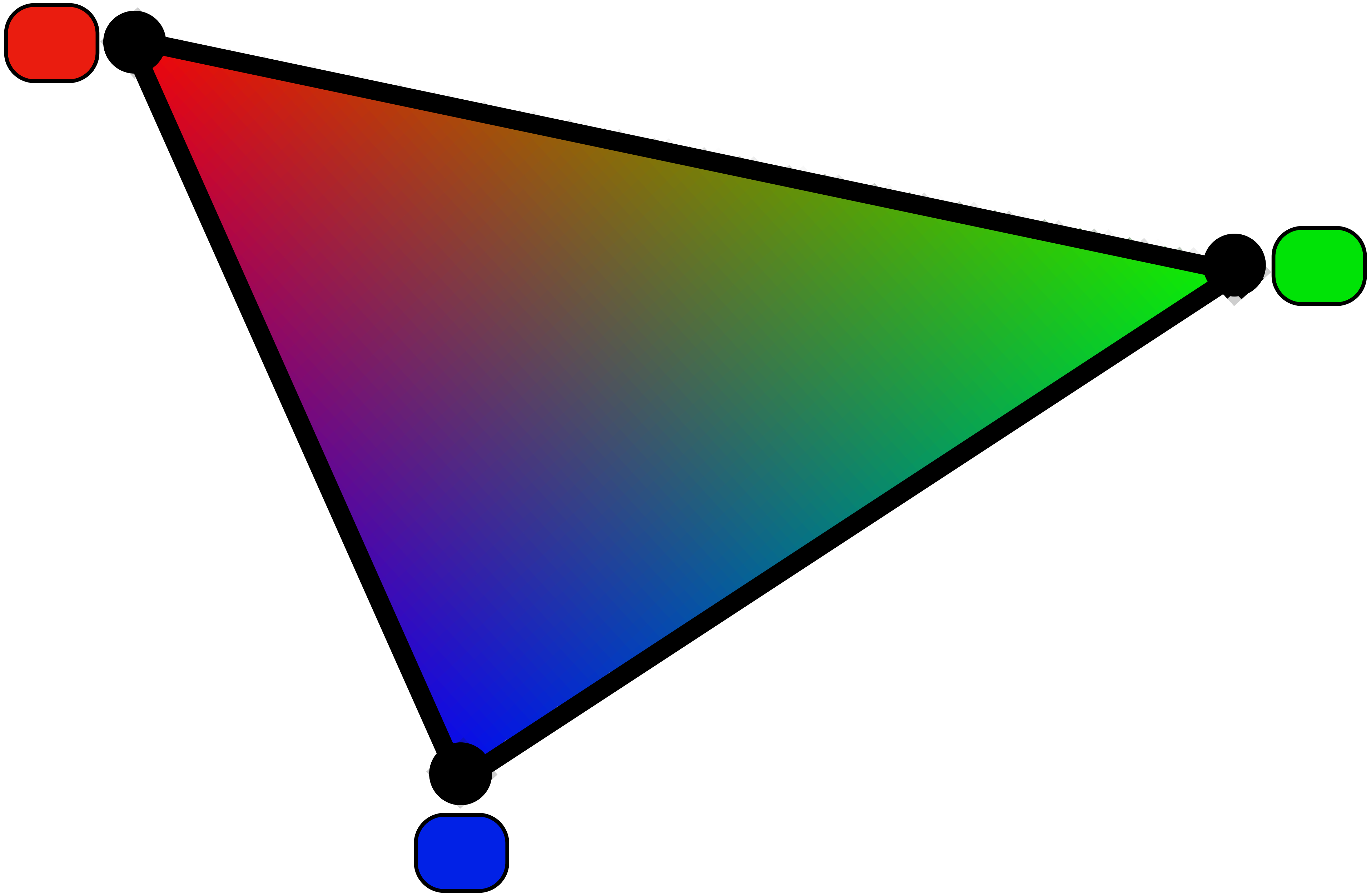
- Vectors
- Equations for curves and surfaces
- Barycentric Coordinates
  - Why barycentric coordinates?
  - What are barycentric coordinates?

Why barycentric coordinates?

- Triangles are the fundamental primitive used in 3D modeling programs.
- Triangles are stored as a sequence of three vectors, each defining a vertex.
- Often, we know information about the vertices, such as color, that we'd like to interpolate over the whole triangle.

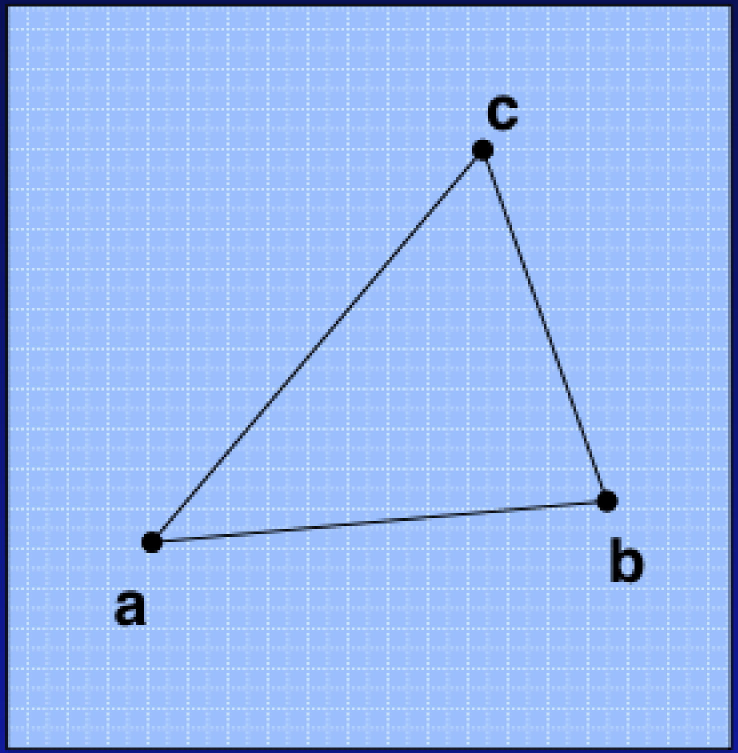


Barycentric Color Interpolation



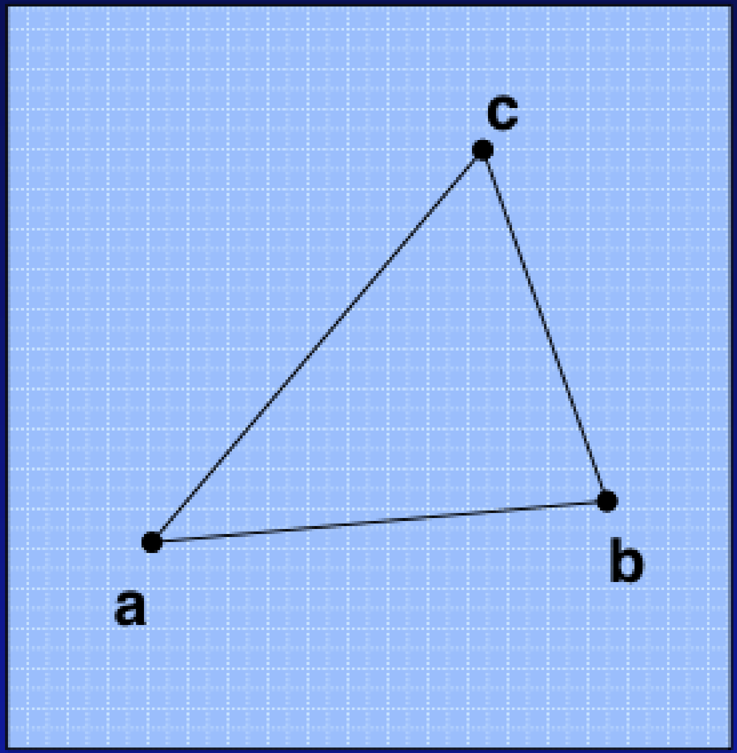
What are barycentric coordinates?

- The simplest way to do this interpolation is *barycentric coordinates*.
- The name comes from the Greek word *barus* (heavy) because the coordinates are weights assigned to the vertices.
- Point **a** on the triangle is the origin of the non-orthogonal coordinate system.
- The vectors from **a** to **b** and from **a** to **c** are taken as basis vectors.

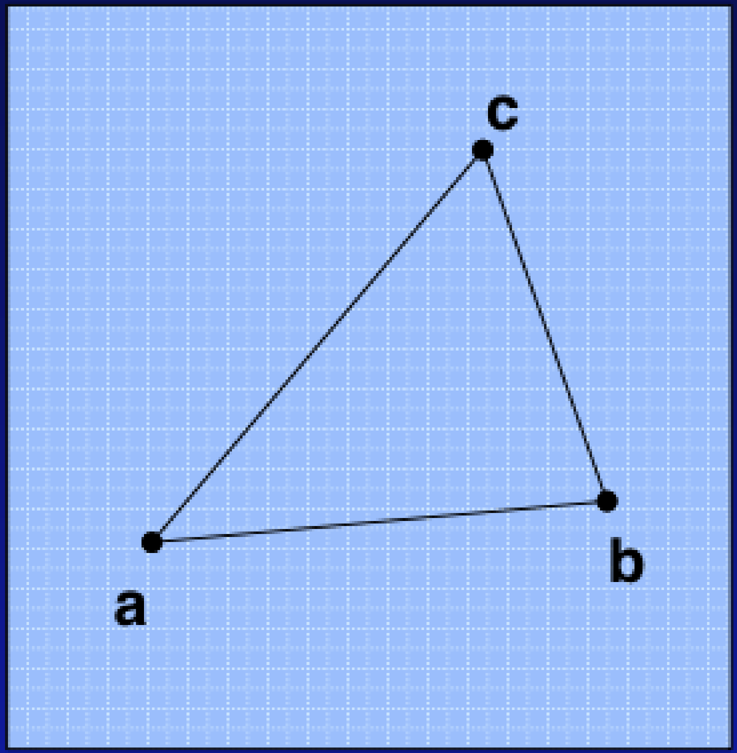


What are barycentric coordinates?

- We can express any point $\mathbf{p}$ coplanar to the triangle as: $\mathbf{p} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$
- Typically, we rewrite this as: $\mathbf{p}(\alpha,\beta,\gamma) = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$ where $\alpha \equiv 1 - \beta - \gamma$
- $\mathbf{a} = \mathbf{p}(1,0,0)$, $\mathbf{b} = \mathbf{p}(0,1,0)$, $\mathbf{c} = \mathbf{p}(0,0,1)$



What are barycentric coordinates?



Some cool properties:

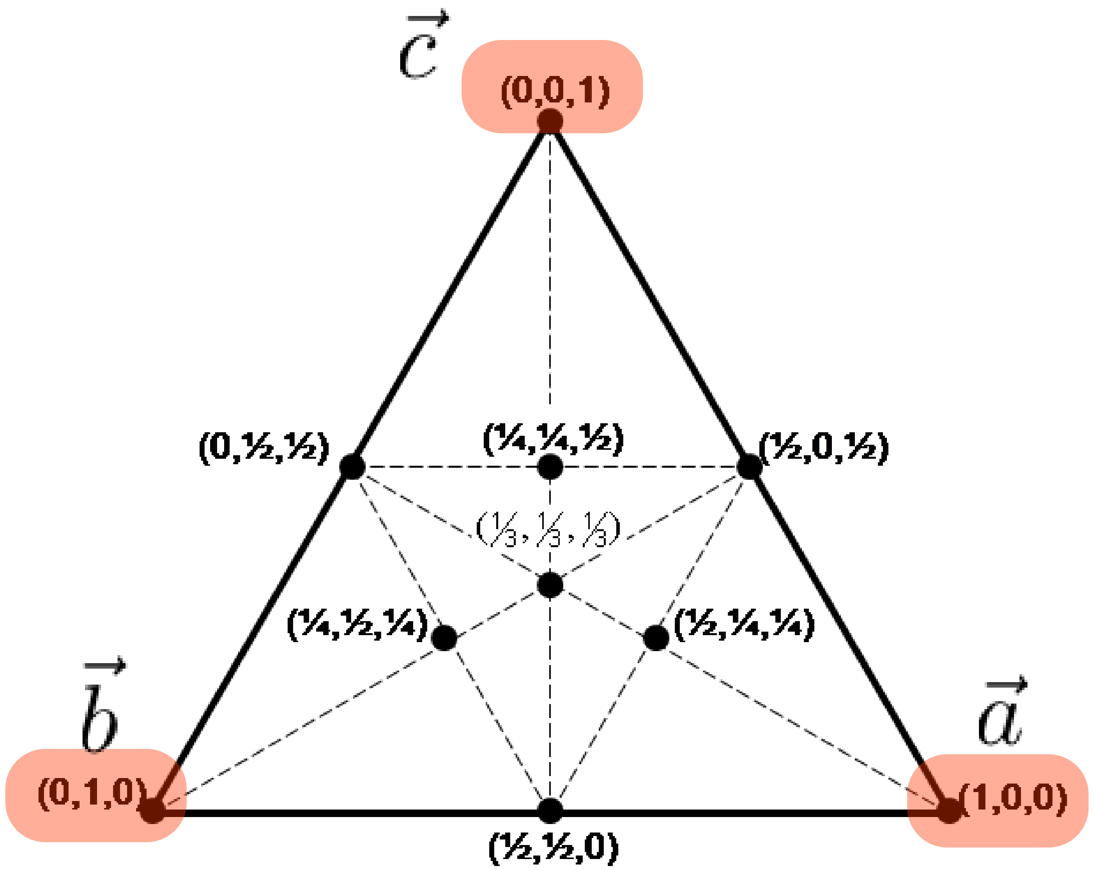
- Point **p** is inside the triangle if and only if

  $0 < \alpha < 1,$

  $0 < \beta < 1,$

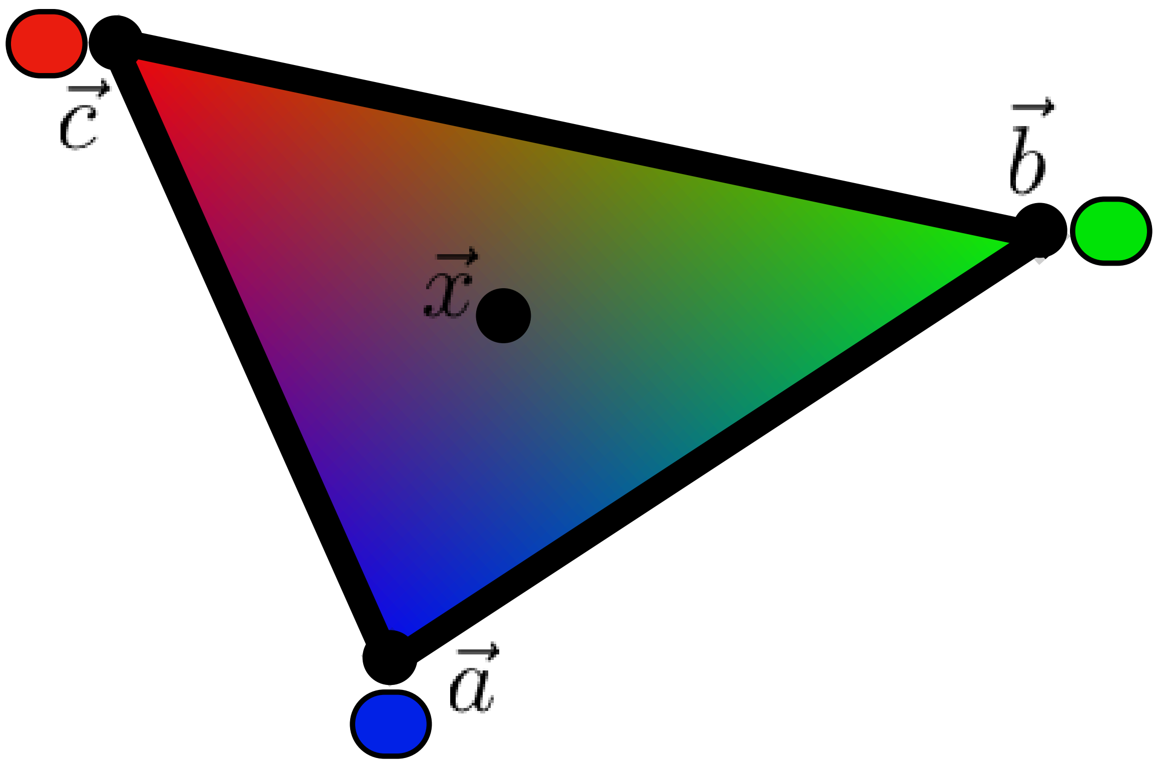
  $0 < \gamma < 1$
- If one component is zero, **p** is on an edge.
- If two components are zero, **p** is on a vertex.
- The coordinates can be used as weighting factors for properties of the vertices, like color.

Barycentric Coordinates



source:
http://en.wikipedia.org/wiki/File:Barycentric_coordinates_1.png

Barycentric Color Interpolation



If: $\vec{x} = \alpha\vec{a} + \beta\vec{b} + \gamma\vec{c}$

Then: $\text{color}(\vec{x}) = \alpha\text{color}(\vec{a}) + \beta\text{color}(\vec{b}) + \gamma\text{color}(\vec{c})$

Barycentric coordinates

Chalkboard examples:

- Conversion from 2D Cartesian
- Conversion from 3D Cartesian

Transformations

Translation, rotation, scaling
- 2D
- 3D

Homogeneous coordinates

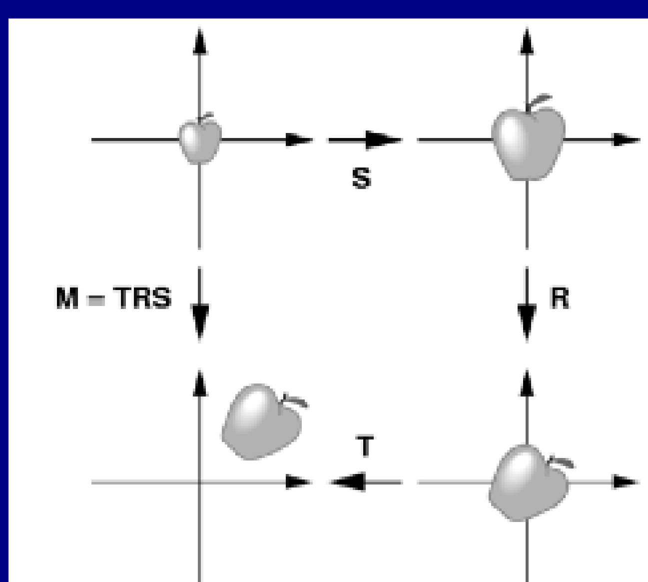
Transforming normals

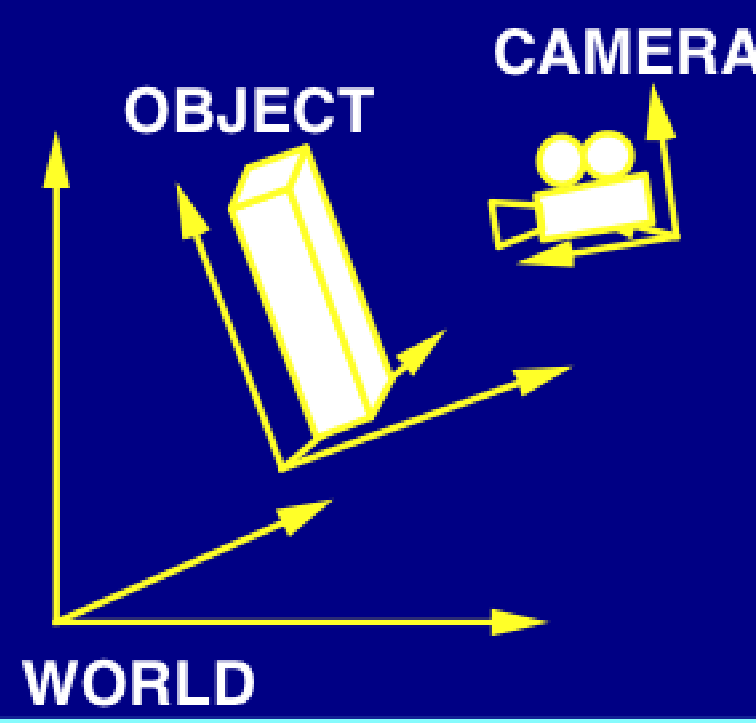
Examples

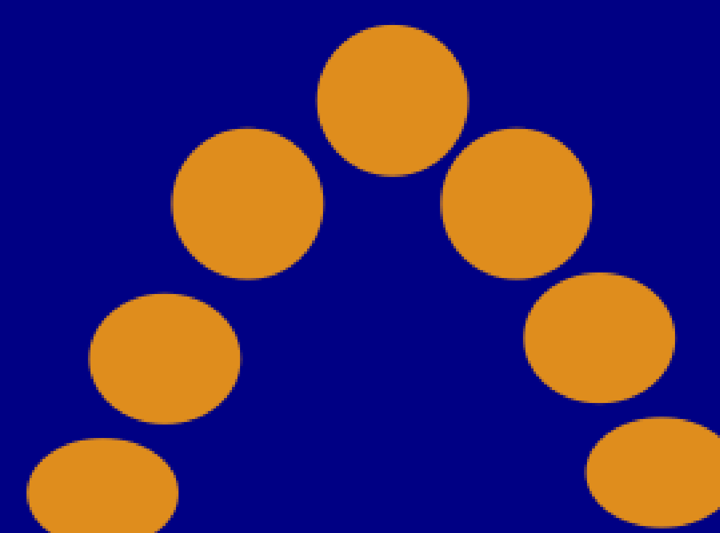
Shirley Chapter 6

Uses of Transformations

- Modeling
  - build complex models by positioning simple components
  - transform from object coordinates to world coordinates
- Viewing
  - placing the virtual camera in the world
  - specifying transformation from world coordinates to camera coordinates
- Animation
  - vary transformations over time to create motion

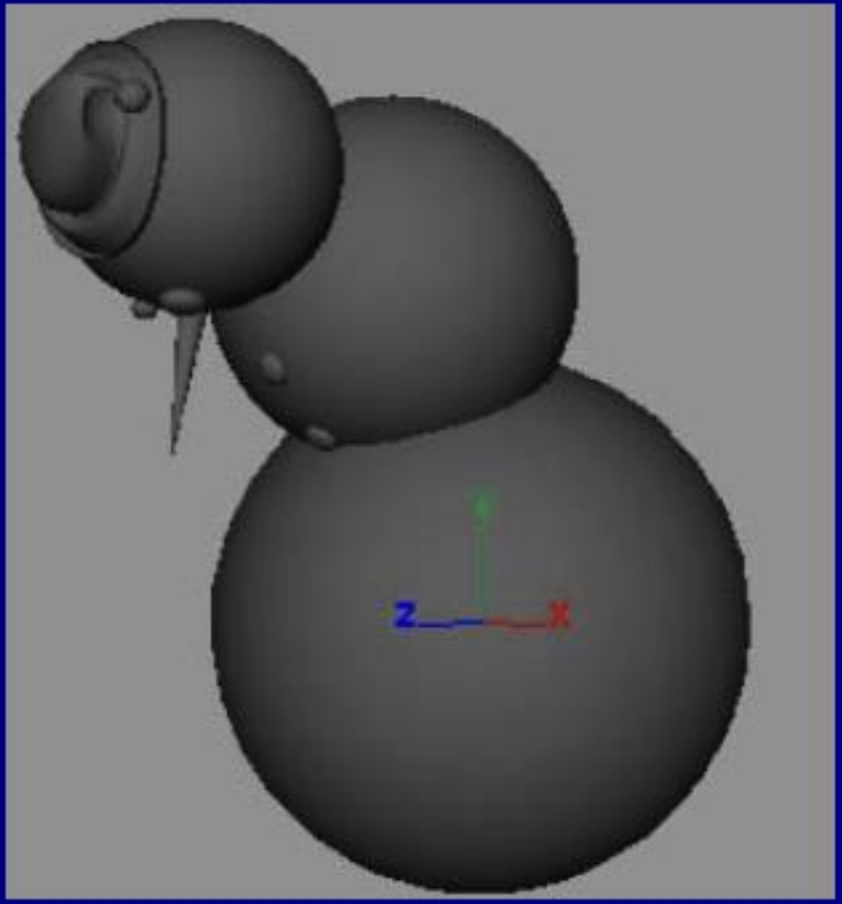




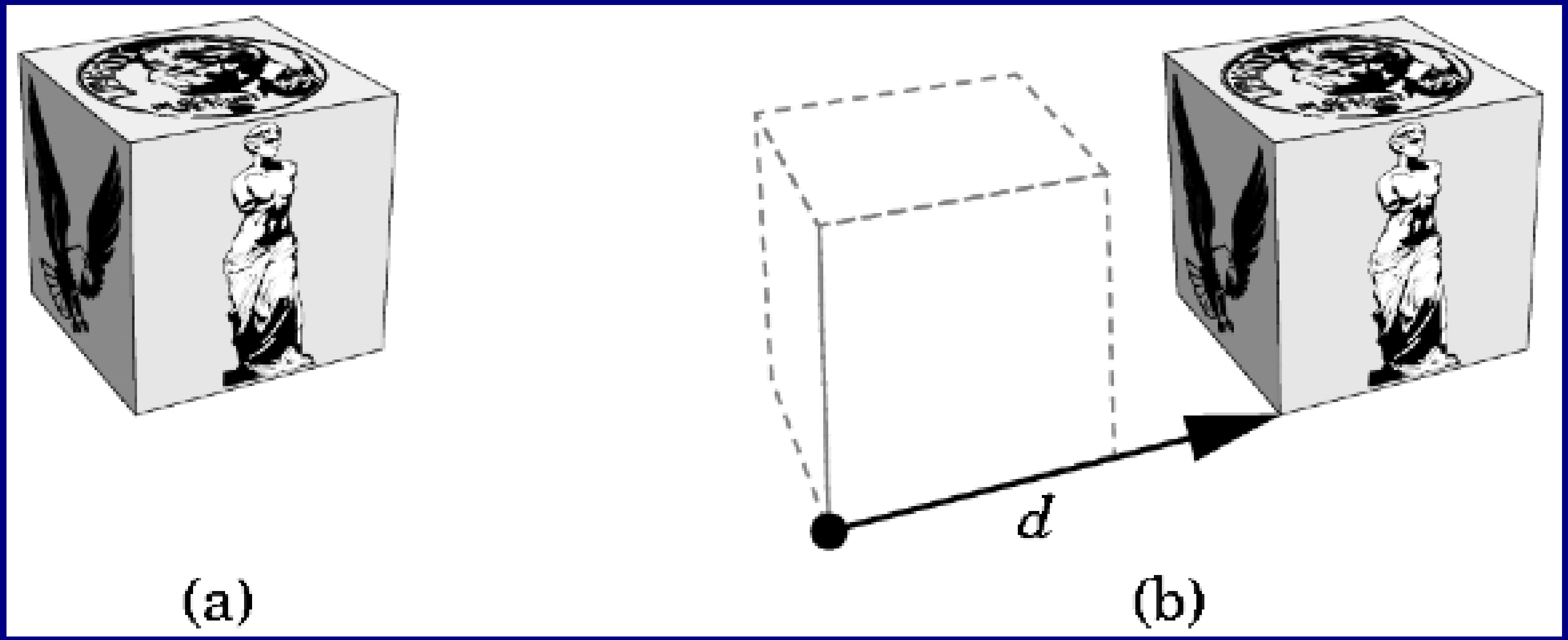


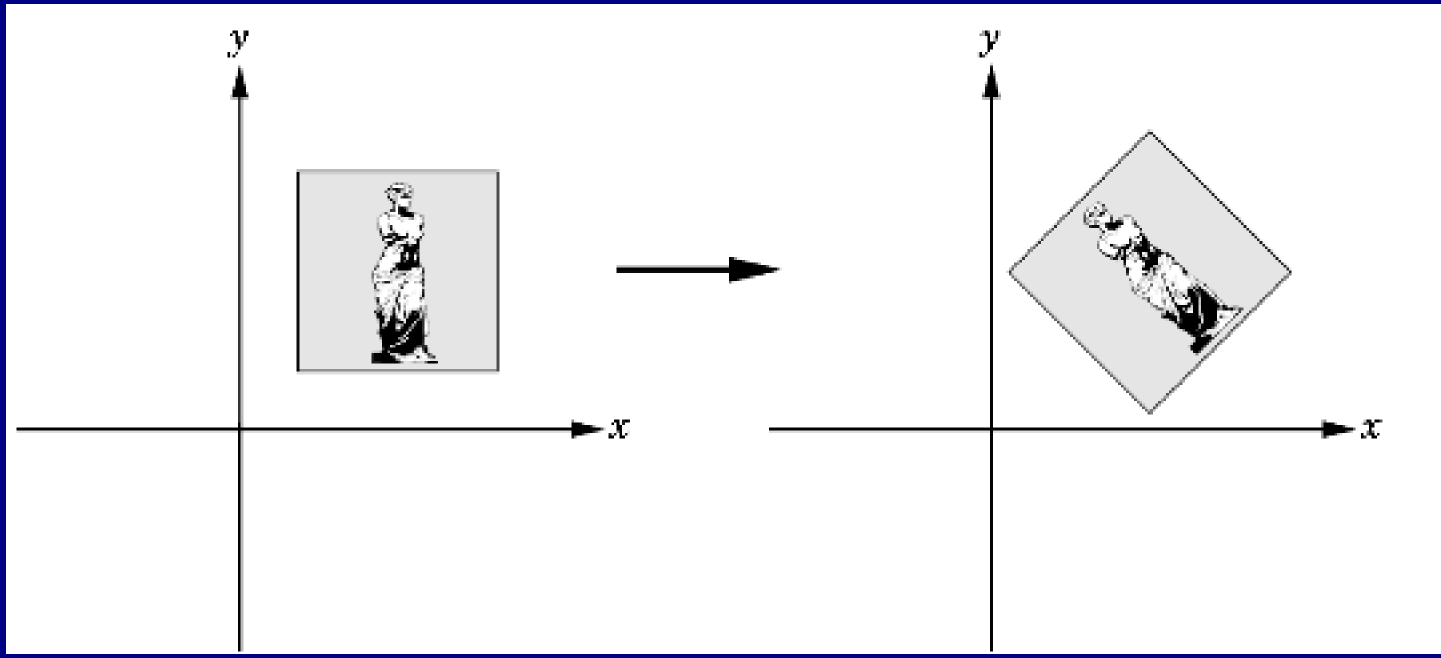
Examples

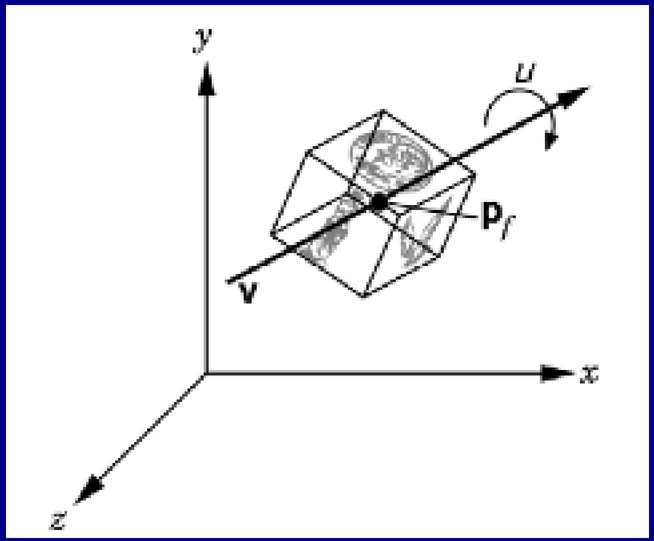
- Modeling with primitive shapes



Rigid Body Transformations

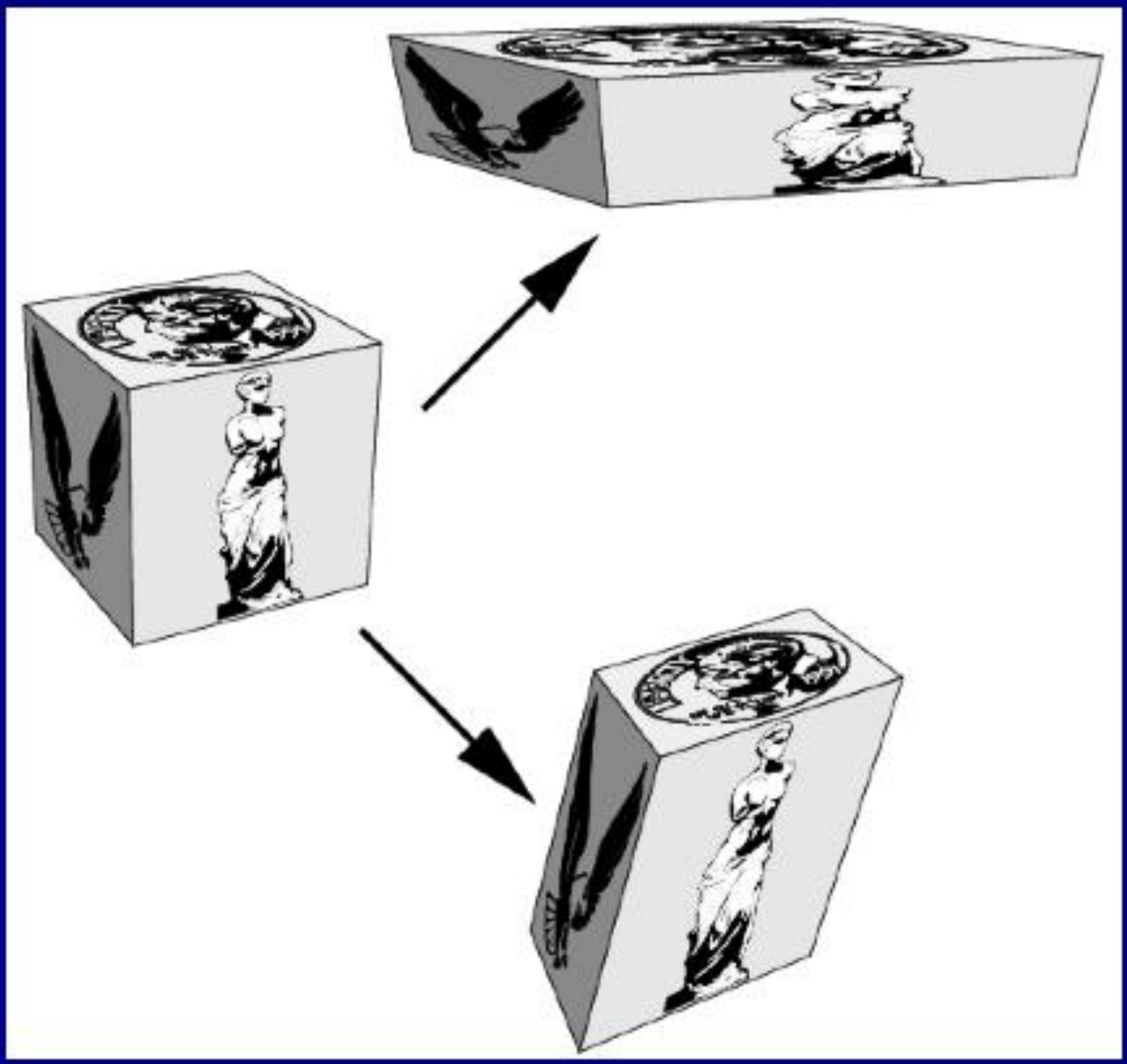


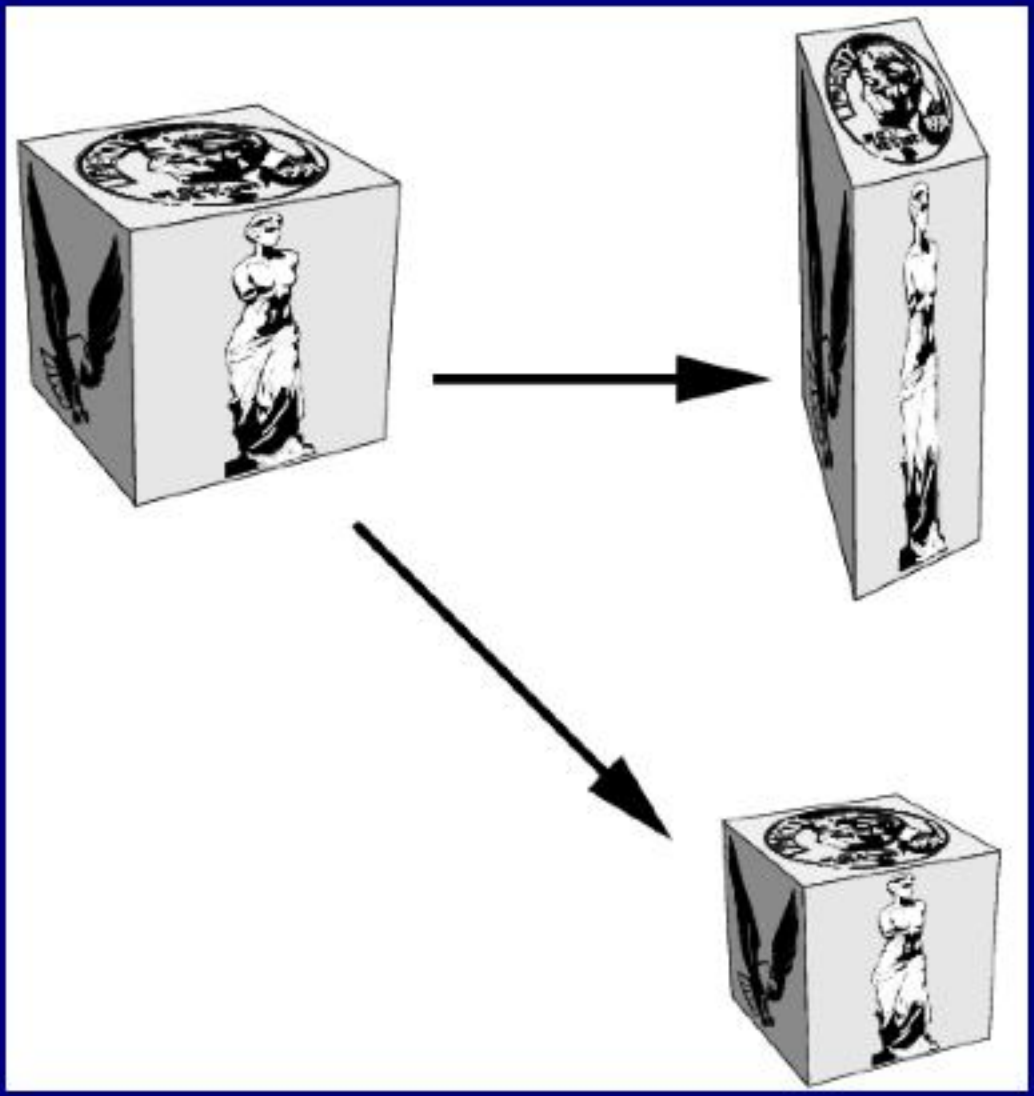




Rotation angle and line about which to rotate

Non-rigid Body Transformations





Distance between points on object do not remain constant

Basic 2D Transformations

Scale

Shear

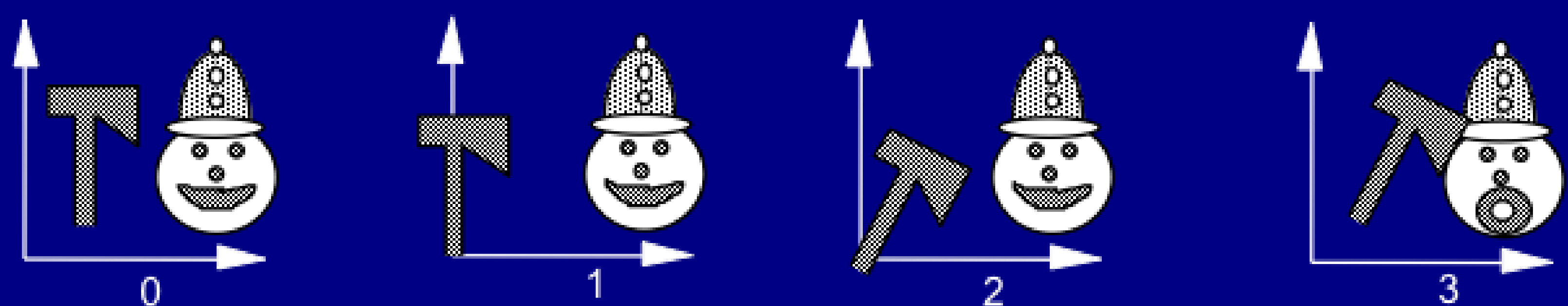
Rotate

chalkboard

Composition of Transformations

- Created by stringing basic ones together, e.g.
  - *"translate p to the origin, rotate, then translate back"*

  can also be described as a rotation about p
- Any sequence of linear transformations can be collapsed into a single matrix formed by multiplying the individual matrices together
- Order matters!
- Can apply a whole sequence of transformations at once



chalkboard

Translate to the origin, rotate, then translate back.

3D Transformations

- 3-D transformations are very similar to the 2-D case
- Scale
- Shear
- Rotation is a bit more complicated in 3-D
  - different rotation axes

chalkboard

But what about translation?

- Translation is not linear--how to represent as a matrix?

chalkboard

But what about translation?

- Translation is not linear--how to represent as a matrix?

chalkboard

- Trick: add extra coordinate to each vector
- This extra coordinate is the *homogeneous* coordinate, or *w*
- When extra coordinate is used, vector is said to be represented in *homogeneous coordinates*
- We call these matrices *Homogeneous* Transformations

Homogeneous 2D Transformations

The basic 2D transformations become

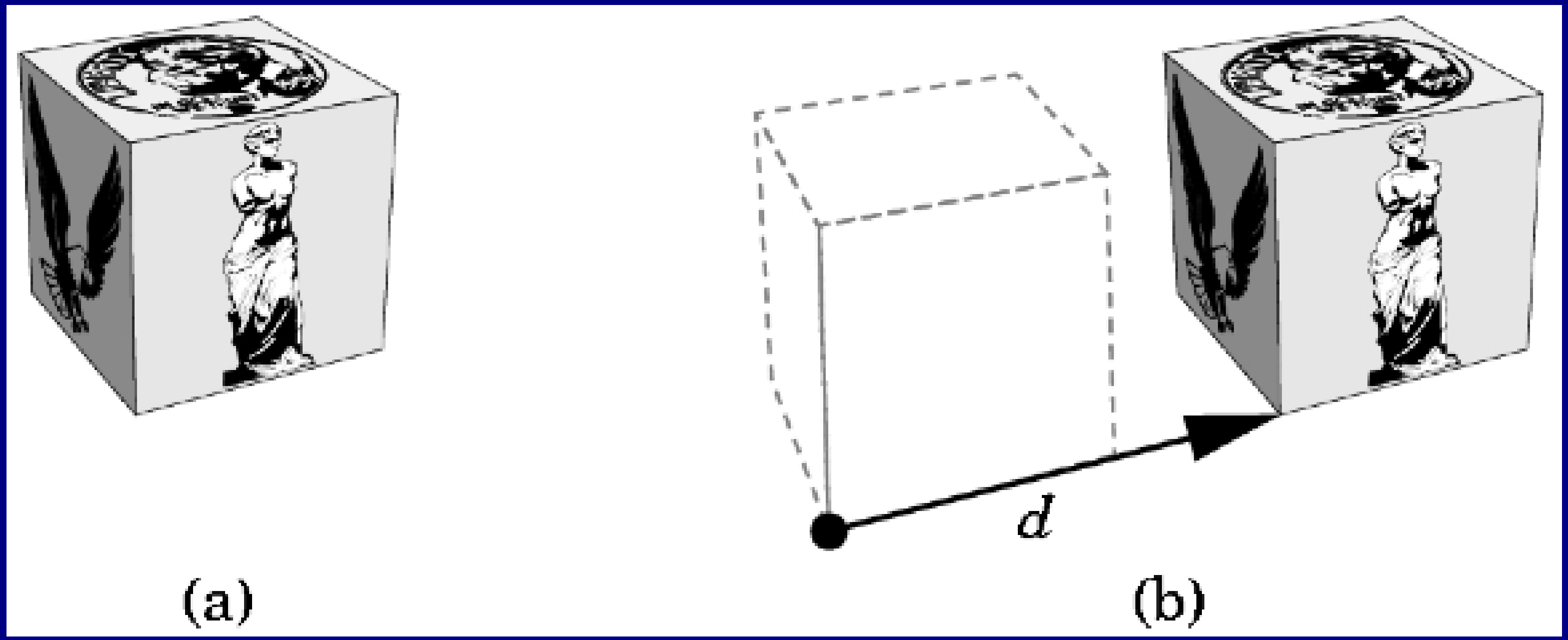
Translate: *Scale:* *Rotate:*

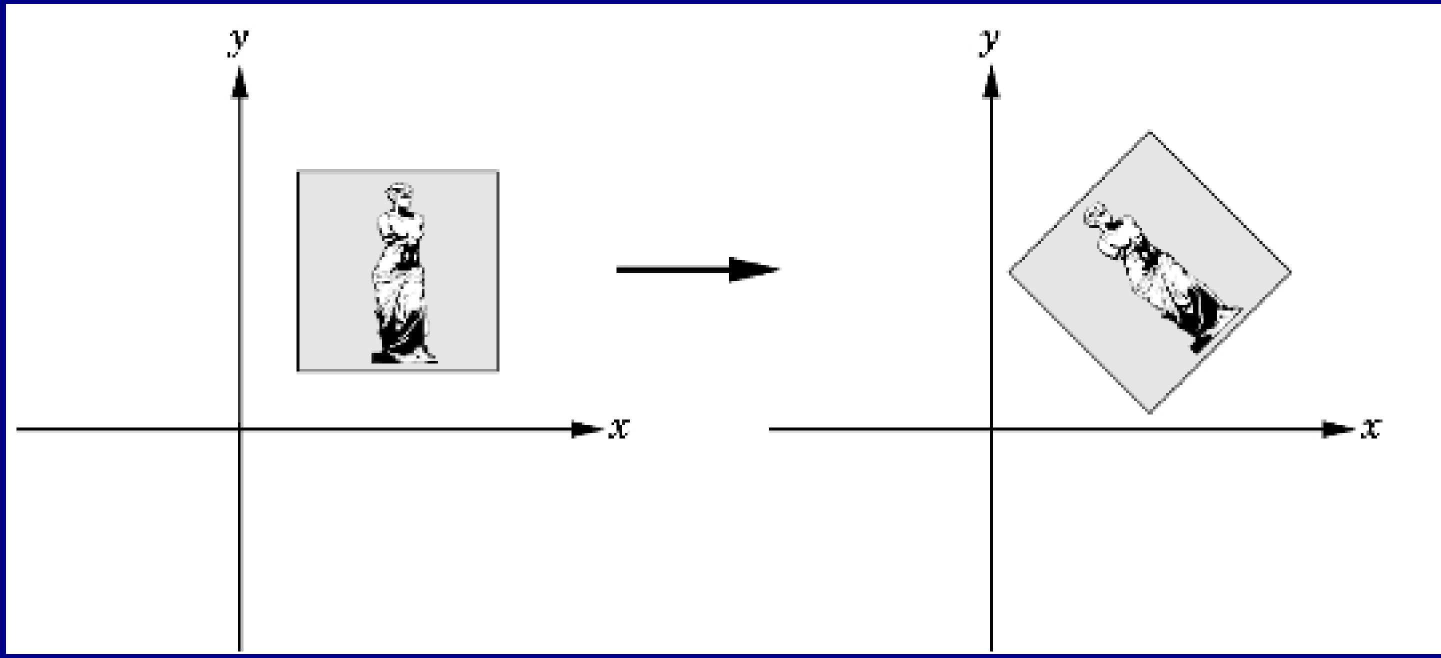
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

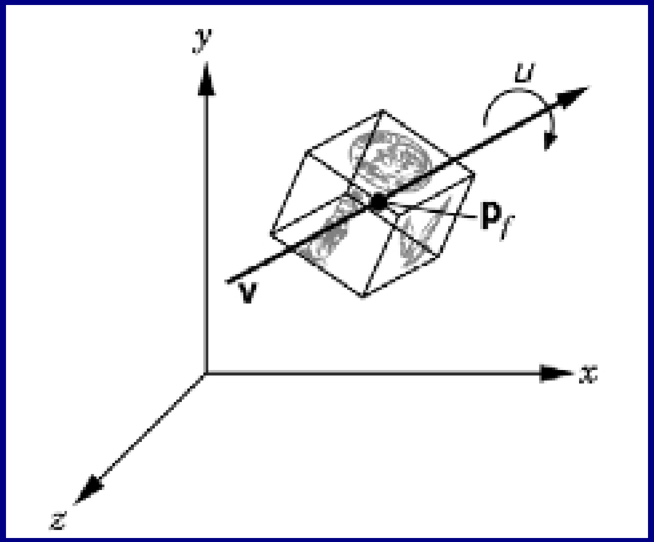
Now *any* sequence of translate/scale/rotate operations can be combined into a single homogeneous matrix by multiplication.

3D transforms are modified similarly

Rigid Body Transformations







Rotation angle and line about which to rotate

Rigid Body Transformations

- A transformation matrix of the form

$$\begin{bmatrix} x_x & x_y & t_x \\ y_x & y_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where the upper 2x2 submatrix is a rotation matrix and column 3 is a translation vector, is a *rigid body transformation.*

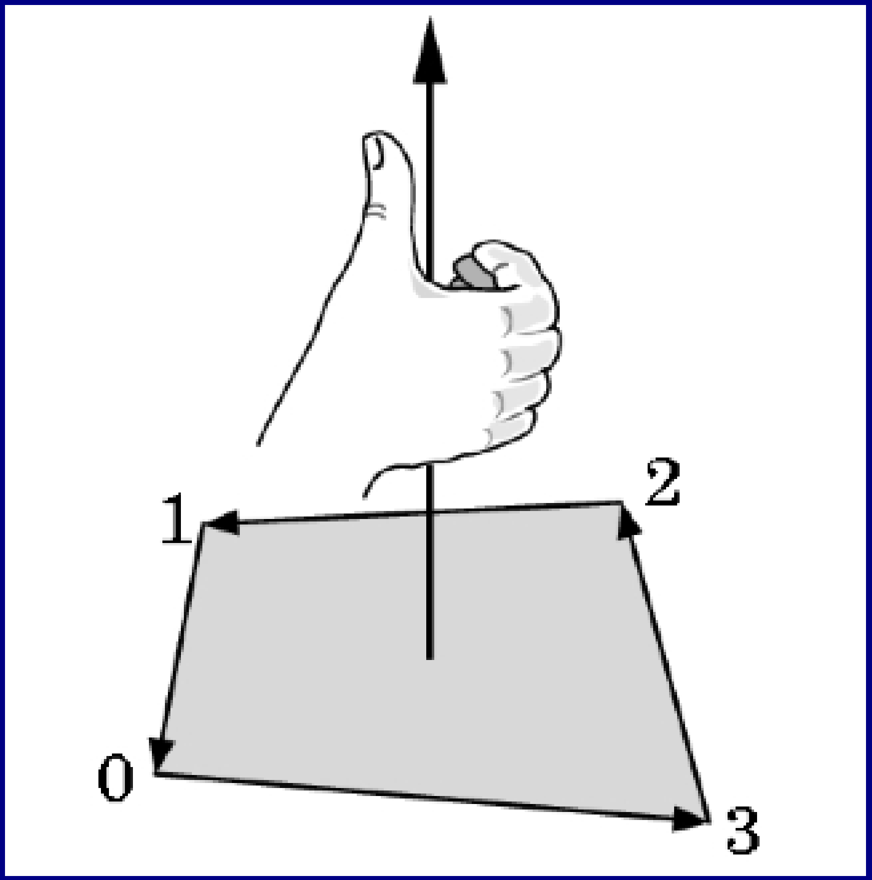
- Any series of rotations and translations results in a rotation and translation of this form (and no change in the distance between vertices)

What is a Normal?– refresher

Indication of outward facing direction for lighting and shading

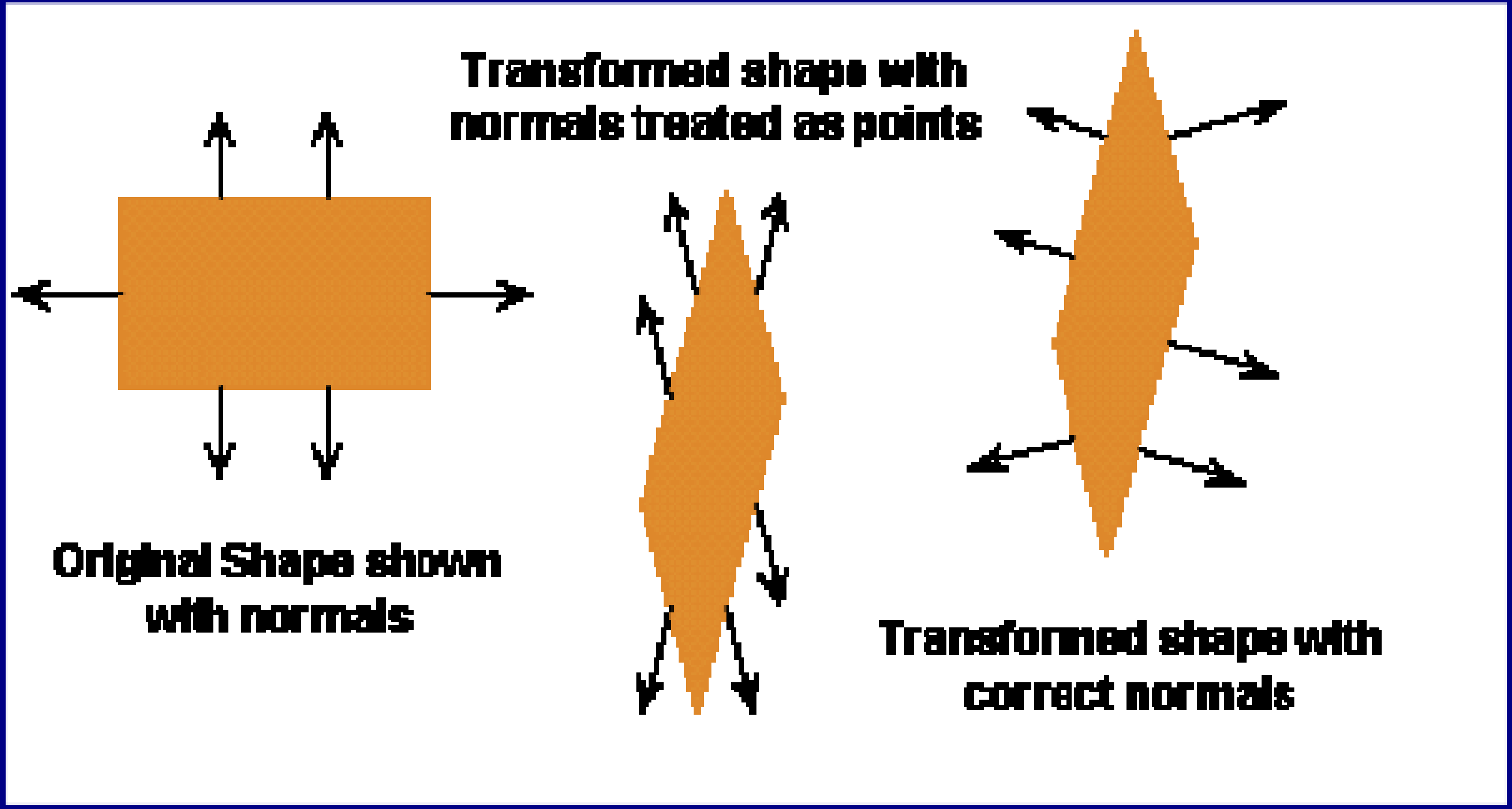
Order of definition of vertices in OpenGL

Right hand rule



Transforming Normals

- It's tempting to think of normal vectors as being like porcupine quills, so they would transform like points
- Alas, it's not so.
- We need a different rule to transform normals.



chalkboard

Announcements

- Reading for Tuesday: Shirley Ch: 2,6, & 7