3D transformations

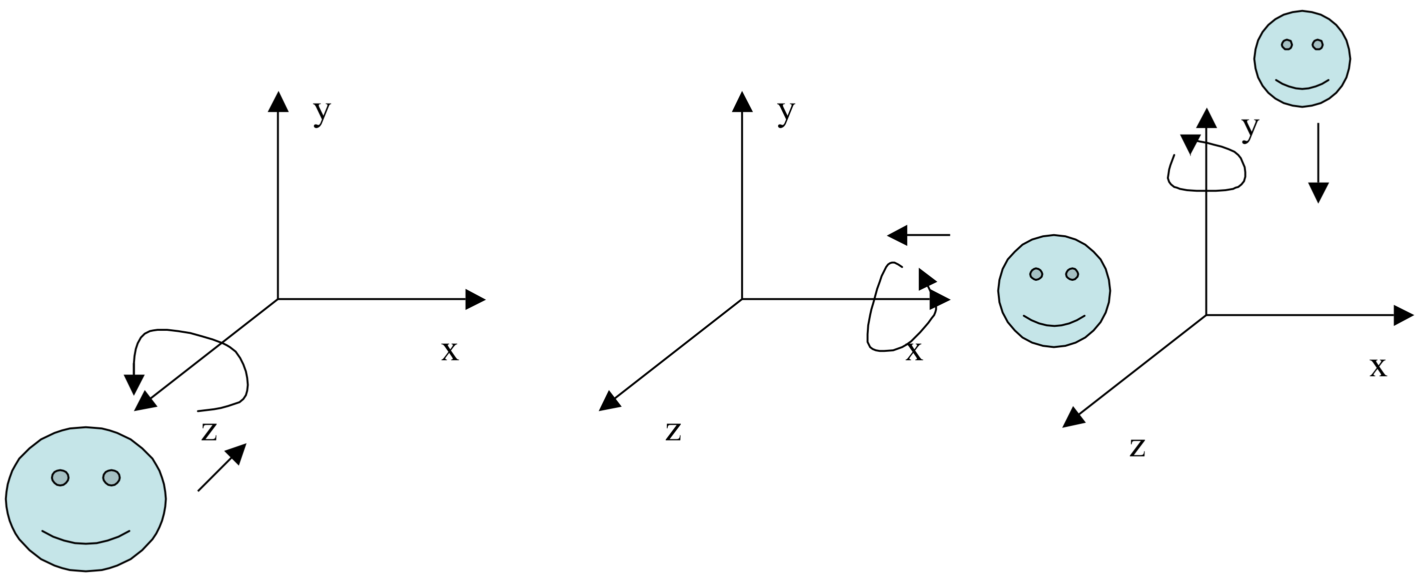
CS116A
Chris Pollett
Oct 18, 2004.

Outline

- 3D Rotations
- 3D Scaling

3D Rotations

- Easiest to describe such rotations in terms of rotations about the coordinate axes:



- Standard convention is to do these rotations counterclockwise along the axis looking into origin

Coordinate Axes Rotations

- z-axis rotations:

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

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

  $z' = z$

- This gives the matrix $R_z(\theta)$:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Other coordinates

- To figure out the rotations about the other axes we can cyclically permute the variables using: x-> y -> z -> x
- So a rotation about the x-axis is given by:

  $y' = y \cos \theta - z \sin \theta$

  $z' = y \sin \theta + z \cos \theta$

  $x' = x$
- What is the corresponding matrix?
- What is the matrix for a rotation about the y axis?

General 3D Rotations

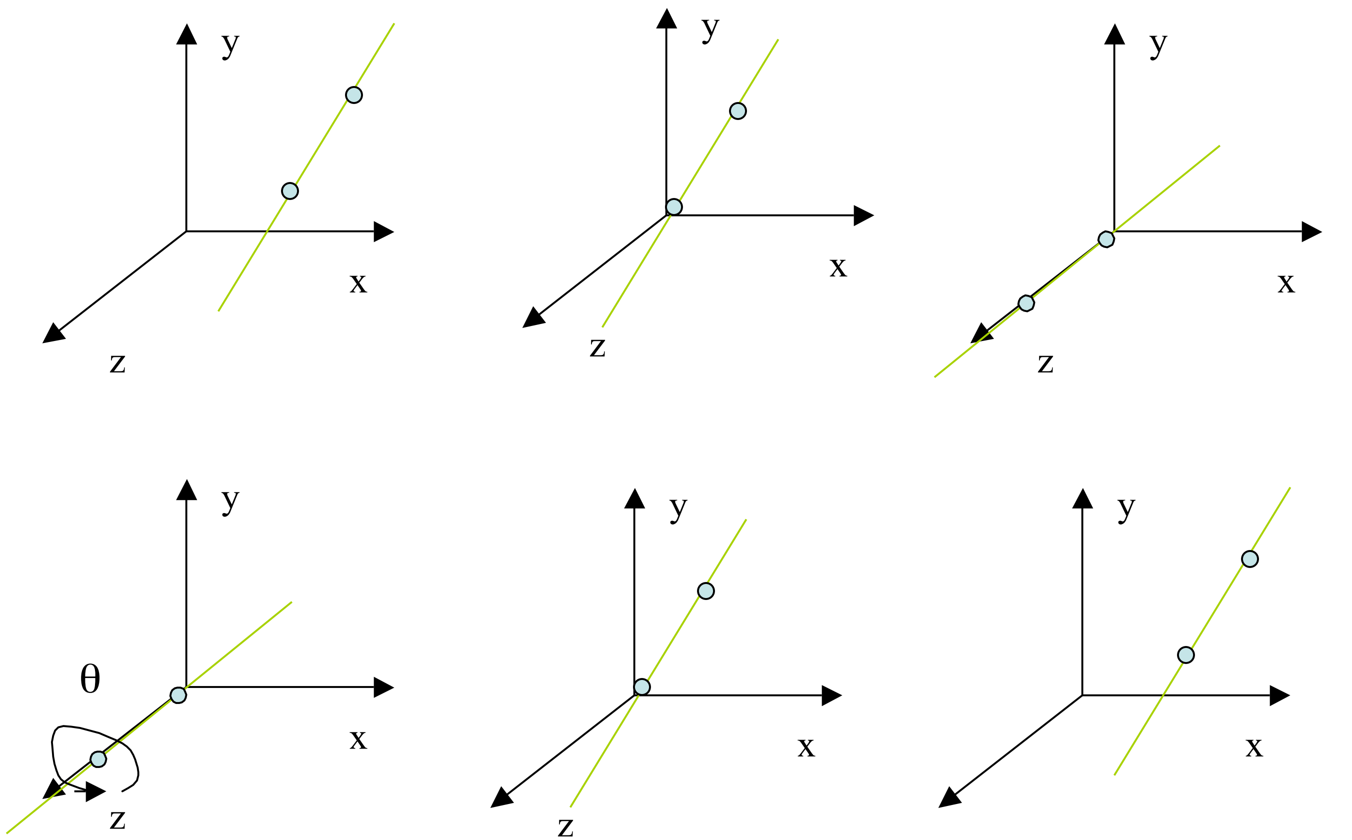
- A useful special case is when line we want to rotate about is parallel to one of the coordinate axes. In which case:
  - Translate object so that the rotation axis coincides with the parallel coordinate axis
  - Perform the rotation
  - Translate back.
- For example, if the parallel axis were the x axis, the sequence of matrices might look like:

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

More on general 3D rotations

- If the axis of rotation is not parallel to one of the coordinate axes, the procedure is a little more complicated:
  - Translate the object so that the rotation axis passes through the coordinate origin
  - Rotate the object so that the axis of rotation coincides with the z-coordinate axes. To do this:
    - rotate around x-axis until point is in xz-plane
    - rotate around y-axis until point is aligned with z-axis
  - Perform the specified rotation
  - Perform the inverses of the first two steps.

Some attempts at pictures



3D Scaling

- Matrix for 3D scaling in xyz and directions looks like:

$$\begin{bmatrix} s\_x & 0 & 0 & 0 \\ 0 & s\_y & 0 & 0 \\ 0 & 0 & s\_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To preserve the shape of the original figure you can do a so-called uniform scaling by setting s_x=s_y=s_z.
- To do a scaling respect to some point can do:
  T(x,y,z) S(s_x, s_y, s_z) T(-x, -y, -z)
- What is the inverse of this?