

CS559 – Lecture 12 Transformations, Viewing



These are course notes (not used as slides)
Written by Mike Gleicher, Oct. 2005
With some slides adapted from the notes of Stephen Cheney

Final version

© 2005 Michael L. Gleicher

Quaternions



- Not really covered here
- Use 4 numbers
 - 4D extension of complex numbers
- Use UNIT quaternions
 - $S(3) \sim SO(3)$
- A calculus is defined for quaternions
 - Multiplication, interpolation, ...
- Popular in advanced graphics
 - Stable
 - Compose
 - Efficient

Representation of 3D Rotations



- Two Theorems of Euler
 - Any rotation can be represented by a single rotation about an arbitrary axis (axis-angle form)
 - Any rotation can be represented by 3 rotations about fixed axes (Euler Angle form)
 - XYZ, XZX, any non-repeating set works
 - Each set is different (gets different singularities)
- Building rotations
 - Pick a vector (for an axis)
 - Pick another perpendicular vector (or make one w/cross product)
 - Get third vector by cross product

Viewing Transformations



- How do we transform the 3D world to the 2D window?
- Concepts:
 - World Coordinates
 - View (or window) VOLUME
 - Need 3rd dimension later to get occlusions right
 - Viewing Coordinates
 - 3D Viewing coordinates
- Separate Issues
 - Visibility (what's in front)
 - Clipping (what is outside of the view volume)

Euler Angles

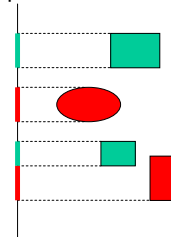


- Pick convention
 - Are axes local or global?
 - What order?
- Apply 3 rotations
- Good news: 3 numbers
- Bad news:
 - Can't add, can't compose
 - Many representations for any rotation
 - Singularities

Orthographic Projection



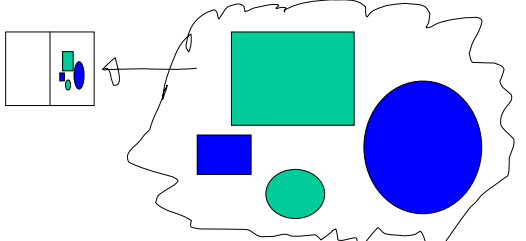
- Projection = transformation that reduces dimension
- Orthographic = flatten the world onto the film plane



View Volumes / Transformations



- Viewing transformation puts the world into the viewing volume
- A box aligned with the screen/image plane



Orthographic Projection



- Rotate / Translate / Scale View volume
 - Can map any volume to view volume
- Sometimes pick skews
- Things far away are just as big
 - No perspective
- Easy – and we can make measurements
 - Useful for technical drawings
 - Looks weird for real stuff
 - Far away objects too big

Canonical View Volume



- -1 to 1 (zero centered)
- XY is screen (y-up)
- Z is towards viewer (right handed coordinates)
 - Negative Z is into screen
- For this reason, some people like left-handed

Perspective Projection



- Farther objects get smaller
- Eye (or focal) point
- Image plane
- View frustum (truncated pyramid)
- Two ways to look at it:
 - Project world onto image plane
 - Transform world into rectangular view volume (that is then orthographically projected)

2 Views of Viewing Transform

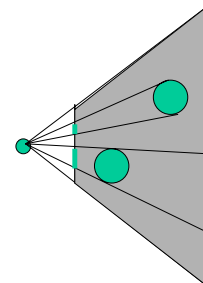


- Put world into viewing volume
- Position camera in world (view volume into world)
- Clip stuff that is outside of the volume
- Somehow get closer stuff to show up instead of farther things (if we want solid objects)

Perspective

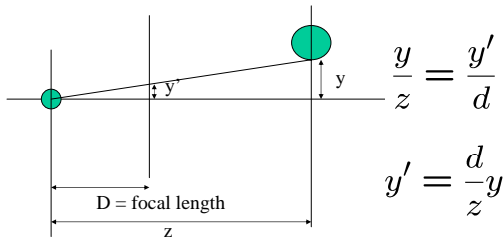


- Eye point
- Film plane
- Frustum
- Simplification
 - Film plane centered with respect to eye
 - Site down negative Z axis
 - Can transform world to fit



Basic Perspective

- Similar Triangles
- Warning = using d for focal length (like book)
 - F will be “far plane”



Shirley's Perspective Matrix

- After we do the divide, we get an unusual thing for z – it does preserve the order and keeps n&f

$$Px = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \frac{n}{z} \\ y \frac{n}{z} \\ n + f - \frac{fn}{z} \\ 1 \end{bmatrix}$$

Use Homogeneous coordinates!

- Use divide by w to get perspective divide
- Issues with simple version:
 - Font / back of viewing volume
 - Need to keep some of Z in Z (not flatten)

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

Camera Model

- The “window coordinate” system is all the we really know
- In a sense, it is the camera coordinate system
- Easiest to think about it as a camera taking a picture of the work
- Transform world coordinates into camera coordinates
 - Or, think about it the other way...

The real perspective matrix

- N = near distance, F = far distance
- Z = n put on front plane, z=f put on far plane

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{n+f}{n} & -f \\ 0 & 0 & -n & 0 \end{bmatrix}$$

How to describe cameras?

- Rotate and translate (and scale) the world to be in view
- The camera is a physical object (that can be rotated and translated in the world)
- Easier ways to specify cameras
 - Lookfrom/at/vup