Lecture 15 – 3D Transformations

Michael Gleicher
October 2007
Notes not for display

3D

- 3D coordinate system & handedness
- Prefer right-handed coordinate systems
- Right-hand rule

What happens in 3D?

- 4D Homogeneous Points
  - 4x4 matrices
- Basic transforms are the same
  - Translate
  - Scale
  - Skew
- Rotation is different
  - Rotation in 3D is more complicated?

What is a rotation?

- A transformation that:
  - Preserves distances between points
  - Preserves the zero
  - Preserves "handedness" (in 2D clockwiseness)
- A subset of linear transformations
- Some things that come out of these:
  - Axes remain perpindicular
  - Axes remain unit length
  - Cross product holds

Parameterizing Rotations

- Rotations are Linear Transformations
  - 2x2 matrix in 2D
  - 3x3 matrix in 3D
- The set of rotations = set of OrthoNormal Matrices
- Inconvenient way to deal with them
  - Can't work with them directly
  - Not stable (small change makes it not a rotation)
- Is there an easier way to parameterize the

Measuring rotation in 2D

- Pick 1 point (1,0)
- Any rotation must put this on a circle
- If you know where this point goes, can figure out any other point
  - Distances (w/point & origin) + handedness says where things go
- Parameterize rotations by distance around circle
  - Angle
- Issues with wrap around

Much harder in 3D

- Any point can go to a sphere
- That one point doesn't uniquely determine things
- No vector in R^n can compactly represent rotations
  - – Singularities
    - nearby rotations / far away numbers
    - Nearby numbers / far away rotations
- Hairy-Ball Theorem

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 perpindicular vector (or make one w/cross product)

Euler Angles

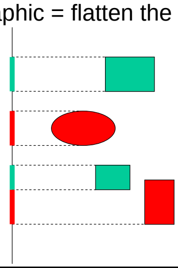
- Pick convention
  - – Are axes local or global?
  - – Local: roll, pitch, yaw
  - – What order?
- Apply 3 rotations
- Good news: 3 numbers
- Bad news:
  - – Can't add, can't compose
  - – Many representations for any rotation
  - – Singularities

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)

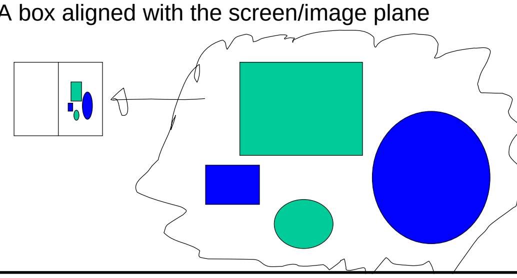
Orthographic Projection

- 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



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

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)

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