

# **Drawing in 3D: Visibility**

**CSCI 4229/5229  
Computer Graphics  
Fall 2007**

# Differences from 2D

- The third dimension (duh!)
- Depth perception
- Hidden lines and surfaces
- Realism
  - Lighting
  - Shading
  - Texture

# What is Visible?

- We see only the OUTSIDE of objects
  - Approximated by flat planes
- We cannot see the BACK side of objects
  - About half the planes are invisible
- We cannot see obscured objects
  - Interference zones
  - Hidden Line/Hidden Surface Algorithms

# Single Convex Object

- Only surfaces with +Z normal are visible
  - How do we tell front from back on the same surface
- Order of drawing surfaces are not important

# Multiple Convex Objects

- Order objects from far to near
- Rely on *painter's algorithm* to *paint over* obscured part of objects
  - Device must support erasing
- Does not work for concave objects
- Does not work when objects intersect

# Projection Z for Simple Rotation

```
glRotatef(ph , 1,0,0);  
glRotatef(th , 0,1,0);
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$
$$= \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ \sin \phi \sin \theta & \cos \phi & -\cos \theta \sin \phi \\ -\sin \theta \cos \phi & \sin \phi & \cos \theta \cos \phi \end{pmatrix}$$

# Z-Buffering

- Store depth (z) on a pixel by pixel basis
- Draw a new pixel only if it is nearer than existing z value for that pixel
  - Must initialize z-buffer before drawing
- Requires hardware support
  - 8/16/24/32 bits
  - “z-fighting” when dz too large

# Z-buffering + Face Culling

- Z-buffering ensures correct rendering
- Face Culling eliminates entire backward facing polygons (possibly lots of pixels)
  - Performance gain (on average 2x)
  - Requires more care in constructing objects



# OpenGL Notes

- Enable Z-buffer
  - `glutInitDisplayMode(GLUT_DEPTH);`
  - `glEnable(GL_DEPTH_TEST);`
    - Typically on for whole scene
- Enable face culling
  - `glEnable(GL_FACE_CULL);`
  - `glFrontFace(dir)`
    - `GL_CCW` (default) or `GL_CW`
  - `glCullFace(face)`
    - `GL_BACK` (default), `GL_FRONT` or `GL_FRONT_AND_BACK`