CS130 : Computer Graphics
Lecture 6: Graphics Pipeline (cont.)

Tamar Shinar
Computer Science & Engineering
UC Riverside
Graphics Pipeline

Geometric Pipeline:
- Transform
- Project
- Clip

Pixel Pipeline:
- OpenGL application program
- Pixel operations
- Rasterizer

Frame buffer
Transform
“Modelview” Transformation

Object coordinates

World coordinates

Eye coordinates

Model

View
Project

Geometric Pipeline

Transform → Project → Clip

Pixel Pipeline

OpenGL application program → Pixel operations → Rasterizer → Frame buffer
Projection:
map 3D scene to 2D image

OpenGL Super Bible, 5th Ed.
Orthographic projection

- parallel lines appear parallel (unlike perspective proj.)
- equal length lines appear equal length (unlike perspective proj.)
OpenGL Orthogonal Viewing

\texttt{glOrtho(left,right,bottom,top,near,far)}

View volume for an orthographic projection

\textit{near} and \textit{far} measured from camera

Orthographic, or parallel projection
- square or rectangular viewing volume
- anything outside volume is not drawn
Perspective projection
OpenGL Perspective Viewing

```c
glFrustum(xmin, xmax, ymin, ymax, near, far)
```

View volume (frustrum) for a perspective projection
Clip
Clip against view volume
Clipping against a plane

What’s the equation for the plane through \( \mathbf{q} \) with normal \( \mathbf{N} \)?
Implicit line equation:

\[ f(X) = N \cdot (X - X_0) = 0 \]
Clipping against a plane

What’s the equation for the plane through $\mathbf{q}$ with normal $\mathbf{N}$?

$$f(p) = ? = 0$$

<whiteboard>
Clipping against a plane

What's the equation for the plane through \( \mathbf{q} \) with normal \( \mathbf{N} \)?

\[
f(p) = \mathbf{N} \cdot (p - q) = 0
\]
Intersection of line and plane

How can we distinguish between these cases?
Intersection of line and plane

\[ f(a)f(b) \geq 0 \]

\[ f(a)f(b) < 0 \]
Intersection of line and plane

How can we find the intersection point?

<whiteboard>
Clip against view volume

We write down the line equations $p_{cb}(s)$ and $p_{ab}(t)$ and find the $s$ and $t$ where they intersect the plane.

$$s = \frac{N \cdot (q - c)}{N \cdot (b - c)}$$

$$t = \frac{N \cdot (q - a)}{N \cdot (b - a)}$$

need to generate new triangles
Hidden Surface Removal

Geometric Pipeline:
- Transform
- Project
- Clip

Pixel Pipeline:
- OpenGL application program
- Pixel operations
- Rasterizer
- Frame buffer
Occlusion

“painter’s algorithm”

draw primitives in back-to-front order
Occlusion

“painter’s algorithm”

draw primitives in back-to-front order

problem:
triangle intersection

who’s in front of whom?
“painter’s algorithm” draw primitives in back-to-front order

**problem**: occlusion cycle

also, sorting primitives by depth is slow
Use a z-buffer for hidden surface removal

at each pixel, record distance to the closest object that has been drawn in a depth buffer
Use a z-buffer for hidden surface removal

at each pixel, record distance to the closest object that has been drawn in a depth buffer

- assume both spheres of the same size, red drawn last
Use a z-buffer for hidden surface removal

Figure 1. Block diagram of OpenGL.

- each fragment must carry a depth
Use a z-buffer for hidden surface removal

http://www.beyond3d.com/content/articles/41/