Computer Graphics
Z Sweedyk
Lecture 8

Graphics Pipeline 5

1. Build Primitives (Model Coordinates)
2. Assemble Scene (World Coordinates)
3. Project into 2D (Projection Coordinates)
4. Normalize (Normalized Coordinates)
5. Scan Convert with Hidden Surface Removal

Now for clipping

Overview
- Vertex clipping
- Line clipping
- Polygon clipping

Vertex Clipping:
Is P in (or on)?
P is in iff it is in wrt each plane of the frustum
Vertex Clipping:
Is \( P \) in wrt a plane of the frustum?

\[ P \text{ is in iff } n \cdot v \geq 0 \]

\( n \) is the inward facing normal.
\( v \) is a vector from some point on the plane to \( P \).

What is an inward facing normal?

Line (Segment) Clipping:
Is \( L \) in, out, or both?

If both return \( L' \)

Line Clipping

- Slow test
- Speed up

Clipping a line segment against a plane of the frustum
Clipping a line segment against a plane of the frustum

Case: $p_0 \& p_1$ go on with next test

Clipping a line segment against a plane of the frustum

Case: $p_0 \& p_1$ out then $L$ is out

Clipping a line segment against a plane of the frustum

Case: $p_0$ in $\& p_1$ out _____ 
Case: $p_0$ out $\& p_1$ in _____

Replace out with intersection point

Equation of a Plane

This plane is the set of points $(x,y,z)$ where $ax + by + cz=d$

$\mathbf{n}=\langle a,b,c \rangle$

$\mathbf{d}=ax_0+by_0+cz_0$, where $(x_0,y_0,z_0)$ is any point on the plane.

Speed up

- Assign an "out-code" to the endpoints, $p_0$ and $p_1$, of $L$.
- The out-code for $p_i$ is $b_0b_1b_2b_3b_4b_5$
  - $b_j$ is 1 if $p_i$ is out wrt plane $j$
  - 0 otherwise

Speed Up Test

- $L$ is "definitely out" if the bitwise AND of the out-codes is non-zero.
- $L$ is "definitely in" if the bitwise OR of the out-codes is zero.
- Else let $j$ be a bit set to 1 in the OR of the outcodes. Clip to plane $j$. Compute the out-code for the new endpoint.
- Restart Test
Polygon Clipping

- If $v_0$ is in then write $v_0$
- For $i=1...n-1$
  - If $v_i$ in and $v_{i+1}$ in then write $v_{i+1}$
  - If $v_i$ out and $v_{i+1}$ out then do nothing
  - If $v_i$ in and $v_{i+1}$ out then write intersection point
  - If $v_i$ out and $v_{i+1}$ in then write intersection point and $v_{i+1}$

Speed Up

- Use line clipping speed up.