Topic 8: Visibility

• Elementary visibility computations:
  - Clipping
  - Backface culling

• Algorithms for visibility determination
  - Z-Buffer
  - Painter’s algorithm
  - Space partitions: BSP, AABB, OBB, octrees

Visibility Problem

What is NOT visible?

primitives outside of the field of view
back-facing primitives
primitives occluded by other objects closer to the camera

Polygon Clipping (wrt to a single plane)

Clip with respect to each plane of the volume in sequence!
Does the order of the planes matter?
Does it work for concave polygons?
Does it work for concave volumes?
Backface culling

\[ \mathbf{N} \cdot (\mathbf{P} - \mathbf{E}) > 0 \] is a back face?

Where in the graphics pipeline can we do backface culling?

@alec: Would be nice to redo this image
Occluded faces

Does backface culling always determine visibility completely for a single object?

In typical scenes some polygons will overlap, we must determine which portion of each polygon is visible to eye!

Painters Algorithm

Sort primitives in Z.
Draw primitives back to front (CBA).

Problems
• Large faces
• Intersecting faces
• Cycles

BSP tree

AABB tree
Visibility Problem: Z-buffer, A-buffer

**Z-buffer**: rasterize each polygon in the scene, keeping track of the polygon closest to the eye at each pixel.

**A-buffer**: accumulate pixel contribution to handle transparent polygons.

Visibility Algorithms

**Image space algorithms**
- Operate in display terms pixels.
- Visibility resolved to display resolution
- Examples: Z-buffer, ray-tracing
- $O(n \times \text{resolution})$

**Object Space algorithms**
- Analytically compute visible fragments
- Examples: painters algorithm, BSP
- $O(n^2)$