Hidden Surface Removal (or, visibility)

Adam Finkelstein
Princeton University
COS 426, Spring 2003

Overview
- Motivation
- Algorithms for HSR
  - Back-face detection
  - Depth sort
  - Ray casting
  - Scan-line
  - Z-buffer
  - Area subdivision
- Tradeoffs

Motivation
- Surfaces may be back-facing.
- Surfaces may be occluded.
- Surfaces may overlap in the image plane.
- Surfaces may intersect.

3D Rendering Pipeline

3D Primitives
  3D Modeling Coordinates
  3D World Coordinates
  Lighting
  3D World Coordinates

Viewing Transformation
  3D Camera Coordinates
  3D Screen Coordinates

Projection Transformation
  2D Screen Coordinates
  2D Image Coordinates

Clipping
  2D Screen Coordinates
  2D Image Coordinates

Scan Conversion
  2D Image Coordinates
  Image

Visibility algorithms

[Ref. Sutherland '74]
### Back-face detection

**Q:** When does this method break down?  
**A:** More than one object. Object not closed. Interreflect?

![Back-facing polygon](image)

**Q:** How do we test for back-facing polygons?  
**A:** Dot product of the normal and view directions.

### Depth sort

“Painter’s algorithm”
- Sort surfaces in order of decreasing maximum depth
- Scan convert surfaces in back-to-front order

![Depth sort](image)

### BSP Tree

- Binary space partition with solid cells labeled
  - Constructed from polygonal representations
  - Provides linear-time depth sort for arbitrary view

![BSP Tree](image)

(We’ll come back to this…)

### 3D Rendering Pipeline

![3D Rendering Pipeline](image)

**Depth sort comments**
- $O(n \log n)$
- Better with frame coherence?
- Implemented in software
- Render every polygon
- Often use BSP-tree or static list ordering

### Ray Casting

- Fire a ray for every pixel
  - If ray intersects multiple objects, take the closest

![Ray Casting](image)
Ray Casting Pipeline

Ray casting comments
- \( O(p \log n) \) for \( p \) pixels
- May (or not) use pixel coherence
- Simple, but generally not used

3D Modeling Coordinates
3D World Coordinates
Ray casting
Lighting
2D Image Coordinates
Image

Z-Buffer

- Color & depth of closest object for every pixel
- Update only pixels whose depth is closer than in buffer
- Depths are interpolated from vertices, just like colors

Z-Buffer

3D Rendering Pipeline

Z-buffer comments
- Polygons rasterized in any order
- Requires lots of memory
  - 1K x 1K x 24bits
  - Was expensive, cheap now
- Subject to aliasing (A-buffer)
- Commonly in hardware

Scan Line

Polygons

Scan-line comments
- Fully compute only visible pixels
- Coherence among along scans
- Commonly in software

Scan Line

3D Rendering Pipeline

3D Modeling Coordinates
3D World Coordinates
Lighting
2D Screen Coordinates
Clipping
2D Image Coordinates
Scan Conversion
Image

Area Subdivision

Warnock’s algorithm
- Fill area if:
  - All surfaces are outside area, or
  - Only one surface intersects area, or
  - One surface occludes other surfaces in area
- Otherwise, subdivide

Scan Line

Example spans
\( z=1.8 \)
\( z=3.8 \)
\( z=6 \)
3D Rendering Pipeline

Area subdivision comments
- Augments scan conversion
- Polygon coherence
- Commonly in software

Conclusions

Algorithms for HSR
- Back-face detection
- Depth sort
- Ray casting
- Scan-line
- Z-buffer
- Area subdivision

• Where in pipeline?
• Hardware / Software?
• Trends in hardware.