Hidden Surface Removal (or, *visibility*)

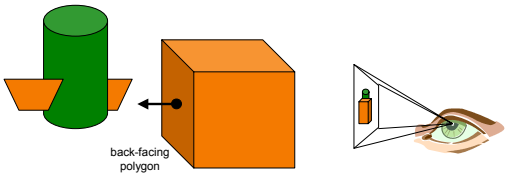
Adam Finkelstein
Princeton University
COS 426, Spring 2003

Overview

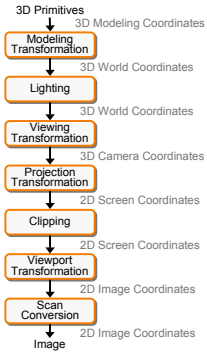
- Motivation
- Algorithms for HSR
  - Back-face detection
  - Depth sort
  - Ray casting
  - Scan-line
  - Z-buffer
  - Area subdivision
- Tradeoffs

Motivation

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



3D Rendering Pipeline

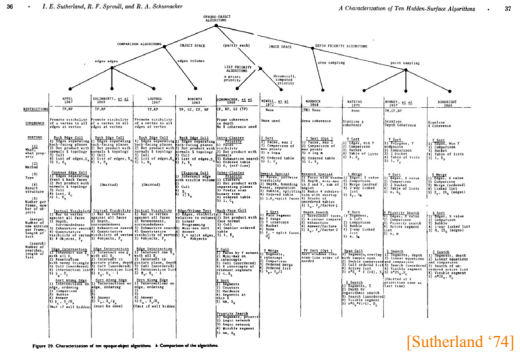


Somewhere in here we have to decide which objects are visible, and which objects are hidden.

Overview

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

Visibility algorithms

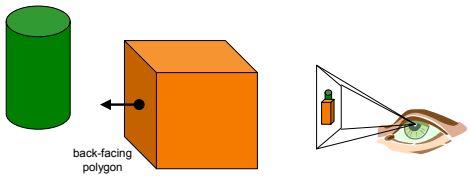


[Sutherland '74]

Back-face detection

Q: When does this method break down?

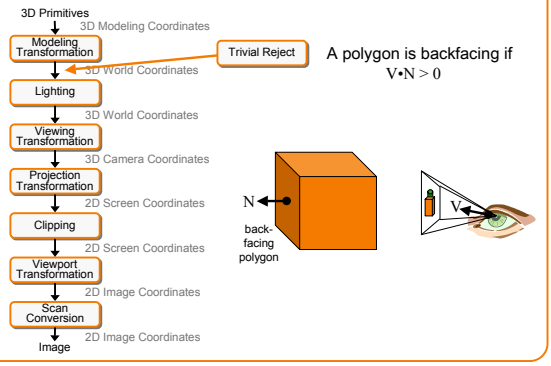
A: More than one object. Object not closed. Interreflect?



Q: How do we test for back-facing polygons?

A: Dot product of the normal and view directions.

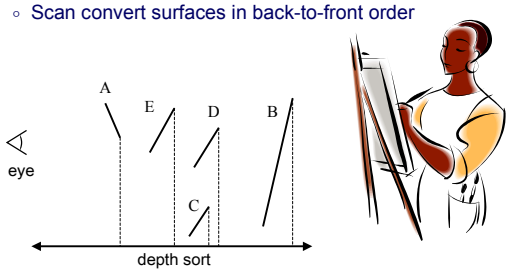
3D Rendering Pipeline



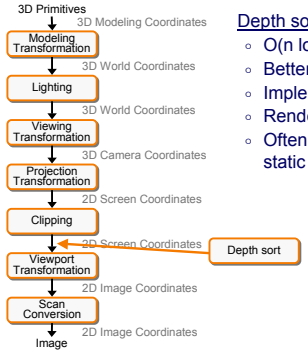
Depth sort

"Painter's algorithm"

- Sort surfaces in order of decreasing maximum depth
- Scan convert surfaces in back-to-front order



3D Rendering Pipeline

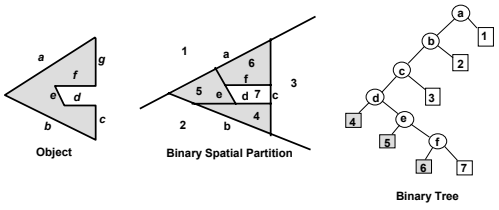


Depth sort comments

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

BSP Tree

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

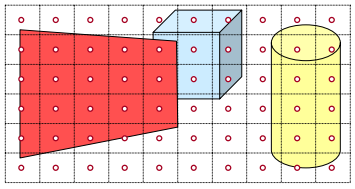


(We'll come back to this…)

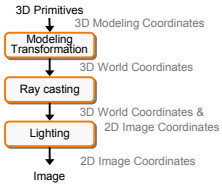
Naylor

Ray Casting

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



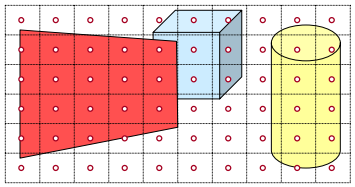
Ray Casting Pipeline



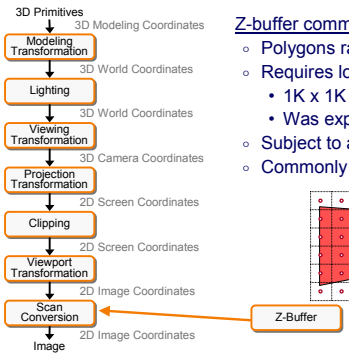
Ray casting comments
- O(p log n) for p pixels
- May (or not) use pixel coherence
- Simple, but generally not used

Z-Buffer

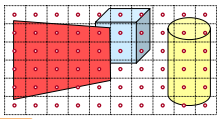
- Color & depth of closest object for every pixel
  - Update only pixels whose depth is closer than in buffer
  - Depths are interpolated from vertices, just like colors



3D Rendering Pipeline

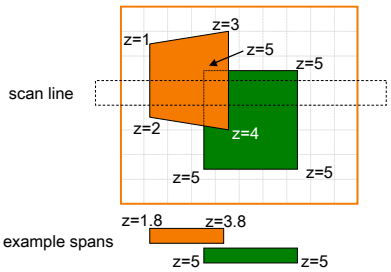


Z-buffer comments
- Polygons rasterized in any order
- Requires lots of memory
  - 1K x 1K x 24bits
  - Was expensive, cheap now
- Subject to aliasing (A-buffer)
- Commonly in hardware

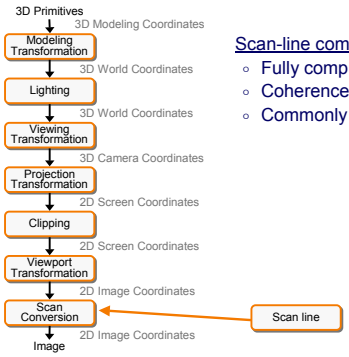


Scan-Line Algorithm

- For each scan line, construct spans
  - Sort by depth



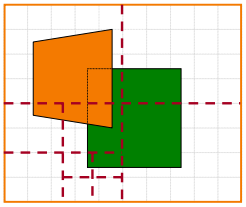
3D Rendering Pipeline



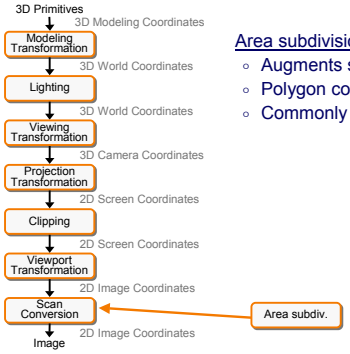
Scan-line comments
- Fully compute only visible pixels
- Coherence among along scans
- Commonly in software

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



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.