Visible Surface Detection

Visible Surface Detection

- Visible surface detection or hidden surface removal.
- Realistic scenes: closer objects occludes the others.
- Classification:
  - Object space methods
  - Image space methods

Object Space Methods

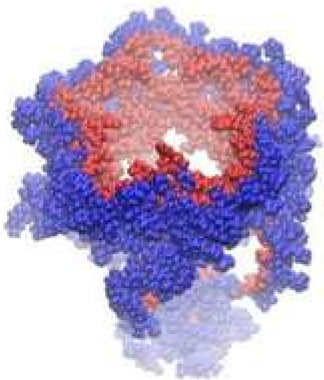
- Algorithms to determine which parts of the shapes are to be rendered in 3D coordinates.
- Methods based on comparison of objects for their 3D positions and dimensions with respect to a viewing position.
- For *N* objects, may require *N***N* comparision operations.
- Efficient for small number of objects but difficult to implement.
- Depth sorting, area subdivision methods.

Image Space Methods

- Based on the pixels to be drawn on 2D. Try to determine which object should contribute to that pixel.
- Running time complexity is the number of pixels times number of objects.
- Space complexity is two times the number of pixels:
  - One array of pixels for the frame buffer
  - One array of pixels for the depth buffer
- Coherence properties of surfaces can be used.
- Depth-buffer and ray casting methods.

Depth Cueing

- Hidden surfaces are not removed but displayed with different effects such as intensity, color, or shadow for giving hint for third dimension of the object.
- Simplest solution: use different colors-intensities based on the dimensions of the shapes.



Back-Face Detection

- Back-face detection of 3D polygon surface is easy
- Recall the polygon surface equation:

$$Ax + By + Cz + D < 0$$

- We need to also consider the viewing direction when determining whether a surface is back-face or front-face.
- The normal of the surface is given by:

$$\mathbf{N} = (A, B, C)$$

Back-Face Detection

- A polygon surface is a back face if:

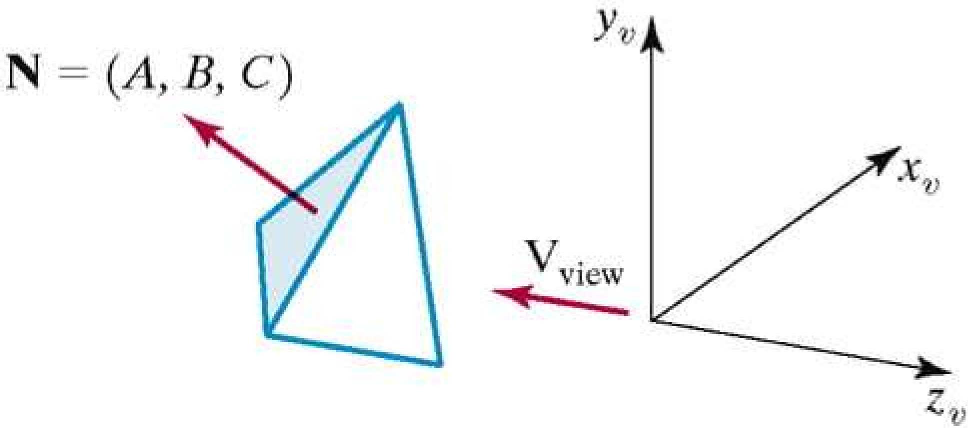
$$\mathbf{V}_{\text{view}} \cdot \mathbf{N} > 0$$

- However, remember that after application of the viewing transformation we are looking down the negative *z*-axis. Therefore a polygon is a back face if:

$$(0,0,-1) \cdot \mathbf{N} > 0$$

$$\text{or if } C < 0$$

Back-Face Detection



- We will also be unable to see surfaces with C=0. Therefore, we can identify a polygon surface as a back-face if:

$$C \leq 0$$

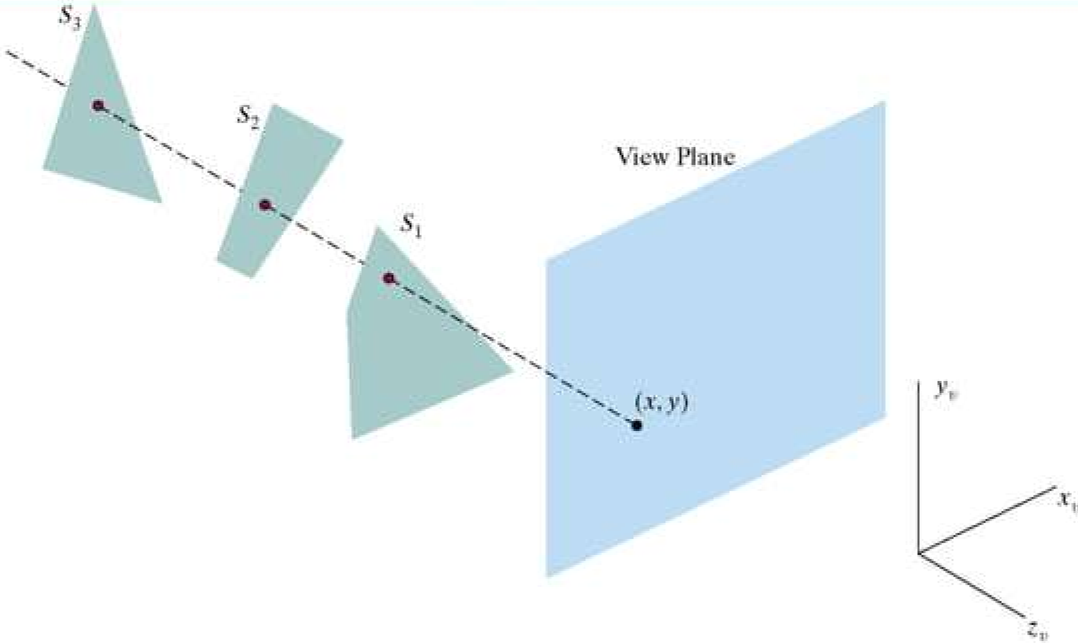
Back-Face Detection

- Back-face detection can identify all the hidden surfaces in a scene that contain non-overlapping convex polyhedra.
- But we have to apply more tests that contain overlapping objects along the line of sight to determine which objects obscure which objects.

Depth-Buffer Method

- Also known as *z*-buffer method.
- It is an image space approach
  - Each surface is processed separately one pixel position at a time across the surface
  - The depth values for a pixel are compared and the closest (smallest z) surface determines the color to be displayed in the frame buffer.
  - Applied very efficiently on polygon surfaces
  - Surfaces are processed in any order

Depth-Buffer Method



Depth-Buffer Method

- Two buffers are used
  - Frame Buffer
  - Depth Buffer
- The z-coordinates (depth values) are usually normalized to the range [0,1]

Depth-Buffer Algorithm

- Initialize the depth buffer and frame buffer so that for all buffer positions (x,y),

  depthBuff (x,y) = 1.0, frameBuff (x,y) =bgColor

- Process each polygon in a scene, one at a time
  - For each projected (x,y) pixel position of a polygon, calculate the depth z.
  - If z < depthBuff (x,y), compute the surface color at that position and set

  depthBuff (x,y) = z, frameBuff (x,y) = surfCol (x,y)

Calculating depth values efficiently

- We know the depth values at the vertices. How can we calculate the depth at any other point on the surface of the polygon.
- Using the polygon surface equation:

$$z = \frac{-Ax - By - D}{C}$$

Calculating depth values efficiently

- For any scan line adjacent horizontal *x* positions or vertical *y* positions differ by 1 unit.
- The depth value of the next position (x+1,y) on the scan line can be obtained using

$$z' = \frac{-A(x+1) - By - D}{C}$$

$$= z - \frac{A}{C}$$

Calculating depth values efficiently

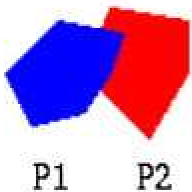
- For adjacent scan-lines we can compute the x value using the slope of the projected line and the previous x value.

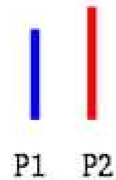
$$x' = x - \frac{1}{m}$$

$$\Rightarrow z' = z + \frac{A/m + B}{C}$$

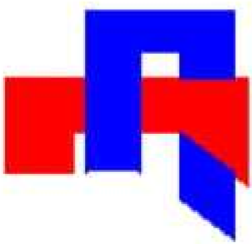
Depth-Buffer Method

- Is able to handle cases such as





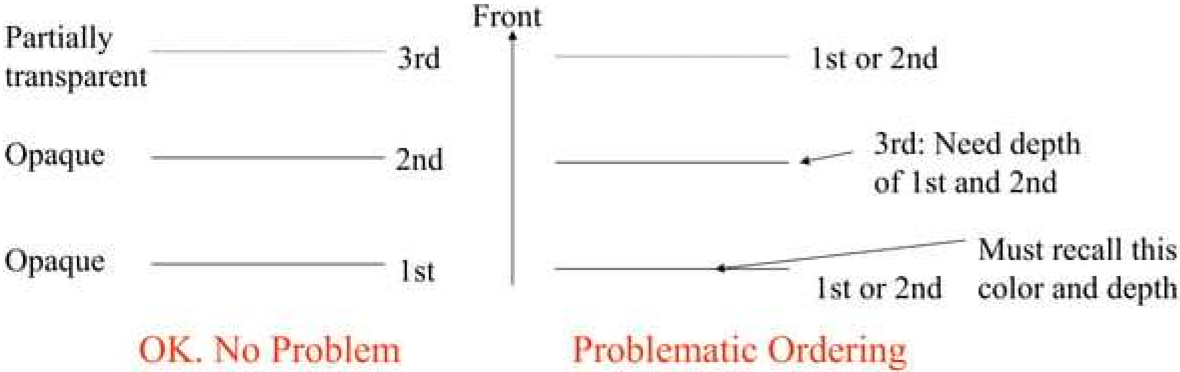
View from the Right-side



These polygons are both in front of and behind one another.

Z-Buffer and Transparency

- We may want to render transparent surfaces (alpha $\neq 1$) with a z-buffer
- However, we must render in back to front order
- Otherwise, we would have to store at least the first opaque polygon behind transparent one

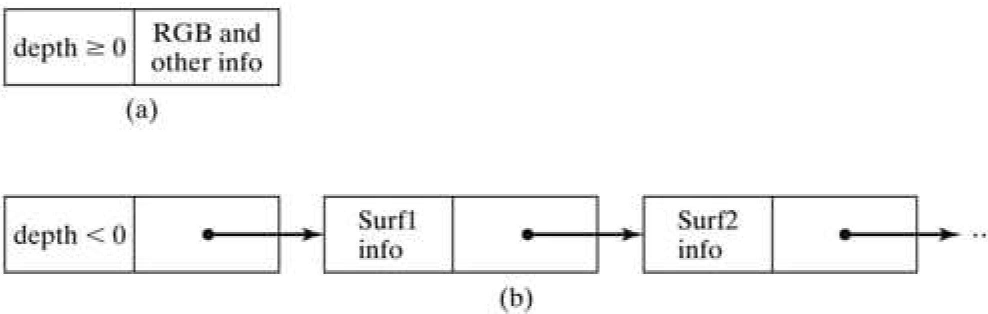


A-Buffer Method

- Extends the depth-buffer algorithm so that each position in the buffer can reference a linked list of surfaces.
- More memory is required
- However, we can correctly compose different surface colors and handle transparent surfaces.

A-Buffer Method

- Each position in the A-buffer has two fields:
  - a depth field
  - surface data field which can be either surface data or a pointer to a linked list of surfaces that contribute to that pixel position



Scan Line Method

- Extension of the scan-line algorithm for filling polygon interiors
  - For all polygons intersecting each scan line
    - Processed from left to right
    - Depth calculations for each overlapping surface
    - The intensity of the nearest position is entered into the refresh buffer

Tables for The Various Surfaces

- Edge table
  - Coordinate endpoints for each line
  - Slope of each line
  - Pointers into the polygon table
    - Identify the surfaces bounded by each line
- Polygon table
  - Coefficients of the plane equation for each surface
  - Intensity information for the surfaces
  - Pointers into the edge table

Active List & Flag

- Active list
  - Contain only edges across the current scan line
  - Sorted in order of increasing x
- Flag for each surface
  - Indicate whether inside or outside of the surface
  - At the leftmost boundary of a surface
    - The surface flag is turned on
  - At the rightmost boundary of a surface
    - The surface flag is turned off