Unit-6: Visible Surface Detection

Hidden surface removal or visible surface determination is the process used to determine which surfaces and parts of surfaces are not visible from a certain viewpoint.

Hidden surface determination is necessary to render an image correctly, so that one may not view features hidden behind the model itself, allowing only the naturally viewable portion of the graphics to be visible.

Visible surface detection methods are broadly classified according to whether they deal with objects or with their projected images.
These two approaches are:

1) ***Object space method:*** An object space method compares objects and parts of objects to each other within scene definition to determine which surfaces are visible.
2) ***Image space method:*** Visibility is decided point by point at each pixel position on the projection plane.

Most visible surface detection algorithm use image-space-method but in some cases object space methods are also used for it.

Algorithms

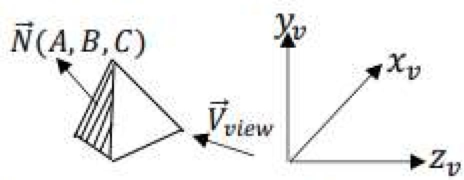
1. Back-face Detection (Plane equation Method)

A fast and simple object space method used to remove hidden surface from a 3D object drawing is known as "Plane equation method".
A point (x, y, z) is inside a surface with plane parameter A, B, C & D if

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

i.e. it is in back face.
Let $\vec{N}$ be the normal vector for a polygon and $\vec{V}_{view}$ be the vector in viewing direction, then a polygon surface is back if

$$\vec{V}_{view} \cdot \vec{N} > 0$$



- For the left handed viewing system if the 'z' component of the normal vector is positive, then it is back face. If the 'z' component of the vector is negative then it is a front face.
- For the right handed viewing system if the 'z' component of the normal vector is negative, then it is back face. If the 'z' component of the vector is positive then it is a front face.

Limitations:

- This method works fine for convex polyhedral, but not necessarily for concave polyhedral or overlapping objects. So, we need to apply other methods to further determine where the obscured faces are partially or completely hidden by other objects (e.g. Using Depth-Buffer Method or Depth-sort Method).
- This method can only be used on solid objects modeled as a polygon mesh.

2. Depth Buffer Method (Z-Buffer Method)

- It is an image space method for detecting visible surface.
- The Z-buffer method compares surface depths of each pixel position on the projection plane. Normally z-axis is represented as depth.
- In this method, two buffers are used:
    1) Depth Buffer: Stores depth values for each (x, y) position.
    2) Frame Buffer: Stores the intensity values for each position.

Algorithm:

1) Set the buffer values:

DepthBuffer(x, y)=0,

FrameBuffer(x, y)=Background color

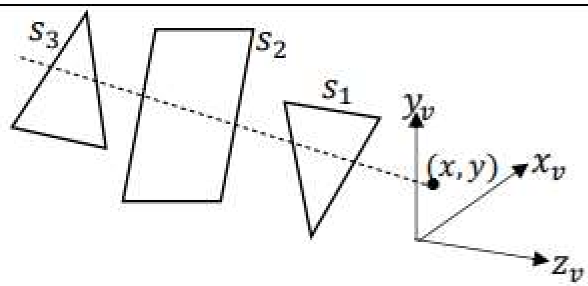
2) Process each polygon, one at a time as follows:

- *For each projected (x, y) pixel position of a polygon, calculate depth 'z'.*
- *If z>DepthBuffer(x,y), compute surface color and set*

  DepthBuffer(x, y)=z,

  FrameBuffer(x, y)=surfacecolor(x, y)

After all surface have been processed, the depth buffer contains depth values for visible surface and frame buffer contains corresponding color values for those surface.



In the figure, at view plane position (x, y), surface S_1 has the smallest depth from the view plane, so it is visible at that position.

Depth value for a surface position (x, y) are calculated from the plane equation as;

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

Now, the depth z' of the next position (x+1, y) is obtained as; $z' = z - \frac{A}{C}$

Advantages:

- It is easy to implement.
- It reduces the speed problem if implemented in hardware.
- It processes one object at a time

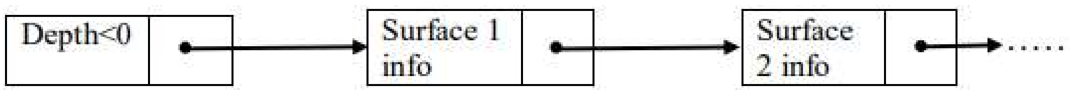
Disadvantages:

- It requires large memory
- It is a time consuming process as it requires comparison for each pixel instead of for the entire polygon.
- Deals only with opaque object but not for transparent object.

3. A-Buffer Method

- The A-buffer method is an extension of the depth-buffer method.
- This method is also known as **anti-aliased** or **area-averaged** or **accumulation buffer**.
- The A-buffer method is a visibility detection method developed at Lucas film studio for the surface rendering system REYES (Render Everything You Ever Saw).
- The A-buffer expands on the depth buffer method to allow transparencies.
- It consists of accumulation(A) buffer which as two fields:
    - ***a) Depth field:*** It stores a positive or negative real number.
    - ***b) Intensity field:*** It stores surface data or a pointer value.

depth ≥ 0	RGB and other info



If depth field is non negative (>=0), it indicates single surface. The intensity field then stores the RGB components of the surface color at that point and the percent of pixel coverage.

If depth is negative (<0) it indicates multiple surface. The intensity field then stores a pointer to a linked list of surface data.

Surface information in A-buffer includes:

- Depth
- Surface identifier
- Opacity parameter
- Percentage of area coverage
- RGB intensity component
- Pointer to the next surface

Advantage:

- It provides anti-aliasing in addition to what Z-buffer does.

Disadvantage:

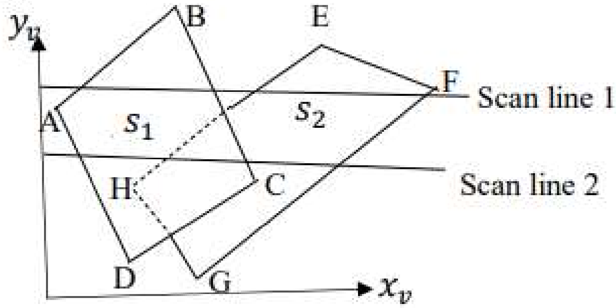
- It is slightly costly than Z-buffer method because it requires more memory in comparison to the Z-buffer method.

4. Scan-Line Method

- It is an image-space method for identifying visible surface.
- It computes and compares depth values along the various scan lines for the scene.
- Surfaces are processed using information stored in the polygon table.
- An active list of edges is formed for each scan line which stores only those edges that crosses the scan line in order of increasing 'x'.
- Also a flag is set for each surface that is set on or off to indicate whether a position along a scan line is either inside or outside the surface.
- Pixel position across each scan-line are processed from left to right.
- At the left intersection with a surface the surface flag is turned on and at the right intersection point the flag is turned off.
- We only need to perform depth calculation when more than one surface has its flag turned on at a certain scan-line position

The active list for scan line 1 contains edges AB, BC, EH & FG. Similarly for scan line 2, active list contains edges AD, EH, BC &FG.

	Edge pairs	Flags for S_1	Flags for S_2
For scan line 1	AB & BC	on	off
	BC & EH	off	off
	EH & FG	off	on
For scan line 2	AD & EH	on	off
	EH & BC	on	on
	BC & FG	off	on



Here, between EH & BC, the flags for both surfaces are ON so depth calculation is necessary. But between other edge pairs no depth calculation is necessary.

Advantage:

- Any number of overlapping polygon surfaces can be processed with this method.

Limitation:

- The scan line method runs into trouble when surface cut through each other or otherwise cyclically overlap. Such surface need to be divided.