Visible Surface Detection Methods

1. Back-Face Detection

A fast and simple object-space method for identifying the back faces of a polyhedron is based on the "inside-outside" tests. A point x,y,z is "inside" a polygon surface with plane parameters A, B, C, and D if When an inside point is along the line of sight to the surface, the polygon must be a back face we are inside that face and cannot see the front of it from our viewing position. We can simplify this test by considering the normal vector N to a polygon surface, which has Cartesian components A,B,C.

In general, if V is a vector in the viewing direction from the eye or "camera" position, then this polygon is a back face if

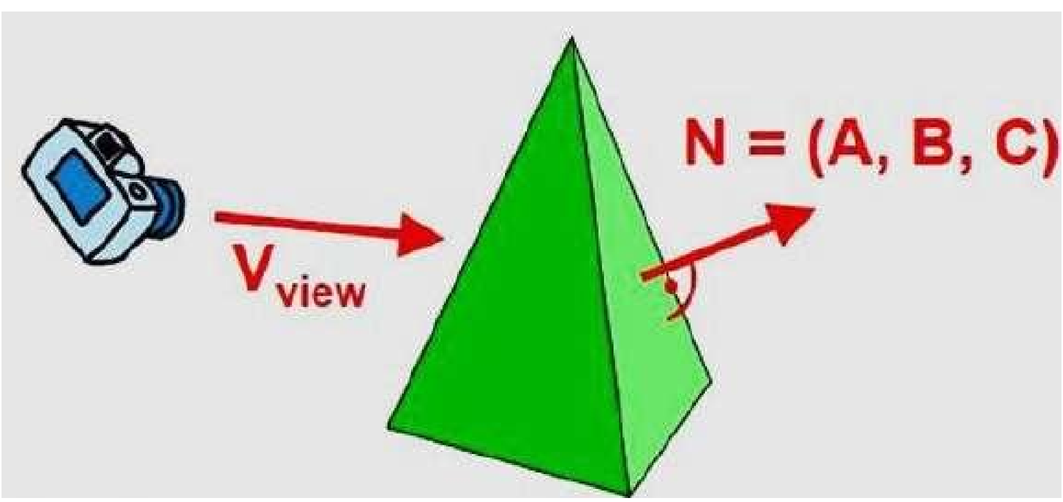
V.N > 0

Furthermore, if object descriptions are converted to projection coordinates and your viewing direction is parallel to the viewing z-axis, then −

V = (0, 0, Vz) and V.N = VZC

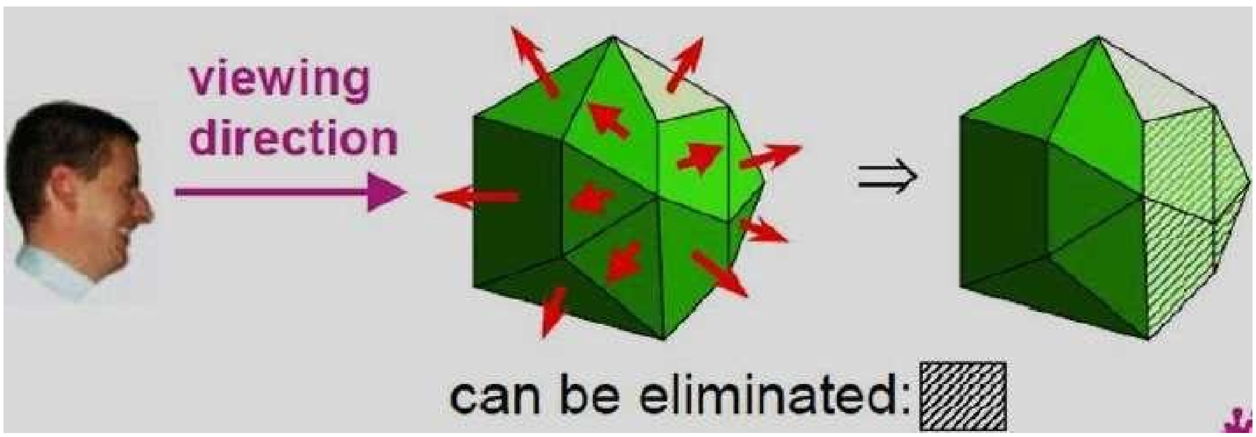
So that we only need to consider the sign of C the component of the normal vector N.

In a right-handed viewing system with viewing direction along the negative Z_v axis, the polygon is a back face if C < 0. Also, we cannot see any face whose normal has z component C = 0, since your viewing direction is towards that polygon. Thus, in general, we can label any polygon as a back face if its normal vector has a z component value − C <= 0



Similar methods can be used in packages that employ a left-handed viewing system. In these packages, plane parameters A, B, C and D can be calculated from polygon vertex coordinates specified in a clockwise direction unlike the counterclockwise direction used in a right–handed system.

Also, back faces have normal vectors that point away from the viewing position and are identified by C >= 0 when the viewing direction is along the positive Z_v axis. By examining parameter C for the different planes defining an object, we can immediately identify all the back faces.



Considering (a),

V.N = |V||N|Cos(angle)

if 0 <= angle =< 90   and V.N > 0

Hence, Back-face.

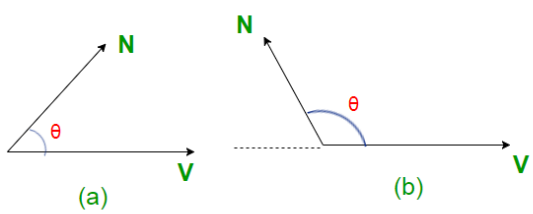
Considering (b),

V.N = |V||N|Cos(angle)

if 90 < angle <= 180,

then cos(angle) < 0 and V.N < 0

Hence, Front-face.



2. Depth Buffer / Z−Buffer Method

This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest visible surface.

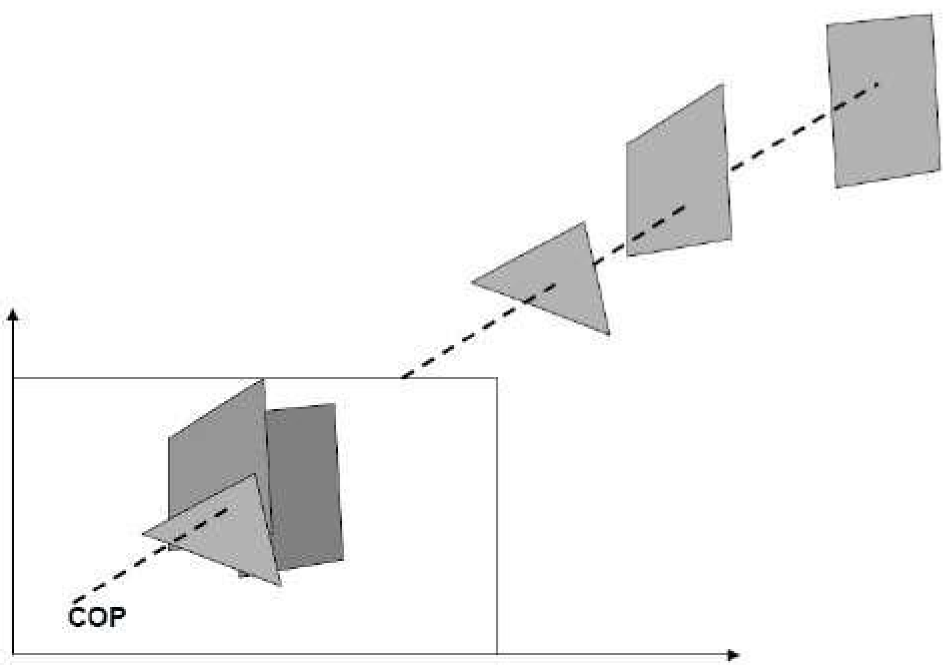
In this method each surface is processed separately one pixel position at a time across the surface. The depth values for a pixel are compared and the closest smallest z surface determines the color to be displayed in the frame buffer.

It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order. To override the closer polygons from the far ones, two buffers named frame buffer and depth buffer, are used.

Depth buffer is used to store depth values for x,y position, as surfaces are processed $0 \le depth \le 1$.

The frame buffer is used to store the intensity value of color value at each position x,y.

The z-coordinates are usually normalized to the range [0, 1]. The 0 value for z-coordinate indicates back clipping pane and 1 value for z-coordinates indicates front clipping pane.



Algorithm

Step-1 − Set the buffer values −

Depthbuffer x,y = 0

Framebuffer x,y = background color

Step-2 − Process each polygon, One at a time

For each projected x,y pixel position of a polygon, calculate depth z.

If Z > depthbuffer x,y

Compute surface color,

set depthbuffer x,y = z,

framebuffer x,y = surfacecolor x,y

Advantages

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

Disadvantages

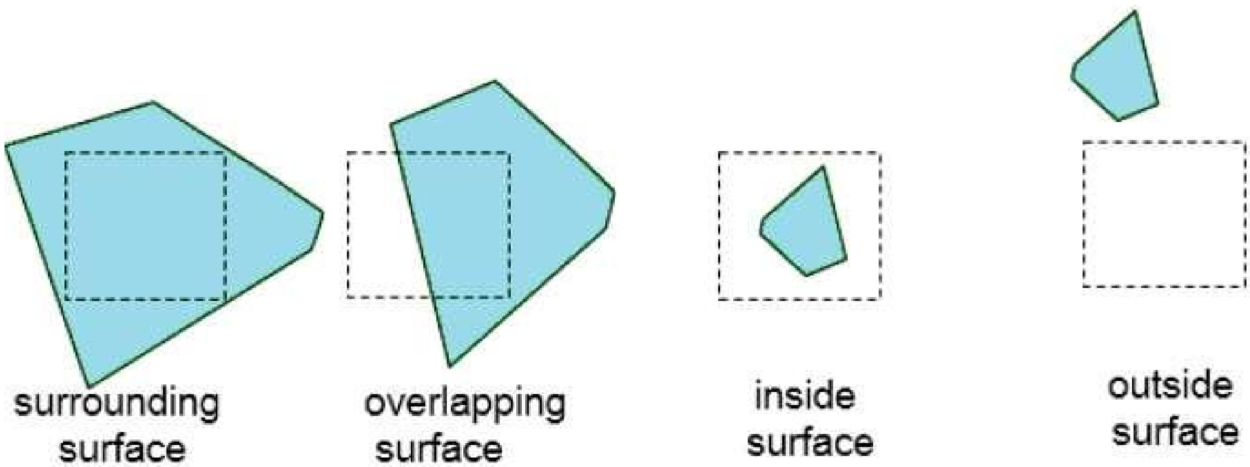
- It requires large memory.
- It is time consuming process

3. Area-Subdivision Method

The area-subdivision method takes advantage by locating those view areas that represent part of a single surface. Divide the total viewing area into smaller and smaller rectangles until each small area is the projection of part of a single visible surface or no surface at all.

Continue this process until the subdivisions are easily analyzed as belonging to a single surface or until they are reduced to the size of a single pixel. An easy way to do this is to successively divide the area into four equal parts at each step. There are four possible relationships that a surface can have with a specified area boundary.

- Surrounding surface − One that completely encloses the area.
- Overlapping surface − One that is partly inside and partly outside the area.
- Inside surface − One that is completely inside the area.
- Outside surface − One that is completely outside the area.



The tests for determining surface visibility within an area can be stated in terms of these four classifications. No further subdivisions of a specified area are needed if one of the following conditions is true −

- All surfaces are outside surfaces with respect to the area.
- Only one inside, overlapping or surrounding surface is in the area.
- A surrounding surface obscures all other surfaces within the area boundaries.