Module 6 - Visible Surface Detection and Animation

By Poonam M. Pangarkar
Department of CSE-DS
APSIT

Introduction

- In the real scene there may be many 3D objects placed randomly, completely or partially hiding some other objects.
- At any given viewing direction some of the surfaces may not be visible. identify only visible surfaces from the scene is a major concern of Computer Graphics.
- For displaying a realistic scene it is often necessary to render visible surfaces only.
- Many algorithms have been devised to find visible surfaces efficiently these algorithms are also known as hidden surface removal algorithm

Classification

Visible Surface Detection Algorithms can be classified into

1. **Object Space Algorithms**
2. **Image Space Algorithms**

Object Space Method

- Object space method determine the visible surface of an object by comparing the entire object or part of an object with other objects within the scene.
- These algorithms operate in physical coordinate system.
- The classification this class of algorithms is faster and performs less computation following methods belong to this category -
  - Back Face Detection Algorithm
  - Painter's Algorithm
  - Roberts Algorithm

Image Space Algorithms

- In image space method the visibility of object is decided by pixel by pixel comparison of overlapping objects.
- The pixel of the nearest object to the viewer is selected for display these algorithms operate in a screen coordinate system.
- Such algorithms are more precise but computationally very intense
- The following methods belong to this category
    - Depth buffer method
    - Area subdivision method
    - Octree method
    - Scan line method
    - Ray-tracing algorithm

Image Space Algorithms

- Most of the visible surface detection algorithms are image space algorithms, however, choice depends on the application generally object space algorithms are preferable
- Performance of algorithm can be improved with sorting and coherence property all objects in the scene are arranged from front to back in increasing order of their depth from the viewer.
- The object nearest to the viewers may hide the part of the scene so we do not have to render a portion of hidden objects behind the visible front object.
- Thus, coherence property is used to take advantage of regularity in the scene
- In the same way the individual scan line can also be assumed to have some interval either hidden or visible. Moreover, two consecutive scan lines also exhibit almost similar properties

Back Surface Detection Method (Object Space Method)

- In a solid object, there are surfaces which are facing the viewer (front faces) and there are surfaces which are opposite to the viewer (back faces).
- These back faces contribute to approximately half of the total number of surfaces. Since we cannot see these surfaces anyway, to save processing time, we can remove them before the clipping process with a simple test.
- Each surface has a normal vector. If this vector is pointing in the direction of the center of projection, it is a front face and can be seen by the viewer. If it is pointing away from the center of projection, it is a back face and cannot be seen by the viewer.
- The test is very simple, suppose the z axis is pointing towards the viewer, if the z component of the normal vector is negative, then, it is a back face.
- If the z component of the vector is positive, it is a front face. Note that this technique only caters well for non-overlapping convex polyhedra.

The polygon surface equation :

Ax + By + Cz + D < 0

While determining whether a surface is back-face or front face, also consider the viewing direction. The normal of the surface is given by :

N = (A, B, C)

A polygon is a back face if Vview.N > 0. But it should be kept in mind that after application of the viewing transformation, viewer is looking down the negative Z-axis. Therefore, a polygon is back face if :

(0, 0, -1).N > 0

or if C < 0

Viewer will also be unable to see surface with C = 0, therefore, identifying a polygon surface as a back face if : C <= 0.

Considering (a),

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

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

Hence, Back-face.

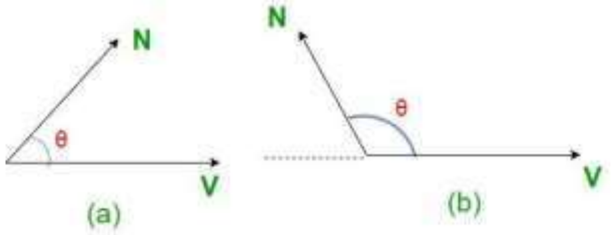
Considering (b),

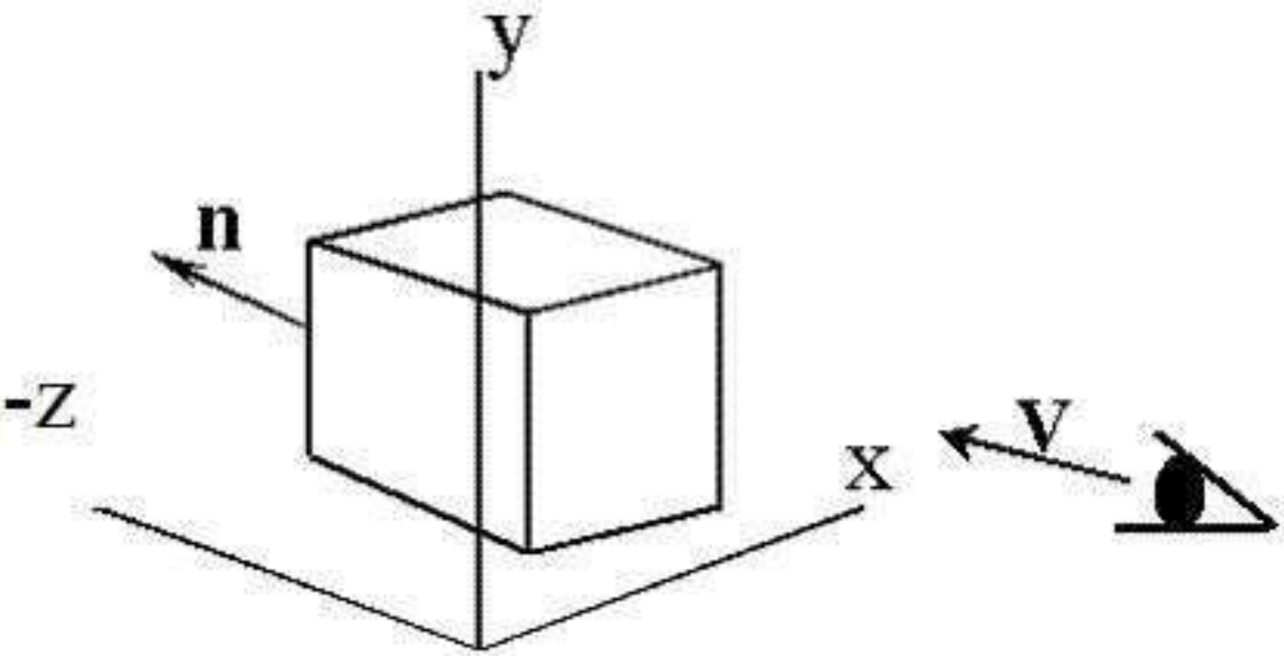
V.N = |V||N|Cos(angle)

if 90 < angle <= 180,

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

Hence, Front-face.

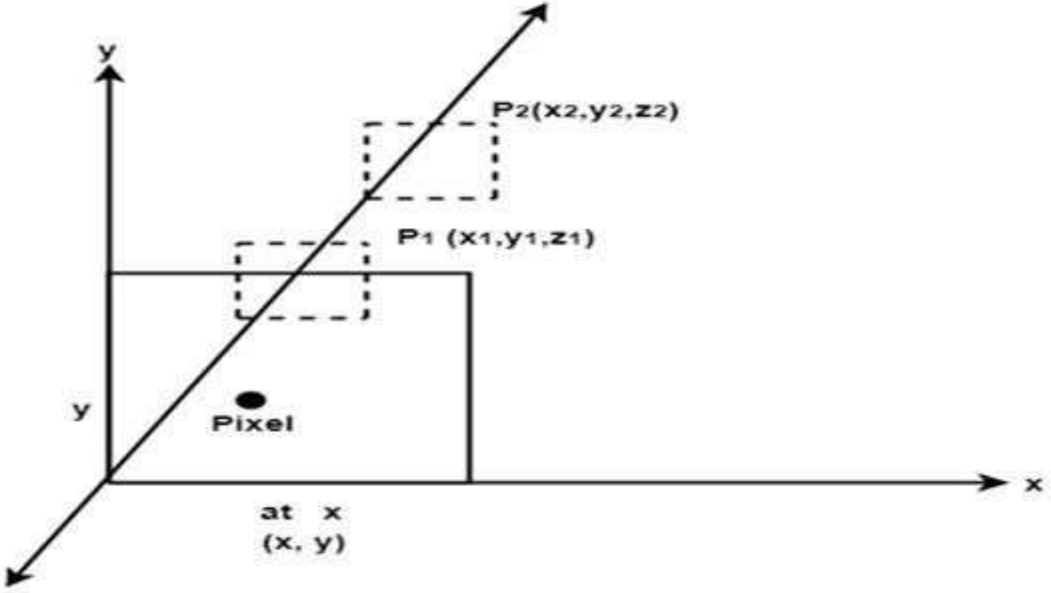



y
n
-z
x
v

Depth Buffer Method or z-buffer Method

- This approach compares surface depths at each pixel position on the projection plane.
- Object depth is usually measured from the view plane along the z axis of a viewing system.
- This method requires 2 buffers: one is the image buffer and the other is called the z-buffer (or the depth buffer).
- Each of these buffers has the same resolution as the image to be captured.
- As surfaces are processed, the image buffer is used to store the color values of each pixel position and the z-buffer is used to store the depth values for each (x,y) position.

Depth Buffer Method or z-buffer Method



Algorithm

1.For all pixels on the screen, set depth(z-buffer) [x, y] to 1.0 and intensity(Image buffer) [x, y] to a background value.

2.For each polygon in the scene, find all pixels (x, y) that lie within the boundaries of a polygon when projected onto the screen. For each of these pixels:

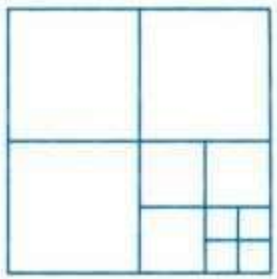
(a) Calculate the depth z of the polygon at (x, y)

(b)If z < depth [x, y], this polygon is closer to the observer than others already recorded for this pixel. In this case, set depth [x, y] to z and intensity [x, y] to a value corresponding to polygon's shading. If instead z > depth [x, y], the polygon already recorded at (x, y) lies closer to the observer than does this new polygon, and no action is taken.

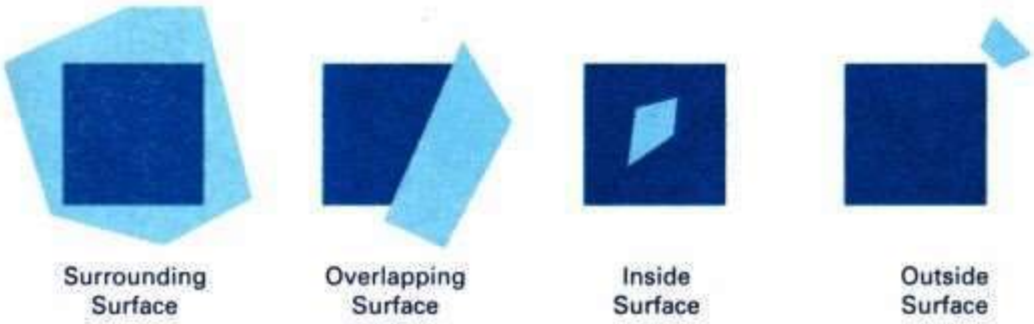
3. After all, polygons have been processed, each pixel of the image buffer represents the color of a visible surface at that pixel.

Area Subdivision Method

The area-subdivision method takes advantage of area coherence in a scene by locating those view areas that represent part of a single surface.
The total viewing area is successively divided into smaller and smaller rectangles until each small area is simple, ie. it is a single pixel, or is covered wholly by a part of a single visible surface or no surface at all.



Dividing a square area into equal-sized quadrants at each step.



Possible relationships between polygon surfaces and a rectangular area.

Area Subdivision Method

The procedure to determine whether we should subdivide an area into smaller rectangle is:

1. We first classify each of the surfaces, according to their relations with the area:

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

To improve the speed of classification, we can make use of the bounding rectangles of surfaces for early confirmation or rejection that the surfaces should be belong to that type.

2.Check the result from 1., that, if any of the following condition is true, then, no subdivision of this area is needed.

a. All surfaces are outside the area.

b. Only one surface is inside, overlapping or surrounding surface is in the area.

c. A surrounding surface obscures all other surfaces within the area boundaries.

For cases b and c, the color of the area can be determined from that single surface.

References

- Hearn & Baker, “Computer Graphics C version”, 2nd Edition, Pearson Publication