- idden Surface Removal Techniques.

Visible surfaces detection

(i) Why we need to detect such surfaces

→ In a complex screen shot (scene) there may be many overlapping objects or some objects are being fully or partially hidden by other objects.

Rendering all the objects in computer graphics takes considerable time and hence to reduce this time we need to detect the visible or hidden surfaces.

Broadly these techiques are classified as
(1) Object space (2) Image space.

→ An object space methods compare objects and parts of objects to each other within the scene definition to determine which surfaces, as a whole, should be labeled as visible or hidden.

for(each object in the screen shot (scene))

    for(each surface of an object)

    { determine those parts of the object whose view is unobstructed by other parts of its or any other object.

    }

computational effort n^2

where $n \Rightarrow$ no of objects in the scene.

Image space method

→ In such method the visibility is decided point by point at each pixel position on the projection plane.

for (each object in the scene)
for (each pixel in the image)

{
determine the object closest to the viewer that is intercepted by the projection plane through the pixel.

}

computational effort : np.

$n \Rightarrow$ no of objects

$p \Rightarrow$ no of pixels.

Object space method

BACK - FACE DETECTION

It is based on the inside-outside test.

Let the polygon surface is

$$Ax + By + Cz + D$$

A point (x', y', z') is inside a polygon surface

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

and outside the polygon surface

if $Ax' + By' + Cz' + D > 0$

→ We can simply test by considering the normal vector N to a polygon surface for detecting the back face.
In general, if V is a vector in the viewing direction from the eye position.
then this polygon is a back face if

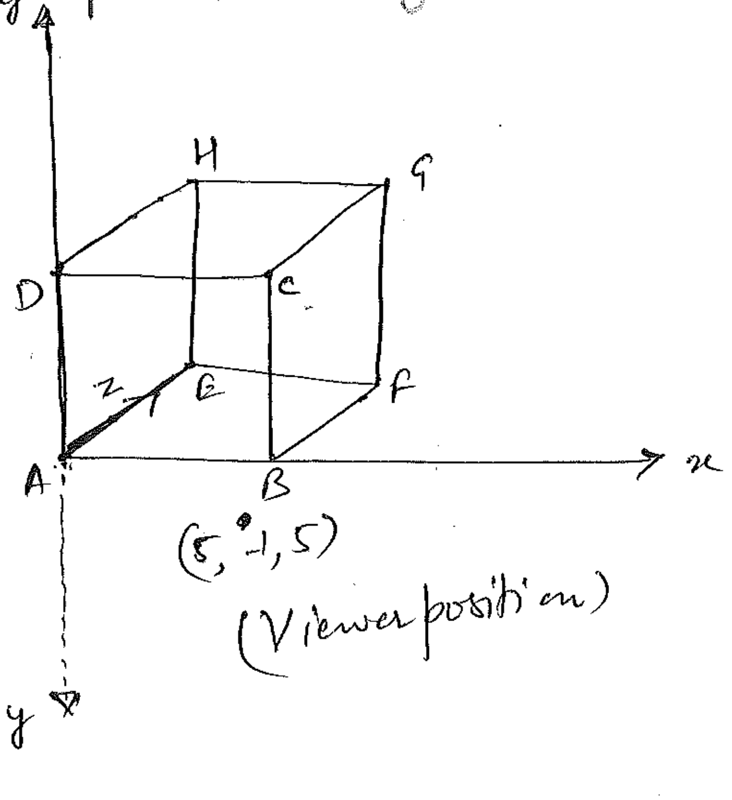
$$V \cdot N > 0$$

if viewing direction is parallel to the viewing axis, then $V_Z = (0, 0, V_z)$

and $$V \cdot N = V_z C.$$

So we need to consider the sign of C
(i) we can't see any face whose normal has z component $C = 0$.
(ii) The polygon is a back face if $C < 0$.

Q. Let the viewer is viewing from (5, -1, 5), using back face removal algorithm find the visible faces of the unit cube placed at origin.



View vector, if viewer is looking towards origin.

$V = (0,0,0) - (5,-1,5)$

$= (-5, 1, -5)$

Now using back face techique if $V \cdot N \geq 0$ then faces are back face otherwise visible.

Suface	Normal(N)	View Vector V	V·N
ABCD	(0, 0, -1)	(-5, 1, -5)	5 (+ve)
EFGH	(0, 0, 1)	(-5, 1, -5)	-5 (-ve)
ABFE	(0, -1, 0)	(-5, 1, -5)	-1 (-ve)
DCGH	(0, 1, 0)	(-5, 1, -5)	1 (+ve)
AEHD	(-1, 0, 0)	(-5, 1, -5)	5 (+ve)
BFGC	(1, 0, 0)	(-5, 1, -5)	-5 (-ve)

Hence the faces ABCD, DCGH, AEHD will be the back faces and EFGH, ABFE and BFGC are visible faces.

Q. Assume that viewer is observing the unit cube placed at origin from (20, 40, 20). Determine which faces are visible or hidden.

Viewing vector $V = (0,0,0) - (20, 40, 20)$

$= (-20, -40, -20)$

Sufaces		V	V·N
ABCD	(0, 0, -1)		
EFGH	(0, 0, 1)		
ABFE	(0, -1, 0)		
DCGH	(0, 1, 0)		
AEHD	(-1, 0, 0)		
BFGC	(1, 0, 0)		

Depth buffer (z-buffer)

Image space method

→ In this approach the surface depths are compared at each pixel position on the projection plane. The surface depth (z) is calculated from the view plane along the z-axis.

→ Each surface of a scene is processed separately, one point at a time across the surface.
The algorithm can be summerized as follows:

1. Initialize the depth buffer and refresh buffer so that for all buffer positions (x, y)

$$depth(x,y) = 0, \quad refresh(x,y) = I_{background}.$$

2. For each position on each polygon surface, compare depth values to previously stored values in the depth buffer to determine visibility.

if $z > depth(x,y)$, then set

$$depth(x,y) = z, \quad refresh(x,y) = I_{surf}(x,y)$$

depth values for a surface position (x, y) can be calculated from the plane equation for each surface

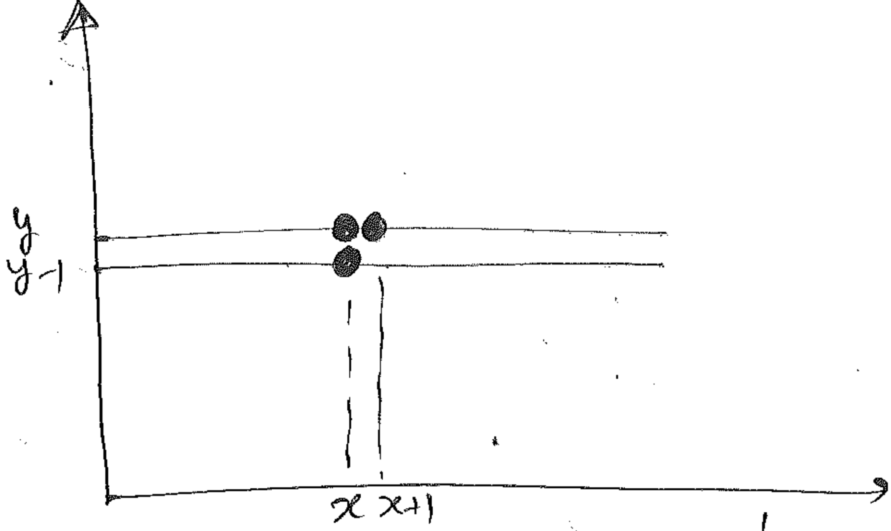
$$Ax + By + Cz + D = 0$$

$$\boxed{z = -\frac{(Ax + By + D)}{C}}$$

For any scan line, adjacent horizontal position $(x+1)$ the depth will be

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

or $\boxed{z' = z - \frac{A}{C}}$



To process down a vertical edge – $x' = x - 1/m$.

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

For a vertical edge the slope is infinite and hence

$$\boxed{z' = z + B/C}$$

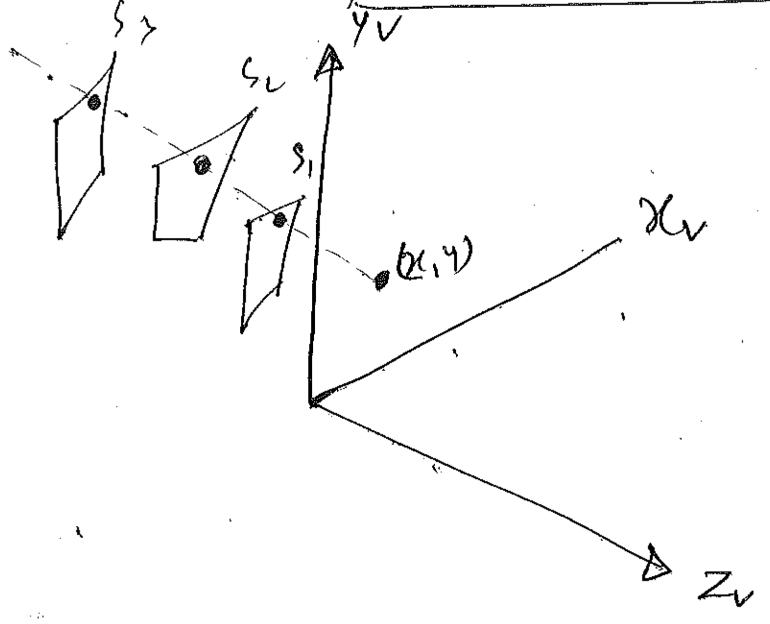


fig: At view plane position (x, y), surface S_1 has the smallest depth from the view plane and so is visible at that position.