Page No.
Date: / /

Z-buffer algorithm

1. Start.
2. Initialize the buffer values
   - Depthbuffer $(x,y) = 0$
   - Framebuffer $(x,y) =$ Ibackground.
3. process each polygon (one at a time)
   3.1. For each projected (x,y) pixel position of a polygon, calculate depth z.
      3.1.1. If $z >$ depthbuffer (x,y)
         - Compute surface color (i.e. Isurface(x,y)).
         - Set depthbuffer $(x,y) = z$,
         - framebuffer $(x,y) =$ Isurface(x,y).
4. Stop.

⇒ After all the surfaces are processed, the depth buffer contains the depth value of the visible surface and refresh buffer contains the corresponding intensity value for that surface.

⇒ The depth value of the surface position (x,y) is calculated by plane equation of surface.

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

Let, depth z' at position $(x+1, y)$

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

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

Here, $-A/C$ is constant for each surface, so corresponding depth value across a scan line are obtained from preceeding values by simple calculations

Advantages of Z-buffer algorithm.

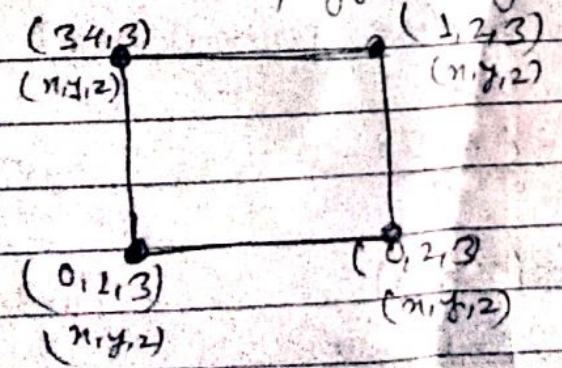
→ It is easy to implement.
→ It can be implemented in hardware to overcome the speed problem.
→ The Z-value of a polygon can be calculated incrementally.
→ No object-object comparison is required.

Dis-advantage.

→ This method requires additional buffer (if compared with depth sort method) and the overhead involves in updating the buffer.
→ It requires large memory.
→ It is time consuming process.

Example:

Assume the polygon is given as follows



In starting, assume that the depth of each pixel is infinite

∞	∞	∞
∞	∞	∞
∞	∞	∞
∞	∞	∞

the depth value at every place in the given polygon is 3, on applying the algorithm, the result is:

3	3	3
3	3	3
3	3	3
3	3	3

Now, let's change the Z-values. In the given figure, Z values goes from 0 to 3.

ie (3,4,3) (1,2,3)

(0,1,0) (0,2,0)

Now, the Z-value generated on the pixel will be

3	3	3
2	2	2
1	1	1
0	0	0