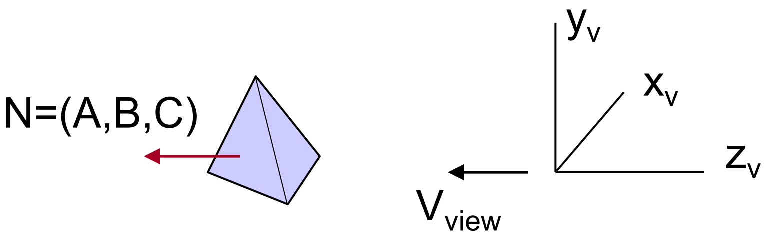
Back-Face Detection



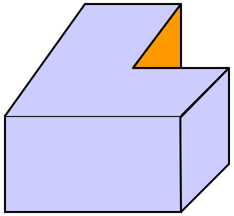
(x,y,z) is behind the polygon if

$Ax+By+Cz<0$

or

A polygon is a backface if

$V_{view} \cdot N > 0$

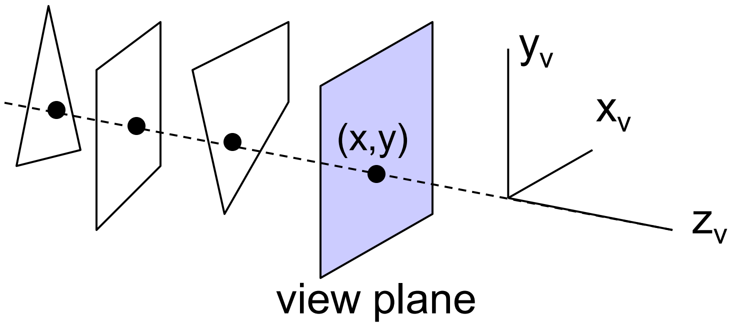


if V_{view} is parallel to z_v axis:

if C<0 then backface

if C=0 then polygon cannot be seen

Depth-Buffer Method (z-Buffer)



Object space method

Compares depths of the surfaces and uses the color of the closest one to the view plane

Depth buffer – depth values of surfaces for (x,y)

$0 \le depth \le 1$

Frame buffer (refresh buffer) – color value for (x,y)

Depth-Buffer Method (z-Buffer)

1. depthbuffer(x,y) = 1.0

   framebuffer(x,y) = background color

2. Process each polygon one at a time

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

   2.2. If z < depthbuffer(x,y)

   compute surface color,

   set depthbuffer(x,y) = z,

   framebuffer(x,y) = surfacecolor(x,y)

Depth-Buffer Method (z-Buffer)

Calculating Depth:

At (x,y): $Ax+By+Cz+D=0$

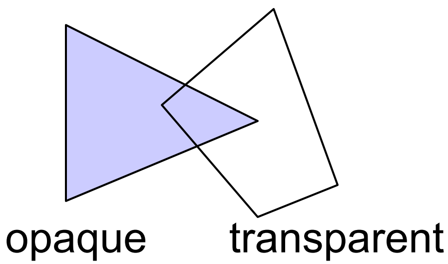
$z=(-Ax-By-D)/C$

For (x+1, y): $z' = z-(A/C)$

For $x'=x-1/m$, $y'=y-1$:

$z'= z+(A/m+B)/C$

A-Buffer Method



Linked list:

depth	surface info

Depth: a real number

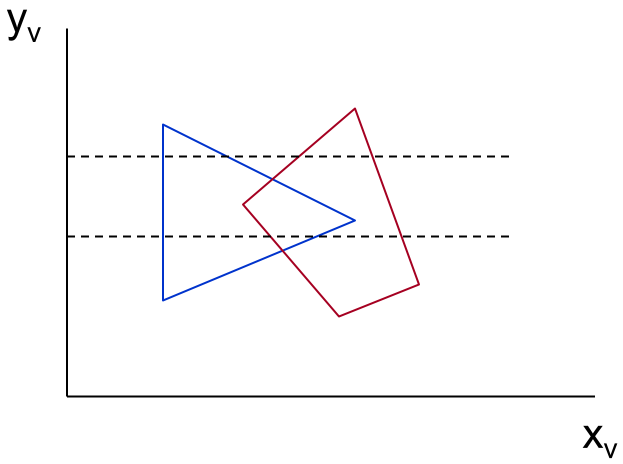
- ≥0 : single surface
- <0 : multiple surfaces

Surface info: surface data or pointer

Surface data:

- RGB intensity
- opacity
- depth
- percent of area coverage
- surface identifier
- etc.

Scan-Line Method



For each scan-line, examine all polygon surface projections intersecting that scan line to determine which are visible. Then enter the surface color of that position in the frame buffer.

Image space method

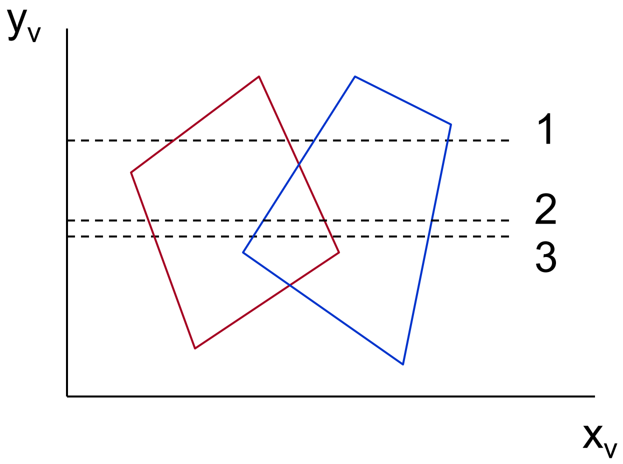
Edge table:

- coordinate endpoints of each line
- inverse slope of each line
- pointers to surface table

Surface table:

- plane coefficients (A,B,C)
- surface material properties
- pointers to edge table

Scan-Line Method



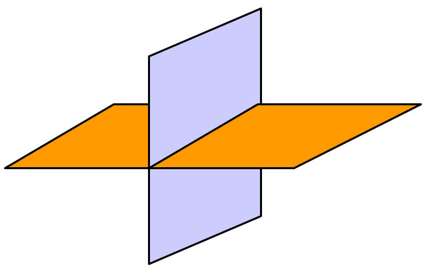
Algorithm:

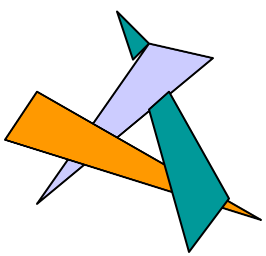
1. Form an active edge list that contains only the edges that cross the current scan line, sorted in order of increasing x.

2. Define a flag for each surface to indicate whether a position along a scan line is inside or outside the surface.

3. Process pixel positions across each scan line from left to right. Locate visible positions of surfaces along the scan line.

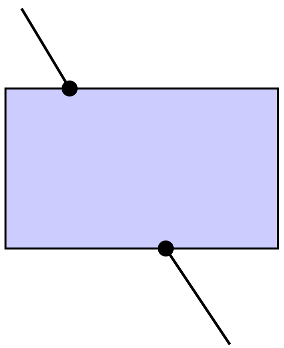
Scan-Line Method





Divide surfaces to eliminate the overlap.

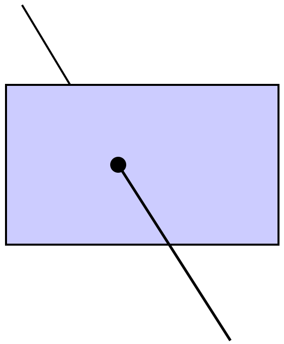
Visible Line Detection (Wireframe visibility)



If the projected edge endpoints of a line segment are both within the projected area of a surface, compare the depth of the endpoints to the surface depth at those (x,y) positions.

If both endpoints are behind the surface => hidden edge

If both endpoints are in front of the surface => visible edge



Otherwise calculate the intersections and the depth value of the intersection point.

- If for both intersection points, edge has greater depth than the surface => part of the edge is behind the surface
- If one edge endpoint has greater depth and the other has less depth than the surface => edge penetrates the surface

  Then, calculate the penetration point

Depth Cueing

$f_{depth}(d)$ is multiplied by each pixel's color

$$f_{depth}(d) = (d_{max}-d)/(d_{max}-d_{min})$$

d: distance of a point from the viewing position

d_{min}: 0.0

d_{max}: 1.0