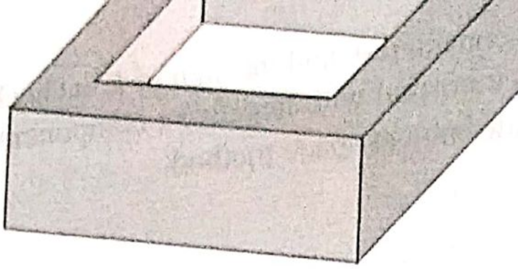
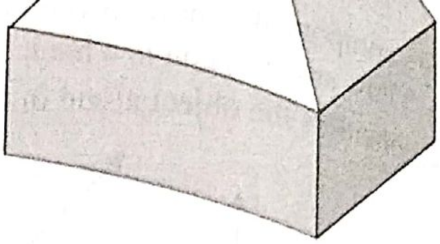


(b) A concave-faced object



(c) A convex-faced object

Fig. 10.5(b) and (c)

idden Surface removal / Visible Surface detection

10.2.2 Depth Buffer Method or Z–Buffer Algorithm

This method falls under the category of image space method, in which visibility is decided on point-to-poi basis for each pixel position on the projection plane. Along each projection line emanating perpendicu to the viewing plane, the depth value of the nearest surface to the viewing plane is checked and the colo and intensity value of that projected pixel is made similar to the closest detected surface, as shown Fig. 10.6. These depth values relative to the viewing plane (normally taken as the x–y plane: the screen a the viewing plane is the viewer/camera plane considered parallel to the computer screen) can be calcula using the plane equation, which is determined using any three points on the plane as done using Cramer's ru in the previous section in the form $Ax + By + Cz + D = 0$.

Hence the depth $$z_{x,y} = \frac{-Ax - By - D}{C}$$

The algorithm is as follows:

Step 1: Initialize the depth array for all pixel positions for the given screen resolution by

depth $(x, y) = 0$, and colour_intensity(x, y) = background colour.

Step 2: For each x ranging from 0 to $x$max_resolution and

For each y ranging from 0 to $y$max_resolution

Calculate the depth z for each polygon and compare

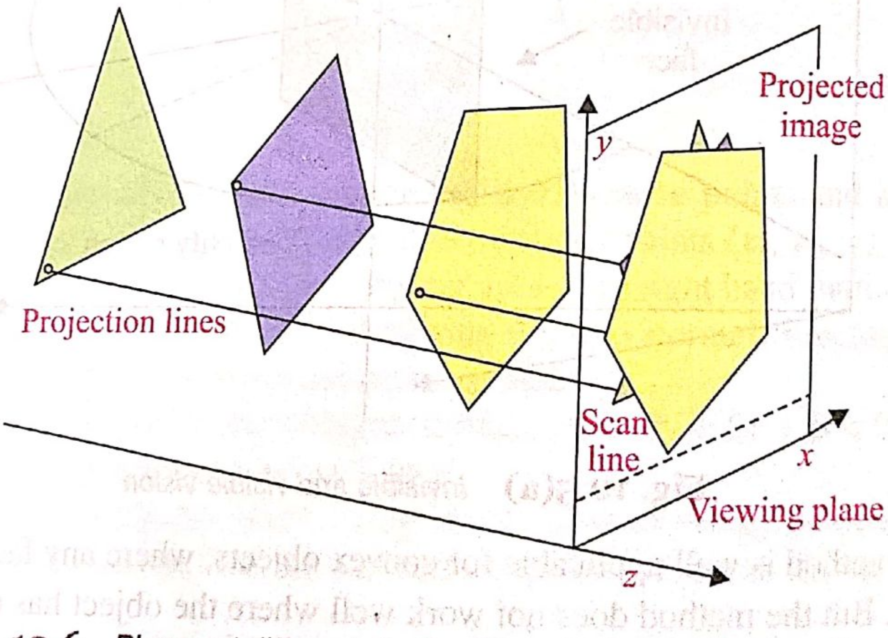


Fig. 10.6 *Planes at different depth with projection on viewing plane*

Scanned by CamScanner

If z > depth (x, y), then depth (x, y) = z and colour_intensity(x, y) = colour_intensity of that surface. For the colour to be replaced, the pixel position in consideration should have a projection line emanating from the selected polygon, that is, the projected point should lie within the projected polygon; if not the background colour will be maintained. For this an inside polygon test is performed which is as follows.

Once all the pixels have been processed, the depth array contains the nearest polygon depth value and the colour_intensity(x, y) array contains the respective colour/intensity value.

For each horizontal scan line on the screen the successive depth value can be calculated as

$$z_{x+1,\,y} = \frac{-A(x+1) - By - D}{C} = \frac{-Ax - By - A}{C} - \frac{A}{C}$$

or

$$z_{x+1,y} = z_{x,\,y} - \frac{A}{C}$$

The scan lines can also be chosen to be vertical ones and move horizontally, and $z_{x,\,y+1}$ can also be successively calculated in a similar manner for each x.

Inside Polygon (Subdivided to Triangle) Test

Triangle here means triangle's projection to $z = 0$ plane, that is, the 2D screen. Any pixel which lies inside the triangle is on the same side as that of the vertex which is opposite the edge line. This is true for all the edges as the triangle is always a convex polygon. Knowing the three vertices of the triangle, the equation of the line passing through any two vertices, that is, the edges, can be known by the common equation of a line passing through two points.

$$\frac{x - x_1}{x_1 - x_2} = \frac{y - y_1}{y_1 - y_2}$$

or,
$$(x - x_1)(y_1 - y_2) = (y - y_1)(x_1 - x_2)$$

or,
$$x(y_1 - y_2) - y(x_1 - x_2) - x_1(y_1 - y_2) + y_1(x_1 - x_2) = 0$$

which is in the form $Ax + By + C = 0$.

The sign of $f(x, y) = Ax + By + C$ is the same for any inside point and the vertex opposite to the edge. If this is true for all the edges of the triangle, the pixel lies inside the triangle.

10.2.3 A–Buffer Method

... drawback in that it can only find visible surfaces ... buffer method; it can accumulate ...

Solved Exercises

10.1 Find the equation of a plane passing through the points (2, 4, 3), (4, 4, 5) and (8, 9, 3).

Solution

Let the general equation through the points be $Ax + By + Cz + D = 0$. This can be expressed as $ax + by + cz + 1 = 0$ by dividing the general equation by D.

Each point must satisfy this equation

Hence,

$$2a + 4b + 3c = -1$$

$$4a + 4b + 5c = -1$$

$$8a + 9b + 3c = -1$$

In matrix form, it can be written as

$$\begin{bmatrix} 2 & 4 & 3 \\ 4 & 4 & 5 \\ 8 & 9 & 3 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \times \begin{bmatrix} 2 & 4 & 8 \\ 4 & 4 & 9 \\ 3 & 5 & 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Hence, $$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2 & 4 & 3 \\ 4 & 4 & 5 \\ 8 & 9 & 3 \end{bmatrix}^{-1}\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.5690 & 0.2586 & 0.1379 \\ 0.4828 & -0.3103 & 0.0345 \\ 0.0690 & 0.2414 & -0.1379 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

or,
$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.1724 \\ -0.2069 \\ -0.1724 \end{bmatrix}$$

Hence the equation of the plane is $0.1724x - 0.2069y - 0.1724z + 1 = 0$

Alternate Method:

Using method discussed in 10.2.1.

$$\begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} x + \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix} y + \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0$$

$$\begin{vmatrix} 0 & -2 \\ -5 & 2 \end{vmatrix} x + \begin{vmatrix} -2 & -2 \\ 2 & -4 \end{vmatrix} y + \begin{vmatrix} -2 & 0 \\ -4 & -5 \end{vmatrix} z - \begin{vmatrix} 2 & 4 & 3 \\ 4 & 4 & 5 \\ 8 & 9 & 3 \end{vmatrix} = 0$$

$$-10x + 12y + 10z - 58 = 0$$

or,
$$0.1724x - 0.2069y - 0.1724z + 1 = 0$$

or,

which is same as calculated above.

-4.

10.2 Calculate the mean unit surface normal to a polygon described by four position vectors **A**(2, 0, 0), **B**(0, 2, 0), **C**(0, 0, 2) and **D**(2.2, – 2, 2).

Solution

Calculating edges of the polygon **ABCD**

$$\mathbf{AB} = (0-2)\mathbf{i} + (2-0)\mathbf{j} + (0-0)\mathbf{k} = -2\mathbf{i} + 2\mathbf{j} + 0\mathbf{k}.$$

$$\mathbf{BC} = (0-0)\mathbf{i} + (0-2)\mathbf{j} + (2-0)\mathbf{k} = 0\mathbf{i} - 2\mathbf{j} + 2\mathbf{k}.$$

$$\mathbf{CD} = (2.2-0)\mathbf{i} + (-2-0)\mathbf{j} + (2-2)\mathbf{k} = 2.2\mathbf{i} - 2\mathbf{j} + 0\mathbf{k}.$$

$$\mathbf{AD} = (2.2-2)\mathbf{i} + (-2-0)\mathbf{j} + (2-0)\mathbf{k} = 0.2\mathbf{i} - 2\mathbf{j} + 2\mathbf{k}.$$

Calculating normals at the vertices **A**, **B**, **C** and **D**

$$\mathbf{n}_A = \mathbf{AB} \times \mathbf{AD} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -2 & 2 & 0 \\ 0.2 & -2 & 2 \end{vmatrix} = 4\mathbf{i} + 4\mathbf{j} + 3.6\mathbf{k}$$

$$\mathbf{n}_B = \mathbf{BC} \times \mathbf{BA} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & -2 & 2 \\ 2 & -2 & 0 \end{vmatrix} = 4\mathbf{i} + 4\mathbf{j} + 4\mathbf{k}$$

$$\mathbf{n}_C = \mathbf{CD} \times \mathbf{CB} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 2.2 & -2 & 0 \\ 0 & 2 & -2 \end{vmatrix} = 4\mathbf{i} + 4.4\mathbf{j} + 4.4\mathbf{k}$$

$$\mathbf{n}_D = \mathbf{DA} \times \mathbf{DC} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -0.2 & 2 & -2 \\ -2.2 & 2 & 0 \end{vmatrix} = 4\mathbf{i} + 4.4\mathbf{j} + 4\mathbf{k}$$

Taking mean of all surface normals

$$\mathbf{n}_{\text{mean}} = 4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}$$

Hence unit surface normal is $= \dfrac{4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}}{|4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}|} = \dfrac{4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}}{\sqrt{4^2 + 4.2^2 + 4^2}}$

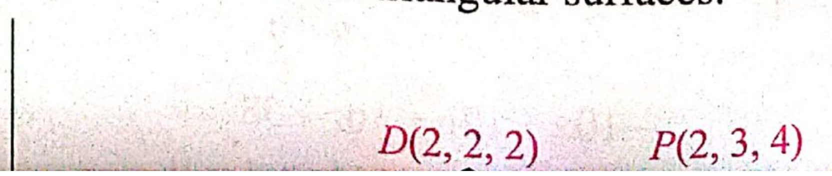
$$\hat{\mathbf{n}}_{\text{mean}} = \frac{4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}}{49.64} = 0.0806\mathbf{i} + 0.0846\mathbf{j} + 0.0806\mathbf{k}$$

Note: A plane can be described by a surface normal and a point on it.

·3 A solid tetrahedron is given by position vectors **A**(1, 1, 1), **B**(3, 1, 1), **C**(2, 1, 3) and **D**(2, 2, 2), a point light source is kept at **P**(2, 3, 4). Using back face detection method, find the surfaces on w the light falls and the surfaces which are to be shadowed.

ıtion

This is a convex object consisting of four flat triangular surfaces.



$\ldots - 0 + 8 + 0 = 8 > 0$. Hence, shadowed.

Note: If the surface normals are in the plane of the object, the surface is to be shadowed if the dot product is negative.

If the camera is placed at the position of light source the surface which are illuminated will be visible and the shadowed surfaces will be invisible to the camera.

The above method is suitable for nonintersecting polygons.

A tetrahedron is given by position vectors A(1, 1, − 1), B(3, 1, − 1), C(2, 1, − 3) and D(2, 2, − 2). Use Depth buffer method (or Z–Buffer method) to find the visible planes of the tetrahedron if the viewing plane is xy − plane i.e. $z = 0$. Take screen resolution of 4×4, and background colour as black (colour value = 0). The colour of plane ACD is blue (1), CBD is green (2), BAD is cyan (3) and ACB is red (4).

Solution

Finding equation of the planes using method discussed in Section 10.2.1

$$\begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} x + \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix} y + \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0$$

Inside polygon test is to be performed as discussed in Section 10.2.2.

For plane ACD:

The plane equation is

$$2x - y + z - 0 = 0$$

$$Z_{ACD} = -2x + y$$

Hence,

Projected triangle vertices are **A**(1, 1), **C**(2, 1) and **D**(2, 2)

Equation of the projected line **AC** is $y - 1 = \dfrac{1-1}{1-2}(x-1)$

or,

$$y = 1$$

or, $f(x, y) = y - 1$, and the checking point for inside polygon test is **D**(2, 2)

$$f(x, y) \text{ at } \mathbf{D} = 2 - 1 = 1 > 0$$

Equation of the projected line **AD** is $y = x$

or, $f(x, y) = x - y$, and the checking point for inside polygon test is **C**(2, 1)

$$f(x, y) \text{ at } \mathbf{C} = 2 - 1 = 1 > 0$$

Equation of the projected line **CD** is $(y - 1)(2 - 2) = (x - 2)(1 - 2)$

or

$$x = 2$$

or, $f(x, y) = 2 - x$, and the checking point for inside polygon test is **A**(1, 1)

$$f(x, y) \text{ at } \mathbf{A} = 2 - 1 = 1 > 0$$

For plane CBD:

The plane equation is

$$-2x - y + z - (-8) = 0$$

or,

$$Z_{CBD} = 2x + y - 8.$$

Projected triangle vertices are **C**(2, 1), **B**(3, 1) and **D**(2, 2)

Equation of the projected line **CB** is $y - 1 = \dfrac{1-1}{3-2}(x-3)$

or,

$$y = 1$$

or, $f(x, y) = y - 1$, and the checking point for inside polygon test is **D**(2, 2)

$$f(x, y) \text{ at } \mathbf{D} = 2 - 1 = 1 > 0$$

Equation of the line **CD** is $x = 2$

or, $f(x, y) = x - 2$, and the check point for inside polygon test is **B**(3, 1)

$$f(x, y) \text{ at } \mathbf{B} = 3 - 2 = 1 > 0$$

Equation of the projected line **BD** is $y - 1 = \frac{1-2}{3-2}(x - 3)$

$$y - 1 = 3 - x,$$

or,

or, $f(x, y) = 4 - x - y$, and the check point for inside polygon test is **C**(2, 1)

$$f(x, y) \text{ at } \mathbf{C} = 4 - 2 - 1 = 1 > 0$$

For plane BAD:

The plane equation is

$$0x - 2y - 2z - 0 = 0$$

$$Z_{BAD} = -y$$

or,

The projected triangle vertices are **B**(3, 1), **A**(1, 1) and **D**(2, 2)

Equation of the projected line **BA** is $y - 1 = \frac{1-1}{1-3}(x - 1)$

$$y = 1$$

or,

or, $f(x, y) = y - 1$, and the checking point for inside polygon test is **D**(2, 2)

$$f(x, y) \text{ at } \mathbf{D} = 2 - 1 = 1 > 0$$

Equation of the projected line **AD**, $x = y$

or, $f(x, y) = x - y$, and the check point for inside polygon test is **B**(3, 1)

$$f(x, y) \text{ at } \mathbf{B} = 3 - 1 = 2 > 0$$

Equation for the line **BD** is $y - 1 = \frac{1-2}{3-2}(x - 3)$

$$y - 1 = 3 - x$$

or,

or, $f(x, y) = 4 - x - y$, and the check point for inside polygon test is **A**(1, 1)

$$f(x, y) \text{ at } \mathbf{A} = 4 - 1 - 1 = 2 > 0$$

Any point inside any triangle should make all the three functions $f(x, y)$ for any triangle > 0.

For plane ACB:

$$0x - 4y + 0z - (-4) = 0$$

$$y = 1.$$

or,

As it is independent of z and x, the plane is perpendicular to xy plane. Thus, the projection of the plane will be a thin line ranging from (min. x, min. y) to (max. x, max. y). This plane will not take part in algorithm.

This type of polygons (plane parallel to xz or yz plane) should be compared by maximum and minimum values of z against the depths of other polygons.

As the parallel projected pixels vary from $x = 1$ to $x = 3$ and $y = 1$ to 2, it is required to check within that range only, other part of the screen remains with its initial background colour.