```
        }
    }
    if (t == (float)x[npts+order])          //pick up last point
        temp[npts] = 1;
//Calculate sum for denominator of rational blend functions
    sum = 0;
    for (i = 1; i <= npts; i++)
            sum = sum + temp[i]*w[i];
//Form rational blend functions and put in array r
    for (i = 1; i <= npts; i++)
    {
        if (sum != 0)
            r[i] = (temp[i]*w[i])/(sum);
        else
            r[i] = 0;
    }
}
```

CHAPTER 8

Solved Exercises

8.1 Determine eleven points on a Bezier curve with equidistant parametric values, having control points $(x_0, y_0) = (50, 180)$, $(x_1, y_1) = (250, 100)$, $(x_2, y_2) = (600, 300)$ and $(x_3, y_3) = (500, 50)$, distributed over a screen of resolution 640 × 350.

Solution

Standard Bezier curve is given by $\mathbf{P}(u) = [x(u), y(u)] = \sum_{k=0}^{n} \frac{n!}{(n-k)!\,k!} u^k (1-u)^{n-k} \mathbf{p}_k \; 0 \le u \le 1$

Here $n = 3$, since there are four vertices

Hence,

$$x(u) = \sum_{k=0}^{3} \frac{3!}{(3-k)!\,k!} u^k (1-u)^k x_k$$

$$= (1)u^0(1-u)^3 x_0 + 3u(1-u)^2 x_1 + 3u^2(1-u)x_2 + (1)\,u^3 x_3$$

or, $$x(u) = 50(1-u)^3 + 750u(1-u)^2 + 1800u^2(1-u) + 500u^3$$

And,

$$y(u) = \sum_{k=0}^{3} \frac{3!}{(3-k)!\,k!} u^k (1-u)^k y_k$$

$$= (1)u^0(1-u)^3 y_0 + 3u(1-u)^2 y_1 + 3u^2(1-u)y_2 + (1)u^3 y_3$$

or, $$y(u) = 180(1-u)^3 + 300u(1-u)^2 + 900u^2(1-u) + 50u^3$$

For eleven points on the curve, u must be varied form 0 to 10 stepping through steps of 0.1, and the corresponding points can be calculated from the above equations.

Point	u	$[x(u), y(u)]$
1	0	(50, 180)
2	0.1	(113.9, 163.67)
3	0.2	(183.2, 159.76)
4	0.3	(254.3, 163.89)
5	0.4	(323.6, 171.68)
6	0.5	(387.5, 178.75)
7	0.6	(442.4, 180.72)
8	0.7	(484.7, 173.21)
9	0.8	(510.8, 151.84)
10	0.9	(517.1, 112.23)
11	1.0	(500, 50)

8.2 Find the equation of the Bezier curve which passes through (0, 0) and (– 4, 2) and controlled through (14, 10) and (4, 0).

Solution

As the curve passes through (0, 0) and (– 4, 2), these two points are considered as end points $\mathbf{P}_0$ and $\mathbf{P}_3$; while the curve is controlled through (14, 10) and (4, 0), they are considered as intermediate points $\mathbf{P}_2$ and $\mathbf{P}_3$, respectively.

As per the equation $\mathbf{P}(u) = [1 \;\; u \;\; u^2 \;\; u^3]\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$, the corresponding Bezier curve is given by

$$\mathbf{P}(u) = [x(u) \;\; y(u)] = [1 \;\; u \;\; u^2 \;\; u^3]\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 14 & 10 \\ 4 & 0 \\ -4 & 2 \end{bmatrix} \qquad 0 \le u \le 1$$

$$= [1 \;\; u \;\; u^2 \;\; u^3]\begin{bmatrix} 0 & 0 \\ 42 & 30 \\ -72 & -60 \\ 26 & 32 \end{bmatrix}$$

or, $$\mathbf{P}(u) = [(42u - 72u^2 + 26u^3) \;\; (30u - 60u^2 + 32u^3)]$$

8.3 Four control points $\mathbf{P_0}$ (a, b), $\mathbf{P_1}$ (2, 5), $\mathbf{P_2}$ (4, 4) and $\mathbf{P_3}$ (7, c) are available for drawing a uniform quadratic B-spline curve segment. Compute the values of a, b and c such that the curve starts from the point (1, 4) and terminates with slope – 0.5.

Solution

Using equation $\mathbf{P}(u) = \sum_{k=0}^{M} \mathbf{p}_k B_{k,n}(u) = \sum_{k=0}^{3} \mathbf{p}_k B_{k,3}(u)$

The number of points are 4, hence $M + 1 = 4$, which implies $M = 3$

As the curve is required to be quadratic i.e. $n - 1 = 2$, or $n = 3$.

The number of knot values in uniform knot vector will be $M + n + 1 = 3 + 3 + 1 = 7$

The uniform knot vector will be {0, 1, 2, 3, 4, 5, 6}

Using recursive relation $B_{k,3}(u) = \dfrac{u - u_k}{u_{k+n-1} - u_k} B_{k,2}(u) + \dfrac{u_{k+n} - u}{u_{k+n} - u_{k+1}} B_{k+1,2}(u)$, each of the four blending functions of spans $n = 3$ subintervals of total range of u are:

$$B_{0,3} = \begin{cases} \frac{1}{2}u^2 & 0 \le u < 1 \\ \frac{1}{2}u(2-u) + \frac{1}{2}(u-1)(3-u) & 1 \le u < 2 \\ \frac{1}{2}(3-u)^2 & 2 \le u < 3 \end{cases}$$

$$B_{1,3} = \begin{cases} \frac{1}{2}(u-1)^2 & 1 \le u < 2 \\ \frac{1}{2}(u-1)(3-u) + \frac{1}{2}(u-2)(4-u) & 2 \le u < 3 \\ \frac{1}{2}(4-u)^2 & 3 \le u < 4 \end{cases}$$

$$B_{2,3} = \begin{cases} \frac{1}{2}(u-2)^2 & 2 \le u < 3 \\ \frac{1}{2}(u-2)(4-u) + \frac{1}{2}(u-3)(5-u) & 3 \le u < 4 \\ \frac{1}{2}(5-u)^2 & 4 \le u < 5 \end{cases}$$

$$B_{3,3} = \begin{cases} \frac{1}{2}(u-3)^2 & 3 \le u < 4 \\ \frac{1}{2}(u-3)(5-u) + \frac{1}{2}(u-4)(6-u) & 4 \le u < 5 \\ \frac{1}{2}(6-u)^2 & 5 \le u < 6 \end{cases}$$

$$\mathbf{P}(u) = B_{0,3}(u)\,\mathbf{P_0} + B_{1,3}(u)\,\mathbf{P_1} + B_{2,3}(u)\,\mathbf{P_2} + B_{3,3}(u)\,\mathbf{P_3}$$

Note that all blending functions below 2 and above 4 are not present and, within this interval, Eq. 8.22 is valid. The sum of all blending functions within this range is 1. As the blending functions outside this range are not present, the sum of blending functions cannot be computed. Thus the starting and ending points will be defined by $u_k = 2$ and $u_k = 4$.

Hence $\mathbf{P}(u) = \frac{1}{2}(3 - u)^2\ \mathbf{P}_0 + \left[\frac{1}{2}(u-1)(3-u)+\frac{1}{2}(u-2)(4-u)\right]\mathbf{P}_1 + \frac{1}{2}(u - 2)^2\ \mathbf{P}_2$ for $2 \le u < 3$

And, $\mathbf{P}(u) = \frac{1}{2}(4 - u)^2\mathbf{P}_1 + \left[\frac{1}{2}(u-2)(4-u)+\frac{1}{2}(u-3)(5-u)\right]\mathbf{P}_2 + \frac{1}{2}(u - 3)^2\ \mathbf{P}_3$ for $3 \le u < 4$

Hence the start point $\mathbf{P}_{start} = ½\ (\mathbf{P}_0 + \mathbf{P}_1)$ and the end point will be $\mathbf{P}_{end} = ½\ (\mathbf{P}_2 + \mathbf{P}_3)$

Also, the derivatives can be obtained by differentiating the above two equations with respect to u and substituting $u = 2$ and 4 for derivatives at the start and end points, respectively.

Thus, $\mathbf{P}'_{start} = \mathbf{P}_1 - \mathbf{P}_0$ and for the end point $\mathbf{P}'_{end} = \mathbf{P}_3 - \mathbf{P}_2$.

The given points are $\mathbf{P}_0$(a, b), $\mathbf{P}_1$(2, 5), $\mathbf{P}_2$(4, 4) and $\mathbf{P}_3$(7, c), start point is $\mathbf{P}_{start}$(1, 4) and slope at the end point is $\mathbf{P}'_{end} = -0.5$.

Thus, by substitution of these values, we get

$$(1, 4) = ½\ [(a, b) + (2, 5)] \text{ and } -½ = [(7, c) - (4, 4)]$$

or, $1 = ½\ (a + 2),\ 4 = ½\ (b + 5)$

or, $a = 0,\ b = 3$

And $-½ = [(7 - 4)\ \ (c - 4)]$

or, $(c - 4)/3 = -½$

or, $c = 2.5$

Hence a = 0, b = 3 and c = 2.5.

8.4 A cubic Bezier curve segment is described by control points $\mathbf{P}_0$ (2, 2), $\mathbf{P}_1$(4, 8), $\mathbf{P}_2$(8, 8) and $\mathbf{P}_3$(9, 5). Another curve segment is described by $\mathbf{Q}_0$ (a, b), $\mathbf{Q}_1$(c, 2), $\mathbf{Q}_2$(15, 2) and $\mathbf{Q}_3$(18, 2). Determine the values of a, b and c so that the two curve segments join smoothly.

Solution

For C^0 continuity at the point of contact, $\mathbf{P}_3$(9, 5) and $\mathbf{Q}_0$ (a, b) must be the same.

$$\mathbf{P}_3(9, 5) = \mathbf{Q}_0\ (a, b)$$

Hence, a = 9 and b = 5

As the curves join smoothly at the point of coincident, C^1, continuity at this point should be the same.

Thus, $\mathbf{P}'_3 = \mathbf{Q}'_0$, i.e. the tangent vector at the end point of the first curve should be identical to the tangent vector at the starting point of the second curve.

From equation

$$3(\mathbf{P}_3 - \mathbf{P}_2) = 3(\mathbf{Q}_1 - \mathbf{Q}_0) \qquad \text{(refer Eqs 8.10 and 8.11)}$$

or, $\mathbf{P}_3 - \mathbf{P}_2 = \mathbf{Q}_1 - \mathbf{Q}_0$

or, $\{[9\ \ 5] - [8\ \ 8]\} = \{[c\ \ 2] - [a\ \ b]\}$

or, $[c\ \ 2] = 2[9\ \ 5] - [8\ \ 8] = [10\ \ 2]$

or, $c = 2.$

Thus, a = 9, b = 5 and c = 2.

CHAPTER 8

8.5 Design a piecewise quadratic polynomial f (u) using $a(u) = (1/2)\,u^2$, $b(u) = 3/4 - (u - 3/2)^2$ and $c(u) = 1/2\,(3 - u)^2$ sequentially, having support of [0 3]. Find the span of curves within the support range and the order of continuity which exists at the existing knots. Draw the curve on the graph.

Solution

Finding the intersection points

Solving $a(u)$ and $b(u)$

$$\frac{1}{2}u^2 = \frac{3}{4} - \left(u - \frac{3}{2}\right)^2$$

or, $$2u^2 = 3 - (2u - 3)^2$$

or, $$2u^2 = 3 - (4u^2 - 12u + 9)$$

or, $$u^2 = 2u - 1$$

or, $$(u - 1)^2 = 0$$

or, $$u = 1$$

Thus, the curves $a(u)$ and $b(u)$ meet at the single intersection point $u = 1$, with $a(1) = b(1) = ½$.

Solving $b(u)$ and $c(u)$

$$\frac{3}{4} - \left(u - \frac{3}{2}\right)^2 = \frac{1}{2}(3 - u)^2$$

or, $$3 - (2u - 3)^2 = 2(3 - u)^2$$

or, $$3 - (4u^2 - 12u + 9) = 2(9 - 6u + u^2)$$

or, $$u^2 - 4u + 4 = 0$$

or, $$(u - 2)^2 = 0$$

or $$u = 2$$

Thus, the curves $b(u)$ and $c(u)$ meet at the single intersection point $u = 2$, with $b(1) = c(1) = ½$.

The curves have C^0 continuity everywhere on the curve and also at the point of contact as the curves meet $u = 1$ and $u = 2$.

Now, finding C^1 continuity at the knots $u = 1$ for $a(u)$ and $b(u)$ and at $u = 2$ for $b(u)$ and $c(u)$.

$$a'(u) = u \text{ and } b'(u) = -2\left(u - \frac{3}{2}\right)$$

or, $$a'(1) = b'(1) = 1$$

Similarly, $$b'(u) = -2\left(u - \frac{3}{2}\right) \text{ and } c'(u) = (3 - u) \text{ and}$$

or, $$b'(2) = c'(2) = -1$$

Again, since the curves are quadratic polynomial, the first order derivative exists at every point of the curve within the segmented spans [0 1], [1 2] and [2 3]. Also, first order derivative at points of contact is the same for a pair of joining curves.

Hence, the curves have C^1 continuity everywhere on the curve.

Now, finding C^2 continuity at the points of coincidence.

$$a''(u) = 0,\ b''(u) = -2 \text{ and } c''(u) = -1$$

Thus, it is observed that the curves do not have second order continuity at the point of coincidence, and higher orders are not possible with quadratic curves.

CHAPTER 8

Thus, the curve a(u), b(u) and c(u) can be defined over spans [0 1], [1 2] and [2 3], respectively, with C^0 and C^1 continuity. The curve can be plotted as below:

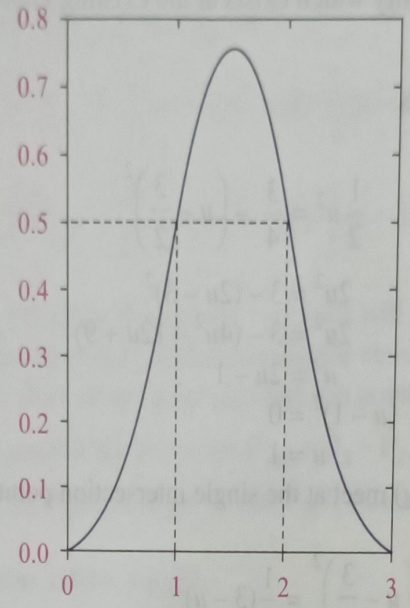


Fig. 8.32

Review Questions

8.1 Find the points on a Bezier curve which has starting and ending points $\mathbf{P}_0(2, 3)$ and $\mathbf{P}_3(4, -3)$, respectively, and is controlled by $\mathbf{P}_1(6, 6)$ and $\mathbf{P}_2(8, 1)$, for $u = 0.2$, 0.5 and 0.9.

8.2 A cubic Bezier curve segment is described by control points $\mathbf{P}_0$ (1, 2), $\mathbf{P}_1(2, 7)$, $\mathbf{P}_2(10, 9)$ and $\mathbf{P}_3(9, b)$. Another curve segment is described by $\mathbf{Q}_0$ (a, 6), $\mathbf{Q}_1(8, c)$, $\mathbf{Q}_2(13, 2)$ and $\mathbf{Q}_3(15, 2)$. Determine the values of a, b and c so that the two curve segments join smoothly.

8.3 Prove the recursive relation for the Bezier blending function $B_{k,n}(u) = C(n, k).u_k(1-u)^{n-k}$, where $C(n, k) = n!/(n-k)!\,k!$.

$$B_{k,n}(u) = (1-u)B_{k,n-1}(u) + u\,B_{k-1,n-1}(u), \qquad n > k \geq 1.$$

8.4 Show that the Bezier curve always touches the starting point (for $u = 0$) and the ending point (for $u = 1$).

8.5 Show that the curve defined is the control polygon itself for ($n = 2$), that is, a second order B-spline curve, by suitably selecting three control points for a B-spline curve. Also show that no curve results when $n = 1$, and the discrete points generated are also the control points.

8.6 Prove that when two knot values are the same, the denominator of the B-spline blending function,

$$B_{k,\mu} = \frac{u-u_k}{u_{k+n-1}-u_k} B_{k,n-1}(u) + \frac{u_{k+n}-u}{u_{k+n}-u_{k+1}} B_{k+1,n+1}(u)$$

becomes zero.

8.7 Use a quadratic B-spline curve with five control points to prove that B-spline blending functions sum to unity.

8.8 Show that $B_{k,n}(u)$ is zero outside the interval $[u_k, u_{k+n}]$, so that the support of an nth order B-spline is n spans in the knot vector.

8.9 Calculate the open uniform knot vector for a B-spline of 8 control points and order 4.

8.10 Show that when $n = M + 1 = 4$, B-spline functions with knot vector $\{0, 0, 0, 0, 1, 1, 1, 1\}$ are Bernstein polynomials (i.e. a open uniform cubic B-spline curve with four control points).

8.11 Show that when weight factors are unity, a rational B-spline curve coincides with a B-spline curve with same control points and order.

8.12 Is the effect of taking weight factor 2 for a rational B-spline the same as that of taking two control points at the same location on a B-spline curve?

8.13 Use the C program 8.3 to design a 2D cup. Use a graph sheet to decide the coordinates on the 2D plane of a 640 × 480 space. Judge the weight factor and knot vector as per the shape of different parts of the curves designed.

8.14 Write a C program to design a parabola using a NURBS curve.

8.15 Show that duplicate interior control points on a B-spline curve does not affect the C^2 continuity at the joining point. [*University Question*]

8.16 Enumerate the major differences between Bezier curve B-spline curves and B-spline curve–NURBS.

8.17 Using two Bezier curves with four sets of control points, design a figure of 8, that is, the shape 8.

8.18 Construct a uniform B-spline curve of third order with 4 polygon vertices $\mathbf{P}(1, 1)$, $Q(2, 3)$, $R(4, 3)$ and $S(6, 4)$. Also show the knot values on the curve. [*University Question*]

8.19 Using the Hermite interpolation technique, determine ten points on a cubic spline curve, given $\mathbf{P_0}$ $(0, 1)$, $\mathbf{P_1}(1, 2)$ and $\mathbf{P_0}'(0, 1)$, $\mathbf{P_1}'(0, -1)$. [*University Question*]

8.20 Four control points $\mathbf{P_0}(a, b)$, $\mathbf{P_1}(2, 5)$, $\mathbf{P_2}(4, 4)$ and $\mathbf{P_3}(7, c)$ are available for drawing a periodic cubic B-spline curve segment. Compute the values of a, b and c such that the curve starts from the point (2.1, 4.3) and terminates with slope – 0.5.

CHAPTER 8

```
        else
                d = 0;
        if (temp[i+1] != 0)
                e = ((x[i+k]-t)*temp[i+1])/(x[i+k]-x[i+1]);
        else
                e = 0;
        temp[i] = d + e;
        }
    }
    if (t == (float)x[nplusc])
        temp[npts] = 1;
//put in output n array
    for (i = 1; i <= npts; i++)
    n[i] = temp[i];

void knot(int n,int c,int *x)
{
    int i;
    x[1] = 0;
        for (i = 2; i <= n + c; i++)
        {
            if ( (i > c) && (i < n + 2) )
                    x[i] = x[i-1] + 1;
            else
                    x[i] = x[i-1];
        }
}
```

CHAPTER 9

Solved Exercises

9.1 A sphere of radius 5 units and centre at (5, 5, 5) given by parametric equations

$$x - x_c = r \cos\phi \cos\theta,$$
$$y - y_c = r \cos\phi \sin\theta, \qquad -\pi/2 \le \phi \le \pi/2 \text{ and } -\pi \le \theta \le \pi$$
and
$$z - z_c = r \sin\phi,$$

where ϕ and θ are latitude and longitude angle, respectively. Calculate the points on all the octants of the sphere that are symmetrical to the point at ($\phi = \pi/4$ and $\theta = \pi/4$) for a sphere.

Solution

Calculating for the sphere with centre at origin (0, 0, 0)

$$x = 5 \cos \pi/4 . \cos \pi/4 = 2.5$$
$$y = 5 \cos \pi/4 . \sin \pi/4 = 2.5$$
$$z = 5 \sin \pi/4 = 3.535$$

Point on the first quadrant will be (2.5, 2.5, 3.535)

By symmetry, points on all octants will be

(2.5, 2.5, 3.535), (2.5, 2.5, –3.535), (2.5, –2.5, 3.535), (–2.5, 2.5, 3.535), (2.5, –2.5, –3.535),

(–2.5, –2.5, 3.535), (–2.5, 2.5, –3.535), (–2.5, –2.5, –3.535),

Thus, for sphere with centre at (5, 5, 5) these points can be shifted by 5 units along the positive direction of each axis.

And the points are

(7.5, 7.5, 8.535), (7.5, 7.5, 1.465), (7.5, 2.5, 8.535), (2.5, 7.5, 8.535), (7.5, 2.5, 1.465), (2.5, 2.5, 8.535), (2.5, 7.5, 1.465), (2.5, 2.5, 1.465).

9.2 For a Bezier surface patch given by control points [**P**] find the point on the surface at $u = v = 0.5$.

$$[\mathbf{P}] = \begin{bmatrix} (-3,0,3) & (-3,1,1) & (-3,1,-1) & (-3,0,-3) \\ (-1,1,3) & (-1,1,1) & (-1,1,-1) & (-1,1,-3) \\ (1,1,3) & (1,1,1) & (1,1,-1) & (1,1,-3) \\ (3,0,3) & (3,1,1) & (3,1,-1) & (3,0,-3) \end{bmatrix}$$

Solution

Recalling Eq. 9.18

$$\mathbf{P}(u, v) = [(1-u)^3 \quad 3u(1-u)^2 \quad 3u^2(1-u)u^3]\,[\mathbf{P}] \begin{bmatrix} (1-v)^3 \\ 3v(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{bmatrix}$$

For $u = v = 0.5$

$\mathbf{P}(0.5, 0.5) =$

$$[0.125 \quad 0.375 \quad 0.375 \quad 0.125] \begin{bmatrix} (-3,0,3) & (-3,1,1) & (-3,1,-1) & (-3,0,-3) \\ (-1,1,3) & (-1,1,1) & (-1,1,-1) & (-1,1,-3) \\ (1,1,3) & (1,1,1) & (1,1,-1) & (1,1,-3) \\ (3,0,3) & (3,1,1) & (3,1,-1) & (3,0,-3) \end{bmatrix} \begin{bmatrix} 0.125 \\ 0.375 \\ 0.375 \\ 0.125 \end{bmatrix}$$

Hence, $\mathbf{P}(0.5, 0.5) = (0, 0.9375, 0)$.

9.3 Find the point on a surface at its centre for an open uniform B-spline surface given by 4 × 4 polygon net [**P**].

$$[\mathbf{P}] = \begin{bmatrix} (-3,0,3) & (-3,1,1) & (-3,1,-1) & (-3,0,-3) \\ (-1,1,3) & (-1,1,1) & (-1,1,-1) & (-1,1,-3) \\ (1,1,3) & (1,1,1) & (1,1,-1) & (1,1,-3) \\ (3,0,3) & (3,1,1) & (3,1,-1) & (3,0,-3) \end{bmatrix}$$

The surface is fourth order in u direction and third order in v direction.

Solution

Recalling Eq. 9.22

$$\mathbf{P}(u, v) = [B_{0,k}(u) \quad B_{1,k}(u) \quad \ldots \quad B_{n,k}(u)][\mathbf{P}] \begin{bmatrix} B_{0,l}(v) \\ B_{1,l}(v) \\ \vdots \\ B_{m,l}(v) \end{bmatrix}$$

Since the curve is open uniform, the knot vector will have multiplicity at the ends equal to the order of the B-spline. $m = n = 3$, order = 4 along u and 3 along v.

Thus, the number of knot vector elements along u and v directions will be $3 + 4 + 1 = 8$ along u and $3 + 3 + 1 = 7$ along v, respectively.

The knot vector along u will be [0 0 0 0 1 1 1 1]

And along v knot vector will be [0 0 0 1 2 2 2]

Thus, the centre point will have parametric value $u = 0.5$ and $v = 1.0$.

$$\text{Thus, } \mathbf{P}(u, v) = [B_{0,4}(0.5) \quad B_{1,4}(0.5) \quad B_{2,4}(0.5) \quad B_{3,4}(0.5)][\mathbf{P}]\begin{bmatrix} B_{0,3}(1.0) \\ B_{1,3}(1.0) \\ B_{2,3}(1.0) \\ B_{3,3}(1.0) \end{bmatrix}$$

$$\mathbf{P}(0.5, 1.0) =$$

$$[0.125 \quad 0.375 \quad 0.375 \quad 0.125]\begin{bmatrix} (-3,0,3) & (-3,1,1) & (-3,1,-1) & (-3,0,-3) \\ (-1,1,3) & (-1,1,1) & (-1,1,-1) & (-1,1,-3) \\ (1,1,3) & (1,1,1) & (1,1,-1) & (1,1,-3) \\ (3,0,3) & (3,1,1) & (3,1,-1) & (3,0,-3) \end{bmatrix}\begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

Hence, $\mathbf{P}(0.5, 1.0) = (0, 1, 0)$

Review Questions

9.1 Find the point on a surface at $u = 0.45$ and $v = 0.25$ for a Bezier surface patch given by control points [**P**].

$$[\mathbf{P}] = \begin{bmatrix} (-3,0,3) & (-1,1,3) & (1,1,3) & (3,0,3) \\ (-3,1,1) & (-1,2,1) & (1,2,1) & (3,1,1) \\ (-3,1,-1) & (-1,2,-1) & (1,2,-1) & (3,1,-1) \\ (-3,0,-3) & (-1,1,-3) & (1,1,-3) & (3,0,-3) \end{bmatrix}$$

9.2 For an open uniform B-spline surface given by 4 × 4 polygon net [**P**] find the point on the surface at its centre.

$$[\mathbf{P}] = \begin{bmatrix} (-3,0,3) & (-3,1,1) & (-3,1,-1) & (-3,0,-3) \\ (-1,1,3) & (-1,1,1) & (-1,1,-1) & (-1,1,-3) \\ (1,1,3) & (1,1,1) & (1,1,-1) & (-1,1,-3) \\ (3,0,3) & (3,1,1) & (3,1,-1) & (3,0,-3) \end{bmatrix}$$

The surface is fourth order in both parametric directions.

9.3 Use the free form surface designer (A NURBS surface generator C program) to design the hull of a boat of any imaginary dimensions. Adjust the coordinates and the weightages only. Study the effect of changing these parameters.

9.4 Find the point on a surface at its centre for an rational B-spline surface given by 4 × 4 polygon net [**P**] = [x, y, z, w].

$$[\mathbf{P}] = \begin{bmatrix} (-3,0,3,1) & (-3,1,1,1) & (-3,1,-1,1) & (-3,0,-3,1) \\ (-1,1,3,1) & (-1,1,1,0.5) & (-1,1,-1,0.5) & (-1,1,-3,1) \\ (1,1,3,1) & (1,1,1,0.5) & (1,1,-1,0.5) & (1,1,-3,1) \\ (3,0,3,1) & (3,1,1,1) & (3,1,-1,1) & (3,0,-3,1) \end{bmatrix}$$

The surface is quadratic in both the parametric direction and the knot vectors are open and uniform.

9.5 Explain the advantages and disadvantages of B-spline surface over Bezier surface.

9.6 Show that a NURBS surface can be converted to a B-spline surface by taking the weight factor for each point to unity, by taking any suitable example of 4 × 4 polygon net.

9.7 Is the effect of taking a weight factor of 2 on the NURBS surface at any particular location the same as that of a B-spline surface having two control points at the same location?

9.8 What conditions are required to convert a B-spline surface to a Bezier surface and a NURBS surface to a B-spline surface?

9.9 Compare the advantages and disadvantages of a NURBS surface over B-spline surface.

9.10 Write a C program to generate a bicubic B-spline surface of 5 × 6 polygon net. Calculate 50 parametric points in each parametric direction and join them to form a wireframe with rectangular grid.

9.11(a) Rotate the set of 4 × 4 control polygon net [**P**] about any of its diagonal by 45° and generate a bicubic open uniform B-spline surface point grid of 20 × 20, using any suitable programming language in 640 × 480 screen resolution.

$$[\mathbf{P}] = \begin{bmatrix} (260,250,300) & (300,250,300) & (340,250,300) & (380,250,300) \\ (260,250,260) & (300,150,260) & (340,150,260) & (380,250,260) \\ (260,250,220) & (300,150,220) & (340,150,220) & (380,250,220) \\ (260,250,180) & (300,250,180) & (340,250,180) & (380,100,180) \end{bmatrix}$$

9.11(b) Are the surface points grid the same, if 20 × 20 points on the surface are generated by polygon net without rotation and the surface points grid rotated about the same axis by the same angle?

9.12 Generate a surface polygon net with the control polygon net [**P**] as given in Exercise 9.11 using bicubic, open uniform, rational B-spline surface formulation with weight factor taken as unity. Now rotate the generated surface points about the same axis as used in Exercise 9.11. Will the resultant points coincide with the points calculated in the previous example?

9.13 Use the B-spline surface Program 9.2 to design the hull of a boat of any imaginary dimension by adjusting the control points, and study the effect of changing the parameters.

```
                if(inside_triangle_check(ptA, ptC, ptB, x, y) == TRUE)
                                depthxy[3] = depth(surface[3], x, y);
                        z = depth_background;
                        clr = clr_background;
                        for(numsurf = 0; numsurf < 4; numsurf++)
                        {
                                if(depthxy[numsurf] > z)
                                {
                                        z = depthxy[numsurf];
                                        clr = surf_clr[numsurf];
                                }
                        }
                        putpixel(x, y, clr);
                }
        }
        getch();
        restorecrtmode();
}
```

Solved Exercise 10.4 describes analytical solution of the above C Program demonstration with a tetrahedron scaled to 1/200 times and reflected about xy plane.

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

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

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.

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

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

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

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

which is same as calculated above.

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

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

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

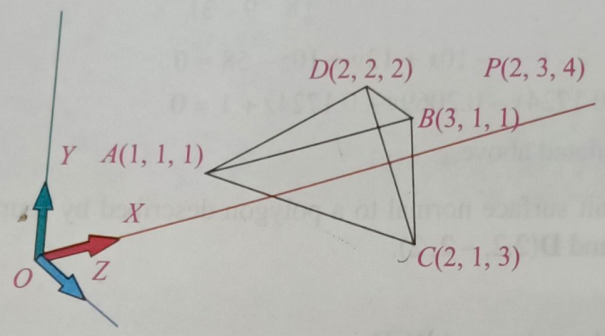
$$\hat{\mathbf{n}}_{mean} = \frac{4\mathbf{i} + 4.2\mathbf{j} + 4\mathbf{k}}{49.64} = 0.0806\mathbf{i} + 0.0846\mathbf{j} + 0.0806\mathbf{k}$$

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

10.3 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), and a point light source is kept at **P**(2, 3, 4). Using back face detection method, find the surfaces on which the light falls and the surfaces which are to be shadowed.

Solution

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



Calculating the surface normals projecting out of the object surfaces:

$$\mathbf{N}_{ACD} = \mathbf{AC} \times \mathbf{AD} = [(2-1)\mathbf{i} + (1-1)\mathbf{j} + (3-1)\mathbf{k}] \times [(2-1)\mathbf{i} + (2-1)\mathbf{j} + (2-1)\mathbf{k}]$$

$$= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & 2 \\ 1 & 1 & 1 \end{vmatrix} = -2\mathbf{i} + \mathbf{j} + \mathbf{k}.$$

$$\mathbf{N}_{CBD} = \mathbf{CB} \times \mathbf{CD} = [(3-2)\mathbf{i} + (1-1)\mathbf{j} + (1-3)\mathbf{k}] \times [(2-2)\mathbf{i} + (2-1)\mathbf{j} + (2-3)\mathbf{k}]$$

$$= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & -2 \\ 0 & 1 & -1 \end{vmatrix} = 2\mathbf{i} + \mathbf{j} + \mathbf{k}.$$

$$\mathbf{N}_{BAD} = \mathbf{BA} \times \mathbf{BD} = [(1-3)\mathbf{i} + (1-1)\mathbf{j} + (1-1)\mathbf{k}] \times [(2-3)\mathbf{i} + (2-1)\mathbf{j} + (2-1)\mathbf{k}]$$

$$= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -2 & 0 & 0 \\ -1 & 1 & 1 \end{vmatrix} = 0\mathbf{i} + 2\mathbf{j} - 2\mathbf{k}.$$

$$\mathbf{N}_{ACB} = \mathbf{AB} \times \mathbf{AC} = [(3-1)\mathbf{i} + (1-1)\mathbf{j} + (1-1)\mathbf{k}] \times [(2-1)\mathbf{i} + (1-1)\mathbf{j} + (3-1)\mathbf{k}]$$

$$= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 2 & 0 & 0 \\ 1 & 0 & 2 \end{vmatrix} = 0\mathbf{i} - 4\mathbf{j} - 0\mathbf{k}.$$

CHAPTER 10

Calculating light vectors to each surface, by joining the point light source **P**(2, 3, 4) to the mean position on the surface:

Mean position of **ACD** = $\left(\frac{1+2+2}{3}\ \frac{1+1+2}{3}\ \frac{1+3+2}{3}\right) = (1.667, 1.333, 2)$

$L_{ACD} = (1.667 - 2)\mathbf{i} + (1.333 - 3)\mathbf{j} + (2 - 4)\mathbf{k} = -0.333\mathbf{i} - 1.667\mathbf{j} - 2\mathbf{k}$.

Mean position of **CBD** = $\left(\frac{2+3+2}{3}\ \frac{1+1+2}{3}\ \frac{3+1+2}{3}\right) = (2.333, 1.333, 2)$

$L_{CBD} = (2.333 - 2)\mathbf{i} + (1.333 - 3)\mathbf{j} + (2 - 4)\mathbf{k} = 0.333\mathbf{i} - 1.667\mathbf{j} - 2\mathbf{k}$.

Mean position of **BAD** = $\left(\frac{3+1+2}{3}\ \frac{1+1+2}{3}\ \frac{1+1+2}{3}\right) = (2, 1.333, 1.333)$

$L_{BAD} = (2 - 2)\mathbf{i} + (1.333 - 3)\mathbf{j} + (1.333 - 4)\mathbf{k} = 0\mathbf{i} - 1.667\mathbf{j} - 2.667\mathbf{k}$.

Mean position of **ACB** = $\left(\frac{1+2+3}{3}\ \frac{1+1+1}{3}\ \frac{1+3+1}{3}\right) = (2, 1, 1.667)$

$L_{ACB} = (2 - 2)\mathbf{i} + (1 - 3)\mathbf{j} + (1.667 - 4)\mathbf{k} = 0\mathbf{i} - 2\mathbf{j} - 2.333\mathbf{k}$.

Since the normal vectors are protruding out of the surface planes, the surface is a back face if the dot product of light vector with the surface normal positive. Any such back face is to be shadowed, leaving other surfaces illuminated.

Checking each face for shadowing

For face **ACD**:

$N_{ACD} \cdot L_{ACD} = (-2\mathbf{i} + \mathbf{j} + \mathbf{k}) \cdot (-0.333\mathbf{i} - 1.667\mathbf{j} - 2\mathbf{k}) = 0.666 - 1.667 - 2 = -3.001 < 0$. Hence, illuminated.

For face **CBD**:

$N_{CBD} \cdot L_{CBD} = (2\mathbf{i} + \mathbf{j} + \mathbf{k}) \cdot (0.333\mathbf{i} - 1.667\mathbf{j} - 2\mathbf{k}) = 0.666 - 1.667 - 2 = -3.001 < 0$. Hence, illuminated.

For face **BAD**:

$N_{BAD} \cdot L_{BAD} = (0\mathbf{i} + 2\mathbf{j} - 2\mathbf{k}) \cdot (0\mathbf{i} - 1.667\mathbf{j} - 2.667\mathbf{k}) = 0 - 3.334 + 5.334 = 2 > 0$. Hence, shadowed.

For face **ACB**:

$N_{ACB} \cdot L_{ACB} = (0\mathbf{i} - 4\mathbf{j} - 0\mathbf{k}) \cdot (0\mathbf{i} - 2\mathbf{j} - 2.333\mathbf{k}) = 0 + 8 + 0 = 8 > 0$. 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.

4 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

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

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

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 $(y - 1)(2 - 2) = (x - 2)(1 - 2)$

$y - 1 = \frac{-2}{2-2}(x-2)$

$y - 1 = -x + 2$

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 $x = 2$

CHAPTER 10

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 $y - 1 = \frac{1-2}{3-2}(x-3)$

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

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

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

or,
$$Z_{BAD} = -y$$

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

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

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**, $x = y$

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 $y - 1 = \frac{1-2}{3-2}(x-3)$

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

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:

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

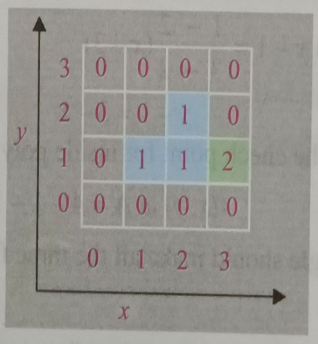
or,
$$y = 1.$$

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.

X	Y	Z		Inside Polygon Test	Z(Maximum)	Colour
1	1	ACD	– 1	True	–1	ACD
		CBD	– 5	False		
		BAD	– 1	True		
	2	ACD	0	False	0	Background
		CBD	– 4	False		
		BAD	– 2	False		
2	1	ACD	– 3	True	– 1	ACD
		CBD	– 3	True		
		BAD	– 1	True		
	2	ACD	– 2	True	– 2	ACD
		CBD	– 2	True		
		BAD	– 2	True		
3	1	ACD	– 5	False	– 1	CBD
		CBD	– 1	True		
		BAD	– 1	True		
	2	ACD	– 4	False	0	Background
		CBD	0	False		
		BAD	– 2	False		



Note: It is useful to keep all inside polygon test function to give positive result for any point inside e polygon. If any position is overlapped by two planes, any one should be selected throughout the heme. For example, position (2, 1) and (1, 2) is commonly occupied by **ACD** and **BAD**. Priority is ven to **ACD** for no particular reason.

$m \times n$ screen resolution has coordinates varying from 0 to $(m - 1)$ along x axis and 0 to $(n - 1)$ ong y axis.

The above method is Image space method and is suitable for intersecting polygons also.

rectangular window has the corners (0, 0) and (8, 8). First polygon has vertices **A**(9, 2), **B**(12, 6) nd **C**(14, 1) and the second polygon has vertices $P(2, 1)$, $Q(3, 6)$ and $R(6, 3)$. Find the relationship etween polygon and window using bounding box method.

Solution

The window has (xw_{min}, yw_{min}) and (xw_{max}, yw_{max}) as (0, 0) and (8, 8), respectively.

The bounding rectangle for the first polygon has corners (x_{min}, y_{min}) and (x_{max}, y_{max}) as (9, 1) and (14, 6).

As for the first polygon $(x_{min} = 9) > (xw_{max} = 8)$, the first polygon is disjoint from the window.

Now, for the second polygon the bounding rectangle has corners (x_{min}, y_{min}) and (x_{max}, y_{max}) as (2, 1) and (6, 6). Here,

$$(x_{min} = 2) > (xw_{min} = 0) \text{ and } (x_{max} = 6) < (xw_{max} = 8) \text{ and}$$

$$(y_{min} = 1) > (yw_{min} = 0) \text{ and } (y_{max} = 6) < (yw_{max} = 8)$$

Hence, the second polygon is contained within the window.

Review Questions

10.1 Find the equation of the plane passing through the points (1, 3, 4), (5, 5, 5) and (7, 9, 4).

10.2 Calculate the mean unit surface normal to a polygon described by four position vectors **A**(3, 0, – 1), **B**(1, 2, – 1), **C**(1, 0, 1) and **D**(3.2, – 2, 1).

10.3 A solid tetrahedron given by position vectors **A**(1, 1, 1), **B**(3, 1, 1), **C**(2, 1, 3) and **D**(2, 2, 2), and a parallel beam of light source is given by – (**i** + 5**j** + 6**k**) that falls on tetrahedron. Find the surfaces which are illuminated and the surfaces which are to be shadowed, using Back face detection method.

10.4 A solid tetrahedron given by position vectors **A**(0, 2, 1), **B**(2, 2, 1), **C**(1, 2, 3) and **D**(1, 3, 2), and a point light source is kept at **P**(1, 4, 4). Find the surfaces on which the light falls and the surfaces which are to be shadowed, using Back face detection method.

10.5 A tetrahedron is given by position vectors **A**(2, 2, – 1), **B**(4, 2, – 1), **C**(3, 2, – 3) and **D**(3, 4, – 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 6 × 6, and background colour as black (colour value = 0). The colour of the plane **ACD** is blue (1), **CBD** is green (2), **BAD** is cyan (3) and **ACB** is red (4).

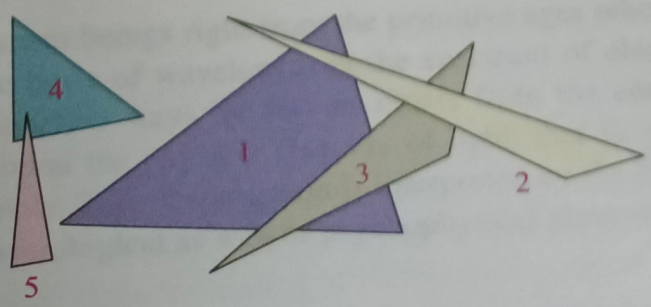
Will the visible planes change if it is rotated about z axis by 45°? Find the visible planes, if it is rotated about x axis by 30° and y axis by 45°.

What will be the effect if perspective projection is used for the same?

10.6 A rectangular window has the corners (0, 0) and (16, 24). First polygon has vertices **A**(18, 6), **B**(24, 18) and **C**(28, 3) and the second polygon has vertices P(4, 3), Q(6, 18) and R(12, 9). Find the relationship between polygon and window using bounding box method.

10.7 Differentiate between the object space method and image space method of detecting visible surface.

10.8 The depths of the triangles shown below numbered as 1, 2, 3, 4, and 5 are 10, 7, 9, 10 and 4 units, respectively. Show the flow of Painter's algorithm to plot these triangles.



CHAPTER 10

10.9 A cube is given by $[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$. Scale the cube by a factor of three about its centre. Rotate the cube about an axis parallel to x axis through its centre by 45°, and about an axis parallel to y axis through its centre by 45°. Use any of the suitable method to find the hidden lines. Draw the projected cube on xy plane i.e. $z = 0$, with hidden lines in different attribute.

10.10 Enumerate the drawbacks of Painter's algorithm, and how it is overcome by BSP tree method?

```
        }
    }
    getch();
    setmode(0x3);
}
```

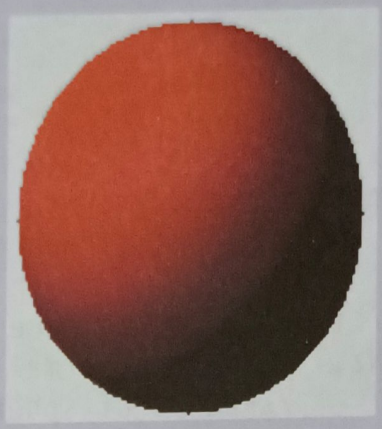


Fig. 11.19 *Output of the above-listed program*

Solved Exercises

11.1 Calculate the mean unit surface normal to a polygon described by four position vectors **A**(0, 0, 0), **B**(1, 0, 0), **C**(1, 1, 0) and **D**(0, 1, 0.2) as shown in Fig. 11.20.

Solution

$$\mathbf{AB} = \mathbf{OB} - \mathbf{OA} = (1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) - (0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) = \mathbf{i}$$

$$\mathbf{BC} = \mathbf{OC} - \mathbf{OB} = (1\mathbf{i} + 1\mathbf{j} + 0\mathbf{k}) - (1\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) = \mathbf{j}$$

$$\mathbf{CD} = \mathbf{OD} - \mathbf{OC} = (0\mathbf{i} + 1\mathbf{j} + 0.2\mathbf{k}) - (1\mathbf{i} + 1\mathbf{j} + 0\mathbf{k}) = -1\mathbf{i} + 0.2\mathbf{k}$$

$$\mathbf{AD} = \mathbf{OD} - \mathbf{OA} = (0\mathbf{i} + 1\mathbf{j} + 0.2\mathbf{k}) - (0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) = \mathbf{j} + 0.2\mathbf{k}$$

$$\check{\mathbf{n}}_A = \mathbf{AB} \times \mathbf{AD} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & 0 \\ 0 & 1 & 0.2 \end{vmatrix} = -0.2\mathbf{j} + \mathbf{k}$$

$$\check{\mathbf{n}}_B = \mathbf{BC} \times \mathbf{BA} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{vmatrix} = \mathbf{k}$$

$$\check{\mathbf{n}}_C = \mathbf{CD} \times \mathbf{CB} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 0 & 0.2 \\ 0 & -1 & 0 \end{vmatrix} = 0.2\mathbf{i} + \mathbf{k}$$

$$\check{\mathbf{n}}_D = \mathbf{DA} \times \mathbf{DC} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & -1 & -0.2 \\ 1 & 0 & -0.2 \end{vmatrix} = 0.2\mathbf{i} - 0.2\mathbf{j} + \mathbf{k}$$

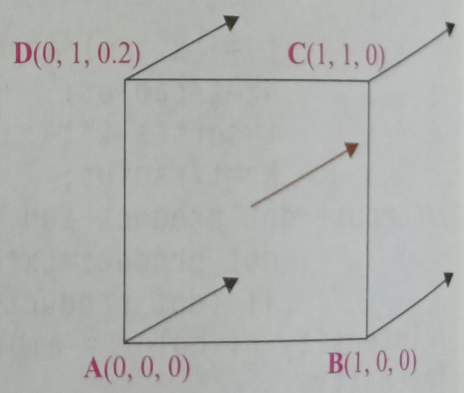


Fig. 11.20

Hence, the mean surface normal is

$$\check{\mathbf{n}}_{mean} = \frac{\check{\mathbf{n}}_A + \check{\mathbf{n}}_B + \check{\mathbf{n}}_C + \check{\mathbf{n}}_D}{4} = \frac{(-0.2\mathbf{j} + \mathbf{k}) + (\mathbf{k}) + (0.2\mathbf{i} + \mathbf{k}) + (0.2\mathbf{i} - 0.2\mathbf{j} + \mathbf{k})}{4}$$

or, $\check{\mathbf{n}}_{mean} = 0.1\mathbf{i} - 0.1\mathbf{j} + \mathbf{k}$

Mean unit surface normal $\hat{\mathbf{n}}_{mean} = \frac{\hat{\mathbf{n}}_{mean}}{|\hat{\mathbf{n}}_{mean}|} = \frac{0.1\mathbf{i} - 0.1\mathbf{j} + \mathbf{k}}{\sqrt{0.1^2 + 0.1^2 + 1^2}}$

or, $\hat{\mathbf{n}}_{mean} = \frac{0.1\mathbf{i} - 0.1\mathbf{j} + \mathbf{k}}{1.00995} = 0.099\mathbf{i} - 0.099\mathbf{j} + 0.99\mathbf{k}$

11.2 Calculate the vertex normal at the point P for the tetrahedron shown in Fig. 11.21.

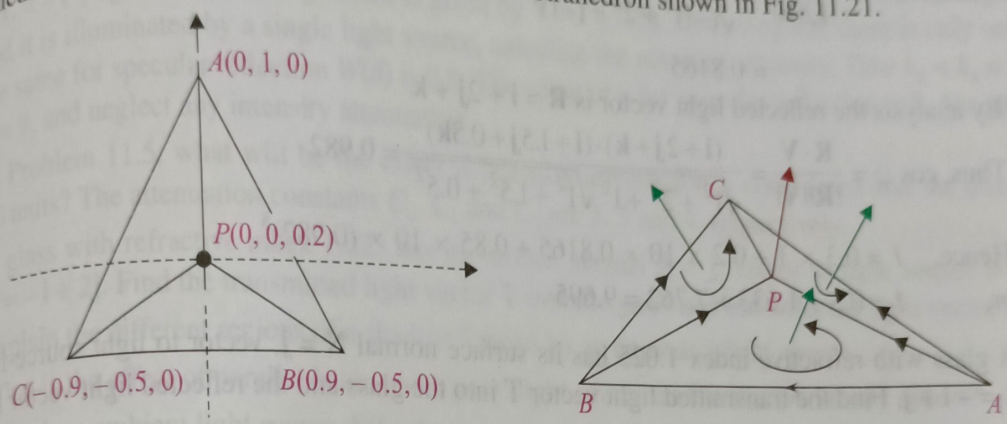


Fig. 11.21

Solution

Finding the surface normal to the surfaces connected to the point P.

As the surfaces attached to the point P are plane surface, hence the mean surface normal is same as vertex normal to any plane.

$\mathbf{N}_{APB} = \mathbf{AP} \times \mathbf{AB} = (0\mathbf{i} - 1\mathbf{j} + 0.2\mathbf{k}) \times (0.9\mathbf{i} - 1.5\mathbf{j} + 0\mathbf{k}) = 0.3\mathbf{i} + 0.18\mathbf{j} + 0.9\mathbf{k}$.

$\mathbf{N}_{ACP} = \mathbf{AC} \times \mathbf{AP} = (-0.9\mathbf{i} - 1.5\mathbf{j} + 0\mathbf{k}) \times (0\mathbf{i} - 1\mathbf{j} + 0.2\mathbf{k}) = -0.3\mathbf{i} + 0.18\mathbf{j} + 0.9\mathbf{k}$.

$\mathbf{N}_{BPC} = \mathbf{BP} \times \mathbf{BC} = (-0.9\mathbf{i} + 1.5\mathbf{j} + 0.2\mathbf{k}) \times (-1.8\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}) = 0\mathbf{i} - 0.36\mathbf{j} + 2.7\mathbf{k}$.

The vertex normal is mean of the adjacent plane normals

Thus,

$$\mathbf{V}_P = \frac{\mathbf{N}_{APB} + \mathbf{N}_{ACP} + \mathbf{N}_{BPC}}{3}$$

$$= \frac{(0.3\mathbf{i} + 0.18\mathbf{j} + 0.9\mathbf{k}) + (-0.3\mathbf{i} + 0.18\mathbf{j} + 0.9\mathbf{k}) + (0\mathbf{i} - 0.36\mathbf{j} + 2.7\mathbf{k})}{3}$$

Hence, $\mathbf{V}_P = 0\mathbf{i} + 0\mathbf{j} + 1.5\mathbf{k}$.

11.3 Three points on a shiny metallic surface is given **A**(2, 0, 4), **B**(15, 0, 6) and (9, 0, 9). The light vector is $\mathbf{L} = -\mathbf{i} + 2\mathbf{j} - \mathbf{k}$ and the viewing vector is given by $\mathbf{V} = \mathbf{i} + 1.5\mathbf{j} + 0.5\mathbf{k}$. Assuming that there is only one object and that it is illuminated by a single light source, calculate the resultant intensity. Take $k_d = 0.2$, $k_a = 0.3$ and the same for specular reflection $W(\theta)$ is 0.85. The value of n for specular reflection is 5. Assume $I_a = 1$, $I_l = 10$, and neglect any intensity attenuation.

Solution

As the three surface points completely lie on xz plane the normal vector is perpendicular to it, i.e. along y axis.

$$\mathbf{N} = \mathbf{j}$$

Following illumination model $I = k_a I_a + \Sigma k_d I_l$

$$(\mathbf{N} \cdot \mathbf{L}) + W(\theta) I_1 \cos^n \varphi$$

Where **L** and **N** are unit normal vectors, thus the dot product is the cosine of angle between them.

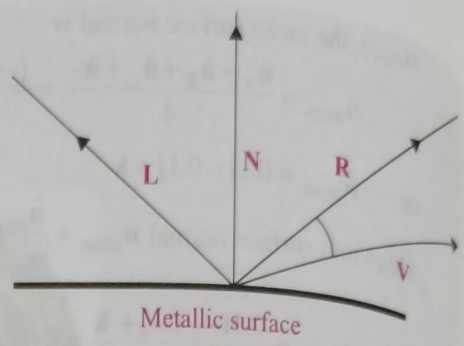


Fig. 11.22

$$\text{Thus, } \cos\theta = \frac{\mathbf{L}\cdot\mathbf{N}}{|\mathbf{L}||\mathbf{N}|} = \frac{\mathbf{j}\cdot(-\mathbf{i}+2\mathbf{j}-\mathbf{k})}{\sqrt{(-1)^2+2^2+(-1)^2}}$$

$$= 0.8165$$

By analysis the reflected light vector is $\mathbf{R} = \mathbf{i} + 2\mathbf{j} + \mathbf{k}$

$$\text{Thus, } \cos\varphi = \frac{\mathbf{R}\cdot\mathbf{V}}{|\mathbf{R}||\mathbf{V}|} = \frac{(\mathbf{i}+2\mathbf{j}+\mathbf{k})\cdot(\mathbf{i}+1.5\mathbf{j}+0.5\mathbf{k})}{\sqrt{1^2+2^2+1^2}\sqrt{1^2+1.5^2+0.5^2}} = 0.982$$

Hence, $I = 0.3 \times 1 + 0.2 \times 10 \times 0.8165 + 0.85 \times 10 \times (0.982)^5$

or, $I = 0.3 + 1.633 + 7.762 = 9.695$

11.4 A glass with refractive index 1.025 has its surface normal $\mathbf{N} = \mathbf{j}$, vector to light source is given by $\mathbf{L} = -\mathbf{i} + \mathbf{j}$. Find the transmitted light vector **T** into the glass and the reflected light vector **R**.

Solution

$$\cos\theta_i = \frac{\mathbf{L}\cdot\mathbf{N}}{|\mathbf{L}||\mathbf{N}|} = \frac{(-\mathbf{i}+\mathbf{j})\cdot\mathbf{j}}{\sqrt{(-1)^2+1^2}} = \frac{1}{\sqrt{2}} = \cos 45°.$$

By inspection the reflected ray is inclined in positive x direction to the same angle as the light ray is inclined in negative x direction.

Hence, $\mathbf{R} = \mathbf{i} + \mathbf{j}$.

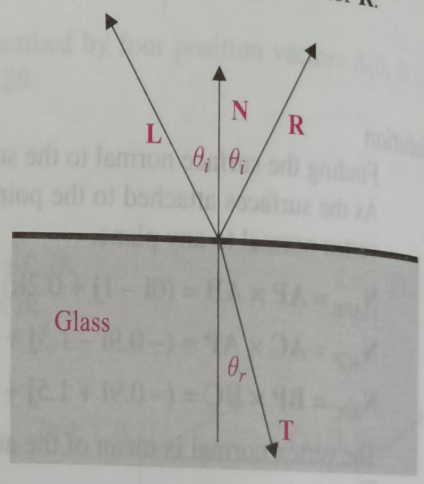


Fig. 11.23

By, Snell's law ${}^i\eta_r = \dfrac{\sin\theta_2}{\sin\theta_r} = \dfrac{{}^a\eta_r}{{}^a\eta_i} = \eta_{glass}$ where ${}^a\eta_b$

= Refractive index of the medium "b" with respect to medium "a".

Hence, $\sin\theta_r = \dfrac{\sin 45°}{1.025}$

or, $\theta_r = 43.62°$.

The transmitted light vector is given by $\mathbf{T} = \left(\dfrac{\eta_i}{\eta_r}\cos\theta_i - \cos\theta_r\right)\mathbf{N} - \dfrac{\eta_i}{\eta_i}\mathbf{L}$, (refer Section 11.7)

$$\text{Thus, } \mathbf{T} = \left(\frac{1}{1.025}\cos 45° - \cos 43.62°\right)\mathbf{N} - \frac{1}{1.025}\mathbf{L}$$

or, $\mathbf{T} = -0.03407\,\mathbf{N} - 0.9756\,\mathbf{L} = -0.03407\,\mathbf{j} - 0.9756\,(-\mathbf{i} + \mathbf{j}) = 0.9756\,\mathbf{i} - 1.00967\,\mathbf{j}$.

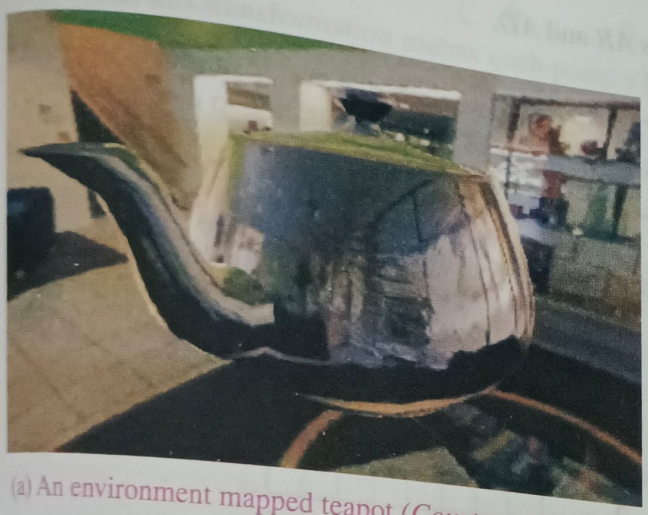
CHAPTER 11

surrounding the object onto a 2D environment map (for example cube faces) as a geometric map shape that surrounds an object, reflections may be defined for a single point in space. Figure 12.14(a) shows a teapot mapped with environment mapping and global lighting effects.

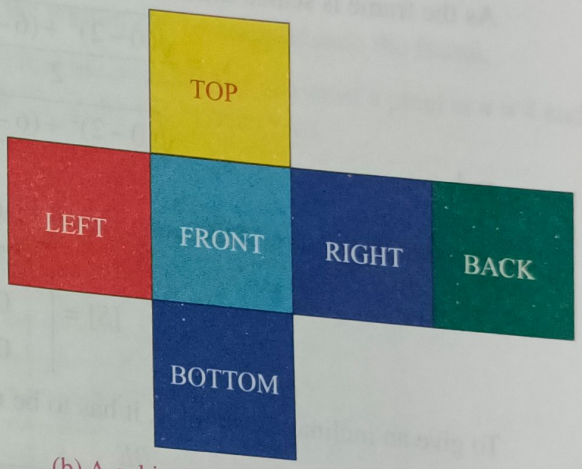
Using a cube as the environment map shape is similar to planar mapping. Instead of using one texture map, box mapping uses six faces of the texture bitmap, one each for the left, right, front, back, top and bottom sides of the object, as shown in Fig. 12.14(b). To texture map the front and back sides, the z-component of an objects point is eliminated and the remaining x and y components are used to locate the colour in the corresponding texture maps.

Other common shapes used for environment array preparation are cylinders and spheres, which are mapped on to the object by the methods discussed in previous sections. The areas of a sphere or a cylinder can be subdivided similar to the cube, which contains the surrounding images uniformly. The adaptation of any shape depends on the dominance of the shape of the final object on which it is to be mapped, so as to avoid any unnatural distortion of the mapped image. For example, a spherical mapping is preferred for mapping over a kettle, cubical mapping over a box or a cylindrical mapping over a can.

This is a quick approximation of a more accurate global effect of the illumination technique discussed in Chapter 11.



(a) An environment mapped teapot *(Courtesy: NVIDIA)*



(b) A cubical environment array elements to be mapped over the objects surfaces

Fig. 12.14 *Environment mapping*

SUMMARY

This chapter presents a method for giving images a realistic appearance by adding patterns or textures to their surface. Surface texture reviews parametric surface description as a means of establishing a one-to-one relationship between the objects surface and the texture to be mapped. It then covers pasting a predefined flat texture onto a surface. A number of algorithms have been discussed showing how texture mapping on a cylindrical surface, spherical surface and rectangular frame is done. Bump mapping which gives an uneven appearance to a smooth surface has been explained. Environment mapping is also discussed for environment array mapping on an object in relation to the viewing direction.

Solved Exercises

12.1 Find the transformation matrix that maps a texture whose diagonally opposite corners are (0, 0), (10, 10) onto a frame whose corners are (2, 2), (0, 6), (8, 10) and (10, 6).

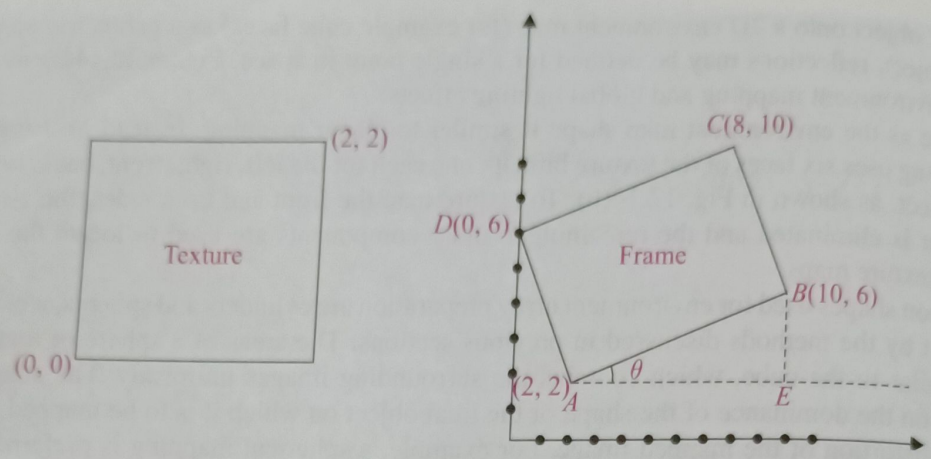


Fig. 12.15

Solution

As the frame is scaled differently along its sides AB and AD,

$$s_x = \frac{\sqrt{(0-2)^2+(6-2)^2}}{2} = 4.2721$$

and

$$s_y = \frac{\sqrt{(0-2)^2+(6-2)^2}}{2} = 2.2360$$

To make the texture of the same size as that of the frame, the scaling matrix will be

$$[S] = \begin{bmatrix} 4.2721 & 0 & 0 \\ 0 & 2.2360 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To give an inclination about A, it has to be rotated anticlockwise about (0, 0) by angle θ

where,

$$\sin\theta = \frac{BE}{AB} = \frac{6-2}{\sqrt{(10-2)^2+(6-2)^2}}$$

$$= \frac{4}{\sqrt{64+16}} = \frac{4}{\sqrt{80}}$$

$$= \frac{4}{4\sqrt{5}} = \frac{1}{\sqrt{5}} = 0.4472$$

and

$$\cos\theta = \frac{AE}{AB} = \frac{10-2}{\sqrt{(10-2)^2+(6-2)^2}}$$

$$= \frac{8}{\sqrt{64+16}} = \frac{2}{\sqrt{5}} = 0.8944$$

The transformation matrix for an anticlockwise rotation will be

$$[R] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.8944 & -0.4472 & 0 \\ 0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now to bring the lower corner to A, the translation matrix will be

$$[T] = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

Thus the composite transformation matrix for a combined transformation of scaling, rotation and translation will be

$$[X] = [S][R][T]$$

or,

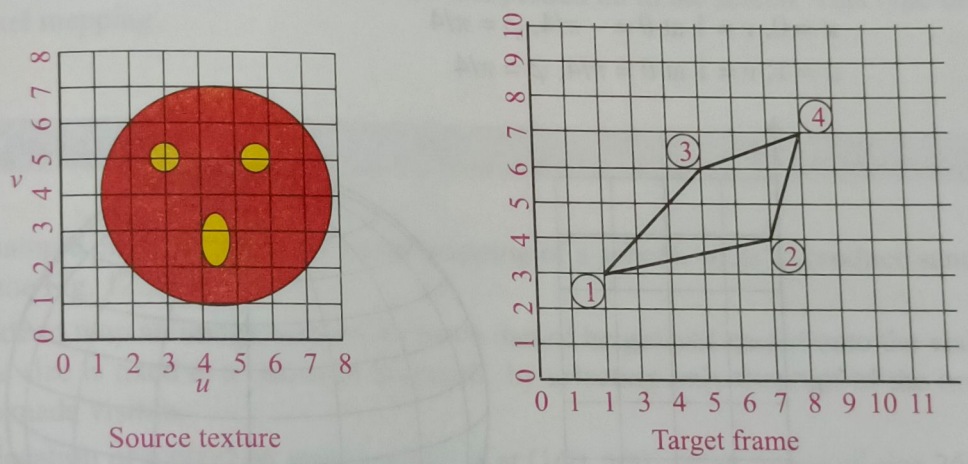
$$[X] = \begin{bmatrix} 4.2721 & 0 & 0 \\ 0 & 2.2360 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & -0.4472 & 0 \\ 0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

or,

$$[X] = \begin{bmatrix} 3.8210 & -1.9105 & 3.8210 \\ 0.9999 & 1.9999 & 5.9996 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

With this transformation matrix each point within the texture can be mapped onto the frame.

12.2 For a texture of size 8 × 8 given below (Fig. 12.16) find the location and colour of a pixel at $u = 4$ and $v = 5$ on a quadrilateral frame having corners as (2, 3), (7, 4), (8, 7) and (5, 6).



Source texture

Target frame

Fig. 12.16

Solution

It can be seen that at $u = 4$ and $v = 5$, the colour is red.

Thus, the pixel colour value on the target frame will also be red.

The location of this pixel on the target frame can be found by

$$X = \left[\frac{x_2 - x_1}{w}u + x_1\right]\frac{h - v}{h} + \left[\frac{x_4 - x_3}{w}u + x_3\right]\frac{v}{h}$$

$$Y = \left[\frac{y_2 - y_1}{w}u + y_1\right]\frac{h - v}{h} + \left[\frac{y_4 - y_3}{w}u + y_3\right]\frac{v}{h}, \text{ where } w = \text{width} = 8 \text{ and } h = \text{height} = 8.$$

Substituting u and v with the given constants

CHAPTER 12

$$X = \left[\frac{7-2}{8}4+3\right]\frac{8-5}{8} + \left[\frac{8-5}{8}4+5\right]\frac{5}{8} = 5.75$$

and

$$Y = \left[\frac{4-3}{8}4+3\right]\frac{8-5}{8} + \left[\frac{7-6}{8}4+6\right]\frac{5}{8} = 5.375$$

the required location on the frame is (5.75, 5.375)

12.3 Map a rectangular image of dimension a × b on to a spherical surface patch of radius r units given by

$$x - x_c = r \sin\theta \sin\varphi$$
$$y - y_c = r \cos\varphi$$
$$z - z_c = r \cos\theta \sin\varphi$$

where $-\pi/4 \le \theta \le \pi/4$ and $-\pi/4 \le \varphi \le \pi/4$.

Solution

Assuming linear mapping of texture onto the surface

$$\theta = Au + B \quad \text{and} \quad \varphi = Cv + D$$

Applying the boundary conditions to the given sphere

$u = 0, v = 0$ at $\theta = -\pi/4, \varphi = -\pi/4$

$u = 1, v = 0$ at $\theta = \pi/4, \varphi = -\pi/4$

$u = 0, v = 1$ at $\theta = -\pi/4, \varphi = \pi/4$

$u = 1, v = 1$ at $\theta = \pi/4, \varphi = \pi/4$

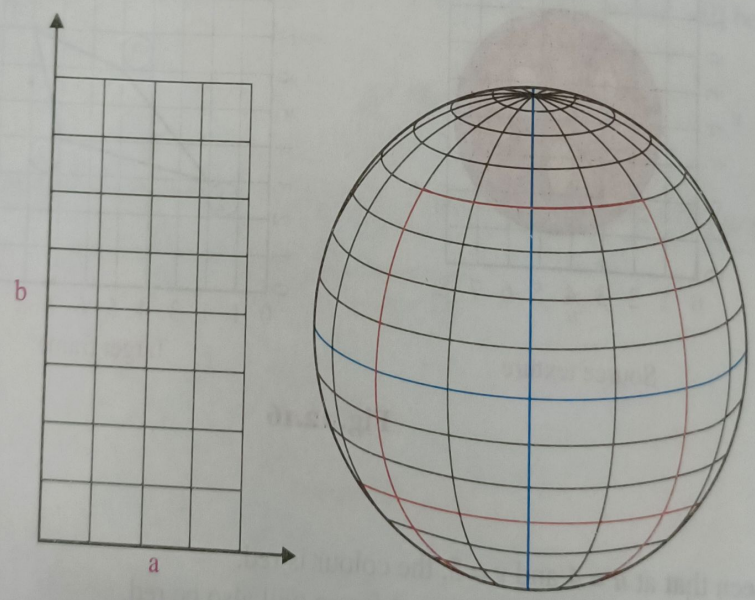


Fig. 12.17

or,

$$\theta = \frac{\pi}{2}u - \frac{\pi}{4} \quad \text{and} \quad \varphi = \frac{\pi}{2}v - \frac{\pi}{4}$$

So, any point uv on a texture space is to be mapped to the corresponding $\theta\varphi$ on the spherical surface patch. u and v parameters correspond to a and b sides of the image, which is made to vary between normalized value 0 and 1.

Hence $u = x/a$ and $v = y/b$ for any pixel xy on the image.

x (Image)	y (Image)	u (Parameter)	v (Parameter)	θ (Sphere)	φ (Sphere)
0	0	0	0	$-\pi/4$	$-\pi/4$
0	b/2	0	0.5	$-\pi/4$	0
0	b	0	1	$-\pi/4$	$\pi/4$
a/2	0	0.5	0	0	$-\pi/4$
a/2	b/2	0.5	0.5	0	0
a/2	b	0.5	1	0	$\pi/4$
a	0	1	0	$\pi/4$	$-\pi/4$
a	b/2	1	0.5	$\pi/4$	0
a	B	1	1	$\pi/4$	$\pi/4$

With this value of θ and φ, the corresponding value on the sphere with centre (x_c, y_c, z_c) and radius r is calculated. Using orthographic projection, only x, y is to be plotted directly to the screen, neglecting the z value.

Note: With any projected pixels of any known sphere on screen, the values of θ and φ can be calculated. Now, by incrementing the screen pixels, corresponding θ and φ can be calculated. With this θ and φ the corresponding colour values on image can be obtained and plotted on to the screen. This type of mapping is known as inverse pixel mapping.

CHAPTER 12

Review Questions

12.1 Modify the sample program given for bump mapping of a sphere, so as to produce sandstone walls, as shown in the Fig. 12.2 (d).

12.2 Write a C code to map an image with width twice that of height and map it onto the visible face of a sphere whose size is fixed at a radius of 80 pixels, by selecting only that half of the image which is desired to be made visible.

12.3 What is the location of a pixel on screen which is at (160, 100) on an image of size 264 × 198. The projected quadrilateral on which it has to be mapped has device or screen coordinates of the corners (90, 40), (170, 70), (90, 160) and (170, 130).

12.4 Map a rectangular image of dimension 100 × 100 on to a spherical surface patch of radius r units given by

$$x = 160 - 50 \cos\phi \sin\theta$$
$$y = 100 + 50 \sin\phi$$
$$z = 100 + 50 \cos\phi \sin\theta$$

where $0 \le \theta \le \pi/2$ and $-\pi/4 \le \varphi \le \pi/4$.

Find x, y, z on screen coordinates on 320 × 200 resolution using orthographic projection.

12.5 When does skipping or overwriting of pixels occur while converting a texture space coordinates to object space coordinates.

12.6 Devise a bump mapping function using a lookup table to generate a brick wall.

12.7 Explain the physical significance of bump mapping. Does the effect change if the light direction is changed?

12.8 While mapping a cylindrical object with a rectangular texture, which parametric direction has the maximum probability for skipping or overwriting of a pixel? What is the effect of such a situation and how can it be reduced?

12.9 How does environment mapping differ from a surface texturing process? What is the effect of any directional light source?

12.10 Map a rectangular texture of dimension a × b to the visible front portion of a cylinder of height b and radius r. The axis of the cylinder is inclined at θ to the device x coordinate.

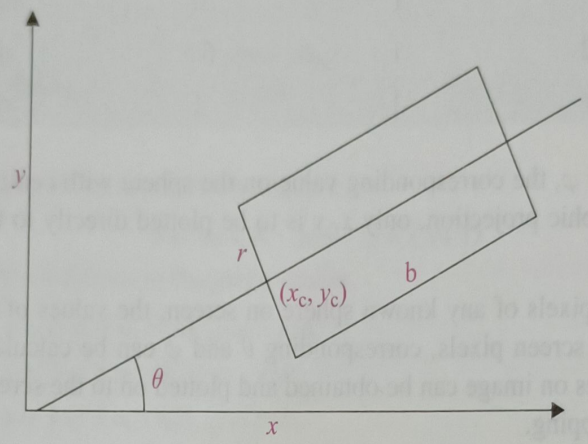


Fig. 12.18

12.11 Write a pseudo code for mapping a cubical environment array on to a spherical surface.

12.12 Develop a C program for texture mapping of polyhedrons.

12.13 Which part of a sphere is more likely to undergo skipping or overwriting of colour information when a rectangular bitmap is mapped over its surface?

12.14 Develop a C program for mapping a cubical environment array on cubical surfaces. What will be the effect of rotation of the cube?

Phong shading requires more calculations, but produces better results for specular reflection than Gouraud shading, in that it creates more realistic highlights. With Gouraud shading, nowhere inside the polygon can you have a brighter colour than a vertex. Unless the highlight occurs on or near the vertex, it will be missed altogether. When it is near the vertex it will be spread over the whole polygon. With Phong shading, the internal point can definitely have a higher value than the vertex. The highlight will occur tightly focused near the correct position. Moreover, the Phong shading model greatly reduces the Mach band effect. This is a effect whereby the contrast between the two areas of different shades is enhanced and appears as the border between the shades.

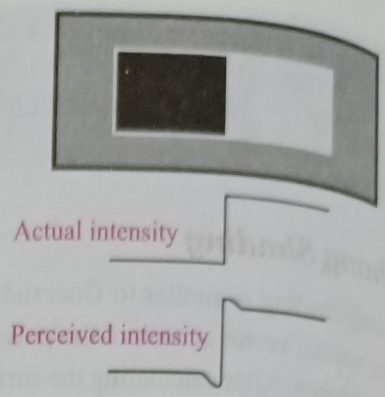
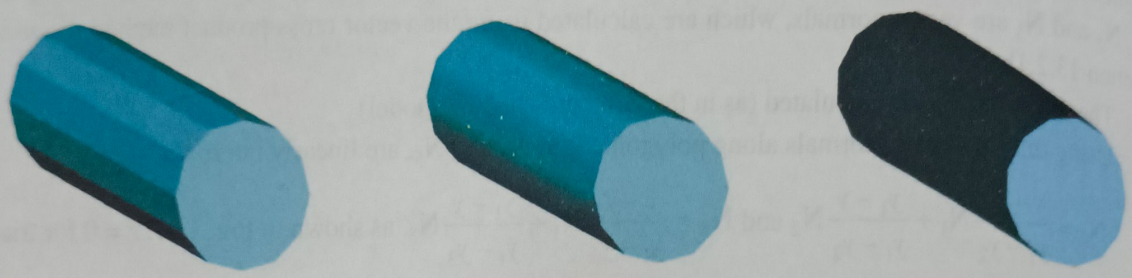


Fig. 13.8 *Mach band effect*



(a) Flat shading (b) Gouraud shading (c) Phong shading

Fig. 13.9 *Various shading effects on a cylinder*

Additional source codes in C for this chapter are available on the accompanying website (http://www.mhhe.com/ansinha/cg)

1. Gouraud intensity interpolation program using 256 colours.
2. Z–Flat shading of the mesh model and also the B-spline surface.
3. Lambert flat shaded wire mesh figure of the B-spline and NURBS surface.
4. Gouraud shading model of the B-spline and NURBS surface.

SUMMARY

Shading creates greater realism. A shading model describes how light reflects off a surface, depending on the surface and its orientation to both light sources and the cameras eye. In this chapter, a number of shading techniques have been discussed. Lambert shading is the simplest shading method that takes light sources and illumination models into consideration. Objects do not appear smoothly rounded. Phong shading is a technique for rendering the surface of displayed objects, making them appear smoothly rounded, creased or peaked as desired. Gouraud shading is another technique for smooth rendering but is less powerful than Phong shading.

Solved Exercises

13.1 Calculate the pixel colour value at the centroid P of the triangle as shown below. The figure shows the coordinates and the colour value of the vertices of the triangle. Use the Gouraud interpolation technique for interpolation.

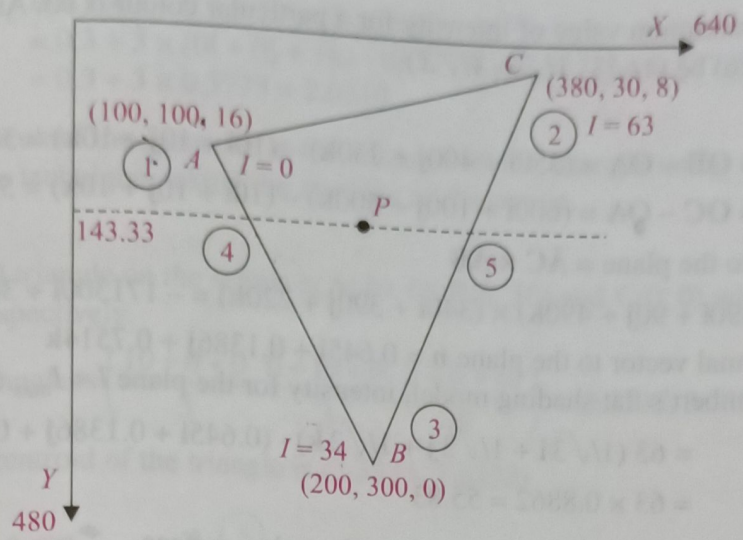


Fig. 13.10

Solution

The centroid of the triangle is at

$$\left(\frac{100+200+380}{3}, \frac{100+300+30}{3}, \frac{16+0+8}{3}\right) = (226.67, 143.33, 8).$$

The parallel projected pixels can be obtained by assigning $z = 0$ to all the points, and considering only the projected (x, y).

Using Gouraud bilinear interpolation for the scan line $y = 143.33$

$$I_4 = \frac{y_3 - y_4}{y_3 - y_1} I_1 + \frac{y_4 - y_1}{y_3 - y_1} I_3 = \frac{300 - 143.33}{300 - 100} 0 + \frac{143.33 - 100}{300 - 100} 34$$

or, $$I_4 = 7.3661$$

and $$I_5 = \frac{y_3 - y_5}{y_3 - y_2} I_2 + \frac{y_5 - y_2}{y_3 - y_2} I_3 = \frac{300 - 143.33}{300 - 30} 63 + \frac{143.33 - 30}{300 - 30} 34$$

or, $$I_5 = 36.556 + 14.2711 = 50.8271$$

Left edge along the scan line $(x_4) = \min[x_1 + (y - y_1)/\text{ slope } (1 - 3), x_2 + (y - y_2)/\text{ slope } (2 - 3)]$

slope$(1 - 3) = (300 - 100)/(200 - 100) = 2$

Thus, $x_4 = 100 + (143.33 - 100)/2 = 121.665$

Right edge $(x_5) = \max[x_1 + (y - y_1)/\text{ slope } (1 - 3), x_2 + (y - y_2)/\text{ slope } (2 - 3)]$.

slope$(2 - 3) = (300 - 30)/(200 - 380) = -1.5$

Thus, $x_5 = 380 - (143.33 - 30)/1.5 = 304.446$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

or, $$I_p = \frac{304.446 - 226.67}{304.446 - 121.665} 7.3661 + \frac{226.67 - 121.665}{304.446 - 121.665} 50.8271 = 32.33$$

13.2 Find the unit vectors normal to the plane of the triangle whose vertices are **A**(10, 10, 10), **B**(350, 400, 230) and **C**(600, 100, 500). Find the colour intensity value of the plane using the Lambert flat shading

CHAPTER 13

method, if the maximum value of intensity for a particular colour is 63. Assume the unit light vector out of the plane to be $(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$.

Solution

Edge $\mathbf{AB} = \mathbf{OB} - \mathbf{OA} = (350\mathbf{i} + 400\mathbf{j} + 230\mathbf{k}) - (10\mathbf{i} + 10\mathbf{j} + 10\mathbf{k}) = 340\mathbf{i} + 390\mathbf{j} + 220\mathbf{k}$

$\mathbf{AC} = \mathbf{OC} - \mathbf{OA} = (600\mathbf{i} + 100\mathbf{j} + 500\mathbf{k}) - (10\mathbf{i} + 10\mathbf{j} + 10\mathbf{k}) = 590\mathbf{i} + 90\mathbf{j} + 490\mathbf{k}$

Normal vector to the plane = $\mathbf{AC} \times \mathbf{AB}$

$$\mathbf{N} = (590\mathbf{i} + 90\mathbf{j} + 490\mathbf{k}) \times (340\mathbf{i} + 390\mathbf{j} + 220\mathbf{k}) = -171300\mathbf{i} + 36800\mathbf{j} + 199500\mathbf{k}$$

Hence, unit normal vector to the plane $\mathbf{n} = 0.645\mathbf{i} + 0.1386\mathbf{j} + 0.7514\mathbf{k}$

Hence, with Lambert's flat shading model, intensity for the plane $I = I_{max}(\mathbf{L} \cdot \mathbf{n})$

$$= 63\,(1/\sqrt{3}\,\mathbf{i} + 1/\sqrt{3}\,\mathbf{j} + 1/\sqrt{3}\,\mathbf{k}) \cdot (0.645\mathbf{i} + 0.1386\mathbf{j} + 0.7514\mathbf{k})$$

$$= 63 \times 0.8862 = 55.83$$

13.3 The position vectors for the vertices of a triangular surface are given by **A**(10, 0, 0), **B**(0, 10, 0) and **C**(0, 0, 10). A point light source is at **P**(0, 0, 20). Find the intensities at the vertices of the quadrilateral, if the ambient light intensity is 1 and the directional light intensity is 10. Assume $k_a = k_d = 0.3$. Neglect any intensity attenuation and specular effect.

Solution

Normal vector at the vertices = Vector normal to the plane

$$\mathbf{N} = \mathbf{AB} \times \mathbf{AC} = (-10\mathbf{i} + 10\mathbf{j} + 0\mathbf{k}) \times (-10\mathbf{i} + 0\mathbf{j} + 10\mathbf{k})$$
$$= 100\mathbf{i} + 100\mathbf{j} + 100\mathbf{k}.$$

And the corresponding unit normal vector $\mathbf{n} = \dfrac{\mathbf{N}}{|\mathbf{N}|}$

$$\mathbf{n} = 0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k}$$

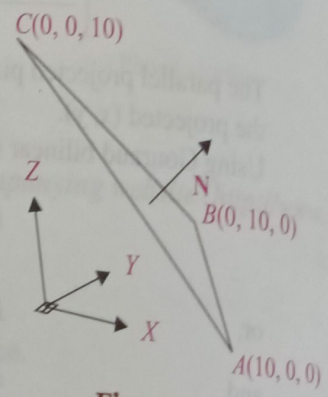


Fig. 13.11

The light vectors from the vertices to the point light source is

$$\mathbf{AP} = -10\mathbf{i} + 0\mathbf{j} + 20\mathbf{k}$$
$$\mathbf{BP} = 0\mathbf{i} - 10\mathbf{j} + 20\mathbf{k}$$
$$\mathbf{CP} = 0\mathbf{i} + 0\mathbf{j} + 10\mathbf{k}$$

Thus, the unit light vectors at the vertices are

$$\mathbf{L}_A = -0.4772\mathbf{i} + 0\mathbf{j} + 0.8944\mathbf{k}$$
$$\mathbf{L}_B = 0\mathbf{i} - 0.4772\mathbf{j} + 0.8944\mathbf{k}$$
$$\mathbf{L}_C = 0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}$$

Applying the illumination model, intensities at any point $I = k_a I_a + k_d I_l (\mathbf{L} \cdot \mathbf{n})$

$$I_A = 0.3 \times 1 + 0.3 \times 10 \times (\mathbf{L}_A \cdot \mathbf{n})$$
$$= 0.3 + 3 \times (-0.4772\mathbf{i} + 0\mathbf{j} + 0.8944\mathbf{k}) \cdot (0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k})$$
$$= 0.3 + 3 \times 0.2408 = 1.0224$$

$$I_B = 0.3 \times 1 + 0.3 \times 10 \times (\mathbf{L}_B \cdot \mathbf{n})$$
$$= 0.3 + 3 \times (0\mathbf{i} - 0.4772\mathbf{j} + 0.8944\mathbf{k}) \cdot (0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k})$$
$$= 0.3 + 3 \times 0.2408 = 1.0224$$

CHAPTER 13

$$I_C = 0.3 \times 1 + 0.3 \times 10 \times (\mathbf{L}_C \cdot \mathbf{n})$$
$$= 0.3 + 3 \times (0\mathbf{i} + 0\mathbf{j} + 1\mathbf{k}) \cdot (0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k})$$
$$= 0.3 + 3 \times 0.5773 = 2.0319$$

13.4 In Program 13.3, calculate the intensity at the centroid of the triangle using the Gouraud bilinear intensity interpolation technique, taking the *xy* plane as the screen.

Solution

The parallel projected triangle on the screen is **A**(10, 0), **B**(0, 10) and **C**(0, 0) with intensities 1.0224, 1.0224 and 2.0319 respectively.

The centroid of the triangle is $\left(\frac{10+0+0}{3}, \frac{0+10+0}{3}, \frac{0+0+10}{3}\right) = (3.3333, 3.3333, 3.3333)$

Hence the projected centroid of the triangle is (3.3333, 3.3333)

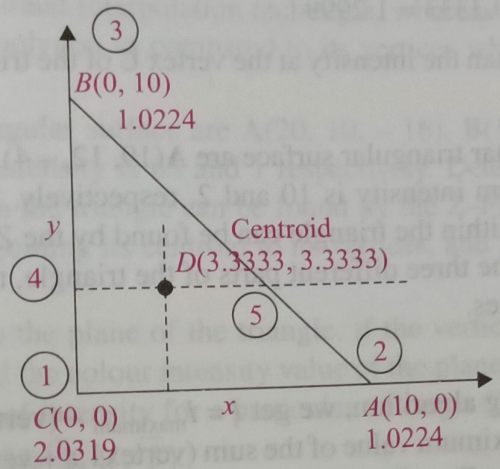


Fig. 13.12 *Projected triangle to xy plane as given in Fig. 13.12*

$$I_4 = \frac{y_3 - y_4}{y_3 - y_1} I_1 + \frac{y_4 - y_1}{y_3 - y_1} I_3 = \frac{10 - 3.3333}{10 - 0} 2.0319 + \frac{3.3333 - 0}{10 - 0} 1.0224 = 1.6954$$

$$I_5 = \frac{y_3 - y_5}{y_3 - y_2} I_2 + \frac{y_5 - y_2}{y_3 - y_2} I_3 = \frac{10 - 3.3333}{10 - 0} 1.0224 + \frac{3.3333 - 0}{10 - 0} 1.0224 = 1.0224$$

(I_5 can be directly estimated as there is no variation on intensity across the line 2 – 3).

$x_4 = 0$ and $x_5 = 10 + (3.3333 - 0)/(-1) = 6.6667$

Interpolating intensity across the scan line $y = 3.3333$, for intensity at the centroid

$$I_D = \frac{x_5 - x_D}{x_5 - x_4} I_4 + \frac{x_D - x_4}{x_5 - x_4} I_5 = \frac{6.6667 - 3.3333}{6.6667 - 0} 1.6954 + \frac{3.3333 - 0}{6.6667 - 0} 1.0224 = 1.3589$$

or, I_D = Intensity at the centroid = 1.3589

Note: Intensity at the centroid is the mean of intensities at the vertices.

13.5 In Program 13.3, interpolate the intensity at the centroid of the triangle using the Phong model.

Solution

As the surface is flat, all the unit normal over the surface is the same as $\mathbf{n} = 0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k}$

CHAPTER 13

(There is no need to interpolate the normal using

$$\mathbf{N}_4 = \frac{y - y_2}{y_1 - y_2}\mathbf{N}_1 + \frac{y_1 - y}{y_1 - y_2}\mathbf{N}_2, \quad \mathbf{N}_5 = \frac{y - y_3}{y_1 - y_3}\mathbf{N}_1 + \frac{y_1 - y}{y_1 - y_3}\mathbf{N}_3$$

and $\mathbf{N} = \frac{x \quad x_5}{x_4 \quad x_5}\mathbf{N}_4 + \frac{x_4 \quad x}{x_4 \quad x_5}\mathbf{N}_5$. It can easily be accessed by observation.)

The light vector at the centroid is the light ray passing through centroid and the point light source. (Refer Fig. 13.12)

$$\mathbf{DP} = (0 - 3.3333)\mathbf{i} + (0 - 3.3333)\mathbf{j} + (20 - 3.3333)\mathbf{k} = -3.3333\mathbf{i} - 3.3333\mathbf{j} + 16.6667\mathbf{k}$$

Thus, the unit light vector is $\mathbf{L}_D = -0.1924\mathbf{i} - 0.1924\mathbf{j} + 0.9622\mathbf{k}$

And the intensity at centroid is $I_D = 0.3 \times 1 + 0.3 \times 10 \times (\mathbf{L}_D \cdot \mathbf{n})$

or, $I_D = 0.3 \times 1 + 0.3 \times 10 \times (-0.1924\mathbf{i} - 0.1924\mathbf{j} + 0.9622\mathbf{k}) \cdot (0.5773\mathbf{i} + 0.5773\mathbf{j} + 0.5773\mathbf{k})$

$= 0.3 + 3 \times 0.3333 = 1.2999$

Note: This is greater than the intensity at the vertex C of the triangle, which is not possible in Gouraud interpolation.

CHAPTER 13

13.6 Three vertices of a planar triangular surface are **A**(19, 12, – 4), **B**(18, 16, – 1) and **C**(26, 11, – 1). The maximum and minimum intensity is 10 and 2, respectively. Determine the constant 'a' so that the intensity at any point within the triangle can be found by the Z–Flat shading method.
Find the intensities in the three different parts of the triangle, the three parts created by a line joining the centroid to its vertices.

Solution

Using the Z–Flat shading algorithm, we get $I = I_{\text{maximum}} - (\text{vertex1}.z + \text{vertex2}.z + \text{vertex3}.z)/a$
I is minimum for the maximum value of the sum (vertex1.z + vertex2.z + vertex3.z). For the subdivided triangle, the centroid is at $\mathbf{D} = (\mathbf{P}_A + \mathbf{P}_B + \mathbf{P}_C)/3 = (21, 13, -2)$.

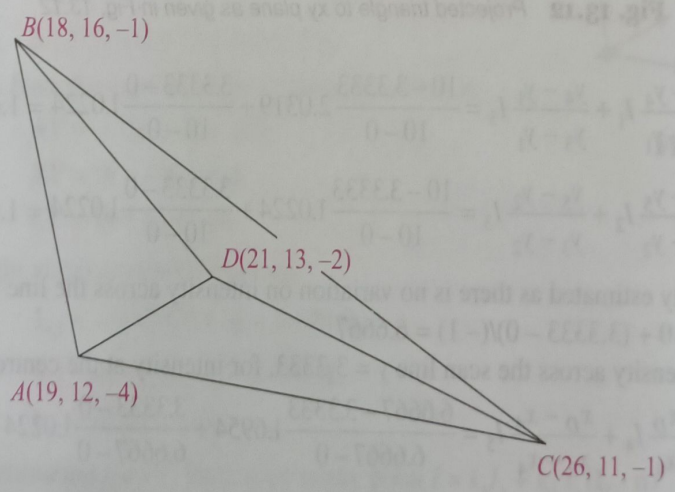


Fig. 13.13

In Fig. 13.13, triangles ABD and ADC fall under maximum depth and it should have the minimum intensity value.

Hence, $I_{\text{minimum}} = I_{\text{maximum}} - (A.z + B.z + D.z)/a$

or, $2 = 10 - (4 + 2 + 1)/a$

or, $7/a = 8$

$$a = 7/8 = 0.875.$$

or,

$$I_{ABD} = I_{minimum} = 2$$

Hence,

$$I_{ADC} = 10 - (4 + 2 + 1)/0.875 = 2$$

$$I_{CBD} = 10 - (2 + 1 + 1)/0.875 = 5.4286$$

Review Questions

13.1 Write a C program to interpolate the colour intensity at any point within the polygon having different colour intensities at the vertices, using the Phong interpolation technique.

13.2 Why is that the intensity within a polygon can never be more or less than the intensity at its vertices when interpolated with the Gouraud interpolation technique, whereas it is possible to have more or less intensity value within the polygon as compared to its vertices when it is interpolated using the Phong model?

13.3 Three vertices of a planar triangular surface are **A**(20, 10, – 16), **B**(16, 15, – 8) and **C**(21, 11, 0). The maximum and minimum intensity is 63 and 1 respectively. Determine the constant 'a' so that the intensity at any point within the triangle can be found by the Z–Flat shading method. Subdivide the triangle into three parts by joining its centroid to its vertices, and find the intensities of the three subparts.

13.4 Find the unit vectors normal to the plane of the triangle, if the vertices of a triangle are **A**(1, 1, 1), **B**(7, 8, 5) and **C**(12, 2, 10). Find the colour intensity value of the plane using the Lambert flat shading method, if the maximum value of intensity for a particular colour is 63? Assume the unit light vector out of the plane to be $(-1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$.

13.5 Calculate the pixel colour value at the centroid *P* of the triangle whose vertices are **A**(0, 3, 5), **B**(6, 10, 23) and **C**(8, 2, 15). The colour values at **A**, **B** and **C** are 0, 12 and 63, respectively. Use the Gouraud technique for interpolation.

13.6 The position vectors for the vertices of a triangular surface are given by **A**(10, 0, 0), **B**(0, 10, 0) and C(0, 0, 10). The normal vector at the vertex **A** is 10**i** + 11**j** + 11**k**, **B** is 11**i** + 10**j** + 11**k** and **C** is 11**i** + 11**j** + 10**k**. The source for a parallel beam of light is given by **L** = – 0.1924**i** – 0.1924**j** + 0.9622**k**. Find the intensity at the parallel projected point (3, 3) within the projected triangle on the *xy* plane of the screen using the Gouraud interpolation technique. The ambient light intensity is 1 and the directional light intensity is 10. Assume $k_a = 0.5$ and $k_d = 0.3$. Neglect any intensity attenuation and specular effect.

13.7 In the above problem use the Phong intensity interpolation technique to find the intensity at projected point (3, 3).

13.8 A sphere is kept at the centre of a space having dimension 640 × 480 × 800. Device a pseudo code to calculate the intensity at every point within the screen space, if a parallel beam of light of intensity 63 is directed along the line joining point *P*(640, 0, 0) to *Q*(0, 480, 800). Assume $k_a = 0.3$ and $k_d = 0.5$.

13.9 Divide the visible surface of the sphere in Program 13.8 with 16 latitudes and 16 longitudes. Now develop a routine to shade the visible surface of the sphere using the Lambert flat shading method.

13.10 Modify the routine generated in Program 13.9 to shade using the Gouraud shading method.

A wide variety of natural objects are modelled by several advanced techniques – procedure, fractals and grammar based modelling. These are discussed in subsequent sections.

14.5 PROCEDURAL MODELS

This type of model is made by following a procedure based on entity representation or by interacting with external events. Interaction is required to modify the models on the coordinate space available. Examples are modelling of spheres, bridges, roads in a countryside, etc. A sphere is modelled by taking a polygon as the entity. The model of a sphere is generated by polygon representation at a requested fitness of subdivision. However, a simple collection of polygons specified by vertices will not result in a sphere and hence does not yield to a procedural model. At the same time a model of a sphere at a desired level of fineness of subdivision is procedural. Therefore, the model which comes into existence is actually determined by the fineness parameter. A model that determines the origin of its coordinate system by requesting information from nearby entities is also procedural.

The important aspect of procedural modelling is interaction with the environment by inter-object communication and inter-object checking. The shape of the object can be controlled by inter-object communication. Suppose a road has to be passed through a wooded terrain. The terrain may be generated by a number of triangles using the fractal method. The trees are generated by grammar-based modelling. The roads are generated by extrusion of a line along a spline path. These objects are to be modelled together by controlling each other such that the base of the trees is placed on the terrain, but not too close to the road. Inter-object control is done by checking the progress of objects against each other in the course of procedure modelling.

The other models discussed in the chapter are similar in some aspects. They are generated by separation of an entity considered to be a subdivision. The subdivisions are terminated based on the models requirement or on amplification of the modellers potential. A tiny change in the manner specifying the entity will lead to a drastic change of form.

14.6 FRACTALS

A natural object is often made of a patterned structure that repeats itself in the form of jagged rough and random edges. Drawing natural objects such as a mountain, tree, river or lighting bolts with lines or polygon requires a lot of specifications. If one were to follow the specifications manually, it would be almost impossible to create even a single natural object as mentioned above. Mathematicians considered these types of problems during the period from 1875 to 1925. They found that the basic elements of the picture are a collection of points that define the geometry of a figure which has zero dimension. The line required to represent this figure is obtained by moving a point which then becomes one dimensional. Further, a square is generated by moving a line at right angle which is considered two dimensional. The 3D figure of a cube is obtained by moving a square.

The question that arises at this point is that if a cube is moved, what happens then? Perhaps, a four-dimensional figure is obtained. Mathematically this can be understood, but what about its physical interpretation in the context of the real world. The second situation arises when a line is drawn continuously and repeatedly to fill a surface (plane). The result appears to be two dimensional whereas a line is one dimensional. However, intuitively one feels that the line must now be two dimensional. Further, a relevant question arises: when did the transition from one dimension to two dimension take place? Probably all these can be answered based on the concept founded by mathematician Felix Hansdroff who conceived of the idea of fractional dimensions. According to Hansdroff, the description of the nature of geometric figures cannot be provided by integer dimensions. On the contrary, a whole spectrum of fractional dimensions is noninteger. For example, the dimension of a particular line may be 1.58496. Hansdroff has given criteria to determine

the dimensional specification of various geometric figures. However, here it is more important to understand the nature of dimension itself. Creating a natural figure requires iterative computation for some time, many thousands of them to be reproduced, which is very tedious when done manually or, rather, impossible. With the advent of digital computer, the subject matter has been brought into focus after many years. This is reflected in Mandelbrot's popular investigations and display of his graphics work of natural figures like coast lines, ferns, etc. which cannot be generated by the Euclidean method. The various pattern that repeat themselves in an object are called fractals.

In current days fractals are used in engineering simulation, calculation and prediction of some complex natural phenomena, generating graphics in video games, astronomical visualizations, flight simulators, naval simulators, military application simulators, television advertisements, and animations. Engineers have begun designing and constructing fractals in order to solve some real world problems like that of fluid mechanics. It is widely used in aerospace, space and ocean sciences.

14.6.1 Basic Characteristics

1. Fractals have infinite details at every region, which can be seen on successive magnification.
2. There is self similarity between the object parts and the overall features of the parent object. Here, similar refers to something very specific. Geometric figures are similar if they have proportional dimensions. For one figure to be similar to another, you must be able to magnify the length of the small figure by the scale factor, and it will be exactly the same size as the larger figure. Now what is self similar? There are many figures which are not fractals but they are self similar, like the ones shown in Fig. 14.6. Each successive small part divides itself and is similar to the main figure.
3.

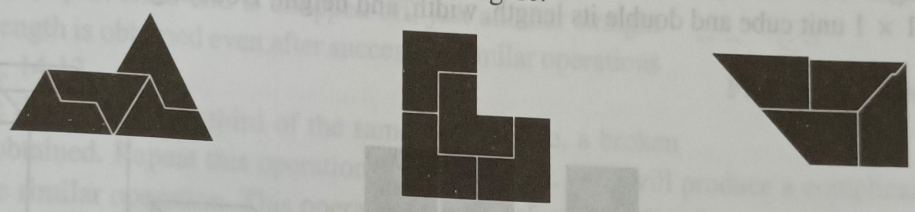
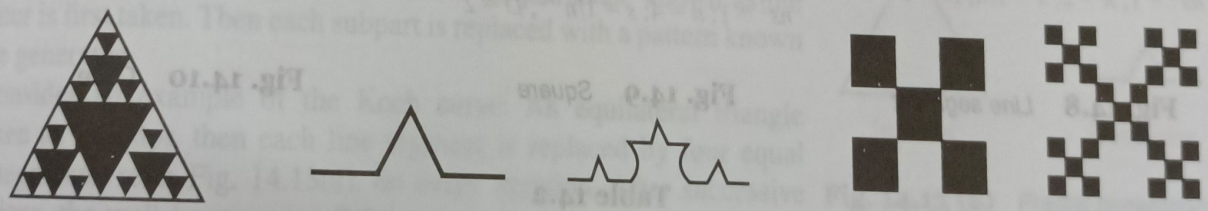


Fig. 14.6 *Self similarity*

Now what is self similarity in fractals? Consider some of the figures as shown below in Fig. 14.7 which are well known fractals.



Sierpinski triangle

Koch curve

Fig. 14.7 *Fractals*

Notice that there are infinitely many smaller and smaller subfigures inside. How many different sized subfigures can one find? All of these are similar to each other and to the original figure in a somewhat technical sense on all scales. The objects need not exhibit exactly the same structure at all scales, but the same "type" of structure must appear on all scales. In Euclidean shape, no matter how complicated it is, it can be zoomed in to smooth out.

A prototype for a fractal may be the length of a coastline measured with different length rulers. The shorter the ruler, the longer the length measured, a paradox known as the coastline paradox. One can measure it with kilometer-long rulers and get a certain measurement. What if on the next day it is measured with meter-long

rulers? Which measurement would give you a larger measurement? Since the coastline is jagged, one could get into the nooks better with the meter-long ruler, so it would yield a greater measurement. Now what if one measured it with a centimeter-long ruler? One could really get into the tiniest and tiniest of nooks there. So the measurement would be even bigger, that is if the coastline is jagged smaller than a centimeter. What if it were jagged at every point on the coastline? One could measure it with shorter and shorter rulers, and the measurement would get longer and longer. You could even measure it with infinitesimally short rulers, and the coastline would be infinitely long. That's what a fractal is!

14.6.2 Fractal Geometry and Dimensions

One of the most important properties of any fractal is fractal dimension.

In Euclidean dimensions, a point has no dimensions: no length, no width, no height. A line has one dimension – length. It has no width and no height, but infinite length. Similarly a plane has two dimensions – length and width, no depth. It extends both ways to infinity. Again a box has three dimensions: length, width, and depth, extending to infinity in all three directions.

Fractals can have fractional (or fractal) dimension. A fractal might have dimension of 1.6 or 2.4. The explanation for this is given below.

Take a self-similar figure like a line segment, as shown in Fig. 14.8 below, and double its length. Doubling the length gives two copies of the original segment.

Take another self-similar figure of a square. This time the square is 1 unit by 1 unit dimension. Now, multiply the length and width by 2. Doubling the sides gives four copies (Fig. 14.9).

Take a 1 × 1 × 1 unit cube and double its length, width, and height. Doubling the side gives eight copies Fig. 14.10.

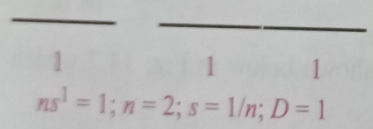


Fig. 14.8 *Line segment*

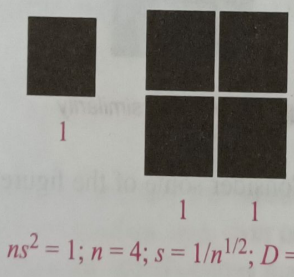


Fig. 14.9 *Square*

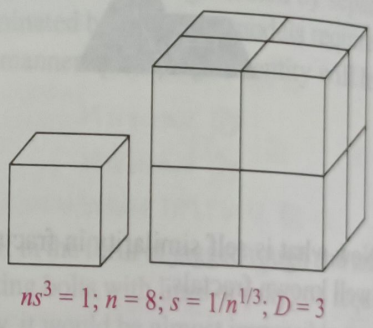


Fig. 14.10 *Cube*

Table 14.2

Figure	*Dimension*	*No. of Copies*
Line Segment	1	$2 = 2^1$
Square	2	$4 = 2^2$
Cube	3	$8 = 2^3$
Doubling	D	$n = 2^d$
Similarity		

For each of these objects the relation between the number of subparts n and the scaling factor s and dimension D is $ns^D = 1$. This D in fractal geometry is called *fractional dimension*.

Thus, fractal similarity dimension can be expressed as

$$D = \frac{\ln n}{\ln (1/s)}$$

For a self similar fractal constructed with different scaling factors for different parts,

$$\sum_{k=1}^{n} s_k^D = 1$$

A plot of this quantity on a log–log graph versus scale gives a straight line, whose slope is said to be the fractal dimension D.

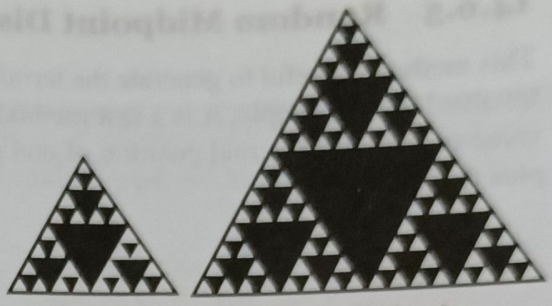


Fig. 14.11 *Sierpinski triangle*

Take a simple fractal, the Sierpinski triangle of 1 unit sides. Double the length of the sides gives us three copies, so $3 = 2^d$, where d = the dimension. Note $2 = 2^1$, and $4 = 2^2$, so what number could this be? It has to be somewhere between 1 and 2, which is a fraction, i.e. 1.58496.

14.6.3 Fractal Generation Procedures

Fractals are often formed by an iterative process by repeatedly applying a specific transformation function to points within the region of space. If $P_o = (x_o, y_o, z_o)$ is a selected initial point, and F is the transformation function, then $P_{n+1} = F(P_n)$.

All iterative processes are not fractals. For example, if a straight line is taken and a third part from end is chopped off, just another straight line of smaller length is obtained even after successive similar operations as shown in Fig. 14.12.

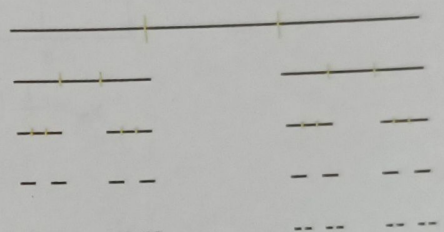


Fig. 14.12 *Simplest fractal*

Now remove the middle one third of the same straight line, a broken straight line is obtained. Repeat this operation to each subpart. This will produce a complicated figure after every successive similar operation. This operation when performed infinitely many times generates a fractal called Cantor dust.

14.6.4 Geometric Construction Methods

To construct a fractal using this method, a simple shape known as the initiator is first taken. Then each subpart is replaced with a pattern known as the generator.

Consider an example of the Koch curve. An equilateral triangle is taken as initiator, then each line segment is replaced by four equal line segments, refer Fig. 14.13(a), on every iteration. After successive iterations, the well-known snowflake pattern is generated, as shown in Fig. 14.13(b).

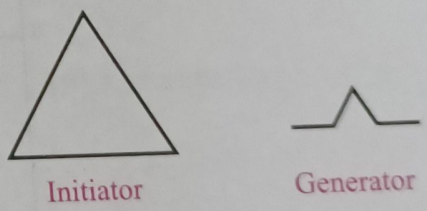


Fig. 14.13 (a) *Fractal generation method*

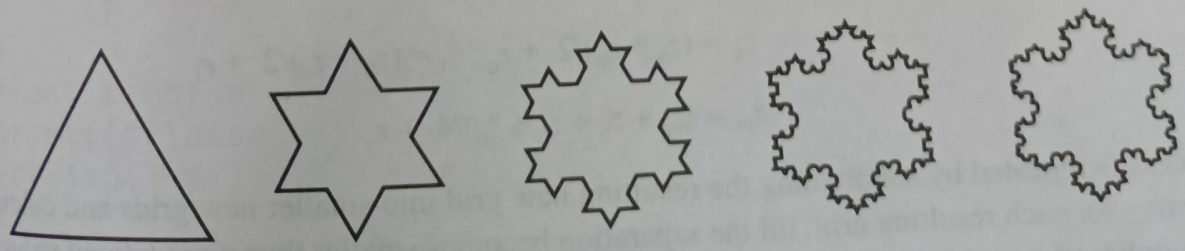


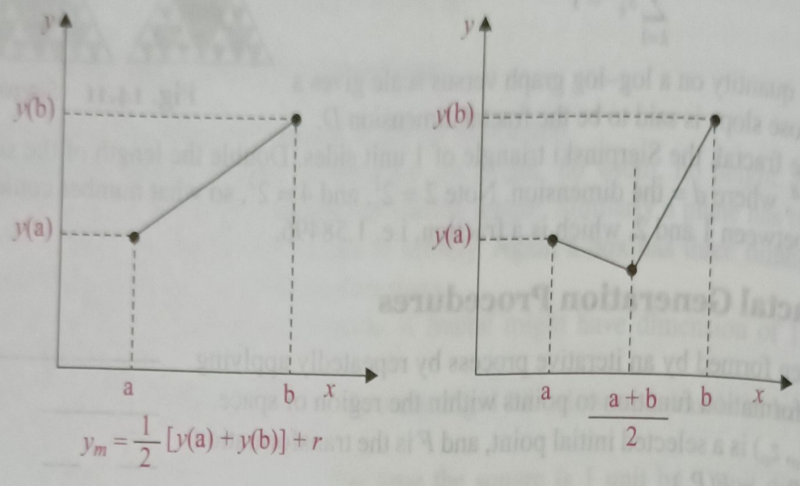
Fig. 14.13 (b) *Generation of a snowflake*

Here, the scaling factor $s = 1/3$ and $n = 4$, so fractal dimension $D = \ln 4/\ln 3 = 1.26185$.

CHAPTER 14

14.6.5 Random Midpoint Displacement Method

This method is useful to generate the terrain used in video games and science fiction films. As the calculation involved is very simple, it is a fast method. A straight line segment is taken into consideration. A displaced value of the y for the mid position of end points along the x axis is the average value of y at the endpoints plus the random offset.



$$y_m = \frac{1}{2}[y(a) + y(b)] + r$$

Fig. 14.14

To give it a realistic look $r = sg\,|b - a|$, where s = *surface roughness factor* and g = *Gaussian random value* with mean 0 and variance 1. At each step, the random displacement decreases, as it is proportional to the distance $|b - a|$. Terrain is created by subdividing the area into a grid and elevation is generated using the midpoint displacement method along both the horizontal axes. For faster generation 'g' can be obtained from a lookup table.

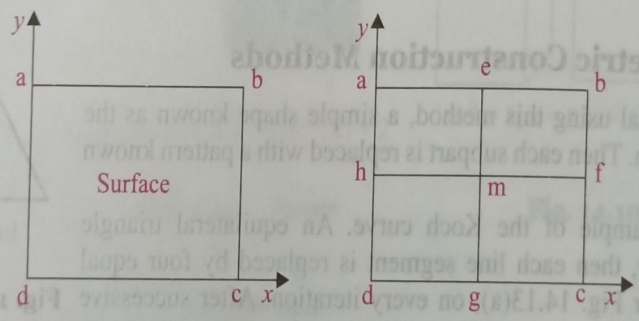


Fig. 14.15

$$z_e = (z_a + z_b)/2 + r_e,\ z_f = (z_b + z_c)/2 + r_f$$

and

$$z_m = (z_a + z_b + z_c + z_d)/4 + r_e$$

This process is repeated by subdividing the resulting new grid into smaller new grids and calculating the new elevations z for each resulting grid, till the separation becomes smaller than a predefined value or after a predefined number of iterations, starting from a defined segment length.