Unit - VI Virtual Reality

Introduction of Virtual Reality : Fundamental Concept, Three I's of virtual reality and Classic Components of VR systems, Applications of VR systems.

Multiple Modals of Input and Output Interface in Virtual Reality : Input - 3D position Trackers and its types, Navigation and Manipulation Interfaces, Gesture Interfaces, Graphics Displays - HMD and CAVE, Sound Displays, Haptic Feedback.

Rendering Pipeline : Graphics rendering Pipeline, Haptics Rendering Pipeline Modeling in Virtual Reality : Concepts of Geometric Modeling, Kinematic Modeling, Physical modeling and Behavior modeling. **(Chapters - 13, 14, 15)**

TABLE OF CONTENTS

Unit - III

Unit - IV

Unit - III

6 WINDOWING AND CLIPPING

6.1 : Concept of Window and Viewport, Viewing Transformations

Q.1 What is windowing and clipping ?

☞[SPPU : May-13, June-22,Dec.-05,12,13, Marks 4]

Ans. : • The process of selecting and viewing the picture with different views is called **windowing,** and a process which divides each element of the picture into its visible and invisible portions, allowing the invisible portion to be discarded is called **clipping.**

Q.2 Describe viewing transformation.

☞[SPPU : Dec.-10, 11, May-12, Marks 8]

OR Explain window to view port transformation.

☞[SPPU : Dec.-10, 11,19, May-12,18,19, June-22, Marks 8]

Ans. : • The picture is stored in the computer memory using any convenient Cartesian co-ordinate system, referred to as **World Co-ordinate System** (WCS).

- When picture is displayed on the display device it is measured in **Physical Device Co-ordinate System** (PDCS) corresponding to the display device. Therefore, displaying an image of a picture involves mapping the co-ordinates of the points and lines that form the picture into the appropriate physical device co-ordinate where the image is to be displayed. This mapping of co-ordinates is achieved with the use of co-ordinate transformation known as **viewing transformation.**
- Sometimes the two dimensional viewing transformation is simply referred to as the **window to view port transformation** or the **windowing transformation.**

- The viewing transformation which maps picture co-ordinates in the WCS to display co-ordinates in PDCS is performed by the following transformations.
  - Converting world co-ordinates to viewing co-ordinates.
  - Normalizing viewing co-ordinates.
  - Converting normalized viewing coordinates to device co-ordinates

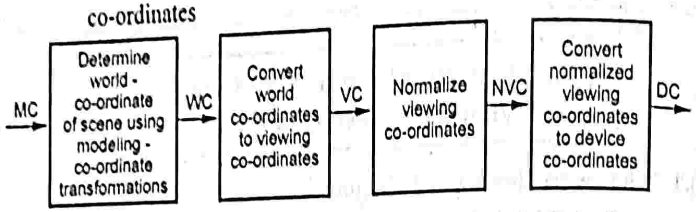


Fig. Q.2.1 Two-dimensional viewing transformation pipeline

- World Co-ordinate System (WCS) is infinite in extent and the device display area is finite.
- To perform a viewing transformation we select a finite world co-ordinate area for display called a **window**.
- An area on a device to which a window is mapped is called a **viewport**.

6.2 : Line Clipping

Q.3 What is line clipping ? **[SPPU : May-12, Marks 2]**

Ans. : • The lines are said to be interior to the clipping window and hence visible if both end points are interior to the window.

- If both end points of a line are exterior to the window, the line is not necessarily completely exterior to the window.
- If both end points of a line are completely to the right of, completely to the left of, completely above, or completely below the window, then the line is completely exterior to the window and hence invisible.
- The lines which across one or more clipping boundaries require calculation of multiple intersection points to decide the visible portion of them.
- Thus deciding visible portion of the line is known as line clipping.

Q.4 Explain Cohen-Sutherland algorithm with the help of suitable example. **[SPPU : May-06,07,08,10,11,12,13,15,16,18, Dec.-10,13,15,16,18,19,22, Marks 8]**

Ans. :

1. Read two end points of the line say P_1 (x_1, y_1) and P_2 (x_2, y_2).
2. Read two corners (left-top and right-bottom) of the window, say $(Wx_1, Wy_1$ and $Wx_2, Wy_2)$.
3. Assign the region codes for two endpoints P_1 and P_2, using following steps :

   Initialize code with bits 0000

   Set Bit 1 - if $(x < Wx_1)$

   Set Bit 2 - if $(x > Wx_2)$

   Set Bit 3 - if $(y < Wy_2)$

   Set Bit 4 - if $(y > Wy_1)$

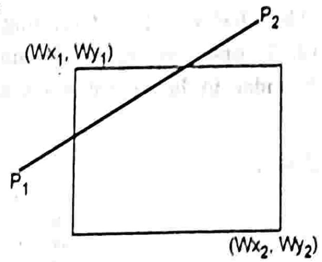


Fig. Q.4.1

4. Check for visibility of line P_1 P_2
   a) If region codes for both endpoints P_1 and P_2 are zero then the line is completely visible. Hence draw the line and go to step 9.
   b) If region codes for endpoints are not zero and the logical ANDing of them is also non-zero then the line is completely invisible, so reject the line and go to step 9.
   c) If region codes for two endpoints do not satisfy the conditions in 4a) and 4b)the line is partially visible.
5. Determine the intersecting edge of the clipping window by inspecting the region codes of two endpoints.
   a) If region codes for both the end points are non-zero, find intersection points P_1' and P_2' with boundary edges of clipping window with respect to point P_1 and point P_2, respectively
   b) If region code for any one end point is non-zero then find intersection point P_1' or P_2' with the boundary edge of the clipping window with respect to it.

6. Divide the line segments considering intersection points.
7. Reject the line segment if any one end point of it appears outsides the clipping window.
8. Draw the remaining line segments.
9. Stop.

Q.5 What are the intersecting point for line P_1 joining (– 1, 0) and (4, 5) and line P_2 (3, 1) and (6, 2) if clipped against a window bounded by line x = 0, y = 0 and x = 5, y = 3.

☞[SPPU : May-14, Marks 3]

Ans. :

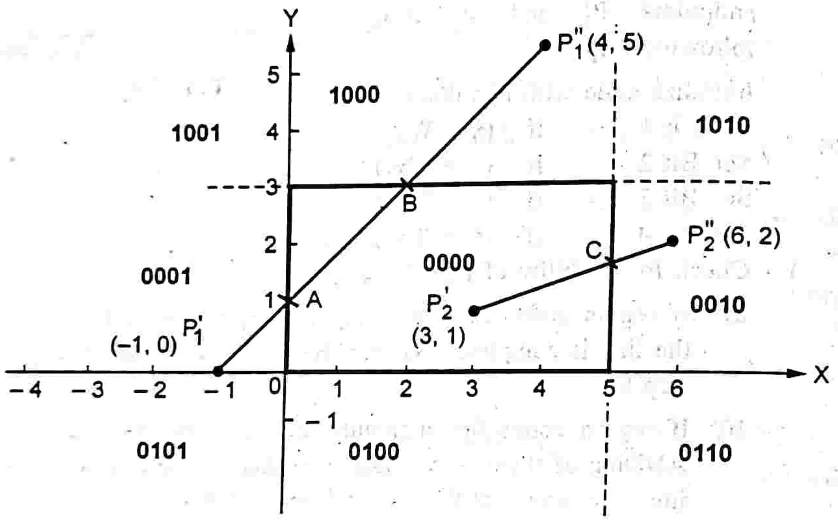


Fig. Q.5.1

1) For line P_1

Region of code for point $P_1' = 0\ 0\ 0\ 1$

Region of code for point $P_1'' = 1\ 0\ 0\ 0$

Since (0 0 0 1) AND (1 0 0 0) = 0 0 0 0, line P_1' P_1'' is partially visible.

Stage of clipping line segment P_1' P_1'' are :

1. $x_{min} = 0$ for window boundary line

2. $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5-0}{4-(-1)} = 1$

3. The intersection points are

$$A = x_{min}, m(x_{min} - x_1) + y_1$$
$$= 0, 1(0 - (-1)) + 0$$
$$A = (0, 1)$$
$$B = x_1 + \frac{1}{m}(y_{max} - y_1), y_{max}$$
$$= (-1) + \frac{1}{1}(3-0), 3$$
$$B = (2, 3)$$

2) For line P_2

Region of code for point $P_2' = 0\ 0\ 0\ 0$

Region of code for point $P_2'' = 0\ 0\ 1\ 0$

Since (0 0 0 0) AND (0 0 1 0) = 0 0 0 0, line P_2' P_2'' is partially visible.

Stages of clipping line segment P_2' P_2'' are :

1. $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-1}{6-3} = \frac{1}{3}$

2. The intersection point is

$$C = x_{max}, y_1 + m \times (x_{max} - x_1) = 5, 1 + \frac{1}{3} \times (5 - 3)$$
$$C = (5, 1.667)$$

Q.6 Use the Cohen-Sutherland algorithm for clipping window having clipping window whose lower left point at (2, 1), upper right point at (7, 5) and line points are (1, 3) and (5, 6). Find the intersection points. **☞[SPPU : June-22, Marks 6]**

Ans. : Line : A(1,3), B(5,6)

Window : $X_L = 2$, $Y_B = 1$, $X_R = 7$ and $Y_T = 5$

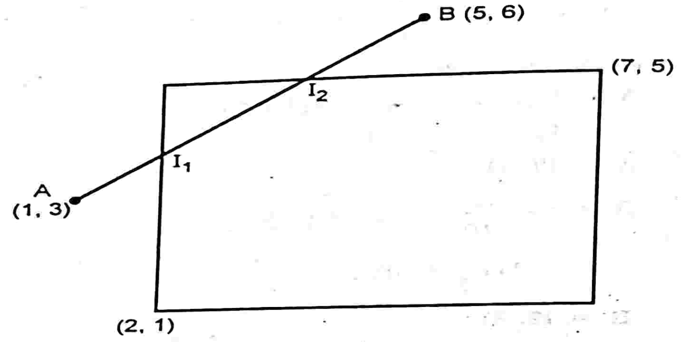


Fig. Q.6.1

Point	End Code	ANDing	Position
A	0001		
B	1000	0000	Partially visible

Lime slope m $= \frac{6-3}{5-1} = \frac{3}{4}$

$$I_1 = X_L,\ m(X_L - X_1) + Y_1 = 2,\ \frac{3}{4}[2-1]+3$$

$$= 2,\ \frac{15}{4}$$

$$I_2 = X_1 + \left(\frac{1}{m}\right)(Y_T - Y_1),\ Y_T$$

$$= 1 + \frac{4}{3}(5-3),\ 5$$

$$= \frac{11}{3},\ 5$$

Q.7 Let ABCD be the rectangle window with A(10, 20), B(100, 20), C(100, 90), D(10, 90). Find the region codes for endpoints and use Cohen Sutherland algorithm to clip the lines $P_1 - P_2$ with P_1 (5, 30) and P_2 (70, 110) and $Q_1 - Q_2$ with Q_1 (50, 70) and Q_2 (80, 30).

☞ **[SPPU : Dec.-18, Marks 6]**

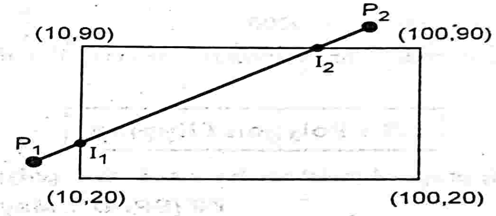
Ans. : **Line 1 :** P_1 (5, 30), P_2 = (70, 110),

x_L = 10 and y_B = 20

∴ x_R = 100, y_T = 90

Region code for point P_1 = 0001

Region code for point P_2 = 1000



$$m = \frac{90-20}{100-10} = \frac{7}{9}$$

Since

(0001) AND (1000) = 0000,

line P_1P_2 is partially visible.

$x_L,\ y = m(x_L - x_1) + y_1$

$$= \frac{7}{9}(10-5)+30 = 33.89$$

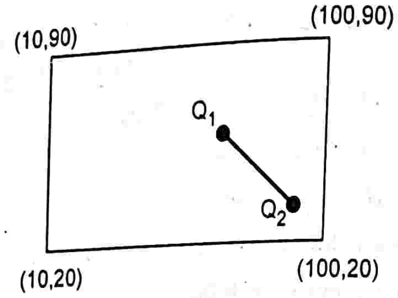
$y_T,\ x = x_1 + \left(\frac{1}{m}\right)(y_T - y_1)$

$$= 5 + \left(\frac{9}{7}\right)(90-30) = 82.14$$

∴ I_1 = (10, 33.89), I_2 = (82.14, 90)

Line 2 :

$Q_1(50, 70)$ $Q_2(80, 30)$ $x_L = 10$ $y_B = 20$, $x_R = 100$, $y_T = 90$



Region code for point $Q_1 = 0000$

Region code for point $Q_2 = 0000$

Since the region codes for endpoints are zero, the line is completely visible.

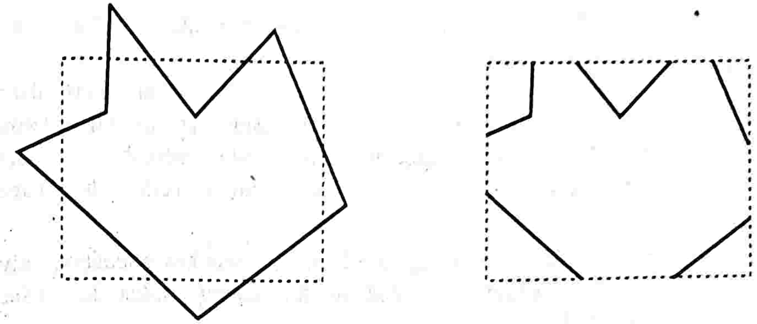
6.3 : Polygon Clipping

Q.8 Can line clipping algorithm be used for polygon clipping ? Justify. **[SPPU : May-10, Marks 8]**

Ans. : • A polygon is nothing but the collection of lines. Therefore, we might think that line clipping algorithm can be used directly for polygon clipping. However, when a closed polygon is clipped as a collection of lines with line clipping algorithm, the original closed polygon becomes one or more open polygon or discrete lines as shown in the Fig. Q.8.1. Thus, we need to modify the line clipping algorithm to clip polygons.

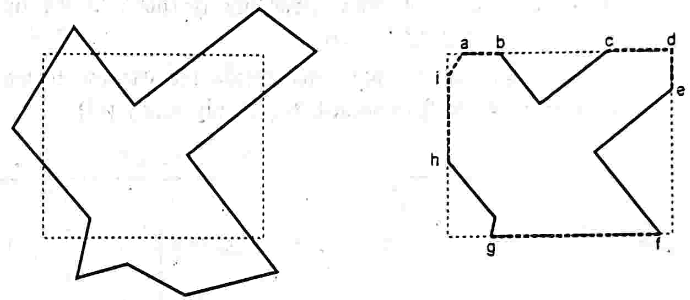
- We consider a polygon as a closed solid area. Hence after clipping it should remain closed. To achieve this we require an algorithm that will generate additional line segment which make the polygon as a closed area.
- For example, in Fig. Q.8.2 the lines a - b, c - d, d - e, f - g, and h - i are added to polygon description to make it closed.
- Adding lines c - d and d - e is particularly difficult. Considerable difficulty also occurs when clipping a polygon results in several disjoint

smaller polygons as shown in the Fig. Q.8.3. For example, the lines a - b, c - d, d - e and g - f are frequently included in the clipped polygon description which is not desired.



(a) Before clipping (b) After clipping

Fig. Q.8.1 Polygon clipping done by line clipping algorithm



(a) (b)

Fig. Q.8.2 Modifying the line clipping algorithm for polygon

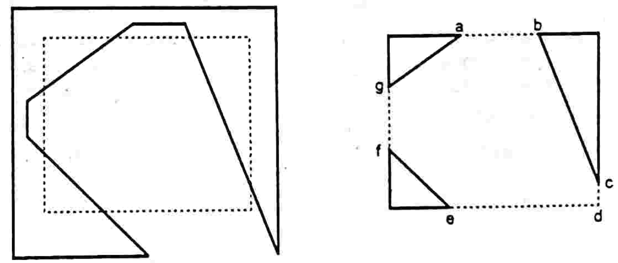
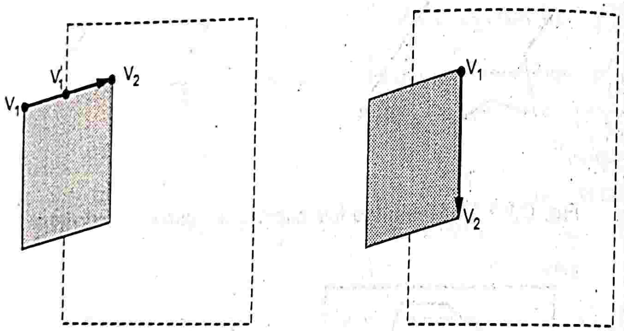


Fig. Q.8.3 Disjoint polygons in polygon clipping

Q.9 State the four possible relationships between the edge and the clipping boundary mentioned in the Sutherland-Hodgeman polygon clipping algorithm.

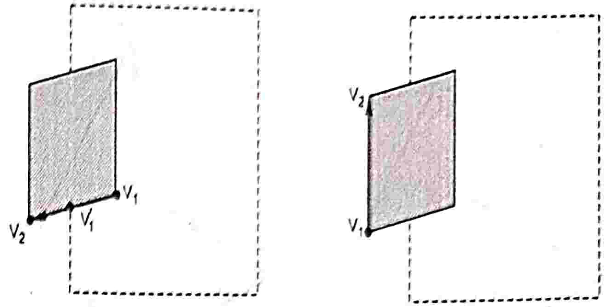
Ans. : • The four possible relationships between the edge and the clipping boundary are as follows :

1. If the first vertex of the edge is outside the window boundary and the second vertex of the edge is inside then the intersection point of the polygon edge with the window boundary and the second vertex are added to the output vertex list (See Fig. Q.9.1 (a)).
2. If both vertices of the edge are inside the window boundary, only the second vertex is added to the output vertex list. (See Fig. 6.8.5 (b)).
3. If the first vertex of the edge is inside the window boundary and the second vertex of the edge is outside, only the edge intersection with the window boundary is added to the output vertex list. (See Fig. Q.9.1 (c)).
4. If both vertices of the edge are outside the window boundary, nothing is added to the output list. (See Fig. Q.9.1 (d)).



V_1 - Outside
V_2 - Inside
$\therefore$ Save V_1' and V_2
(a)

V_1 - Inside
V_2 - Inside
$\therefore$ Save V_2
(b)



V_1 - Inside
V_2 - Outside
$\therefore$ Save V_1'
(c)

V_1 - Outside
V_2 - Outside
$\therefore$ Save nothing
(d)

Fig. Q.9.1 Processing of edges of the polygon against the left window boundary

Q.10 Describe Sutherland-Hodgeman polygon clipping algorithm with example.

☞[SPPU : Dec.-06,07,08,11,14, May-10,14, Marks 8]

Ans. : 1. Read coordinates of all vertices of the polygon.

2. Read coordinates of the clipping window
3. Consider the left edge of the window
4. Compare the vertices of each edge of the polygon, individually with the clipping plane
5. Save the resulting intersections and vertices in the new list of vertices according to four possible relationships between the edge and the clipping boundary.
6. Repeat the steps 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed to process the next edge of the clipping window.
7. Stop.

Let us consider, a polygon and clipping window shown in Fig. Q.10.1 (See Fig. Q.10.1 on next page)

Original polygon vertices are V_1, V_2, V_3, V_4, V_5. After clipping each boundary the new vertices are given in Fig. Q.10.1 (a).

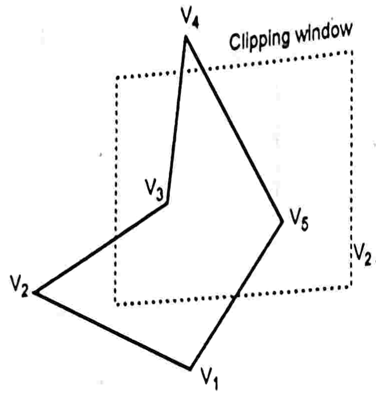


Fig. Q.10.1

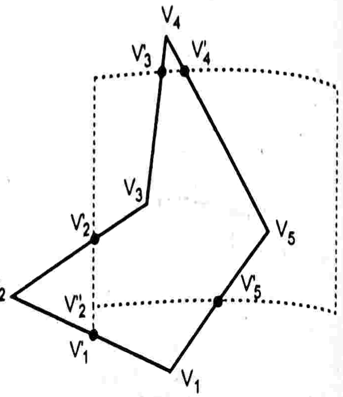


Fig. Q.10.1 (a)

After left clipping : $V_1', V_2', V_3, V_4, V_5$

After right clipping : $V_1', V_2', V_3, V_4, V_5$

After top clipping : $V_1', V_2', V_3, V_3', V_4', V_5$

After bottom clipping : $V_2'', V_2', V_3, V_3', V_4', V_5,$

Q.11 Using Sutherland-Hodgeman method, clip polygon ABCDE against window PQRS. The coordinators of polygon are A(80, 200), B(220, 120), C(150, 100), D(100, 30), E(10, 120). Coordnates of the window are P(200, 50), Q(50, 150), R(200, 150), S(50, 50).

[SPPU : June-22, Marks 6]

Ans. : **Polygon :** A(80, 200), B(220, 120), C(150, 100), D(100, 30), E(10, 120)

Window : P(200, 50), Q(50, 150), R(200, 150), S(50, 50)

$W_L = 50,\ W_R = 200,\ W_T = 150,\ W_B = 50$

After left clipping : $BCDD'E'A$

After right clipping : $B'CDD'E'AB''$

After top clipping : $B'CDD'E''A'B''$

After bottom clipping : $CC'D''D'E''A'B''B'$

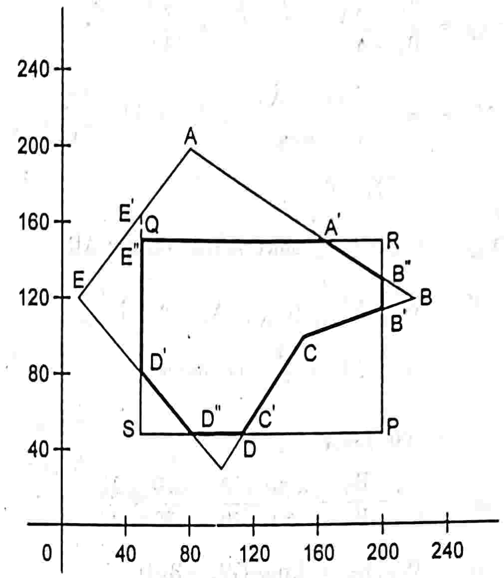


Fig. Q.11.1

$$m_{CD} = \frac{D_Y - C_Y}{D_X - C_X} = \frac{(30-100)}{(100-150)} = \frac{-70}{-50} = \frac{7}{5}$$

$$C' = C_X + \frac{(W_B - C_Y)}{m_{CD}},\ W_B = 150 + \frac{(50-100)}{7/5},\ 50$$

$$= 114.28,\ 50$$

$$m_{DE} = \frac{E_Y - D_Y}{E_X - D_X} = \frac{120-30}{10-100} = \frac{90}{-90} = -1$$

$$D' = W_L,\ D_Y + [m_{DE} \times (W_L - D_X)]$$

$$= 50,\ 30 + [-1 \times (50-100)] = 50,\ 80$$

$$m_{DE} = m_{D'D} = -1 \text{ since } D' \text{ lies on line DE}$$

$$D'' = D_X + \frac{(W_B - D_Y)}{m_{D'D}},\ W_B = 100 + \left(\frac{50-30}{-1}\right),\ 50$$

$$= 80,\ 50$$

$$E'' = 50,\ 150$$

$$m_{AB} = \frac{B_Y - A_Y}{B_X - A_X} = \frac{120-200}{220-80} = \frac{-80}{140} = \frac{-4}{7}$$

$$A' = A_X + \left(\frac{W_T - A_Y}{m_{AB}}\right),\ W_T = 80 + \left(\frac{150-200}{(-4/7)}\right),\ 150$$

$$= 167.5,\ 150$$

$$m_{AB} = m_{A'B} = \frac{-4}{7} \text{ since } A' \text{ lies on line AB}$$

$$B'' = W_R,\ A'_Y + [m_{A'B}(W_R - A'_X)]$$

$$= 200,\ 150 + \left[\frac{-4}{7}(200-167.5)\right]$$

$$= 200,\ 131.43$$

$$m_{BC} = \frac{C_Y - B_Y}{C_X - B_X} = \frac{100-120}{150-220} = \frac{-20}{-70} = \frac{2}{7}$$

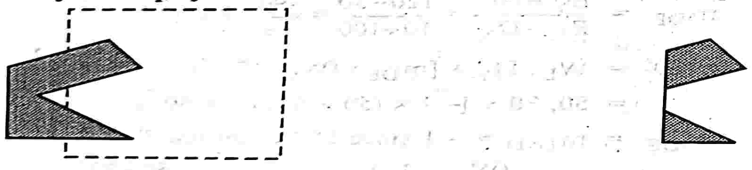
$$B' = W_R,\ B_Y + [m_{BC}(W_R - B_X)]$$

$$= 200,\ 120 + \left[\frac{2}{7}(200-220)\right]$$

$$= 200,\ 114.28$$

Q.12 What is the limitation of Sutherland-Hodgman polygon clipping algorithm ? How can it be resolved ?

Ans. : • The Sutherland-Hodgman polygon clipping algorithm clips convex polygons correctly, but in case of concave polygons, clipped polygon may be displayed with extraneous lines, as shown in Fig. Q.12.1



(a) (b)

Fig. Q.12.1 Clipping the concave polygon in (a) With the Sutherland-Hodgeman algorithm produces the two connected areas in (b)

- The problem of extraneous lines for concave polygons in Fig. Q.12.1.
- Sutherland-Hodgeman polygon clipping algorithm can be solved by separating concave polygon into two or more convex polygons and processing each convex polygon separately.

Q.13 Write a note on concave polygon and its splitting.

☞[SPPU : Winter-18, Marks 4]

Ans. : • For concave polygon-clipping regions, we can apply parametric clipping procedures if we first split the concave polygon into a set of convex polygons. We can identify a concave polygon by calculating the cross products of successive edge vectors in order around the polygon perimeter. If the z component of some cross products is positive while others have a negative z component, we have a concave polygon. Otherwise, the polygon is convex. This is assuming that no series of three successive vertices are collinear, in which case the cross product of the two edge vectors for these vertices is zero.

- A vector method for splitting a concave polygon in the xy plane is to calculate the edge-vector cross products in a counter clockwise order and to note the sign of the z component of the cross products. If any z component turns out to be negative, the polygon is concave and we can split it along the line of the first edge vector in the cross product pair.

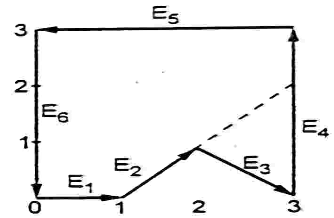


Fig. Q.13.1 Splitting a concave polygon using the vector method

- Fig. Q.13.1 shows a concave polygon with six edges. Edge vectors for this polygon can be expressed as

E1 = (1,0, 0), **E2** = (1,1,0)

E3 = (1, − 1 0), E4 = (0,2,0)

E5 = (− 3,0,0), E6 = (0, − 2,0)

where the z component is 0, since all edges are in the xy plane. The cross product $E_i \times E_j$ for two successive edge vectors is a vector perpendicular to the xy plane, with z component equal to

$E_{ix}E_{jy} - E_{jx}E_{iy}$

$E1 \times E2 = (0,0,1)$, $E2 \times E3 = (0,0,-2)$

$E3 \times E4 = (0,0,2)$, $E4 \times E5 = (0,0,6)$

$E5 \times E6 = (0,0,6)$, $E6 \times E1 = (0,0,2)$

- Since the cross product $E2 \times E3$ has a negative z component, we split the polygon along the line of vector E2. The line equation for this edge has a slope of 1 and a y intercept of -1. We then determine the intersection of this line and the other polygon edges to split the polygon into two pieces. No other edge cross products are negative, so the two new polygons are both convex.

END...

Unit - III

7 3D TRANSFORMATIONS AND PROJECTIONS

7.1 : 3D Translation

Q.1 Obtain the 3D transformation matrices for translation.

[SPPU : Dec.-05,07,08, May-05,07,08,19, June-22, Marks 2]

Ans. : • Three dimensional transformation matrix for translation with homogeneous coordinates is as given below. It specifies three coordinates with their own translation factor.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$\therefore \quad P' = P \cdot T$$

$$\therefore \quad [x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$= [x + t_x \quad y + t_y \quad z + t_z \quad 1] \qquad \text{... (Q.1.1)}$$

7.2 : 3D Scaling

Q.2 Obtain 3D transformation matrix for scaling.

[SPPU : May-05, 07, 08, 16,19, June-22, Dec.-05, 07, 08, 16, Marks 2]

Ans. : • Three dimensional transformation matrix for scaling with homogeneous co-ordinates is as given below.

- It specifies three co-ordinates with their own scaling factor.

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \quad P' = P \cdot S$$

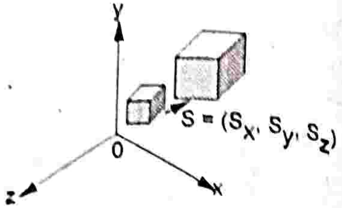


Fig. Q.2.1 3D Scaling

$$\therefore \quad [x' \; y' \; z' 1] = [x \; y \; z \; 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [x \cdot S_x \quad y \cdot S_y \quad z \cdot S_z \quad 1] \qquad \dots (Q.2.1)$$

7.3 : 3D Rotation

Q.3 Obtain the 3D transformation matrix for rotation about z-axis.

☞[SPPU : May-11, Marks 3]

OR Explain the rotation about all co-ordinate axis.

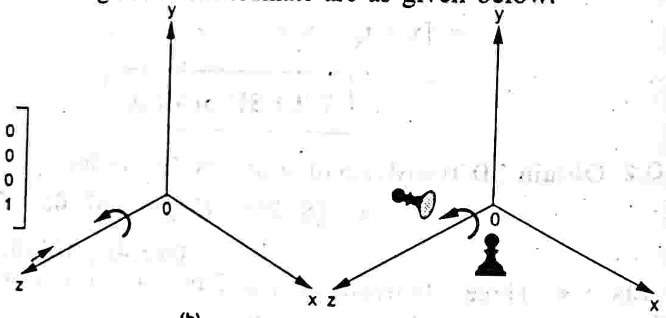
☞[SPPU : May-06, 07,19, June-22, Dec.-16, Marks 8]

OR Give homogenous transformation matrix for 3D rotation with respect to y axis. **☞ [SPPU : May-16, Marks 2]**

Ans. : • Three dimensional transformation matrix for each co-ordinate axes rotations with homogeneous co-ordinate are as given below.

$$R_z = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a)



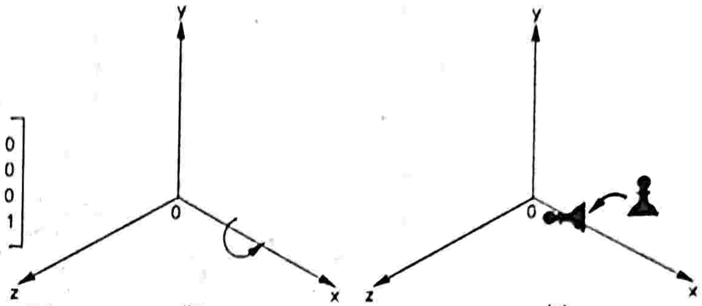
(b) (c)

Fig. Q.3.1 Rotation about z axis

• The positive value of angle θ indicates counterclockwise rotation. For clockwise rotation value of angle θ is negative.

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a)

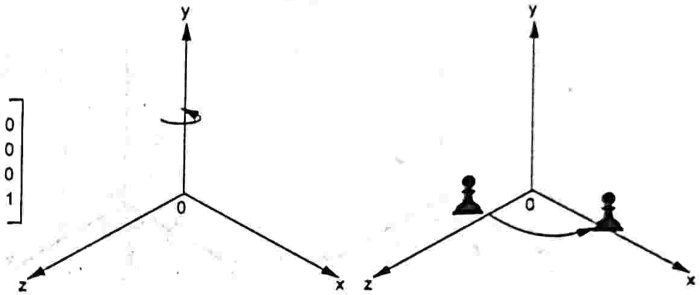


(b) (c)

Fig. Q.3.2 Rotation about x axis

$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a)



(b) (c)

Fig. Q.3.3 Rotation about y axis

7.4 : Rotation about Arbitrary Axis

Q.4 Derive the transformation matrix for rotation about any arbitrary axis.

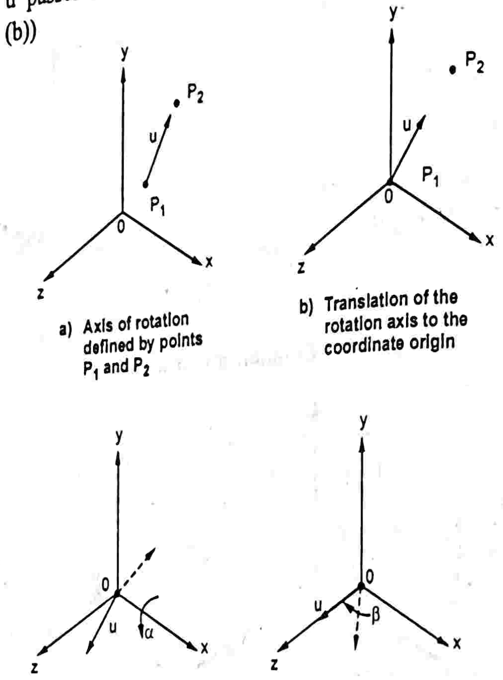
☞ [SPPU : May-11, 12, Marks 10 Dec.-11, Marks 12]

OR Explain the concept of 3D rotation about any arbitrary axis.

☞ [SPPU : May-06,08,12,14, Dec.-06,07,08,14,18, June-22, Marks 8]

Ans. : • When an object is to be rotated about an axis that is not parallel to one of the coordinate axes, we have to perform some additional transformations. The sequence of these transformations is given below.

1. Translate the object so that rotation axis specified by unit vector u passes through the coordinate origin. (See Fig. Q.4.1 (a) and (b))



a) Axis of rotation defined by points P_1 and P_2

b) Translation of the rotation axis to the coordinate origin

c) Rotation of unit vector about x axis to bring it into xz plane

d) Rotation of unit vector u about y axis to align it with the z axis

Fig. Q.4.1

2. Rotate the object so that the axis of rotation coincides with one of the coordinate axes. Usually the z axis is preferred. To coincide the axis of rotation to z axis we have to first perform rotation of unit vector u about x axis to bring it into xz plane and then perform rotation about y axis to coincide it with z axis. (See Fig. Q.4.1 (c) and (d))
3. Perform the desired rotation θ about the z axis.
4. Apply the inverse rotation about y axis and then about x axis to bring the rotation axis back to its original orientation.
5. Apply the inverse translation to move the rotation axis back to its original position.

Derivation of transformation matrix

The translation matrix is given as

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_1 & -y_1 & -z_1 & 1 \end{bmatrix}$$

- To perform the rotation of unit vector u about x axis. The rotation of u around the x axis into the xz plane is accomplished by rotating u′ (0, b, c) through angle α into the z axis and the cosine of the rotation angle α can be determined from the dot product of u′ and the unit vector u_z (0, 0, 1) along the z axis.

$$\cos\alpha = \frac{u' \cdot u_z}{|u'||u_z|} = \frac{c}{d} \qquad \text{where } u'(0, b, c) = b\mathbf{J} + c\mathbf{K} \text{ and}$$

$$u_z (0, 0, 1) = \mathbf{K}$$

$$= \frac{c}{|u'||u_z|} = \frac{c}{|u'|} \qquad \text{Since } |u_z| = 1$$

$$= \frac{c}{d}$$

where d is the magnitude of u′.

$$d = \sqrt{b^2 + c^2}$$

- Similarly, we can determine the sine of α from the cross product of u′ and u_z.

$$u' \times u_z = u_x |u'| |u_z| \sin\alpha \qquad \text{... (Q.4.1)}$$

and the Cartesian form for the cross product gives us

$$u' \times u_z = u_x \cdot b \qquad \text{... (Q.4.2)}$$

- Equating the right sides of equations (Q.4.1) and (Q.4.2) we get

$$u_x |u'| |u_z| \sin\alpha = u_x . b$$

$$\therefore \quad |u'| |u_z| \sin\alpha = b$$

$$\therefore \quad \sin\alpha = \frac{b}{|u'||u_z|}$$

$$= \frac{b}{d} \qquad \text{since } |u_z| = 1 \text{ and } |u'| = d$$

- By substituting values of cos α and sin α the rotation matrix R_x can be given as

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To perform the rotation of unit vector about y axis. This can be achieved by rotating u'' (a, 0, d) through angle β onto the z axis. Using similar equations we can determine cos β and sin β as follows.
- We have angle of rotation = – β

$$\therefore \quad \cos(-\beta) = \cos\beta = \frac{u'' \cdot u_z}{|u''||u_z|} \text{ where } u'' = aI + dK \text{ and}$$

$$u_z = K$$

$$= \frac{d}{|u''||u_z|} = \frac{d}{|u''|} \qquad \because |u_z| = 1$$

$$= \frac{d}{\sqrt{a^2+d^2}}$$

- Consider cross product of u'' and u_z.

$$u'' \times u_z = u_y |u''| |u_z| \sin(-\beta)$$

$$= -u_y |u''| |u_z| \sin\beta \qquad \because \sin(-\theta) = -\sin\theta$$

- Cartesian form of cross product gives us

$$u'' \times u_z = u_y (+a)$$

- Equating above equations,

$$-|u''|\,|u_z| \sin\beta = a$$

$$\therefore \quad \sin\beta = \frac{-a}{|u''||u_z|} = \frac{-a}{|u''|} \qquad \because |u_z| = 1$$

$$= \frac{-a}{\sqrt{a^2+d^2}}$$

but we have, $d = \sqrt{b^2+c^2}$

$$\therefore \quad \cos\beta = \frac{d}{\sqrt{a^2+d^2}} = \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}}$$

and $$\sin\beta = \frac{-a}{\sqrt{a^2+d^2}} = \frac{-a}{\sqrt{a^2+b^2+c^2}}$$

By substituting values of cos β and sin β in the rotation matrix R_y and by substituting $x = \sqrt{b^2+c^2}$ and $|V| = \sqrt{a^2+b^2+c^2}$ we have R_{xy},

∴ Resultant rotation matrix $R_{xy} = R_x \cdot R_y$

$$R_{xy} = \begin{bmatrix} \frac{\lambda}{|V|} & 0 & \frac{a}{|V|} & 0 \\ \frac{-ab}{|V|\lambda} & \frac{c}{\lambda} & \frac{b}{|V|} & 0 \\ \frac{-ac}{|V|\lambda} & \frac{-b}{\lambda} & \frac{c}{|V|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_z = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The overall transformation matrix for rotation about an arbitrary axis then can be expressed as the concatenation of five individual transformations.

$$R(\theta) = T \cdot R_{xy} \cdot R_z \cdot R_{xy}^{-1} \cdot T^{-1}$$

7.5 : Reflection

Q.5 Explain 3D reflection about xy, yz and xz plane.

[SPPU : May-17,18, Dec.-19,22, June-22, Marks 6]

Ans. : Reflection with Respect to xy Plane : • Consider point P(x, y, z). The reflection of this point with respect to xy plane is given by point P′(x, y, – z), as shown in Fig. Q.5.1.

- Corresponding to this reflection the transformation matrix can be given as

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

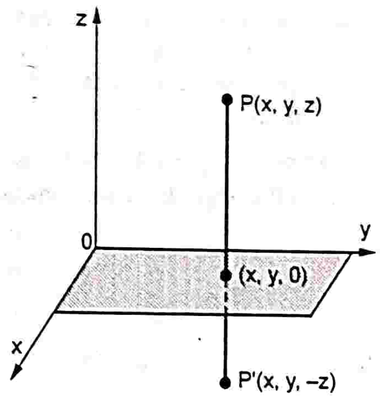


Fig. Q.5.1

- Simimilarly, the reflection matrices with respect to yz and xz planes are given as,

$$M_{yz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad M_{xz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Q.6 Explain the reflection with respect to arbitrary plane.

Ans. : • Often it is necessary to reflect an object through a plane other than x = 0 (yz plane), y = 0 (xz plane) or z = 0 (xy plane). Procedure to achieve such a reflection (reflection with respect to any plane) can be given as follows :

1. Translate a known point P_o, that lies in the reflection plane to the origin of the co-ordinate system.
2. Rotate the normal vector to the reflection plane at the origin until it is coincident with +ve z axis, this makes the reflection plane z = 0 co-ordinate plane i.e. xy plane.
3. Reflect the object through z = 0 (xy plane) co-ordinate plane.
4. Perform the inverse transformation to those given above to achieve the result.

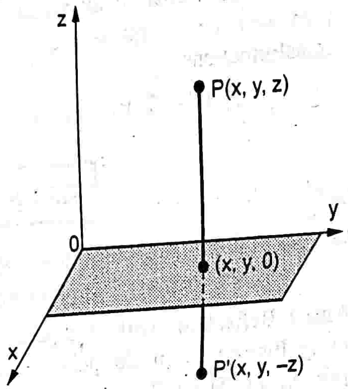


Fig. Q.6.1

• Let P_o (x_o, y_o, z_o) be the given known point. Translate this point to the origin by using corresponding translation matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_o & -y_o & -z_o & 1 \end{bmatrix}$$

• Let the normal vector

$$N = n_1 I + n_2 J + n_3 K$$

$$\therefore \quad |N| = \sqrt{n_1^2 + n_2^2 + n_3^2}$$

and $$\lambda = \sqrt{n_2^2 + n_3^2}$$

• As we want to match this vector with z axis, (so that the plane of reflection will be parallel to xy plane), we will use the same procedure as used in rotation.

$$\therefore \quad R_{xy} = \begin{bmatrix} \frac{\lambda}{|N|} & 0 & \frac{n_1}{|N|} & 0 \\ \frac{-n_1 n_2}{\lambda |N|} & \frac{n_3}{\lambda} & \frac{n_2}{|N|} & 0 \\ \frac{-n_1 n_3}{\lambda |N|} & \frac{-n_2}{\lambda} & \frac{n_3}{|N|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

• As seen earlier for reflection about xy plane we have

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

• Now for inverse transformation we have,

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_o & y_o & z_o & 1 \end{bmatrix}$$

$$R_{xy}^{-1} = \begin{bmatrix} \frac{\lambda}{|N|} & \frac{-n_1 n_2}{\lambda |N|} & \frac{-n_1 n_3}{\lambda |N|} & 0 \\ 0 & \frac{n_3}{\lambda} & \frac{-n_2}{\lambda} & 0 \\ \frac{n_1}{|N|} & \frac{n_2}{|N|} & \frac{n_3}{|N|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ Resultant transformation matrix can be given as

$$R_T = T \cdot R_{xy} \cdot M \cdot R_{xy}^{-1} \cdot T^{-1}$$

Q.7 Find the matrix for mirror reflection with respect to the plane passing through the origin and having a normal vector whose direction is M = I + J + K

Ans. : Here, P_0 (0, 0, 0) and the plane passes through the origin hence translation matrix is not necessary.

The normal vector $N = I + J + K$

$$\therefore \quad n_1 = 1, \quad n_2 = 1, \quad n_3 = 1$$

$$|N| = \sqrt{3} \quad \text{and} \quad \lambda = \sqrt{2}$$

$$\therefore \quad R_{xy} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{xy}^{-1} = \begin{bmatrix} \sqrt{\frac{2}{3}} & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{+1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ The reflection matrix is given by

$$R_T = R_{xy} \cdot M \cdot R_{xy}^{-1}$$

$$\therefore \quad R_T = \begin{bmatrix} 1/3 & -2/3 & -2/3 & 0 \\ -2/3 & 1/3 & -2/3 & 0 \\ -2/3 & -2/3 & +1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7.6 : Projections

Q.8 Explain parallel and perspective projection ?

[SPPU : Dec.-17,18, May-19, June-22, Marks 6]

Ans. : Parallel Projection

- In parallel projection, z co-ordinate is discarded and parallel lines from each vertex on the object are extended until they intersect the view plane.
- The point of intersection is the **projection of the vertex.**
- We connect the projected vertices by line segments which correspond to connections on the original object.
- As shown in the Fig. Q.8.1, a parallel projection preserves relative proportions of objects but does not produce the realistic views.

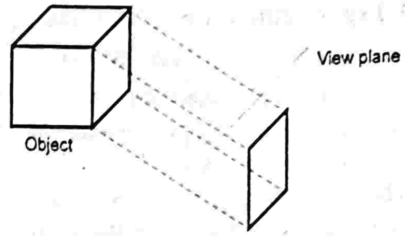


Fig. Q.8.1 Parallel projection of an object to the view plane

- Types of parallel projections are : Orthographic projection and oblique projection.

Perspective Projection : • The perspective projection, on the other hand, **produces realistic views but does not preserve relative proportions.**

- In perspective projection, the lines of projection are not parallel. Instead, they all converge at a single point called the **center of projection** or **projection reference point.**

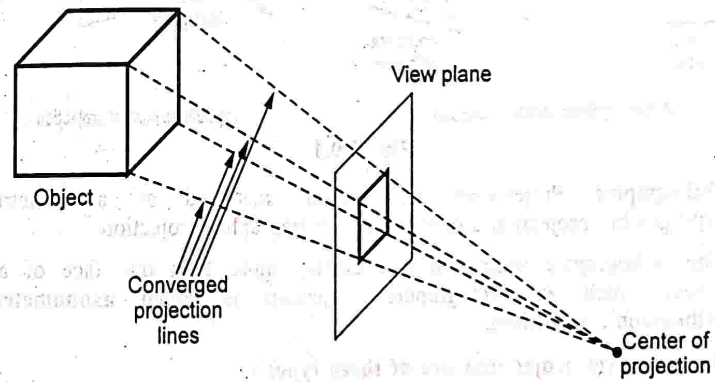


Fig. Q.8.2 Perspective projection of an object to the view plane

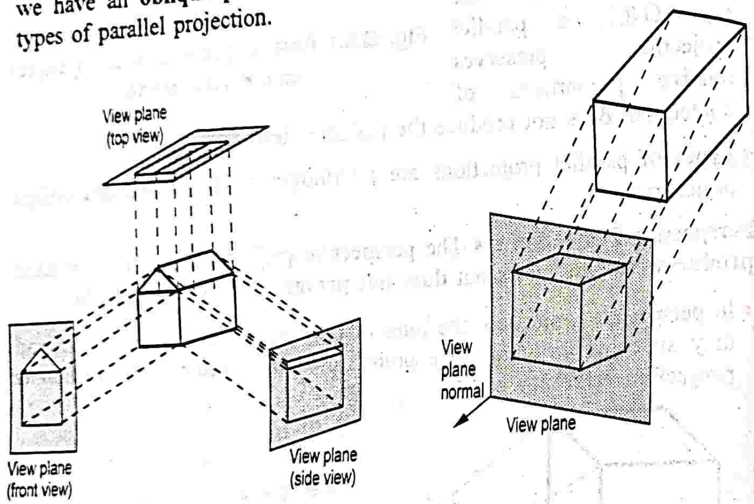
- The object positions are transformed to the view plane along these converged projection lines and the projected view of an object is determined by calculating the intersection of the converged projection lines with the view plane, as shown in the Fig. Q.8.2.

Q.9 Explain different types of parallel projection.

☞ [SPPU : May-14,18, Dec.-19, Marks 6]

Ans. : • Parallel projections are basically categorized into two types, depending on the relation between the direction of projection and the normal to the view plane.

- When the direction of the projection is normal (perpendicular) to the view plane, we have an **orthographic parallel projection.** Otherwise, we have an **oblique parallel projection.** Fig. Q.9.1 illustrates the two types of parallel projection.



(a) Orthographic parallel projection (b) Oblique parallel projection

Fig. Q.9.1

- Orthographic Projections are further classified as axonometric orthographic projection and multiview orthographic projection.
- The orthographic projection can display more than one face of an object. Such an orthographic projection is called **axonometric** orthographic projection.
- **Axonometric projections are of three types :**
  - **Isometric :** All three principle axes are foreshortened equally.
  - **Dimetric :** Two principle axes are foreshortened equally.

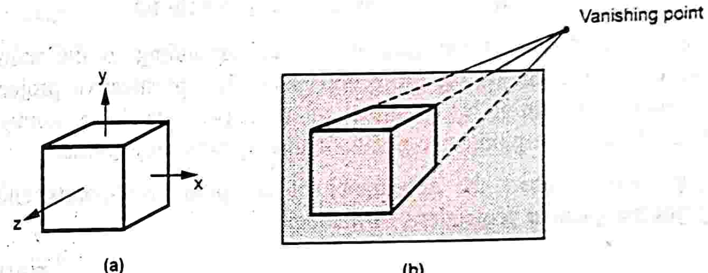
  - **Trimetric :** All three principle axes are foreshortened unequally.
- The oblique projections are further classified as the **cavalier** and **cabinet projections.**
- For the cavalier projection, the direction of projection makes a 45° angle with the view plane.
- When the direction of projection makes an angle of arctan (2) = 63.4° with the view plane, the resulting view is called a **cabinet projection.**

Q.10 What is the concept of vanishing point in perspective projection ? Explain with diagram.

☞[SPPU : Dec.-16,22, May-18, Marks 6]

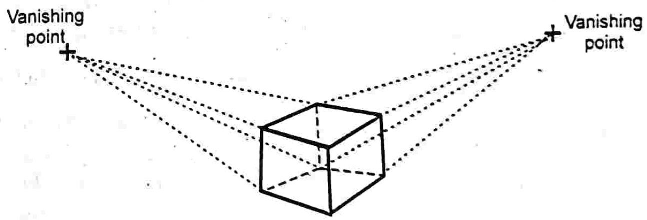
Ans. : • The perspective projection of any set of parallel lines that are not parallel to the projection plane converge to a **vanishing point.**

- The vanishing point for any set of lines that are parallel to one of the three principle axes of an object is referred to as a **principle vanishing point** or **axis vanishing point.**

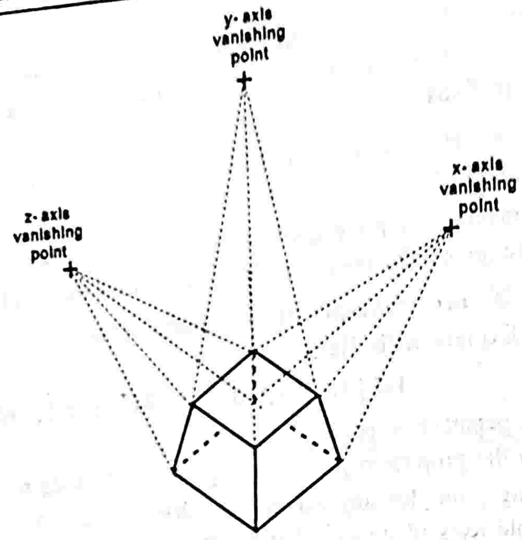


(a) Co-ordinate description

(b) One-point perspective projection of a cube



(c) Two-points perspective projection of a cube



(d) Three-point perspective of a cube

Fig. Q.10.1 Perspective projections

- There are at most three such points, corresponding to the number of principle axes cut by the projection plane. The perspective projection is classified according to number of principle vanishing points in a projection : One-point, two-points or three-points projections.
- Fig. Q.10.1 shows the appearance of one-point, two-points and three point perspective projections of a cube.

END...

Unit - IV

8 SEGMENTS

8.1 : Introduction

Q.1 What is segment ? [SPPU : May-05,06,07,08,16,19,22, Dec.-05,11,13,14,15,17,22 Marks 2]

OR Why do we need segments ? [SPPU : Dec.-11,13, Marks 2]

Ans. : • In practice, the image on the display screen is often composed of several pictures or items of information.

- An image may contain several views of an object and related information.
- It may also contain close up view of a particular component. For example, we may wish to display an internal plan of a living room. The plan may contain various objects such as sofa-set, T.V., show-case, teapot, show-pieces, wall hangings and so on.
- Each object has a set of attributes such as size, colour and its position in the room.
- We might wish to see all these objects simultaneously or a single object at a time.
- To view the entire image or a part of the image with various attributes we need to organize the image information in a particular manner.
- The image information is stored in the display file.
- Existing structure of the display file does not satisfy our requirements of viewing the image. Hence the display structure is modified to reflect the subpicture structure. To achieve this display file is divided into **segments.**
- Each segment corresponds to a component or an object of the overall display and is associated with a set of attributes.

- Along with the attributes the segment is also associated with the image transformation parameters such as scaling along X and/or Y-direction, rotation and shearing.
- Therefore, presence of segment allows :
  - Subdivision of the picture.
  - Visualization of a particular part of the picture.
  - Scaling, rotation and translation of a particular part of the picture.

8.2 : Segment Table

Q.2 Explain a segment table with example.

☞[SPPU : May-05,06,07,08,12,13,16,22, Dec.-05,14,15,17,22, Marks 4]

OR Give the structure of segment table.

Ans. : • To access a particular segment and the information associated with it we must have a unique name assigned to each segment.

- Along with the name we must have its display file position and its attribute information.
- The structure used to organize all this information related to segments is called **segment table.**

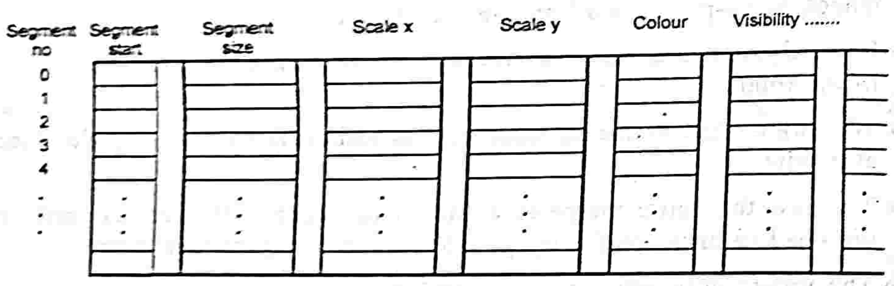


Fig. Q.2.1 Segment table

- First array holds the display file starting location for that segment, the second array holds the segment size information, while the third indicates the scaling and so on. This is illustrated in Fig. Q.2.1.
- Each row in the segment table represents information of one segment including its name, position, size, attributes and the image information parameters.

- If we wish to make segment 4 visible then this is achieved by setting the corresponding entry in the array to 'ON'.
- The display file interpreter initially checks the start, size and visible attribute of the segments and it interprets only those segments which are to be made visible.

8.3 : Functions for Segmenting the Display File

Q.3 Write the algorithm for change of visibility attribute of segments.

☞[SPPU : May-19, Dec.-11,13, Marks 3]

Ans. : Algorithm to change visibility of segment

```
Arguments  SEGMENT-NAME
           ON-OFF                          ... visibility setting
Global     VISIBILITY                      ... array of visibility falgs
Constant NUMBER-OF-SEGMENTS                ... the size of segment table
BEGIN
   IF SEGMENT-NAME<1 OR SEGMENT-NAME>
                                       NUMER-OF-SEGMENTS
   THEN
      RETURN ERROR 'INVNUD SEGMENT NAME';
      VISIBILITY [SEGMENT-NAME] ← ON-OFF;
      IF NOT ON-OFF THEN NEW-FRAME;
      RETURN;
END;
```

Q.4 Write the algorithm for the following :
i) Delete a segment ii) Delete all segments.

☞[SPPU : Dec.-11,13, Marks 5]

Ans. : Algorithm : Delete Segment

1. Read the name of segment which is to be deleted.
2. Check whether that segment name is valid; if not, display error message "Segment not valid" and go to step 8.
3. Check whether the segment is open, if yes, display error message "Can't delete the open segment" and go to step 8.

4. Check whether the size of segment is greater than 0; if no, no processing is required, as segment contains no instructions. Therefore, go to step 8.
5. Shift the display file elements which follow the segment which is to be deleted by its size.
6. Recover the deleted space by resetting the index of the next free instruction.
7. Adjust the starting positions of the shifted segments by subtracting the size of the deleted segment from it.
8. Stop.

Algorithm : Delete All Segments

```
For I = 0 To NUMBER-OF SEGMENTS DO
BEGIN
    SEGMENT-START[I] ← 1;
    SEGMENT-SIZE[I] ← 0;
END;
NOW-OPEN ← 0;
FREE ← 1;
NEW-FRAME;
RETURN;
END;
```

Q.5 Explain operations performed on segment table.

[SPPU : May-06,07,08,12,13,14,15,16, Dec.-11,13,14, Marks 6]

OR How do we get create segment ? [SPPU : Dec.-13,22, Marks 2]

Ans. : Basic operations performed on segments are :

- Create segment
- Close segment
- Delete segment
- Rename segment

Segment Creation

- In a segment creation process, initially, we have to check whether some other segment is still open.
- It is not allowed to open two segments at a same time because it is then difficult to assign the drawing instructions to particular segment. Hence,

segment must be created or opened when no other segment is currently open.

- We must give the segment a name so that we can identify it. While doing this it is important to check whether a given segment name is valid or not and whether there already exists a segment with the same name. If so, we have to assign other valid name to the segment.
- Once a valid segment name is assigned, we have to initialize the items in the segment table under our segment name to indicate that this is a fresh new segment.
- The first instruction of this segment will be located at the next free storage area in the display file.
- As we have not entered any instructions into the segment yet, its size is initialized with value zero.
- The attributes of the segment are initialized as a default attribute values.

Segment Close

- Once a segment is open we can enter the display file instructions in it. The entered commands are then associated with the open segment.
- After completion of entering all display file instructions, the segment must be closed.
- To close a segment it is necessary to change the name of the currently open segment. It can be achieved by changing the name of currently open segment as 0. Now the segment with name 0 is open i.e. unnamed segment is open.
- If there are two unnamed segments in the display file one has to be deleted.

Delete Segment

- When we want to delete a particular segment from the display file then we must recover the storage space occupied by its instructions and make this space free for some other segment. To do this we must not destroy and re-form the entire display file, but we must delete just one segment, while preserving the rest of the display file. The method to achieve this depends upon the data structure used to represent the display file.
- Here, we have used arrays to store the display file information.

- Use of arrays makes the recovery of storage space occupied by the segment very easy and straight forward. However using arrays is not as efficient as some other storage techniques such as link list.
- In case of arrays the gap left by deleted block is filled by shifting up all segments which are following the deleted segment.

Rename Segment

- We can display animated character on the display by presenting sequence of images, each with a slightly different drawing of the character.
- Let us assume that we have a segment in a display file for animated character. Then to display a new image in the sequence we have to delete the current segment and re-create it with the altered character. The problem in this process is that during the time after the first image is deleted and time before the second image is completely entered, only a partially completed character is displayed on the screen. We can avoid this problem by keeping the next image ready in the display file before deleting the current segment.
- This means that both segments, the segment which is to be deleted and the segment which is to be replaced with must exist in the display file at the same time.
- This can be achieved by creating a new invisible image under some temporary segment name.
- Now, when the current segment is deleted we can make the new image visible and rename it with the name of deleted segment.
- These steps can be repeated to achieve the animation.
- The idea of storing two images, one to show and other to create or alter, is called **double buffering**.

8.4 : More about Segments

Q.6 What are advantages of segments ?

[SPPU : May-15, Marks 4]

Ans. : Advantages

1. Segmentation allows to organize display files in subpicture structure.
2. It allows to apply different set of attributes to the different portions of the image.

3. Due to segmentation selective portion of the image can be displayed.
4. Segmentation makes it easier to modify the picture by changing segment attributes or by replacing segments.
5. Segmentation allows application of transformation on selective portion of the image.

END...

Unit - IV

9 ILLUMINATION MODELS AND SHADING ALGORITHMS

9.1 : Light Sources

Q.1 Explain light. ☞[SPPU : Dec.-14, Mark 1]

Ans. : Light is an electromagnetic energy, reaches the eye after interacting with the physical environment.

Q.2 Describe point source illumination.

☞[SPPU : Dec.-07,10,15,16,May-16, Marks 4]

Ans. : • A point source is a direction source, whose all the rays come from the same direction, therefore, it can be used to represent the distant sun by approximating it as an infinitely distant point source.

- The modelling of point sources requires additional work because their effect depends on the surface's orientation. If the surface is normal (perpendicular) to the incident light rays, it is brightly illuminated. The surfaces turned away from the light source (oblique surfaces) are less brightly illuminated.
- For oblique surfaces, the illumination decreases by a factor of cos I, where I is the angle between the direction of the light and the direction normal to the surface plane. The angle I is know as angle of incidence.

Q.3 What is diffuse reflection ?

Ans. : We know that shiny materials reflect more of the incident light, and dull surfaces absorb more of the incident light. Whereas rough or grainy surfaces tend to scatter the reflected light in all directions. This scattered light is called **diffuse reflection.**

9.2 : Ambient Light

Q.4 Write a note on ambient light.

Ans. : A surface of the object that is not exposed directly to a light source still will be visible if nearby objects are illuminated. A simple way

to model the combination of light reflections from various surfaces to produce a uniform illumination called the **ambient light,** or **background light** is to set a general level of brightness for a scene. This level is defined by the parameter I_a and each surface is then illuminated with this constant value. With such illumination, the reflected light is a constant for each surface, independent of the viewing direction and the spatial orientation of the surface. However, the intensity of the reflected light for each surface, depends on the optical properties of the surface, that is; how much of the incident light energy is to be reflected and how much absorbed.

9.3 : Diffuse Illumination and Reflection

Q.5 Describe diffuse illumination.

☞[SPPU : Dec.-07,10,15,16, May-11,12,13,16, Marks 4]

Ans. : • An object illumination is as important as its surface properties in computing its intensity. The object may be illuminated by light which does not come from any particular source but which comes from all directions. When such illumination is uniform from all directions, the illumination is called **diffuse illumination.**

- Usually, diffuse illumination is a background light which is reflected from walls, floor and ceiling.

Q.6 Explain reflectivity. ☞[SPPU : Dec.-14, Mark 1]

Ans. : • In practice, when object is illuminated, some part of light energy is absorbed by the surface of the object, while the rest is reflected.

- The ratio of the light reflected from the surface to the total incoming light to the surface is called **coefficient of reflection** or the **reflectivity.** It is denoted by R.

Q.7 What is Lambert's cosine law ?

Ans. : Lambert's cosine law states that the reflection of light from a perfectly diffusing surface varies as the cosine of the angle between the normal to the surface and the direction of the reflected ray.

9.4 : Specular Reflection

Q.8 Explain the following in detail : Specular reflection.

☞[SPPU : May-13, Marks 3]

Ans. : • When we illuminate a shiny surface such as polished metal or an apple with a bright light, we observe highlight or bright spot on the shiny surface. This phenomenon of reflection of incident light in a concentrated region around the specular-reflection angle is called **specular-reflection.**

• Due to specular-reflection, at the highlight, the surface appears to be not in its original colour, but white, the colour of incident light.

• The Fig. Q.8.1 shows the specular-reflection direction at a point on the illuminated surface. The specular-reflection angle equals the angle of the incident light, with the two angles measured on opposite sides of the unit normal surface vector N.

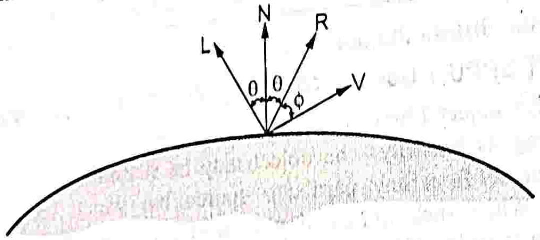


Fig. Q.8.1 Specular-reflection

• As shown in the Fig. Q.8.1, R is the unit vector in the direction of ideal specular-reflection, L is the unit vector directed towards the point light source and V is the unit vector pointing to the viewer from the surface position.

• The angle ϕ between vector R and vector V is called **viewing angle.** For an ideal reflector (perfect mirror), incident light is reflected only in the specular-reflection direction. In such case, we can see reflected light only when vector V and R coincide, i.e., $\phi = 0$.

9.5 : Phong Model

Q.9 Explain the Phong model.

☞[SPPU : Dec.-05, 08,17,18, May-17,18, Marks 4]

Ans. : • Phong Bui-Tuong developed a popular illumination model for nonperfect reflectors.

• It assumes that maximum specular-reflection occurs when ϕ is zero and falls off sharply as ϕ increases.

• This rapid fall-off is approximated by $\cos^n \phi$, where n is the specular-reflection parameter determined by the type of surface. The values of n typically vary from 1 to several hundred, depending on the surface material. The larger values (say, 100 or more) of n are used for very shiny surface and smaller values are used for dull surfaces. For a perfect reflector, n is infinite. For rough surface, such as chalk, n would be near to 1.

• Fig. Q.9.1 and Fig. Q.9.2 show the effect of n on the angular range of specular-reflection.

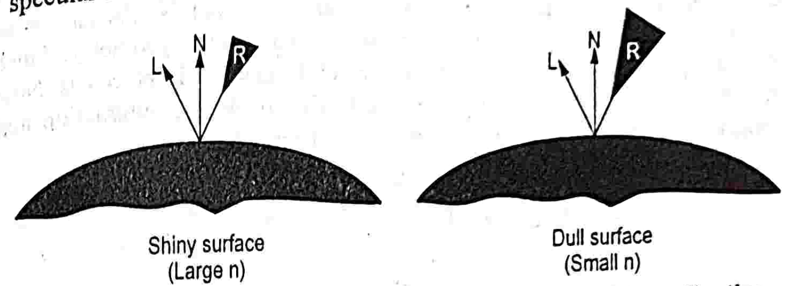


Fig. Q.9.1 Effect of n on the angular range of specular-reflection

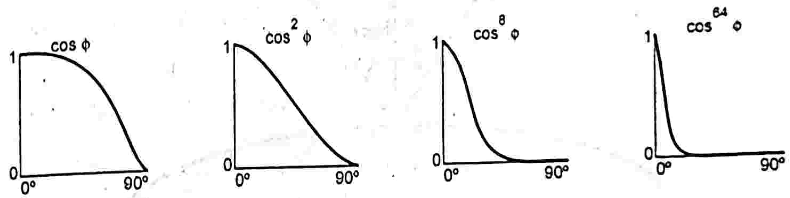


Fig. Q.9.2 Different values of $\cos^n \phi$ used in the Phong illumination model

• The amount of incident light specularly reflected depends on the angle of incidence θ, material properties of surface, polarization and colour of the incident light. The model is approximated for monochromatic specular intensity variations using a specular-reflection coefficient, W(θ), for each surface. We can write the equation for Phong specular-reflection model as,

$$I_{spec} = W(\theta)\, I_l \cos^n \phi$$

where I_l is the intensity of the light source and ϕ is the angle between viewing vector and specular reflection vector R.

- $W(\theta)$ is typically set to a constant k_s, the material's specular-reflection coefficient, which ranges from between 0 to 1. The value of k_s is selected experimentally to produce aesthetically pleasing results. Note that V and R are the unit vectors in the viewing and specular-reflection directions, respectively. Therefore, we can calculate the value of $\cos\phi$ with the dot product $V \cdot R$. Considering above changes we can rewrite the equation for intensity of the specular-reflection as,

$$I_{spec} = k_s I_l (V \cdot R)^n$$

- The vector R in the above equation can be calculated in terms of vector L and N. This calculation requires mirroring L about N. As shown in Fig. Q.9.3, this can be accomplished with some simple geometry. Since N and L are normalized, the projection of L onto N is N cos θ. Note that $R = N\cos\theta + S$, where $|S|$ is $\sin\theta$. But, by vector subtraction and congruent triangles, S is just $N\cos\theta - L$. Therefore,

$$R = N\cos\theta + N\cos\theta - L$$
$$= 2N\cos\theta - L$$

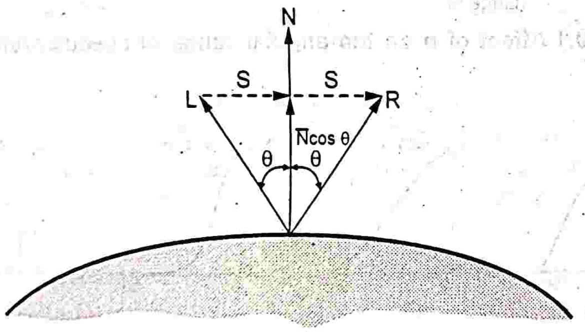


Fig. Q.9.3 Calculating the reflection vector

- Substituting $N \cdot L$ for $\cos\theta$ we have,

$$R = 2N(N \cdot L) - L$$

9.6 : Combined Diffuse and Specular Reflections with Multiple Light Sources

Q.10 Derive the illumination model with combine diffuse and specular reflections.

Ans. : • For a single point light source, the combined diffuse and specular reflections from any point on the illuminated surface is given as,

$$I = I_{diff} + I_{spec}$$
$$= k_a I_a + k_d I_l (N \cdot L) + k_s I_l (N \cdot H)^n$$

- For a multiple point light source the above equation can be modified as,

$$I = k_a I_a + \sum_{i=1}^{M} I_{l_i} [k_d (N \cdot L_i) + k_s (N \cdot H_i)^n]$$

- Therefore, in case of multiple point light sources the light reflected at any surface point is given by summing the contributions from the individual sources.

9.7 : Shading Algorithms

Q.11 Explain shading. ☞[SPPU : Dec.-14,22, May-18, Mark 1]

Ans. : • According to light source, representing the polygon surface is known as shading.

Q.12 How is polygon shading different from polygon filling ? ☞ [SPPU : May-17, Marks 7]

OR What is the need of shading ?

Ans. : • Filling the polygon means highlighting all pixels which lie inside the polygon with any colour other than background colour.

- On the other hand, shading polygon refers to the process of altering the colour of an polygon in the 3D scene, based on its angle to lights and its distance from light to create photorealistic effect.

Q.13 Write a note on constant intensity shading. ☞[SPPU : Dec.-22, Marks 4]

Ans. : • The fast and simplest method for shading polygon is constant shading, also known as **faceted shading** or **flat shading.**

- In this method, illumination model is applied only once for each polygon to determine single intensity value. The entire polygon is then displayed with the single intensity value.
- This method is valid for the following assumptions :

1. The light source is at infinity, so $N \cdot L$ is constant across the polygon face.
2. The viewer is at infinity, so $V \cdot R$ is constant over the surface.

3. The polygon represents the actual surface being modeled and is not an approximation to a curved surface.

- If either of the first two assumptions are not true still we can use constant intensity shading approach; however, we require some method to determine a single value for each of L and V vectors.

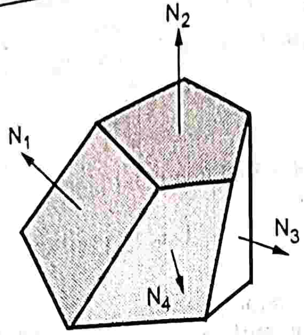


Fig. Q.13.1 Polygons and their surface normals

Q.14 Explain Gourand shading.

[SPPU : May-07,12,18, June-22, Dec.-08,11,12,19, Marks 4]

Ans. : • In this method, the intensity interpolation technique developed by Gouraud is used, hence the name.

- The polygon surface is displayed by linearly interpolating intensity values across the surface. Here, intensity values for each polygon are matched with the values of adjacent polygons along the common edges. This eliminates the intensity discontinuities that can occur in flat shading.
- By performing following calculations we can display polygon surface with Gouraud shading.
  1. Determine the average unit normal vector at each polygon vertex.
  2. Apply an illumination model to each polygon vertex to determine the vertex intensity.
  3. Linearly interpolate the vertex intensities over the surface of the polygon.
- We can obtain a normal vector at each polygon vertex by averaging the surface normals of all polygons sharing that vertex. This is illustrated in Fig. Q.14.1.
- As shown in the Fig. Q.14.1, there are three surface normals N_1, N_2 and N_3 of

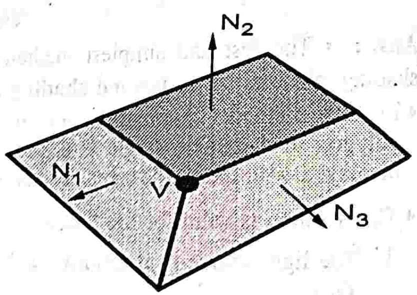


Fig. Q.14.1 Calculation of normal

polygon sharing vertex V. Therefore, normal vector at vertex V is given as

$$N_V = \frac{N_1 + N_2 + N_3}{|N_1 + N_2 + N_3|}$$

- In general, for any vertex position V, we can obtain the unit vertex normal by equation,

$$N_V = \frac{\sum_{i=1}^{n} N_i}{\left|\sum_{i=1}^{n} N_i\right|}$$

where n is the number of surface normals of polygons sharing that vertex.

- The next step in Gouraud shading is to find vertex intensities. Once we have the vertex normals, their vertex intensities can be determined by applying illumination model to each polygon vertex.
- Finally, each polygon is shaded by linear interpolating of vertex intensities along each edge and then between edges along each scan line. This is illustrated in Fig. Q.14.2.

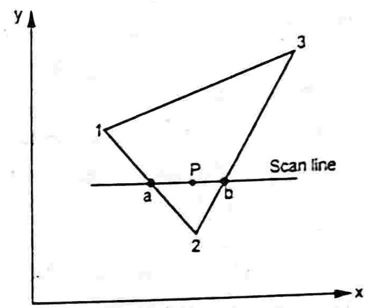


Fig. Q.14.2

Q.15 Explain pseudo C algorithm for Gourand shading.

[SPPU : Dec.-17,18, Marks 6]

Ans. : Pseudo C algorithm for Gouraud shading

$$dx = x_2 - x_1; \quad dv = V_2 - V_1;$$

$$r = dx/2; \quad i = dv/dx;$$

$$dv = dv - i * dx; \quad V = V_1;$$

for (x = x_1; x < = x_2; x++)

{

```
r = r + dv ;
if (r > dx)
    {
        V = V + 1;
        r = r - dx ;
    }
V = V + i ;
}
```

In this algorithm the values to be interpolated are stored in V_1 and V_2. The algorithm determines the integer portion of the increment i which is added at each step. The fractional portion of the increment is represented by r.

Q.16 Explain Phong's shading.

[SPPU : May-07,10,15,18, Dec.-19, June-22, Marks 4]

Ans. : • Phong shading, also known as **normal-vector interpolation shading**, interpolates the surface normal vector N, instead of the intensity.

• By performing following steps we can display polygon surface using Phong shading.

1. Determine the average unit normal vector at each polygon vertex.
2. Linearly interpolate the vertex normals over the surface of the polygon.
3. Apply an illumination model along each scan line to determine projected pixel intensities for the surface points.

• The first steps in the Phong shading is same as first step in the Gouraud shading.

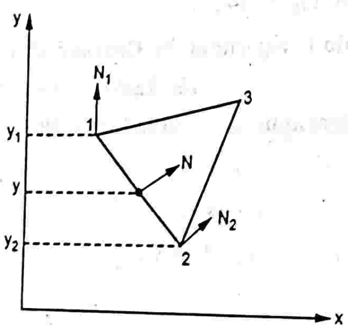


Fig. Q.16.1 Calculation of interpolation of surface normals along a polygon edge

• In the second step the vertex normals are linearly interpolated over the surface of the polygon. This is illustrated in Fig. Q.16.1. As shown in the Fig. Q.16.1, the normal vector N for the scan line intersection point along the edge between vertices 1 and 2 can be obtained by vertically interpolating between edge endpoint normals :

$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$

• Like, Gouraud shading, here also we can use incremental methods to evaluate normals between scan lines and along each individual scan line.

• Once the surface normals are evaluated, the surface intensity at that point is determined by applying the illumination model.

Q.17 Compare Gourand and Phong's method of shading.

[SPPU : Dec.-16,22, Marks 7]

Ans. :

Sr. No.	Gouraud shading	Phong's shading
1.	In this shading model the polygon surface is displayed by linearly interpolating intensity values across the surface.	In this shading model, the surface normal vector N is interpolated, instead of intensity.
2.	It suffers from mach-band effect.	It greatly reduces the mach-band effect.
3.	It requires less calculations than phong's shading.	It requires more calculations, increasing the cost of shading.

Q.18 Explain halftone shading.

[SPPU : May-18, Dec.-19, Marks 3]

Ans. : • Many displays and hardcopy devices are bilevel. They can only produce two intensity levels. In such displays or hardcopy devices, we can create an apparent increase in the number of available intensities. This is achieved by incorporating multiple pixels positions into the display of each intensity value.

• When we view a very small area from a sufficiently large viewing distance, our eyes average fine details within the small area and record only the overall intensity of the area. This phenomenon of apparent increase in the number of available intensities by considering combine intensity of multiple pixels is known as **halftoning.**

- The halftoning is commonly used in printing black and white photographs in newspapers, magazines and books. The pictures produced by halftoning process are called **halftones.**
- In computer graphics, halftone reproductions are approximated using rectangular pixel regions, say 2×2 pixels or 3×3 pixels. These regions are called **halftone patterns** or **pixel patterns.** Fig. Q.18.1 shows the halftone patterns to create number of intensity levels.

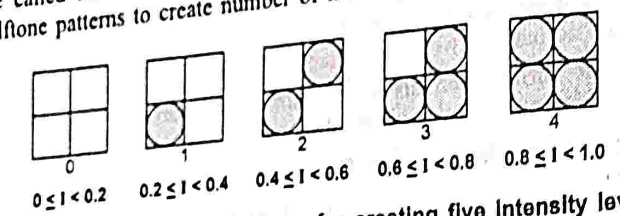


Fig. Q.18.1 (a) 2×2 pixel patterns for creating five intensity levels

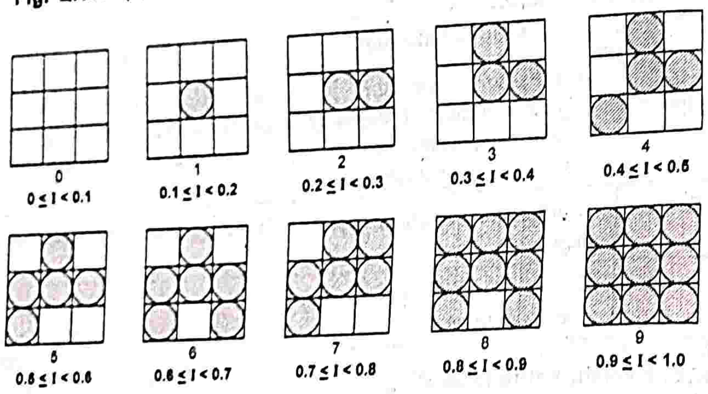


Fig. Q.18.1 (b) 3×3 pixel patterns for creating ten intensity levels

END...

Unit - IV

10 COLOUR MODELS

10.1 : Introduction

Q.1 Discuss the properties of light.

OR What is colour gamut ? ☞ **[SPPU : June-22, Marks 2]**

Ans. : • When this light is incident upon an object, some frequencies are absorbed and some are reflected by the object.

- The combination of reflected frequencies decides the colour of the object.
- The dominant frequency decides the colour of the object. Due to this reason, dominant frequency is also called the **hue** or simply the **colour.**
- The brightness refers to the intensity of the perceived light.
- The saturation describes the purity of the colour.
- Pastels and pale colours are described as less pure or less saturated.
- When the two properties purity and dominant frequency are used collectively to describe the colour characteristics, are referred to as **chromaticity.**
- When two colour sources are combined to produce white colour, they are referred to as **complementary colours.**
- Red and cyan, green and magenta, blue and yellow are complementary colour pairs.
- Usually, the colour model use combination of three colours to produce wide range of colours, called the **colour gamut** for that model.
- The basic colours used to produce colour gamut in particular model are called **primary colours.**

10.2 : CIE Chromaticity Diagram

Q.2 Explain CIE chromaticity diagram.

☞ **[SPPU : May-09, 12, 13, June-22, Dec.-22, Marks 5]**

Ans. : • Matching and therefore defining a coloured light with a combination of three fixed primary colours is desirable approach to specify colour.

- In 1931, the Commission Internationale de I' Eclairage (CIE) defined three standard primaries, called X, Y and Z to replace red, green and blue.
- Here, X, Y and Z represent vectors in a three-dimensional, additive colour space.
- The three standard primaries are imaginary colours. They are defined mathematically with positive colour-matching functions, as shown in Fig. Q.2.1. They specify the amount of each primary needed to describe any spectral colour.

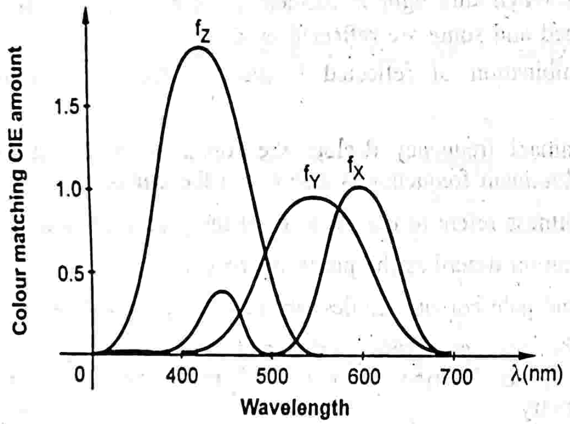


Fig. Q.2.1 Amounts of CIE primaries needed to display spectral colours

- The advantage of using CIE primaries is that they eliminate matching of negative colour values and other problems associated with selecting a set of real primaries.
- Any colour (C_λ) using CIE primaries can be expressed as,

$$C_\lambda = X\,\mathbf{X} + Y\,\mathbf{Y} + Z\,\mathbf{Z}$$

where X, Y and Z are the amounts of the standard primaries needed to match C_λ and **X**, **Y** and **Z** represent vectors in a three-dimensional, additive colour space.

- With above expression we can define chromaticity values by normalizing against luminance (X + Y + Z). The normalizing amounts can be given as,

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}$$

- Notice that x + y + z = 1. That is, x, y and z are on the (X + Y + Z = 1) plane. The complete description of colour is typically given with the three values x, y and z. The remaining values can be calculated as follows :

$$z = 1 - x - y, \qquad X = \frac{x}{y}Y, \qquad Z = \frac{z}{y}Y$$

- Chromaticity values depend only on dominant wavelength and saturation and are independent of the amount of luminous energy.
- By plotting x and y for all visible colours, we obtain the CIE chromaticity diagram shown in Fig. Q.2.2, which is the projection onto the (X, Y) plane of the (X + Y + Z = 1) plane.

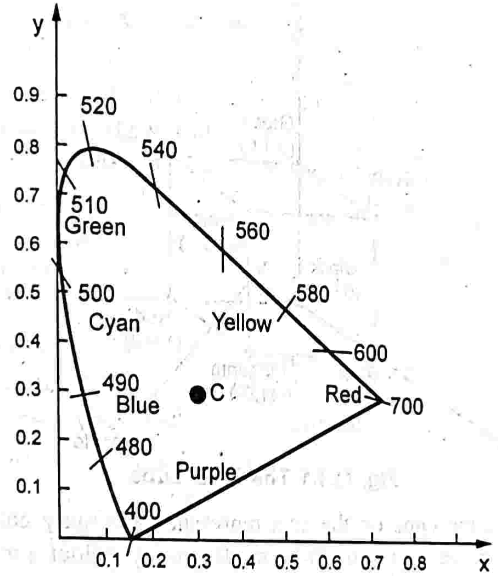


Fig. Q.2.2

- The interior and boundary of the tongue-shaped region represent all visible chromaticity values.
- The points on the boundary are the **pure** colours in the electromagnetic spectrum, labelled according to wavelength in nanometre from the red end to the violet end of the spectrum.

- A standard white light, is formally defined by a light source illuminant C, marked by the center dot.
- The line joining the red and violet spectral points is called the purple line, which is not the part of the spectrum.

10.3 : RGB Colour Model

Q.3 Explain RGB colour model.

[SPPU : Dec.-12,13,15,16, May-16,22, Marks 8]

Ans. : • In this model, the individual contribution of red, green and blue are added together to get the resultant colour.

- We can represent this colour model with the unit cube defined on R, G and B axes, as shown in the Fig. Q.3.1.

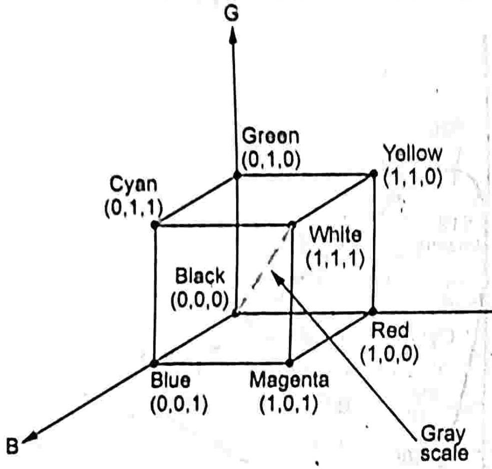


Fig. Q.3.1 The RGB cube

- The vertex of the cube on the axes represent the primary colours and the remaining vertices represent the complementary colour for each of the primary colours.
- The main diagonal of the cube, with equal amounts of each primary represents the gray levels. The end at the origin of the diagonal represents black (0, 0, 0) and other end represents white (1, 1, 1).
- Each colour point within the bounds of the cube is represented as the triple (R, G, B), where value for R, G, B are assigned in the range from 0 to 1.

- As mentioned earlier, it is an additive model.
- Intensities of the primary colours are added to get the resultant colour. Thus, the resultant colour C_λ is expressed in RGB component as,

$$C_\lambda = RR + GG + BB$$

10.4 : CMY Colour Model

Q.4 Explain CMY colour model.

[SPPU : May-06, Dec.-10,13,22, Marks 4]

Ans. : • In this model cyan, magenta and yellow colours are used as a primary colours. This model is used for describing colour output to hard-copy devices.

- Unlike video monitor, which produce a colour pattern by combining light from the screen phosphors, hard-copy devices such as plotters produce a colour picture by coating a paper with colour pigments.
- The subset of the cartesian co-ordinate system for the CMY model is the same as that for RGB except that white (full light) instead of black (no light) is at the origin. Colours are specified by what is removed or subtracted from white light, rather than by what is added to blackness.
- Cyan can be formed by adding green and blue light. Therefore, when white light is reflected from cyan coloured ink, the reflected light does not have red component. That is, red light is absorbed or subtracted, by the ink.
- Magenta ink subtracts the green component from incident light and yellow subtracts the blue component. Therefore, cyan, magenta and yellow are said to be complements of red, green and blue respectively.
- Fig. Q.4.1 shows the cube representation for CMY model.
- As shown in the Fig. Q.4.1, point (1,1,1) represents black, because all components of the incident light are subtracted.
- The point (0, 0, 0), the origin represents white light.
- The main diagonal represents equal amount of primary colours, thus the gray colours.
- A combination of cyan and yellow produces green light, because the red and blue components of the incident light are absorbed.
- Other colour combinations are obtained by a similar subtractive process.

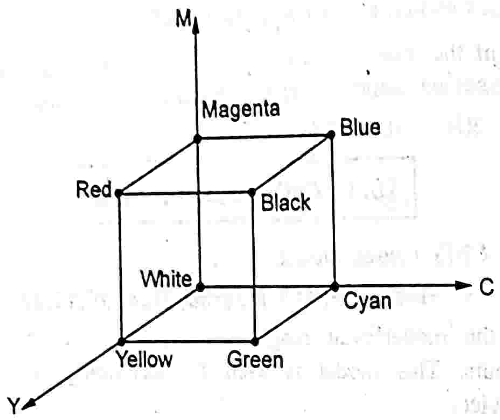


Fig. Q.4.1 The CMY cube

- It is possible to get CMY representation from RGB representation as follows,

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- The unit column vector is the RGB representation for white and the CMY representation for black. The conversion from RGB to CMY is then can be given as,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

10.5 : YCbCr Colour Model

Q.5 Write a note on YCbCr colour model.

Ans. : • YCbCr is another color model. It is used in digital video. In which,

- **Y :** It is the luma component (component which gives luminance information) of the colour. Luma component is the brightness of the colour. That means the light intensity of the colour. The human eye is more sensitive to this component.

- **Cb and Cr :** These are the blue and red components related to the chroma component (component which gives colour information). Component Cb is the difference between the blue component and a reference value. Component Cr is the difference between the red component and a reference value. These components are less sensitive to human eyes.

- Since the Y component is more sensitive to the human eye, it needs to be more correct. Since Cb and Cr are less sensitive to the human eye, they need not be more accurate.
- The YCbCr color model is used for component digital video and was developed as part of the ITU-R BT.601 Recommendation. YCbCr is a scaled and offset version of the YUV color model.
- The YCbCr colour space is used widely in digital video. In this format, luminance information is represented by a single component, Y and color information is stored as two colour-difference components, Cb and Cr. Component Cb is the difference between the red component and a reference value (Poynton [1996]). The transfromation used by IPT to convert from RGB to YCbCr is

Conversion between RGB and YCbCr Models

$$Y' = 0.257*R' + 0.504*G' + 0.098*B' + 16$$

$$Cb' = -0.148*R' - 0.291*G' + 0.439*B' + 128$$

$$Cr' = 0.439*R' - 0.368*G' - 0.071*B' + 128$$

$$R' = 1.164*(Y'-16) + 1.596*(Cr'-128)$$

$$G' = 1.164*(Y'-16) - 0.813*(Cr'-128) - 0.392*(Cb'-128)$$

$$B' = 1.164*(Y'-16) + 2.017*(Cb'-128)$$

10.6 : HSV Colour Model

Q.6 Explain HSV colour model.

☞[SPPU : May-05,06, June-22, Dec.-05,08,10,13,22, Marks 4]

Ans. : • RGB and CMY models are hardware oriented model. In contrast, HSV colour model is user oriented. It uses colour descriptions that have a more intuitive appeal to a user.

- The colour specification in HSV model can be given by selecting a spectral colour and the amounts of white and black that are to be added to obtain different shades tints and tones.

- This model uses three colour parameter : hue (H), saturation (S) and value (V).
- **Hue** distinguishes among colours such as red, green, purple and yellow.
- **Saturation** refers to how far colour is from a gray of equal intensity. For example, red is highly saturated whereas pink is relatively unsaturated. The **value** V indicates the level of brightness.
- This model is also known as **HSL** or **HSI**.
- HSL stands for hue, saturation and lightness.
- HSI stands for hue, saturation and intensity.
- The model uses cylindrical co-ordinate system and the subset of the space within which model is defined is a hexcone, or six-sided pyramid, as shown in the Fig. Q.6.1 (a).

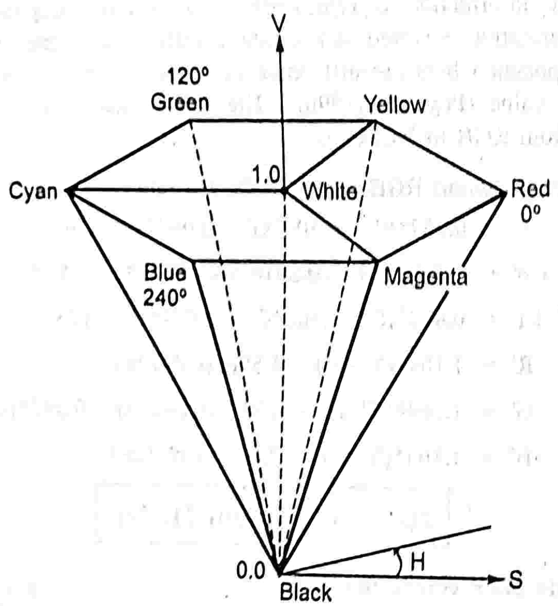
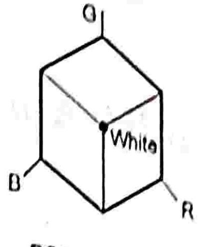


Fig. Q.6.1 (a) The HSV hexcone

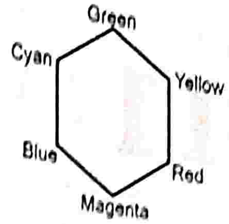
- The top of the hexcone is derived from the RGB cube.
- If we imagine viewing the cube along the main diagonal from the white vertex to the origin (black), we see an outline of the cube that has the

hexagon shape shown in Fig. Q.6.1 (b). This boundary of cube is used as a top if hexcone and it represents various hues.

- Hue or H, is measured by the angle around the vertical axis, with red at 0 °, green at 120 ° and so on as shown in the Fig. Q.6.1 (a).



RGB color cube
(a)



Color hexagon
(b)

Fig. Q.6.1 (b) Top of hexcone

- Complementary colours in the HSV hexcone are 180 ° apart saturation parameter varies from 0 to 1. Its value is the ratio ranging from 0 on the centre line (V axis) to 1 on the triangular sides of the hexcone.
- The value V varies from 0 at the apex of the hexcone to 1 at the top. The apex represents black.
- At the top of the hexcone, colours have their maximum intensity.
- When V = 1 and S = 1, we have the pure hues. For example, pure red is at H = 0, V = 1 and S = 1, pure green is at H = 120, V = 1 and S = 1, pure blue is at H = 240, V = 1 and S = 1 and so on.
- The required colour can be obtained by adding either white or black to the pure hue.
- Black can be added to the selected hue by decreasing the setting for V while S is held constant.
- On the other hand, white can be added to the selected hue by decreasing S while keeping V constant.

END...

Unit - V

11 CURVES AND FRACTALS

11.1 : Curve Generation

Q.1 What is true curve generation ? Write a Pseudo code to implement DDA arc generation.

☞ [SPPU : May-06, Dec.-07,10,11, Marks 10]

OR Write short note on True curve generation.

☞ [SPPU : Dec.-10, Marks 2]

Ans. : In a true curve generation algorithm, rather than approximating a curve by small line segments, we use digital differential analyzer algorithm to determine points those lie on the curve.

Algorithm / Pseudo code

Algorithm

1. Read the centre of a curvature, say (x_0, y_0).
2. Read the arc angle, say θ.
3. Read the starting point of the arc, say (x, y).
4. Calculate $d\theta$.

   $d\theta = \text{Min}\,(0.01,\ 1 / (3.2 \times (\,|x - x_0| + |y - y_0|\,)\,)\,)$
5. Initialize Angle = 0.
6. While (Angle < θ)

```
do
    {   Plot (x, y)
        x = x – (y – y0) × dθ
        y = y + (x – x0) × dθ
        Angle = Angle + dθ
    }
```

7. Stop.

problems in true-curve generation approach

1. To specify a curve, we need more information than just its end points.
2. It is difficult to apply transformations. For example, a circle when scaled in only one direction becomes an ellipse. If our algorithm supports only circular arc generation then ability to scale pictures is limited.
3. New clipping algorithm is required to clip arcs.
4. The curve generation algorithms for curves other than circular or elliptical such as airplane wings or cars or human faces, are complex.

11.2 : Interpolation and Interpolating Algorithms

Q.2 Write a short note on blending functions.

☞[SPPU : Dec.-13, Marks 4]

Ans. : • Suppose we want a polynomial curve that will pass through n sample points.

$(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)$

• We will construct the function as the sum of terms, one term for each sample point. These functions can be given as

$$f_x(u) = \sum_{i=1}^{n} x_i\, B_i(u)$$

$$f_y(u) = \sum_{i=1}^{n} y_i\, B_i(u)$$

$$f_z(u) = \sum_{i=1}^{n} z_i\, B_i(u)$$

• The function $B_i(u)$ is called **'blending function'**. For each value of parameter u, the blending function determines how much the i^{th} sample point affects the position of the curve.

• In other words we can say that each sample points tries to pull the curve in its direction and the function $B_i(u)$ gives the strength of the pull.

- If for some value of u, $B_i(u) = 1$ for unique value of i (i.e. $B_i(u) = 0$ for other values of i) then i^{th} sample point has complete control of the curve and the curve will pass through i^{th} sample point.
- For different value of u, some other sample point may have complete control of the curve. In such case the curve will pass through that point as well.
- In general, the blending functions give control of the curve to each of the sample points inturn for different values of u.

Q.3 What is interpolation ? Explain Lagrange interpolation method.

☞[SPPU : May-07,08,09,16,18,19, Dec.-05,10,11,12,13,16,18, Marks 8]

Ans. : In a curve generation, process of determining the intermediate points between the known sample points is acheived using interpolation. The blending functions give control of the curve to each of the sample points inturn for different values of u.

- Let us assume that the first sample point (x_1, y_1, z_1) has complete control when $u = -1$, the second when $u = 0$, the third when $u = 1$, and so on. i.e.
  - When $u = -1 \Rightarrow B_1(u) = 1$ and 0 for $u = 0, 1, 2, \ldots, n-2$
  - When $u = 0 \Rightarrow B_2(u) = 1$ and 0 for $u = -1, 1, \ldots, n-2$

    $\vdots \quad \vdots \quad \vdots$

    $\vdots \quad \vdots \quad \vdots$
  - When $u = (n-2) \Rightarrow B_n(u) = 1$ and 0 for $u = -1, 0, \ldots, (n-1)$
- To get $B_1(u) = 1$ at $u = -1$ and 0 for $u = 0, 1, 2, \ldots, n-2$, the expression for $B_1(u)$ can be given as

$$B_1(u) = \frac{u(u-1)(u-2)\ldots.[u-(n-2)]}{(-1)(-2)\ldots.(1-n)}$$

where denominator term is a constant used. In general form i^{th} blending function which is 1 at $u = i - 2$ and 0 for other integers can be given as :

$$B_i(u) = \frac{(u+1)(u)(u-1)\ldots[u-(i-3)][u-(i-1)]\ldots.[u-(i-2)]}{(i-1)(i-2)(i-3)\ldots.(1)(-1)\ldots(i-n)}$$

The approximation of the curve using above expression is called **Lagrange interpolation.**

11.3 : Interpolating Polygons

Q.4 Write a note on interpolating polygons.

☞ [SPPU : Dec.-08, Marks 4; June-22, Marks 6]

Ans. : • Blending functions can also be used to round the sides of polygon.

- Smoothing of each side of the polygon is done by replacing it with several small line segments.
- We start out with a polygon that has only a few sides and end up with a polygon which has many more sides and appears smoother due to the interpolation.
- This is illustrated in Fig. Q.4.1.

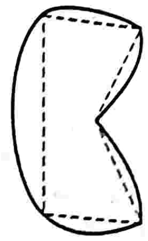


Fig. Q.4.1 Smoothing of a polygon

Algorithm

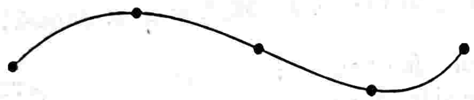
1. Read the original number of polygon sides, say N.
2. Read the vertices of original polygon.
3. Read the blending function values.
4. Read the count for the number of small line segments per original polygon side.
5. Smooth one side of original polygon by stepping through four blending functions.
6. Repeat step 5 according to number of small line segments required per original polygon side.
7. Repeat step 5 and 6 for each side of the original polygon.
8. Draw the polygon using small line segments.
9. Stop.

11.4 : Spline Representation

Q.5 Explain the term control points and order of continuity in curve drawing. **☞[SPPU : May-11, 12, Marks 8, June-22, Marks 6]**

Ans. : • We can specify a spline curve by giving a set of co-ordinate positions, called **control points**, which indicates the general shape of the curve.

- When polynomial sections are fitted so that the curve passes through all control points, as shown in the Fig. Q.5.1, the resulting curve is said to **interpolate** the set of control points.

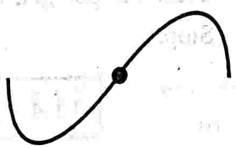


Interpolation spline

Fig. Q.5.1

- To ensure a smooth transition from one section of a piecewise parametric curve to the next, we can impose various continuity conditions at the connection points. We see **parametric continuity** and **geometric continuity** conditions.
- In geometric continuity we require parametric derivatives of two sections to be proportional to each other at their common boundary instead of equal to each other.
- Parametric continuity is set by matching the parametric derivatives of adjoining two curve sections at their common boundary.
- In zero order parametric continuity, given as C^0, it means simply the curve meet and same is for zero order geometric continuity.
- In first order parametric continuity called as C^1 means that first parametric derivatives of the coordinate functions for two successive curve sections are equal at the joining point and geometric first order continuity means the parametric first derivative are proportional at the intersection of two successive sections.
- Second order parametric continuity or C^2 continuity means that both the first and second parametric derivatives of the two curve sections are same at the intersection and for second order geometric continuity or C^2 continuity means that both the first and second parametric derivatives of the two curve sections are proportional at their boundary. Under C^2 continuity curvature of the two curve sections match at the joining positions.

(a) Zero order continuity



(b) First order continuity

(c) Second order continuity

Q.6 Why is cubic form chosen for representing curve ?

[SPPU : Dec.-17, Marks 6]

Ans. : • Polylines and polygons are first-degree, piecewise linear approximation to curves and surfaces, respectively. But this lower degree polynomials give too little flexibility in controlling the shape of the curve.

- The higher-degree polynomials give reasonable design flexibility, but introduce unwanted wiggles and also require more computation.

For this reason the third-degree polynomials are most often used for representation of curves. These polynomials are commonly known as **cubic polynomials.**

Q.7 What is convex hull ?

Ans. : • The convex polygon boundary that encloses a set of control points is called the **convex hull.** One way to envision the shape of a convex hull is to imagine a rubber band stretched around the positions of the control points so that each control point is either on the perimeter of the hull or inside it. Convex hulls provide a measure for the deviation of a curve or surface from the region bounding the control points. Some splines are bounded by the convex hull, thus ensuring that the polynomials smoothly follow the control points without erratic oscillations. Also, the polygon region inside the convex hull is useful in some algorithms as a clipping region.

11.5 : Hermite Interpolation

Q.8 Write a short note on Hermite interpolation.

Ans. : • In Hermite interpolation, cubic polynomial is interpolated piecewise with a specified tangent at each control point and each curve section is only dependent on its endpoint constraints. Thus, unlike natural cubic splines, Hermite splines can be adjusted locally.

- As stated, the Hermite (named after the French mathematician Charles Hermite) curve is an cubic polynomial curve segment determined by

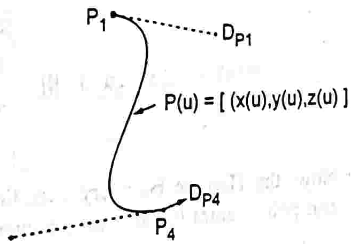


Fig. Q.8.1 Hermite curve

constraints on the endpoints P_1 and P_4 and tangent vectors at the endpoints D_{P1} and D_{P4}, as shown in the Fig. Q.8.1.

- As shown in the Fig. Q.8.1, P(u) represents a parametric cubic point function for the curve section between two endpoints P_1 and P_4. Therefore, the boundary conditions that define this Hermite curve section are

$$P(0) = P_1$$
$$P(1) = P_4$$
$$P'(0) = D_{P1}$$
$$P'(1) = D_{P4} \qquad \ldots (Q.8.1)$$

where D_{P1} and D_{P4} are the values for the parametric derivatives (slope of the curve) at endpoints P_1 and P_4, respectively.

- We can write the vector equivalent of equation (Q.8.1) for this Hermite-curve section as

$$P(u) = au^3 + bu^2 + cu + d, \; 0 \leq u \leq 1 \qquad \ldots (Q.8.2)$$

where,

the x component of P is $x(u) = a_x u^3 + b_x u^2 + c_x u + d$

the y component of P is $y(u) = a_y u^3 + b_y u^2 + c_y u + d$ and

the z component of P is $z(u) = a_z u^3 + b_z u^2 + c_z u + d$

The equation (Q.8.2) can be written in matrix form as

$$P(u) = [u^3 \; u^2 \; u \; 1] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad \ldots (Q.8.3)$$

and the derivative of the point function can be expressed as

$$P'(u) = [3u^2 \; 2u \; 1 \; 0] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad \ldots (Q.8.4)$$

- Now, the Hermite boundary condition can be expressed by substituting end point values 0 and 1 for parameter u. These are specified as follows

$$\begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad \ldots (Q.8.5)$$

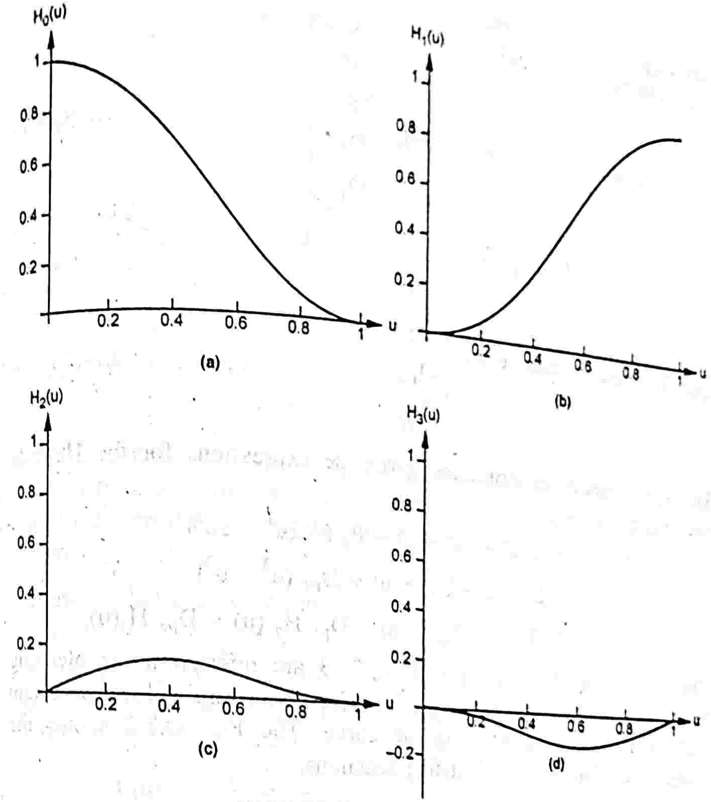


Fig. Q.8.2

- Solving equation (Q.8.5) for the polynomial coefficient, we have

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix}$$

$$= M_H \begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix} \quad \ldots \text{(Q.8.6)}$$

where M_H, the Hermite matrix, is the inverse of the boundary constraint matrix. The equation (Q.8.3) now can be written as

$$P(u) = [u^3 \ \ u^2 \ \ u \ \ 1] \cdot M_H \cdot \begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix} \quad \ldots \text{(Q.8.7)}$$

$$= U \cdot M_H \cdot G_H$$

where G_H is the column vector $\begin{bmatrix} P_1 \\ P_4 \\ D_{P1} \\ D_{P4} \end{bmatrix}$

- The above matrix multiplication gives the expressions for the Hermite blending functions.

$$P(u) = P_1 (2u^3 - 3u^2 + 1) + P_4 (-2u^3 + 3u^2)$$
$$+ D_{P1} (u^3 - 2u^2 + u) + D_{P4} (u^3 - u^2)$$
$$= P_1 H_0(u) + P_4 H_1(u) + D_{P1} H_2(u) + D_{P4} H_3(u)$$

- The polynomials $H_i(u)$ for $i = 0, 1, 2, 3$ are referred to as blending functions because they blend the boundary constrain values to obtain each coordinate position along the curve. The Fig. Q.8.2 shows the shape of the four Hermite blending functions.

11.6 : Bezier Curves

Q.9 Explain the Bezier curve.

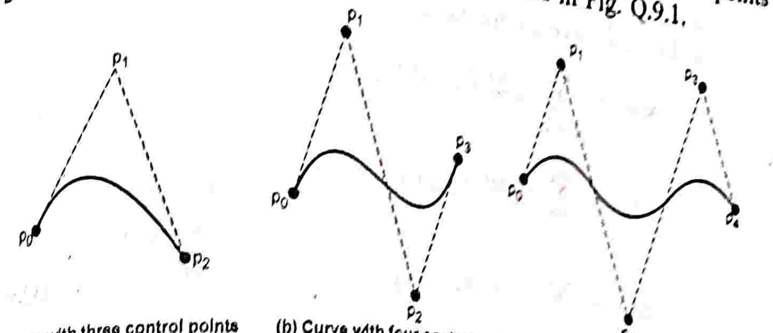
[SPPU : Dec.-05,12,13,14,15,22, May-07,10,11,13,14, June-22, Marks 6]

OR Explain any Bezier curve generation method.

[SPPU : Dec.-17, Marks 3]

Ans. : • In general, a Bezier curve section can be fitted to any number of control points. However, as number of control points increases, the degree of the Bezier polynomial also increases. Because in a Bezier curve a degree of a polynomial is one less than the number of control points used.

- For example, three control points generate a parabola, four points generate cubic curve and so on. This is illustrated in Fig. Q.9.1.



(a) Curve with three control points (b) Curve with four control points (c) Curve with five control points

Fig. Q.9.1

- The Bezier curves can be specified with boundary conditions, with a characterizing matrix or with blending functions. Out of these, blending function specification is the most convenient way for general Bezier curves.
- Consider that the curve has n + 1 control points : $p_k = (x_k, y_k, z_k)$ ... where k varies from 0 to n. The co-ordinates of these control points can be blended to produce position vector P(u), which gives the path of an approximating Bezier polynomial function between p_0 and p_n. The position vector can be given by,

$$P(u) = \sum_{k=0}^{n} p_k \, BEZ_{k,n}(u), \quad 0 \le u \le 1 \quad \ldots \text{(Q.9.1)}$$

- The Bezier blending functions $BEZ_{k,n}(u)$ are the Bernstein polynomials. They are specified as,

$$BEZ_{k,n}(u) = C(n, k)\, u^k (1-u)^{n-k}$$

where the C(n ,k) are the binomial coefficients. Binomial coefficients are given by,

$$C(n, k) = \frac{n!}{k!\,(n-k)!} \quad \ldots \text{(Q.9.2)}$$

- Equivalently, we can define Bezier blending functions with the recursive calculation as,

$$BEZ_{k,n}(u) = (1-u)\,BEZ_{k,n-1}(u) + u\,BEZ_{k-1,n-1}(u), \quad n > k \geq 1 \qquad \dots (Q.9.3)$$

with $BEZ_{k,k} = u^k$ and $BEZ_{0,k} = (1-u)^k$. The position vector equation represents a set of three parametric equations for the individual curve co-ordinates :

$$x(u) = \sum_{k=0}^{n} x_k BEZ_{k,n}(u)$$

$$y(u) = \sum_{k=0}^{n} y_k BEZ_{k,n}(u)$$

$$z(u) = \sum_{k=0}^{n} z_k BEZ_{k,n}(u) \qquad \dots (Q.9.4)$$

- The successive binomial coefficients can be calculated as

$$C(n, k) = \frac{n-k+1}{k} C(n, k-1) \qquad \dots (Q.9.5)$$

Q.10 What are the properties of Bezier curve ?

☞[SPPU : Dec.-17, June-22, Marks 6]

Ans. : Properties of Bezier curve

1. The basis functions are real.
2. Bezier curve always passes through the first and last control points i.e. curve has same end points as the guiding polygon.
3. The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is three, i.e. cubic polynomial.
4. The curve generally follows the shape of the defining polygon.
5. The direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments.
6. The curve lies entirely within the convex hull formed by four control points.
7. The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.

8. The curve exhibits the variation diminishing property. This means that the curve does not oscillate about any straight line more often than the defining polygon.
9. The curve is invariant under an affine transformation.

Q.11 Explain Bezier curve generation using midpoint sub division.

☞[SPPU : May-15,18,19, Dec.-19, Marks 7]

Ans. : • In the **midpoint approach**, the Bezier curve can be constructed simply by taking midpoints.

- In midpoint approach midpoints of the lines connecting four control points (A, B, C, D) are determined (AB, BC, CD). These midpoints are connected by line segments and their midpoints ABC and BCD are determined. Finally these two midpoints are connected by line segments and its midpoint ABCD is determined. This is illustrated in Fig. Q.11.1.
- The point ABCD on the Bezier curve divides the original curve into two sections. This makes the points A, AB, ABC and ABCD are the control points for the first section and the points ABCD, BCD, CD and D are the control points for the second section.

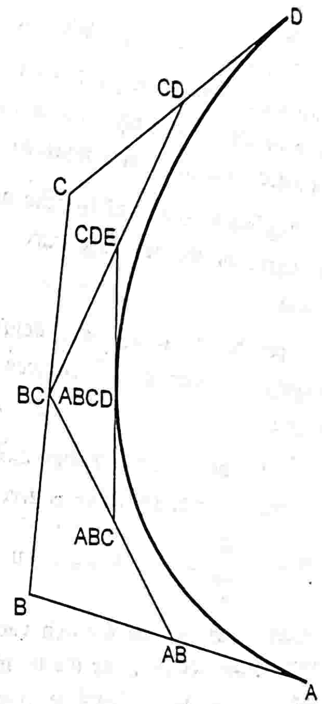


Fig. Q.11.1 Subdivision of a Bezier spline

- By considering two sections separately we can get two more sections for each separate section i.e. the original Bezier curve gets divided into four different curves. This process can be repeated to split the curve into smaller sections until we have sections so short that they can be replaced by straight lines or even until the sections are not bigger than individual pixels.

11.7 : B-Spline Curve

Q.12 Write short note on B-splines.
[SPPU : May-10,11,13,15, Dec.-12,13,22, Marks 8]

OR Explain B-splines for curve generation.
[SPPU : May-16,18, Dec.-16, Marks 4]

Ans. : • There is another basis function, called the B-spline basis, which contains the Bernstein basis as a special case.

- The B-spline basis is nonglobal. It is nonglobal because each vertex B_i is associated with a unique basis function. Thus, each vertex affects the shape of the curve only over a range of parameter values where its associated basis function is nonzero.
- The B-spline basis also allows the order of the basis function and hence the degree of the resulting curve is independent on the number of vertices.
- It is possible to change the degree of the resulting curve without changing the number of vertices (control points) of the defining polygon.
- If P(u) be the position vectors along the curve as a function of the parameter u, a B-spline curve is given by

$$P(u) = \sum_{i=1}^{n+1} B_i N_{i,k}(u) \quad u_{min} \le u < u_{max}, \quad 2 \le k \le n+1 \qquad \text{...(Q.12.1)}$$

where the B_i are the position vectors of the n + 1 defining polygon vertices and the $N_{i,k}$ are the normalized B-spline basis functions. For the i^{th} normalized B-spline basis function of order k, the basis function $N_{i,k}(u)$ are defined as

$$N_{i,1}(u) = \begin{cases} 1 & \text{if} \quad x_i \le u < x_{i+1} \\ 0 & \text{Otherwise} \end{cases}$$

$$\text{and } N_{i,k}(u) = \frac{(u - x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \qquad \text{...(Q.12.2)}$$

- The values of x_i are the elements of a knot vector satisfying the relation $x_i \le x_{i+1}$. The parameter u varies from u_{min} to u_{max} along the curve P(u).
- The choice of knot vector has a significant influence on the B-spline basis functions $N_{i,k}(u)$ and hence on the resulting B-spline curve.

Q.13 Discuss the properties of B-spline curve.
[SPPU : Dec.-17, Marks 3]

Ans. : Properties of B-spline curve

- The sum of the B-spline basis functions for any parameter value u is 1.

  i.e. $\sum_{i=1}^{n+1} N_{i,k}(u) \equiv 1$
- Each basis function is positive or zero for all parameter values, i.e., $N_{i,k} \ge 0$.
- Except for k = 1 each basis function has precisely one maximum value.
- The maximum order of the curve is equal to the number of control points of defining polygon.
- The curve exhibits the variation diminishing property. Thus the curve does not oscillate about any straight line move often than its defining polygon.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

Q.14 State advantages of B-spline curve over Bezier for generating curve.
[SPPU : Dec.-06, Marks 4]

Ans. : Advantages of B-splines over Bezier curves

- The degree of B-spline polynomial is independent on the number of control points of defining polygon (with certain limitations).
- B-spline allows local control over the curve surface because each control point affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.

Q.15 As far as splines are concerned, what do Bezier and B - splines curves indicate ? ☞[SPPU : May-17, Marks 3]

Ans. : • In general, a Bezier curve section can be fitted to any number of control points. However, as number of control points increases, the degree of the Bezier polynomial also increases. Because in a Bezier curve a degree of a polynomial is one less than the number of control points used.

- The degree of B-spline polynomial is independent on the number of control points of defining polygon (with certain limitations).
- B-spline allows local control over the curve surface because each control point affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.

Q.16 Give comparison between Bezier and B-spline curves. ☞ [SPPU : May-08, Dec.-10, 19, June-22, Marks 8]

Ans. :

Sr. No.	Bezier curve	B-spline curve
1.	The degree of the polynomial defining the curve segment is one less that the number of defining polygon point (Control points). Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.	The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
2.	Bezier curve can't be forced to interpolate any of its n control points without repeating it.	B-splines can be forced to interpolate any of its n control points without repeating it.
3.	Bezier curve requires less computation.	B-spline curve requires more computation.
4.	Bezier curves are less flexible.	B-Spline curve offers more control and flexibility.
5.	Bezier curves require less information.	B-Spline curves require more information such as degree of the curve and a knot vector.
6.	In Bezier curves, it is not possible to use lower degree curves and still maintain a large number of control points.	In B-Spline curves, it is possible to use lower degree curves and still maintain a large number of control points.

11.8 : Fractals

Q.17 Define fractals with examples. Given various categories in which fractals are classified. ☞[SPPU : Dec.-17,18,22, June-22, Marks 6]

Ans. : Rough, jagged and random surfaces are called **fractals.**

Classification of fractals

- The fractals can be classified as
  - Self similar
  - Self affine and
  - Invariant

Self similar fractals

- These fractals have parts those are scaled-down versions of the entire object.
- In these fractals object subparts are constructed by applying a scaling parameter s to the overall initial shape. It is a choice of user to use the same scaling factor s for all subparts, or use different scaling factors for different scaled-down parts of the object.
- Another sub class of self similar fractals is a statistically self-similar fractals, in which user can also apply random variations to the scaled-down subparts. These fractals are commonly used to model trees, shrubs, and other plants.

Self-affine fractals

- These fractals have parts those are formed with different scaling parameters, s_x, s_y, s_z in different co-ordinate directions.
- In these fractals, we can also apply random variations to obtain statistically self-affine fractals.
- These fractals are commonly used to model water, clouds and terrain.

Invarient fractals

- In these fractals, **nonlinear transformation** is used.
- It includes **self squaring fractals** such as the Mandelbrot set, which are formed with squaring functions in complex space and self inverse fractals, form with inversion procedures.

Q.18 What do you mean by topological and fractal dimension ?

☞[SPPU : May-05,06,08,09, Dec.-05,07,10,18, Marks 6]

Ans. : Consider an object composed of elastic or clay. If the object can be deformed into a line or line segment we assign its dimension $D_t = 1$. If object deforms into a plane or half plane or disk we assign its dimension $D_t = 2$. If object deforms into all space or half space or sphere, we assign its dimension $D_t = 3$. The dimension D_t is referred to as the **topological dimension.**

Fractal dimension

- It is the second measure of an object dimension. Imagine that a line segment of length L is divided into N identical pieces. The length of each line segment l can be given as

$$l = \frac{L}{N}$$

- The ratio of length of original line segment and the length of each part of the line segment is referred to as scaling factor and is given as

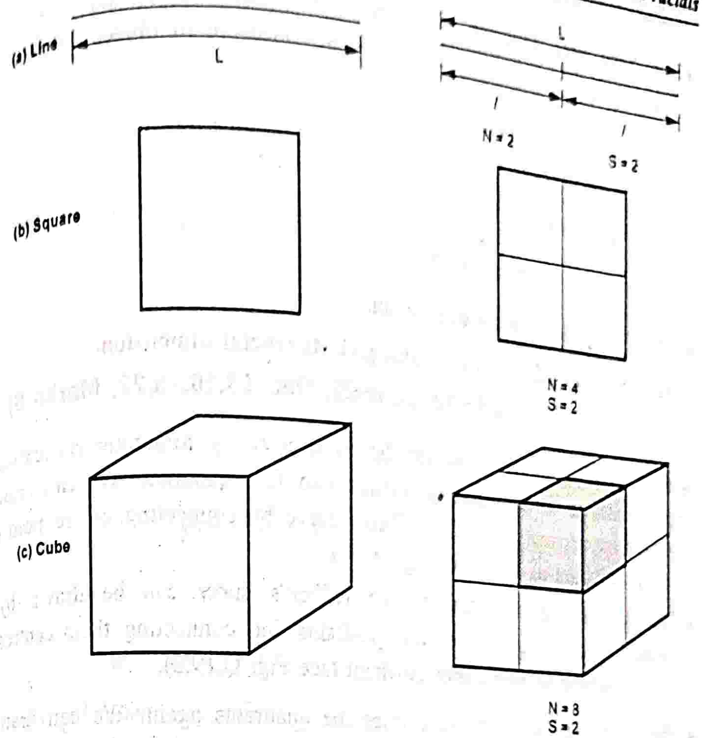


Fig. Q.18.1 Scaling of objects in various dimensions

$$s = \frac{L}{l}$$

- From above two equations we can write

$$N = s$$

i.e.

$$N = s^1$$

- In other words we can say that if we scale a line segment by a factor 1/s then we have to add N pieces together to get the original line segment. If we scale square object by a factor 1/s we will get a small square. In case of s = 2, we require 4 pieces of square to get original square. In general we can write

$$N = s^2$$

- Similarly for cubical object we have, (Refer Fig. Q.18.1)

$$N = s^3$$

- We have seen that we can specify the dimension of the object by variable D. Here the exponent of s is a measure of object dimension. Thus we can write

$$N = s^D$$

Solving for D we get

$$D = \log N/\log s$$

- This D is called **fractal dimension.**

Q.19 Explain Hilbert's curve and give its fractal dimension.

[SPPU : May-06,15,19, June-22, Dec.-08,10,15,22, Marks 6]

Ans. : • The Hilbert's curve can be constructed by following successive approximations. If a square is divided into four quadrants we can draw the first approximation to the Hilbert's curve by connecting centre points of each quadrant as shown in Fig. Q.19.1.

- The second approximation to the Hilbert's curve can be drawn by further subdividing each of the quadrants and connecting their centres before moving to next major quadrant (see Fig. Q.19.2).
- The third approximation subdivides the quadrants again. We can draw third approximation to Hilbert's curve by connecting the centres of finest level of quadrants before stepping to the next level of the quadrant (See Fig. Q.19.3).

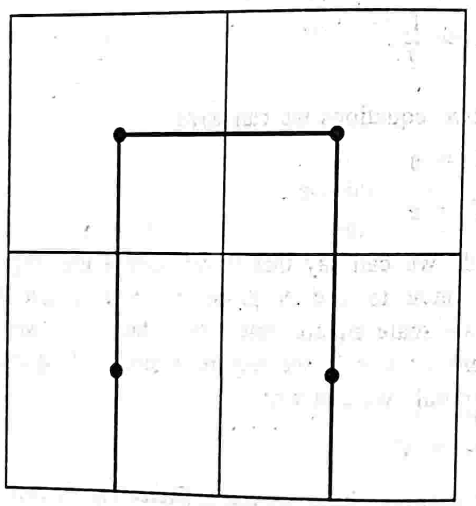


Fig. Q.19.1 The first approximation to Hilbert's curve

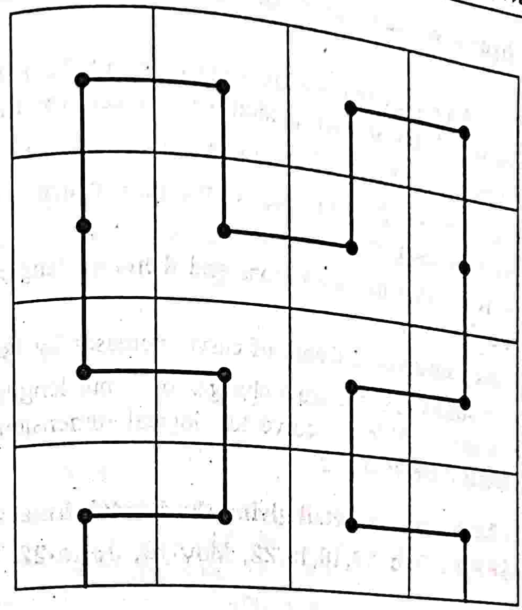


Fig. Q.19.2 Second approximation to Hilbert's curve

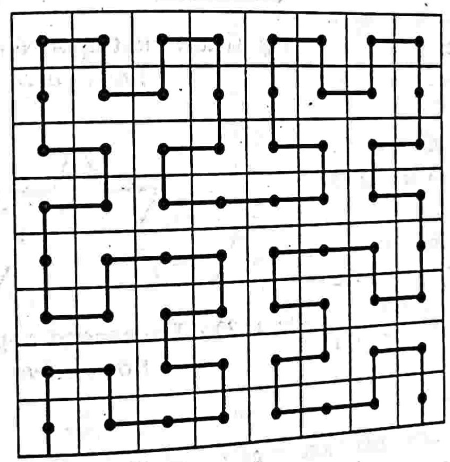


Fig. Q.19.3 Third approximation to Hilbert's curve

From three figures we can easily note following points about Hilbert's curve.

1. If we infinitely extend the approximations to the Hilbert's curve, the curve fills the smaller quadrants but never crosses itself.
2. The curve is arbitrarily close to every point in the square.
3. The curve passes through a point on a grid, which becomes twice as fine with each subdivision.
4. There is no limit to subdivisions and therefore length of curve is infinite.
5. With each subdivision length of curve increases by factor of 4.
6. At each subdivision the scale changes by 2 but length changes by 4 therefore for Hilbert's curve topological dimension is one but the fractal dimension is 2.

Q.20 Explain koch curve in detail giving the fractal dimension.

☞[SPPU : Dec.-17,18,19,22, May-19, June-22, Marks 7]

Ans. : • The Koch curve can be drawn by dividing line into 4 equal segments with scaling factor 1/3 and middle two segments are so adjusted that they form adjacent sides of an equilateral triangle as shown in the Fig. Q.20.1. This is the first approximation to the koch curve.

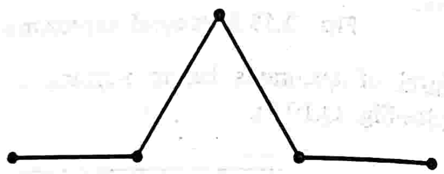


Fig. Q.20.1 First approximation of the Koch curve

• To apply the second approximation to the Koch curve we have to repeat the above process for each of the four segments. The resultant curve is shown in Fig. Q.20.2.

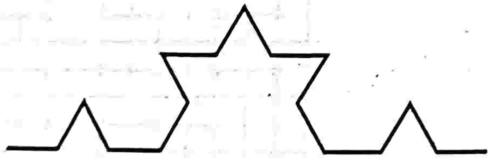


Fig. Q.20.2 The second approximation to Koch curve

• The resultant curve has more wiggles and its length is 16/9 times the original length.

• From the above figures we can easily note following points about the koch curve :

1. Each repetition increases the length of the curve by factor 4/3.
2. Length of curve is infinite.
3. Unlike Hilbert's curve, it doesn't fill an area.
4. It doesn't deviate much from its original shape.
5. If we reduce the scale of the curve by 3, we find the curve that looks just like the original one; but we must assemble 4 such curves to make the original, so we have

$$4 = 3^D$$

Solving for D we get

$$D = \log_3 4 = \log 4 / \log 3 \cong 1.2618$$

• Therefore for koch curve topological dimension is 1 but fractal dimension is 1.2618.

• From the above discussion we can say that point sets, curves and surfaces which give a fractal dimension greater than the topological dimension are called **fractals**. The Hilbert's curve and koch curves are fractals, because their fractal dimensions (respectively, 2 and 1.2618) are greater than their topological dimension which is 1.

11.9 : Fractal Lines

Q.21 Write short note on fractal lines.

☞[SPPU : May-10, 11, 12, Dec.-12,13, Marks 4]

OR Explain fractal lines with example.

☞[SPPU : Dec.-15,18, May-16,19, Marks 3]

Ans. : • Fractal line can be generated by performing following steps :

1. If the straight line segment is specified by points (x_1, y_1, z_1) and (x_2, y_2, z_2), find the midpoint of line by using following expression :

$$\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}, \frac{z_1+z_2}{2}\right)$$

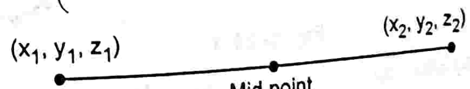


Fig. Q.21.1

2. Add an offset to the midpoint such that the resultant midpoint should not lie on the line itself. This can be achieved by adding offset term to each co-ordinate as follows :

$$\left(\frac{x_1+x_2}{2}+dx,\ \frac{y_1+y_2}{2}+dy,\ \frac{z_1+z_2}{2}+dz\right)$$

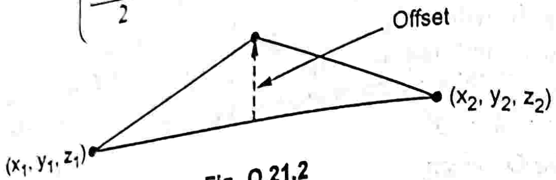


Fig. Q.21.2

In order to get random effect calculate the offset as shown below.

$$dx = L \times W \times GAUSS$$
$$dy = L \times W \times GAUSS$$
$$dy = L \times W \times GAUSS$$

where L : Length of the segment

W : Waiting function governing the curve roughness (i.e. fractal dimension)

GAUSS : Gaussian variable which returns random values between –1 and 1 with 0 mean.

i.e. returned values consist of equal amount of positive and negative values.

3. The shifted midpoint divides the original line into two parts. Repeat the same process for each part separately.
4. Repeat the process 1, 2, 3 until the segments become small enough. By following the above procedure we can get fractal line as shown in the Fig. Q.21.3.

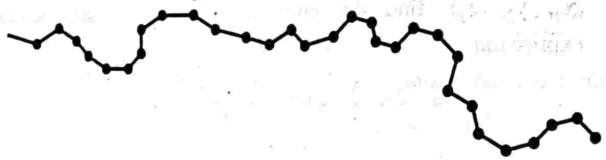


Fig. Q.21.3

- The above implementation can easily be achieved using a recursive procedure.

Q.22 Explain how fractals are used to generate fractal surfaces.

☞ [SPPU : Dec.-16,19, Marks 7]

Ans. : • The concept of fractal lines can be extended to generate fractal surfaces.

- There are many ways to accomplish this. Here we will use method base on triangles.
- We know that triangle has three vertex points. We can generate a fractal surface for the area between them. This is illustrated in Fig. Q.22.1.
- As shown in the Fig. Q.22.1 (a), we can apply fractal line algorithm to each edge of the triangle to divide it into two line segments. By connecting the half way points of each line segments we can divide the original triangle into 4 smaller triangles as shown in Fig. Q.22.1 (b).

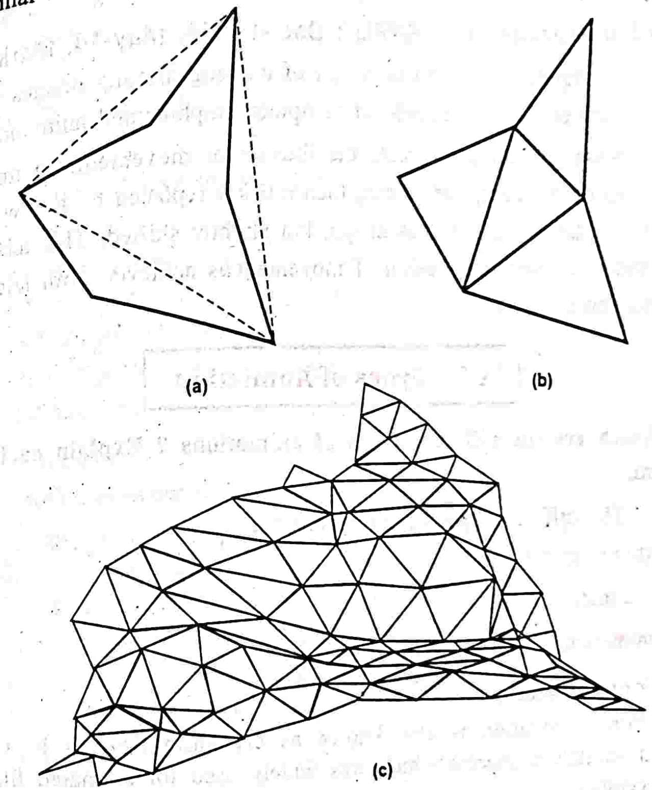


Fig. Q.22.1 Fractal surfaces

END...

UNIT - V

12 ANIMATION

12.1 : Basics of Animation

Q.1 What is computer animation ? ☞ [SPPU : Dec.-10,13,17, Marks 4]

OR Define animation. ☞ [SPPU : Dec.-11, 16, May-14, Marks 2]

Ans. : • Computer animation is the art of creating moving images via the use of computers. It is a subfield of computer graphics and animation.

• In computer animation to create the illusion of movement, an image is displayed on the computer screen, then quickly replaced by a new image that is similar to the previous image, but slightly shifted. This technique is identical to how the illusion of movement is achieved with television and motion pictures.

12.2 : Types of Animation

Q.2 Which are the different types of animations ? Explain each one of them.

Ans. : The different types of animations are :

- Traditional animation
- Path animation
- 3D animation

Traditional animation

- Traditional animation is also known as **cel animation.** It is a hand drawn animation process which was widely used for animated films of 20th century.

- Cels were used for this purpose. They are transparent films on which pictures were drawn, traced or photocopied and used in movies.
- These cels are placed one above the other to produce a single animation frame.
- This is a convenient method when it is required to change or isolate certain parts of images between the frames.
- The look of the traditional cel animation is preserved even now and work has remained same over the past 70 years.
- **'Tradigital'** term is used to describe cel animation by several animators, which makes extensive use of computer technology.
- **Full animation** is a process in which high quality traditionally animated movies that use detailed drawings and plausible movement, have a very smooth animation.
- Fully animated movies having various styles can be made from more realistically animated works. For example Aladdin, Beauty and Beast, etc.
- Most of the **Disney animated** films are very good examples of full animation.
- **Limited animation** is a type of animation which makes less use of detailed and stylized drawings and methods of movement are usually of skippy or choppy animation.
- Pinocchio, The Illusionist and Animal Farm are examples of traditionally animated movies.

Path animation

- Path animation evolved with the advent of computers.
- In this, the animation program stores mathematical entities known as **vectors.** The animation is represented by vectors unlike frames in traditional animation.
- This animation makes use of sprite and spline. **Sprite** is a collection of images which moves like an independent object. For example, a bouncing ball, flying kite, spinning wheel, rotating planet and so on.
- **Spline** is a motion path having any shape, along which the sprite moves.
- These splines are represented as a set of equations and define how the movement of sprite takes place.

- The shape of the curve can be controlled by anchor points, through which the spline passes.
- When the animation is played back, the vectors are used to generate a path along which the sprite moves.
- Its duration and speed can be controlled using other parameters. **Sprite animation** is another name used for path animation.
- Execution of these animations is dependent on software programs which are able to interpret the mathematical equations and display the images on the screen.
- Since **vector representation** is used, which is a compact process, path animation occupies very little space for storage as compared to traditional animation.

2D Vs 3D animation

- Computer animation makes use of digital creation and representation of animated movies, making use of variety of techniques.
- 2D animation is more focused on image manipulation and representation on a flat surface.
- **Cartesian coordinates** are used to represent 2D animations and it does not take into account the depth of objects.
- **2D bitmap graphics** or 2D bitmap graphics is used to create and edit the animated figures.
- It makes use of computerized automated versions of traditional animation techniques like onion skinning, interpolated morphing and interpolated rotoscoping.
- 2D animation has applications in power point animation, flash animation and analog computer animation.
- 3D animation is more complex and is represented using three axis.
- It is used to present a virtual world in which objects and characters interact with each other.
- It models, renders, adds lights and surface properties along with camera motions in a space coordinate.
- In the beginning, the animator creates a **3D polygon mesh**. This mesh is usually made of many vertices which are interconnected by edges and faces.

- Sometimes, a structure known as armature which is given as an internal digital skeletal structure is used to control the mesh by weighting the vertices.
- This process can be used along with keyframes to create movements which are well known as **rigging.**
- Few other techniques, like water and fire simulations, mathematical functions and simulated fur or hair can be used to add special effects. All of these techniques come under the 3D dynamics category.

12.3 : Principles of Animation

Q.3 Which are the basic principles of animation ? Explain each one in detail. ☞ **[SPPU : Dec.-17, Marks 6]**

Ans. : • The basic principles of animation are :

- Squash and stretch
- Slow-in and slow-out
- Maintaining 3D effects.

Squash and stretch

- The first principle, namely, squash and stretch is used to indicate the physical properties that are expected to get changed in the process of animation.
- For example, a rubber ball when dropped on the floor might have to be shown as getting distorted in its shape due to its **elasticity**.
- A **bouncing rubber ball** might be shown as elongating as it approaches the floor, flattening out when it hits and then elongating again as it rises.
- By contrast, a **metal sphere** hitting the floor might distort very little but might wobble after the impact, exhibiting very small, high-frequency distortions.
- These **distortions have to be calculated** depending on the motion and the environments in which the animations are planned.

Slow-in and slow-out

- Second principle is to use slow-in and slow-out to help **smooth interpolations.**

- Sudden, jerky motions are extremely distracting. This is particularly evident in interpolating the camera position.
- An audience viewing an animation identifies with the camera view, so sudden changes in camera position may make the audience feel motion sickness. Thus, camera changes should be as smooth as possible.

Maintaining 3D effects

- Third principle that carries over naturally from the 2D character-animation world to 3D animations is to stage the action properly.
- This includes choosing a view that conveys the most information about the events taking place in the animation and isolating events so that only one thing at a time occupies the viewer's attention.
- These three are the basic thumb rules to be followed for generating a good animation sequence.

12.4 : Design for Animation Sequences

Q.4 What are the steps in design in animation sequence ? Describe about each step briefly.

☞ [SPPU : Dec.-19, 22, May-18, May-19, Marks 7]

Ans. : • **Design of animation sequences** : The steps for designing animation sequences are as follows :

- Storyboard layout
- Object definitions
- Keyframe specifications
- Generation of in-between frames

- **Storyboard layout** : The storyboard is an outline of the action. It defines a set of basic events that are to take place in a specific order. Such an ordered set of events gives the motion sequence. Usually storyboard consists of rough sketches or it could be a list of the basic ideas for the motion.
- **Object definition** : Each active section of the scene is treated as an object. It is defined in terms of basic shapes, such as polygons or splines. Along with the object the motion associated with the object is also defined.
- **Keyframe specification** : A keyframe is a detailed drawing of the scene at a certain time in the animation sequence. They are positioned according to the time for that frame. Usually, some keyframes represent

the extreme positions in the action and others are spaced so that the time interval between keyframes is not too great.

- **In-between frames** : These are the intermediate frames between the keyframes. Usually there are 3 to 5 in-between frames between the keyframes. Suppose we want to design an animation sequence for 10 seconds. For this we need to have 24 × 10 frames because film requires 24 frames per second. Out of these 240 frames we can have 48 keyframes and remaining 192 in-between frames. In this case there are four in-between frames between two keyframes.

12.5 : Animation Languages

Q.5 Write a note on animation languages.

☞ [SPPU : Dec.-12,13,22, Marks 8]

OR Give different types of animation languages.

☞ [SPPU : Dec.-11, Marks 2, June-22, Marks 6]

Ans. : • There are many different languages for describing computer animation, and new ones are constantly being developed. These animation languages can be categorized as :

- Linear-list notations
- General-purpose languages with embedded animation directives.
- Graphical languages

Linear-list notations : • In linear-list notations, each event in the animation is described by a starting and ending frame number and an action that is to take place.

General-purpose languages : • General purpose languages such as C, C++, Pascal or Lisp can be used to design and control the animation sequences.

Graphical languages : • There are several specialized animation languages developed called **graphical languages**.

- These languages provide various animation functions which make it easy to design and control the animation.

12.6 : Keyframes and Morphing

Q.6 What is Morphing ? [SPPU : Dec.-12,22, May-13, Marks 2]

Ans. : • Morphing is a special effect in motion pictures and animations that changes one image into another through a seamless transition.

• It is a transformation of object shapes from one form to another.

Q.7 What are the applications of morphing ? [SPPU : Dec.-12,22, May-15, Marks 3]

Ans. : Morphing is used to produce special effects in motion pictures and animations that changes one image into another through a seamless transition. It is used as a transition technique between one scene and another in television shows, even if the contents of the two images are entirely unrelated. It is commonly used in entertainment industry to produce special effects.

Q.8 What is tweening ?

Ans. : • Tweening is a short for in-betweening. It is the process of generating intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image.

• It is an interpolation technique where an animation program generates extra frames between the key frames that the user has created. This gives smoother animation without the user having to draw every frame.

• In tweening a scene is described by a mathematical model - a set of two- or three-dimensional objects whose positions are given by sets of coordinates. It uses mathematical formulae to generate these coordinates at a sequence of discrete times.

12.7 : Motion Specification

Q.9 Explain various methods to specify the motion of the objects.

Ans. : • The various ways in which the motions of objects can be specified. These are :

- Direct motion specification
- Goal-directed systems
- Kinematics and Dynamics

• **Direct motion specification :** ◦ It is most straightforward method for defining a motion sequence.

◦ In this method, the rotation angles and the translation vectors are specified so that the geometrical transformations can be applied to the objects in the scene to generate animation sequences.

• **Goal-directed systems :** ◦ In these systems, instead of specifying motion parameters, action goal specific instructions are specified. Thus these systems are known as goal directed systems.

◦ For example, we could specify that we want an object to **walk** or to run to a particular destination.

• **Kinematics and Dynamics :** ◦ In case of kinematic descriptions, motion parameters such as position, velocity and acceleration are specified without reference to the forces that causes the motion to generate animation sequences.

12.8 : Methods of Controlling Animation

Q.10 Describe the various methods used to control animation. [SPPU : Dec.-22, Marks 6]

Ans. : There are special methods used for controlling animation. These are :

• **Full explicit control :** In this control, the animator explicitly describes the position and attributes of every object in a scene by means of translation, rotation and other position and attribute changing operators.

• **Procedural control :** In this control, various elements of the model communicate in order to determine their properties.

• **Constraint based system :** In this control, the motion of object is modeled by constraints.

• **Tracking live action :** In this control, the animation is created by replacing the human actors with their animation equivalent. This technique provides realistic motion.

• **Actors :** It is a high-level form of procedural control in which actor (a small program) is invoked once per frame to determine the characteristics of some objects in the animation.

• **Kinematics and dynamics :** In this control, the positions and velocities of points (kinematics) are described and the physical laws that governs the kinematics (dynamics) are considered to control the motion of objects.

- **Physically based animation :** In this control, the physical based models based on simulations of the evolution of physical systems are used to control the motion of objects.

Q.11 Explain Full explicit control method of controlling animation.

Ans. : • Full explicit control is the simplest sort of animation control. Here, the animator provides a description of everything that occurs in the animation. This includes the information about scaling, translation, rotation, key frames, interpolation methods used to generate in-between frames.

- In these systems, a sequence of actions defined between key-frames may be difficult to modify; extending one action may require shortening the neighbouring actions to preserve the coherence of the animation. For example, consider an animation in which one ball roll up and hits another, causing the second ball to roll away. If the first ball is made to move more quickly, the start of the second action (the second ball rolling away) must start early.

Q.12 What is the procedural control method of controlling animation.

Ans. : • In procedural control various elements of the model communicate in order to determine their properties. Such control is ideally suited to control the animation. For example, we can animate the position of grass blades according to wind flow. In such animation wind particles are involved to determine the positions of the grass blades. Thus, the particle system describing the grass was affected by aspects of other objects in the scene. This sort of procedural interaction among objects can be used to generate motion that would be difficult to specify through explicit control.

Q.13 What are the constraint-based systems for controlling animation ?

Ans. : • In physical world, some objects move in a straight line; however, some objects move in a manner determined by the other objects with which they are in contact. Such motion of object is not linear. For example, a ball rolls down an inclined plane. If gravity were the only force acting on the ball, the ball would fall straight down. However, the plane is also pushing up and side ways and so the ball rolls down the plane. It is possible to model such motion by constraints.

- Constraints provide a unified method to build objects and to animate them. The models assemble themselves as the elements move to satisfy the constraints. Constraints provide a way to specify the behavior of physical objects in advance without specifying their exact positions, velocities etc. In other words constraints are partial descriptions of the objects desired behavior. So, given a constraint, we must determine the forces to meet the constraint and then find forces to maintain the constraint. A good deal of research has been done towards the use of constraint-based methods to create realistic animation. Many constraint-based modeling systems have been developed :
- **Point-to-nail constraint :** It is used to fix a point on a model to a user-specified location in space.
- **Point-to-point constraint :** It is used to attach two points on different bodies to create complex models from simpler ones.
- **Point-to-path constraints :** It requires some points on a model to follow an arbitrary user-specified path.
- **Orientation constraints :** It is used to align objects by rotating them.

12.9 : Frame by Frame Animation Techniques

Q.14 What is frame-by-frame animation ?

☞ [SPPU : June-22, Marks 6]

Ans. : • Frame-by-frame animation is a technique that creates the illusion of movement by making incremental changes between every keyframe.

- Frame-by-frame animation is in direct contrast with digital techniques such as rigging, whereby a skeleton model is created for a character to allow it to move.
- Throughout most of the 20th century, animating frame-by-frame was the only method of producing animation. Although it's no longer necessary, it's still used by many animators for the following two purposes :
  - **Tradition :** To replicate the traditional qualities of hand-drawn animation.
  - **Practicality :** Certain types of animation (such as stop motion or rotoscope) can only be produced frame-by-frame.

Q.15 Explain the types of frame-by-frame animation.

Ans. : • There are four main types of frame-by-frame animation :

- Flipbook Animation
- Traditional Animation

- Stop Motion Animation
- Rotoscope Animation

Flipbook Animation

- Flipbook animation is the earliest and most simple form of frame-by-frame animation.
- This is where each frame is drawn on a separate page of a book. The pages are then flipped through rapidly in order to create the illusion of a continuous motion sequence.

Traditional Animation

- Traditional animation (also known as cel animation or hand-drawn animation) is 2D animation that's drawn by-hand, one frame at a time. It served as the dominant form of animation for most of the 20th century.
- As each frame must be drawn entirely from scratch, it's incredibly time-consuming and requires a high-level of artistic skill to produce.

Stop Motion Animation

- Stop motion animation is a technique used to give inanimate objects the illusion of movement. This is done by incrementally moving the objects in-between frames.
- Although any type of inanimate object can be used for stop motion, plasticine models are most-commonly used due to their flexibility of movement. A stop motion production that exclusively uses plasticine models is known as Claymation.

Rotoscope Animation

- Rotoscope animation is a type of 2D animation where animators trace over live action footage frame-by-frame.
- This not only saves a significant amount of time when animating by-hand, but also results in a more realistic sense of movement in the finished product.

Q.16 State the advantages and disadvantages of frame-by-frame animation.

Ans. : Advantages of frame-by-frame animation : The three main advantages of frame-by-frame animation are :

1. **It's Traditional :** As frame-by-frame animation dominated much of the 20th century, it serves as one of the most recognizable and beloved forms of animation.

2. **It Offers More Creative Control :** Creative control is arguably the greatest benefit of frame-by-frame animation.
3. **It's Fantasy-Like**

Disadvantages of Frame-By-Frame Animation

- The three main disadvantages of frame-by-frame animation are :

1. **It's Time-Consuming**
2. **It's Expensive :** Frame-by-frame animation not only takes a significant amount of time, but also requires a talented and experienced animator to produce the project.
3. **It's Not Realistic-Looking :** Frame-by-frame animation by nature is slightly less convincing in terms of realism.

12.10 : Real Time Animation Techniques

Q.17 Explain raster animation. ☞ **[SPPU : Dec.-12, Marks 4]**

Ans. : • We can generate real-time animation using raster operations.

- To generate real-time animation we have to execute sequence of raster operations. For example, we can rotate a object like square box, as shown in the Fig. Q.17.1. In each iteration the box is rotated by some angle. The resulted position of the box after rotation generates new frame. By continuing such rotations we can generate number of frames and the effect is rotation of box.

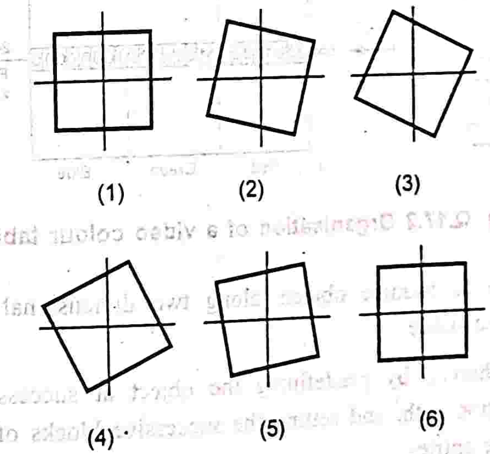


Fig. Q.17.1 Frames generated using raster operations

- We can also animate objects along two-dimensional motion paths using the colour-table transformations.
- Let use see how we can generate such an animation.
- In colour displays, 24-bits per pixel are commonly used, where 8-bits represent 256 levels for each colour. Here, it is necessary to read 24-bits for each pixel from frame buffer.
- This is very time consuming. To avoid this video controller uses look-up-table (LUT) to store many entries of pixel values in RGB format. This look-up-table is commonly known as **colour-table.** With this facility, now it is necessary to only read index to the colour-table from the frame buffer for each pixel. This index specifies the one of the entries in the colour-table. The specified entry in the colour-table is then used to control the intensity or colour of the CRT.
- Usually, colour-table has 256 entries. Therefore, the index to the colour-table has 8-bits, and hence for each pixel frame buffer has to store 8-bits per pixel instead of 24-bits. Fig. Q.17.2 shows the organisation of a video colour-table.

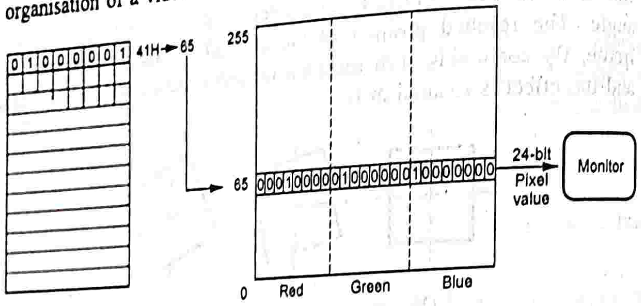


Fig. Q.17.2 Organisation of a video colour table

- It is possible to animate objects along two dimensional motion path using the colour-table.
- It can be achieved by predefining the object at successive positions along the motion path, and setting the successive blocks of pixel values to colour table entries.

- Initially, the pixels at the first position of the object are made ON, and the pixels at the other positions of the object are set to the background colour. The animation is then accomplished by changing the colour-table values so that the object is ON at successively positions along the animation path.
- Each time the preceding pixels at the preceding position are set to the background colour.

END...

Unit - VI

13 INTRODUCTION TO VIRTUAL REALITY

13.1 : Fundamental Concept

Q.1 What is virtual reality ?

Ans. : • Virtual Reality (VR) is a high-end user computer interface that involves real time simulation and interactions through the senses.

- Virtual reality, sometimes referred to as **immersive multimedia,** is a computer-simulated environment that can simulate physical presence in places in the real world or imagined worlds.
- Virtual reality is the term used to describe a three-dimensional, computer generated environment which can be explored and interacted with by a person. That person becomes part of this virtual world or is immersed within this environment and whilst there, is able to manipulate objects or perform a series of actions.
- Virtual reality can recreate sensory experiences, including virtual taste, sight, smell, sound, touch, etc.
- Various sensory stimuli such as **sound, video** and **images** form part of most virtual reality environments.
- The virtual reality has applications that provide solutions to real world problems in engineering, medicine, military, entertainment etc.

13.2 : The Three I's of Virtual Reality

Q.2 Give short note on three I's of virtual reality.

Ans. : • **Interaction :** In terms of functionality, VR is responsive to the user's input. Users not only see and manipulate graphic objects on the screen; they also touch and feel those using input and output devices. The input devices like keyboard, mouse or multimodal devices like wired glove, boom arm, etc. help the user to interact with the virtual environment.

- **Immersion :** Immersion is what makes VR feel real to the user. A computer-simulated environment can simulate physical presence in places in the real world or imagined world. The users feel immersed in a simulated world, via hardware e.g., headsets and software.
- **Imagination :** It is the user's mind capacity that makes it possible to perceive non-existent things and create the illusion of them being real.

13.3 Classic Components of VR Systems

Q.3 Draw block diagram of VR system and explain the classic components of it.

Ans. : • Fig. Q.3.1 shows a typical VR system. It includes six main components :

- The virtual world
- Virtual reality engine (Simulation engine, Graphics engine)
- Software and databases
- User interface
- Input devices
- Output devices

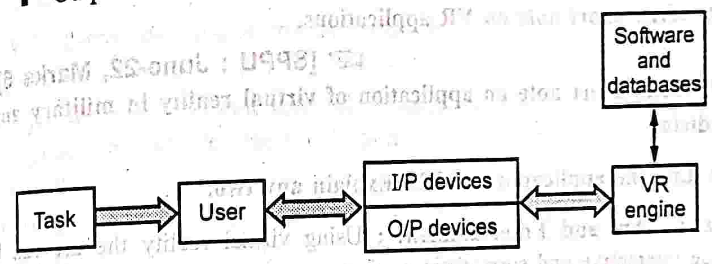


Fig. Q.3.1 VR system

- The virtual reality engine is a key component of any VR system. It consists of simulation engine and graphics engine. It reads its input through user interface using input device, accesses task-dependent databases, performs the required real-time computations to update the state of the virtual world and feeds the results to the output device.
- The components of virtual system are divided onto two parts :
  - **Internal components :** Virtual world, virtual reality engine (Simulation engine, graphics engine), software / databases and user interface.
  - **External components :** Inputs and outputs.

- A **scene database** includes geometric representations and attributes for all objects within the environment. The format of this representation depends on the graphics engine and simulation engine used.
- The **graphic engine** generates the image that is seen by the user. While doing so the scene database, viewer's current position and orientation is taken into consideration. Additional effects such as textures, sounds, special effects etc. enables the user to feel that he / she is observing that object from a particular point.
- The **simulation engine** performs main task i.e. it concerns about the dynamics of the environment. It looks after the change in the environment over time and how it responds to the user's actions. This comprises of handling interactions, programmed object actions, physical simulations or user actions.
- The **user interface** controls the user navigation and interaction with the virtual environment. It acts as buffer between the virtual world software and the input and output devices being used.

13.4 : Applications of VR Systems

Q.4 Write short note on VR applications.

☞ [SPPU : June-22, Marks 6]

OR Write short note on application of virtual reality in military and medicine.

OR List the applications of VR. Explain any two.

Ans. : **Art and Entertainment :** Using virtual reality the art can be made interactive and surrealistic art form.

- Multimedia enables mixing of different media on a computer screen. The VR art can integrate hitherto disparate art from like painting, music and moving pictures.
- VR enables user to visualize and fantasies the art as a science fiction to date.
- VR is used in the games to enable the user to immerse in the game and enjoy it.
- **Education :** VR system enables to visualized the matter to students which makes them easy to understand and grasp the things.

- **Training and Simulation :** In case of training, there is risk of accident causing life or property loss and training is a very costly matter. Use of VR helps in training with low cost and less risk of life or property loss.
- **Classroom teaching :** VR increases the student teacher interaction in the process of learning in a classroom.
- **Virtual classroom :** The telepresence concept is used for virtual classroom, where the students at different cities/towns/nations form a virtual classroom.
- **Manufacturing and Architecture :** The field of architecture and CAD/CAM based projects are benefitted a lot due to virtual reality.
- VR helps the designer as well as the client to visualize the structure that is to be built.
- **Science :** VR system enables scientists and technologists to visualize data in 3D world (for example 3D models of genes, visual impact of 3D equations, etc.)
- **Medicine and Surgery :** VR provides an efficient human-computer interfaces, 3D visualization and modeling tools which helps in medical diagnosis.
- VR system can assist the surgeon while performing operation of a complex anatomy.
- **Robotics :** A telerobot uses telepresence in such a way that a human operator at remote control station controls the movements of the telerobot.
- **Military :** With the help of VR, battlefield training can be given to the solider.
- VR can be used for military purpose for :
  - Flight simulation
  - Battlefield simulation
  - Medical training
  - Vehicle simulation
  - Virtual boot camp.

END...

Unit - VI

14 MULTIPLE MODALS OF INPUT AND OUTPUT INTERFACE IN VIRTUAL REALITY

14.1 : 3D Position Trackers

Q.1 Write a note on 3D position trackers. **[SPPU : June-22, Marks 5]**

Ans. : • **Definition :** Position tracking is a real-time measurement of the three-dimensional position and three-dimensional orientation of a moving object.

• Position tracking is required in virtual environments to control computer-generated stimuli and in teleoperators to control the behaviour of the telerobot.

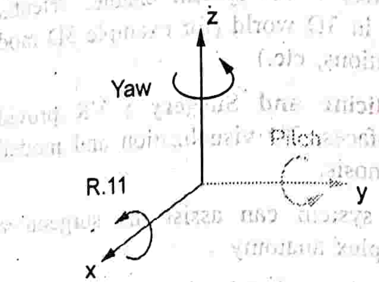


Fig. Q.1.1 System of coordinates of a moving 3D object

• Tracker is a special purpose hardware used in VR to measure the real-time change in a 3D object position and orientation.

• Usually, VR systems require tracking of the head, the hand or fingers. However, in some applications position tracking of the eyes, the torso, and the arms and legs may also be required.

• The position and orientation of a particular body part is monitored by a VR system with the help of **tracking devices.** For example, the HMD device is used to measure position and orientation of head. Data gloves and flying joysticks trackers are used so that the virtual hand can follow the positional and orientation changes of the user's real hand.

• There are two types of tracking technology :

- Position and / or orientation trackers and
- Angle trackers

• **Position and/or orientation** trackers provide the position and/or orientation of object and angle trackers provide the angle between two objects.

• Position and/or orientation trackers are also known as **6-degree-of-freedom (6DOF)** devices. They measure position i.e. x, y and z co-ordinates and the orientation i.e. yaw, pitch and roll with respect to the reference point.

Q.2 What are the hardware components of tracker ?

Ans. : • The hardware of trackers mainly consists of :

- **A source :** It generates signal
- **A sensor :** It receives signal
- **A control box :** It processes the signal and communicates with the computer.

• Either of the source or sensor is attached to the body with the other fixed at a particular spot in the environment and acts as a reference point. The technology being used decides whether the source or the sensor is to be attached to the body.

14.2 : Types of Tracking Devices

Q.3 List the types of tracking devices.

Ans. : • The types of tracking devices are :

- Mechanical trackers
- Magnetic trackers
- Ultrasonic trackers
- Optical trackers
- Hybrid inertial trackers

Q.4 Explain mechanical trackers in detail.

Ans. : • Mechanical tracker/mechanical link trackers was the first tracker developed to use in VR system. A simple mechanical tracker can take the form of mechanical arm jointed at the shoulder, elbow and wrist. When one end is fixed, the 3D position of the other end is readily calculated by measuring the joint angles using suitable sensors.

- Thus mechanical trackers are equivalent to robot arm and includes jointed structure with rigid links, a supporting base and an active end that is connected to the body part that is to be tracked (mostly the hand).

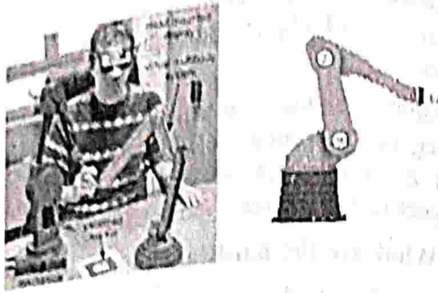


Fig. Q.4.1 Mechanical tracker

Q.5 State the advantages and disadvantages of mechanical tracker.

Ans. : Advantages

- They are highly accurate.
- Have extremely low latencies.
- They are immune to interference from magnetic fields.
- They are not susceptible to jitter.
- They provide high update rate.

Disadvantages

- They are bulky.
- They have restricted area of operation.
- They limit the user's freedom of motion.
- They have usually cumbersome physical structure which can get in the way of task performance.

Q.6 Describe electromagnetic trackers along with their advantages and disadvantages.

Ans. : • **Electromagnetic trackers/magnetic trackers** are non-contact 3D measurement devices. In the 3D measurement procedure, the user's freedom of motion in process tracking led to development of electromagnetic trackers.

- Several body parts can be tracked simultaneously with the help of electromagnetic tracker. The functioning of this tracker is correct until the objects come between the source and the detector.

- Magnetic tracker consists of a stationary transmitter that produces magnetic field. This magnetic field is used to determine the real-time position of a moving receiver.
- Magnetic tracker produces three electromagnetic field that are **mutually perpendicular**. The detector on the user's body measures field attenuation i.e. the strength and direction of electromagnetic field and sends this information back to computer. The computer triangulates the distance and orientation of the three perpendicular axis in the detector corresponding to the three electromagnetic fields produced by the source.

Q.7 State the advantages and disadvantages of electrmagnetic tracker.

Ans. : Advantages

- High update rate
- Low latency
- Good precision
- No line of sight restriction between transmitter and receiver.

Disadvantages

- Sensitive to magnetic interference, metallic objects and ferromagnetic materials.
- Tracking accuracy decreases with increase in distance between detector and source. Thus they have limited range typically 1 to 3 meters.

Q.8 Write a note on ultrasonic trackers.

Ans. : • In this tracker there is a stationary transmitter that generates ultrasonic signals to determine the real-time position of a moving receiver.

- Ultrasonic tracker is a non-contact position measurement device, and it is also known as **acoustic trackers.**
- In ultrasonic trackers, there are three high frequency emitters that emit sound waves in a rigid formation. This sound wave emitters acts as source for the receivers mounted on the user.
- **Two ways of determining position and orientation using acoustic trackers :**
  - Phase coherence
  - Time-of-flight.

- In **phase coherence** method, the phase difference of the source waves received by the receiver and that produced by the emitter is measured by the computer so as to determine the position and orientation. The system can update the position until the distance traveled by the target (sound wave) is less than one wavelength between the two updates.
- In **time-of-flight** method, the time required by the sound transmitted (by transmitter) to reach sensors is measured. The position is calculated by using only one transmitter whereas to determine the orientation the difference between three sensors signals is considered.

Q.9 State the advantages and disadvantages of ultrasonic trackers.

Ans. : Advantages

- Inexpensive
- Not sensitive to magnetic interference, metallic objects and ferromagnetic materials.
- Wireless version of acoustic trackers are feasible.

Disadvantages

- Moderate precision
- Low update rate
- Sensitive to acoustic interference, i.e. acoustic noise and echoes.
- Susceptible to line-of sight error i.e. the transmitter and receiver should be within their line-of-sight.
- Time-of-flight trackers have low update rate whereas phase coherence trackers undergo error accumulation over the time.
- Acoustic trackers are affected by changes in temperature, pressure and humidity level of the work environment.

Q.10 Write a note on optical trackers.

Ans. : • Infrared trackers are also known as **optical trackers.**

- They are non-contact measurement devices that use optical sensors to determine the real-time position or orientation of the object.
- IR trackers have many stationary emitters fixed in a rigid arrangement and cameras or quad cells that receive IR light. The position of tracker is fixed by the computer using triangulation technique.

Advantages

- No effect of metals.
- High update rate and low latency.

Disadvantages

- The emitters and cameras/quad-cells should be within their line-of-sight.
- High intensity lights, other IR light sources or other glares affect the accuracy of IR trackers.

Q.11 Explain hybrid inertial trackers in brief.

Ans. : • The inertial trackers do not have hardware or cabling between a computer and the tracker. Because of this they can be used in a comparatively large working volume.

- Inertial trackers are self-contained sensors used to measure the rate of change in an object orientation and object translation velocity.
- Inertial tracking uses gyroscopes and accelerometers to measure the accelerations of the tracker, which are integrated to provide the position, orientation and velocity.
- **Advantages**
  - Source less operation with theoretically unlimited range.
  - No line-of-sight constraints.
  - Very low sensor noise (jitter).
  - Because of low jitter, there is no need of additional time-consuming filtering which reduces latency.
- **Disadvantages**
  - Bulky
  - Expensive
  - High drift
  - Errors may occur due to continuous addition of relative values.

14.3 : Navigation and Manipulation Interfaces

Q.12 What is navigation and manipulation interface ?

☞ **[SPPU : Dec.-22, Marks 5]**

Ans. : Definition : A device that allows the interactive change in the view of virtual environment and exploration through the selection and manipulation of a virtual object under consideration is known as navigation/manipulation interface.

- **Navigation/Manipulation** involves selection, movement, and sometimes rotation of an object.
- The navigation/manipulation can be done either in absolute co-ordinates or relative co-ordinates. The trackers discussed in the previous section are absolute as they return the position and orientation of a moving object with respect to a fixed system of co-ordinates.
- User programmable pushbuttons and trackers are used as navigation /manipulation interfaces.

Q.13 List the types of navigation / manipulation interfaces.

Ans. : • **Tracker-based interface** : They provide more functionality to virtual reality simulations rather than simply measuring real-time position of the user body parts.

- **Trackballs** : It consists of a cylinder which senses/measures three forces and three torques applied by the user's hand.
- **3D probes** : Intuitive and inexpensive device which allows either absolute or relative position control of objects.

Q.14 Which are the sub task involved in the navigation task ?

Ans. : • The navigation process involves getting information of the current location of the object and its movement. Accordingly, navigation task is subdivided into two categories :

- **Wayfinding :** Finding orientation and setting pathways to travel towards the desired location within the virtual environment.
- **Travel :** It refers to actual movement from present location to desired location.

Q.15 What is Wayfinding ?

Ans. : • Wayfinding is nothing but a combine process of defining a path through an environment by using and acquiring spatial knowledge, aided by both natural and artificial clues.

- Wayfinding is different and more difficult task when performed in a virtual world as compared to that done in real world. This is because, in virtual world most of the times the perceptual clues and movement constraints are missing.

- Two techniques used to support wayfinding are : **User-centered technique** and **Environment-centered technique.**

Q.16 What is travel task ? Explain the types of travel task.

Ans. : • It is expected that the travel technique should allow user to easily move through the environment.

- **Three types of travel tasks are :**
  - **Exploration** : This tasks have no explicit goal of the movement. The user is simply exploring or browsing the 3D space, usually to build knowledge of the environment.
  - **Search** : This tasks involve traveling to a specific target location in the environment.
  - **Maneuvering** : This tasks usually involve short, precise movements where the goal is to change the viewpoint slightly in order to do a particular task.

Q.17 List the categories of travel techniques.

Ans. : • **Five categories of travel techniques :**

- **Physical locomotion technique** : It uses the user's physical exertion to transport himself through the virtual world. It mimics a natural method of locomotion in the real world. The three main techniques are :
  - Walking
  - Walking in place
  - Bicycles
- **Steering technique:** It provides continuous control of the direction of motion by the user.
- **Route planning technique** : It specifies the path. It is responsible for drawing a path, marking points along the path, and manipulating a user representation.
- **Target-based travel technique** : It specifies destination.
- **Manual manipulation technique** : Here, an object manipulation metaphor (e.g. HOMER) is used to manipulate the viewpoint.

14.4 : Gesture Interfaces

Q.18 Explain the gesture interfaces of VR system in detail.

Ans. : • Though navigation and manipulation interfaces are simple, compact and popular, they limit the freedom of motion of users to a small area. To overcome this problem, gesture interfaces are used.

• **Definition** : The devices that measures the real-time position of the user's fingers or wrists are called **gesture interfaces**. They allow natural, gesture-recognition based interaction in the virtual world.

• The examples of gesture interfaces are : Pinch glove, Data glove, Digi glove and cyber-glove.

• The human-machine interface is made possible with the help of gesture recognition done by the gesture interface. Gesture interfaces acts as an efficient substitute for input device like mouse, keyboard and even touch screen.

14.5 : Graphics Displays

Q.19 What is graphics displays.

Ans. : • **Definition** : A graphics display is a computer interface that presents artificial world images to user/users interacting with the virtual world.

Q.20 Give the classification of graphics displays.

Ans. : • Users involvement in the virtual world is mostly dependent on depth perception along with large viewing area and high resolution images. Accordingly, graphics displays are characterized.

- **Monoscopic or stereoscopic** : Monoscopic displays provide depth cues such as interposition, brightness (light and shade), perspective projection(size) and monocular motion parallax. On the other hand, stereoscopic display is capable of conveying depth perception to the viewer by means of stereopsis for binocular vision.
- **Resolution** : It specifies the ability of display number of pixels in the scene
- **Field of view** : It is the portion of the eye's viewing volume that display covers
- **Display technology used** : LCD/LED, CRT based etc.

- Display weight
- Cost

Q.21 Write a note on HMD. [SPPU : June-22, Marks 3]

Ans. : • **Head Mounted Display (HMD)**, as the name indicates it is a computer display that is mounted on user's head like a helmet or a set of goggles.

• A typical HMD has either one or two small displays with lenses and semi-transparent mirrors embedded in a helmet, eyeglasses (also known as data glasses) or visor. The display units are miniaturized and may include CRT, LCDs, Liquid Crystal on silicon (LCos), or OLED. Some vendors employ multiple micro-displays to increase total **resolution** and **field of view**.

• HMDs differ in whether they can display just a Computer Generated Image (CGI), show live images from the real world or a combination of both. Most HMDs display only a computer-generated image, sometimes referred to as a virtual image. Some HMDs allow a CGI to be superimposed on a real-world view. This is sometimes referred to as **augmented reality** or **mixed reality**.

Q.22 State the advantages of LCD in HMD.

OR Why are LCDs preferred in HMD ?

Ans. : **Advantages of LCD in HMD**

- Compact
- Light weight
- Efficient
- Inexpensive than CRT displays

• The CRT displays have more screen resolution and brightness as compared to LCDs. But due to the disadvantages of CRT displays discussed previously, LCDs are preferred as monitor screens in HMDs.

Q.23 Explain CAVE. [SPPU : June-22, Marks 2]

Ans. : • A CAVE is a computer assisted virtual environment. It is an immersive virtual reality environment where projectors are directed to three, four, five or six of the walls (acting as huge monitors) of a room-sized cube.

- It allows multiple users to share in the same experience simultaneously – enhancing teamwork, discovery and decision making.
- Due to large monitor screen (wall), the user get very wide field of view. The user can freely move around in the cave (room) without being connected to a computer at a particular place, wearing a goggles that are same as the 3D glasses.
- The wand is the major input device used to interact with and control a virtual reality (VR) experience in the CAVE. It is essentially a 3D mouse, with a receiving antenna attached which provides the computer with information about the wand's position and orientation. The wand consists of three programmable buttons and a joystick. The joystick is used primarily for navigation in combination with the position and orientation information. The buttons are used to set modes and select options.

14.6 : Sound Displays

Q.24 Describe various sound displays of a VR system.

☞ **[SPPU : Dec.-22, Marks 6]**

Ans. : • In VR, the sounds produced may be mono, stereo or 3D audio.

- **Mono sound :**
  - Recorded with one microphone; signals are the same for both ears.
  - Sound only at one point (0-dimensional), no sense of sound positioning.
- **Stereo sound :**
  - Recorded with two microphones several feet apart separated by empty space; signals from each microphone goes into one of the ear respectively.
  - Heard commonly through stereo headphones or speakers; typical multimedia configuration of personal computers.
  - Gives a better sense of the sound's position as recorded by the microphones, but only varies across one axis (1-dimensional), and the sound sources appear to be at a position inside the listener's head.

- **3D Sound :**
- Often termed as spacial sound, is sound processed to give the listener the impression of a sound source within a three-dimensional environment. This means that, it gives user the feeling that the recorded sound is actually coming from the user's surrounding.

14.7 : Haptic Feedback

Q.25 Write a note on haptic feedback.

☞ **[SPPU : Dec.-22, Marks 6]**

Ans. : • To overcome this limitation of human-computer interface, haptics introduced an additional sensory channel to the computer and that is the **sense of touch**. Due to this addition, the bidirectional communication between the humans and computers has become more efficient.

- Haptic technology is a **tactile feedback technology.**
- In haptic technology or haptics, the sense of touch is recreated by applying forces, vibrations or motions to the user. This mechanical stimulation can be used to assist in the creation of virtual objects in a computer simulation, to control such virtual objects, and to enhance the remote control of machines and devices (telerobotics).
- Haptic information is a combination of :
  - **Tactile Information** : It refers to the information acquired by the sensors connected to the body.
  - **Kinesthetic Information** : It refers to the information acquired by the sensors in the joints.
- Tactile sensors used to measure the forces exerted by the user on the interface can incorporate with haptic devices.
- Haptic technology has enabled human being to understand the working of human sense of touch as it allows creation of carefully controlled haptic virtual objects.
- Devices that use haptic feedback are : Cyber force, omega, delta, spider and haptic master.
- Example of haptic technology is, when user touches the touchscreen a vibration is observed which means that a touchscreen button is pressed.

Q.26 List the types of haptic feedback.

Ans. :

1. **Tactile feedback :** It is nothing but the touch feedback. This feedback allows user to feel things like texture of surfaces, temperature and vibration. Thus, this feedback deals with how a virtual object feels.
2. **Force feedback :** This feedback reproduces directional forces due to solid boundaries, the weight of grasped virtual objects, mechanical compliance of an object and inertia. Thus, this feedback deals with the effect of virtual environment on the user.

END...

Unit - VI

15 RENDERING PIPELINE

15.1 : Graphics Rendering Pipeline

Q.1 Explain graphical rendering pipeline. **[SPPU : June-22, Marks 6]**

Ans. : • The graphical rendering pipeline consist of three conceptual stages :

- Application stage,
- Geometry stage and
- Rasterizer stage.

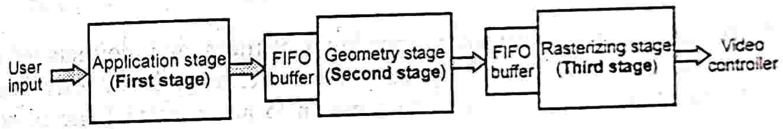


Fig. Q.1.1 Three stages of graphics rendering pipeline

• Each stage can be pipelined or parallelized one. The performance of this pipeline is judged by the slowest stage in it.

Application stage

• It is the first stage in the pipeline. It is executed entirely on the general purpose CPU and the programmer has full control over what happens, i.e. it is implemented by software only.

• The role of the application stage in the graphics rendering pipeline is to feed geometries (i.e. points, lines, triangles) to be rendered to the geometry stage. This stage receives input from various input devices like datagloves, mouse, trackballs, etc.

• This stage may utilize several CPU processing cores to execute in parallel to increase performance.

Geometry stage

- The geometry stage is the second stage in the pipeline. It is responsible for the majority of the per-polygon and per-vertex operations. It can be implemented either in hardware or software.

Q.2 Explain substages of the geometry stage of rendering pipeline.

Ans. :

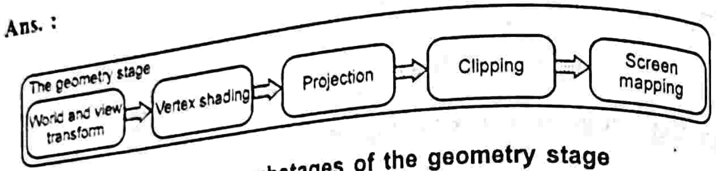


Fig. Q.2.1 The substages of the geometry stage

- The geometry stage is divided into the following substages :
  - Model and view transform,
  - Vertex shading,
  - Projection, clipping and
  - Screen mapping.
- **World and view transform** : Transforms vertices and normals of a model. A world transform is applied to the model to position, orient and scale the model in the world space common to all models of the scene. Once all models reside in the common world space they are transformed by the view transform into camera space (or eye space) to facilitate projection and clipping.
- **Vertex shading** : Models the appearance of each model according to their material and possible light sources shining on them.
- **Projection and clipping** : In projection stage, models are projected from the three dimensional camera space to the two dimensional normalized device space. Clipping stage filters out primitives and parts of primitives that lie outside the view volume.
- **Screen mapping** : Finally, in this stage the clipped primitive's x and y coordinates are transformed into screen coordinates.

Rasterizing stage

- It is the third and the last stage in the pipeline. It converts the inputted transformed and projected vertices with their associated shading data from the geometry stage into pixels on the screen.

Q.3 Explain the substages of the rasterizer stage of rendering pipeline.

Ans. :

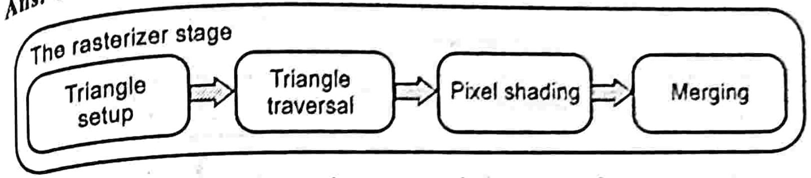


Fig. Q.3.1 The substages of the rasterizer stage

- The rasterizer stage is divided into substages, these are :
  - Triangle setup,
  - Triangle traversal,
  - Pixel shading and
  - Merging.
- **Triangle setup** : In this stage, data needed for interpolating the various shading data produced in the geometry stage is computed.
- **Triangle Traversal** : In this stage, a fragment is generated for each pixel that is at least partially covered by the triangle. The properties of the fragment which is to be sent to the pixel shader are calculated by interpolating among the three triangle vertices.
- **Pixel Shading** : In this stage, the interpolated data from the triangle traversal stage is used to perform perpixel shading calculations.
- **Merging** : The data from the pixel shading stage is stored in rectangular buffers, these are typically color in the color buffer, depth in the depth buffer (or Z-buffer) and alpha in the alpha buffer. These buffers are used in combination in the merging stage to determine the final colors of the pixels.

Q.4 Write a note on OpenGL rendering pipeline.

Ans. : Fig. Q.4.1 shows OpenGL rendering pipeline.

- OpenGL Pipeline has a series of processing stages in order. Two graphical information, vertex-based data and pixel-based data, are processed through the pipeline, combined together then written into the frame buffer.
- From Fig. 15.1.4 we can say that geometric data (for example vertices, lines, polygons) follow **geometry path** whereas pixel data (for example

pixel, images and bitmaps) follow **image path**. But before writing the final pixel data into the frame buffer both the data viz geometric data and pixel data undergo same steps that are rasterization and per fragment operations.

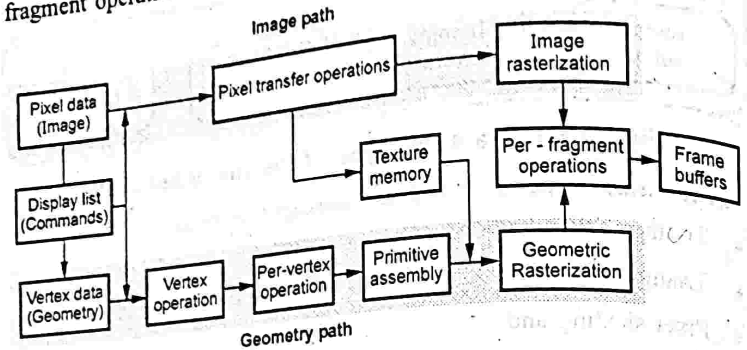


Fig. Q.4.1 OpenGL rendering pipeline

- **Display list** : Display list is a group of OpenGL commands that have been stored (compiled) for later execution. All data, geometry (vertex) and pixel data, can be stored in a display list. It may improve performance since commands and data are cached in a display list. When OpenGL program runs on the network, you can reduce data transmission over the network by using display list. Since display lists are part of server state and reside on the server machine, the client machine needs to send commands and data only once to server's display list.
- **Vertex operation** : Each vertex and normal coordinates are transformed by GL_MODELVIEW matrix (from object coordinates to eye coordinates). Also, if lighting is enabled, the lighting calculation per vertex is performed using the transformed vertex and normal data. This lighting calculation updates new color of the vertex.
- **Primitive assembly** : After vertex operation, the primitives (point, line, and polygon) are transformed once again by projection matrix then clipped by viewing volume clipping planes; from eye coordinates to clip coordinates. After that, perspective division by w occurs and viewport transform is applied in order to map 3D scene to window space coordinates. Last thing to do in Primitive Assembly is culling test if culling is enabled.

- **Pixel transfer operations** : After the pixels from client's memory are read, the data operations are performed such as scaling, bias, mapping and clamping. These operations are called Pixel Transfer Operations. The transferred data are either stored in texture memory or rasterized directly to fragments.
- **Texture memory** : Texture images are loaded into texture memory to be applied onto geometric objects.
- **Rasterization** : It is the conversion of both geometric and pixel data into fragment. Fragments are a rectangular array containing color, depth, line width, point size and antialiasing calculations (GL_POINT_SMOOTH,GL_LINE_SMOOTH,GL_POLYGON_SMOOTH). If shading mode is GL_FILL, then the interior pixels (area) of polygon will be filled at this stage. Each fragment corresponds to a pixel in the frame buffer.
- **Per-fragment operations** : It is the last process to convert fragments to pixels onto frame buffer. The first process in this stage is texel generation. Here, a texture element is generated from texture memory and it is applied to the each fragment. Then fog calculations are applied. After that, there are several fragment tests follow in order : Scissor Test, Alpha Test, Stencil Test and Depth Test. Finally, blending, dithering, logical operation and masking by bitmask are performed and actual pixel data are stored in frame buffer.

15.2 : Haptics Rendering Pipeline

Q.5 What is haptics rendering pipeline.

☞[SPPU : June-22, Marks 6]

Ans. : • Haptic rendering provides a unique, two-way communication between humans and interactive systems, enabling bi-directional interaction via tactile sensory cues. By harnessing the sense of touch, haptic display can further enhance a user's experience in a multi-modal synthetic environment, providing a more natural and intuitive interface with the virtual world.

- The haptic rendering pipeline involves successive software steps such as collision detection, physical simulation and force modelling. Haptic rendering, besides providing contact forces, can convey relevant information about the contact such as the material of the interacting objects.

• The main aim of haptic rendering is to generate tactual displays of the shapes, hardness, surface textures and frictional properties of 3D objects in real time.

Q.6 State the advantages of haptics rendering pipeline.

Ans. : Advantages are :

• Enables unique and bidirectional communication of human-computer interface.

• Improves human-machine interaction due to sense of touch.

• Enhances visual computation.

• Enhances exploration.

• Improves understanding level of geometric and mechanical properties of synthetic models.

• Provides force and tactile presence in tele-operated environments.

• Natural manipulation of simulated objects becomes possible.

Q.7 Explain the block diagram of haptic rendering pipeline.

Ans. : • Fig. Q.7.1 shows the general block diagram of haptic rendering pipeline. The haptic rendering pipeline consist of three stages :

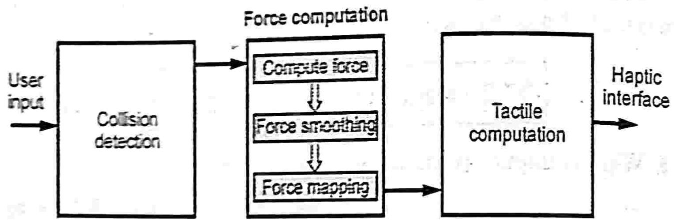


Fig. Q.7.1 General block diagram of haptic rendering pipeline

- Collision detection
- Force computation
- Tactile computation.

• **Collision detection :**

- Loads physical definition of 3D objects from database.
- Performs collision detection to determine which virtual objects collide.

• **Force computation :**

- Compute the collision forces
- Force smoothing
- Force mapping.

• **Tactile computation :**

- Render the touch feedback component of the simulation.
- The computed effects are added to the force vector send to the haptics output display.

Q.8 Explain the working of force-feedback device.

OR Describe collision detection in VR.

Ans. : • Fig. Q.8.1 shows a force-feedback device (in this case, an actuated fingertip probe) commanded by a haptic-rendering algorithm. It generates a tactual representation of contact between a fingertip and a computationally simulated object.

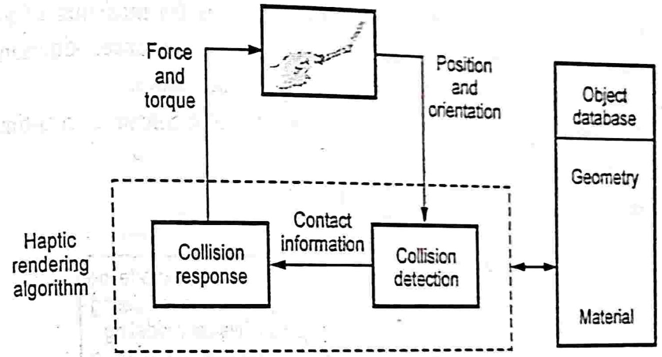


Fig. Q.8.1 Forced feedback device controlled by haptic rendering algorithm

• A haptic-rendering algorithm comprises a collision detection section and a collision-response section.

• When user manipulates (touches) the force feedback device with help of fingertip probe the local position and orientation of the probe are noted. Collision between the fingertip probe and the virtual object that is touched/manipulated is detected by the collision detection algorithm.

• After the collision detection, the **collision response algorithm** computes the force of interaction between the fingertip probe and the virtual objects and thus provides commands to the force-feedback device to generate the tactual representation of the objects.

- To sense the hardness of deformable objects, geometry and physics based mathematical models are used.

Q.9 State the applications of haptic interface.

Ans. : • Applications of haptic Interfaces :

- Medical training
- Molecular docking
- Digital sculpting
- Military applications
- Video games
- Virtual prototyping
- Manipulation of nanomaterial
- Entertainment
- Mechanical design
- Teleoperation or telerobotics

15.3 : Modeling in Virtual Reality

Q.10 What is modeling ? Draw VR modeling cycle.

Ans. : • The important aspect of VR simulations is the modeling of the virtual world. It includes modeling object shape, appearance, kinematic constraints, intelligent behaviour and physical characteristics.

- The model needs to be optimized in order to maintain a real-time interaction with the simulation.
- Fig. Q.10.1 shows VR modeling cycle.

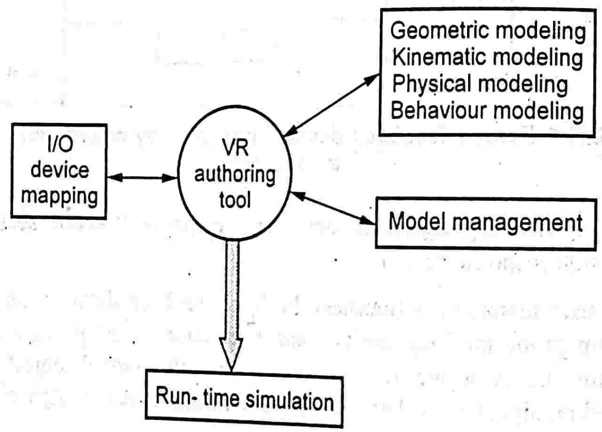


Fig. Q.10.1 VR modeling cycle

- As shown in Fig. Q.10.1, geometric modeling, kinematic modeling, physical modeling, behaviour modeling and model optimization during model management are the aspects of VR modeling.

Q.11 What do you mean by geometric modeling ?

Ans. : • **Geometric modeling** is the term used to describe shape and appearance of virtual objects. Shape of the object indicates rectangle, circle, triangle, polygon, spline, etc. whereas its appearance indicates color, surface pattern, surface texture, surface illumination, etc.

Q.12 With the help of diagram explain triangle meshes, state advantages and disadvantages of using triangle meshes to compose surface of virtual objects.

Ans. : • The 3D surface is used to determine the shape of virtual object. Mostly **triangle meshes** are used to compose surface of virtual objects.

Advantages :

1. Triangle meshes use shared vertices and faster to render.

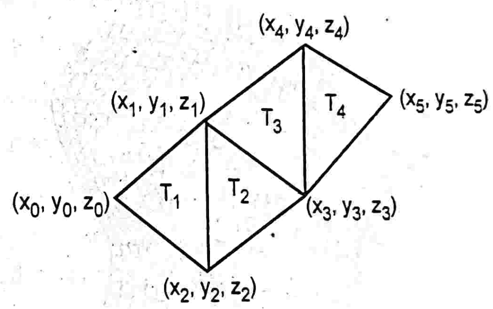


Fig. Q.12.1 Triangle mesh

As shown in Fig. Q.12.1, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the shared vertices for triangle T_1 and triangle T_2.

2. Less memory will be required to store the model using triangle meshes. It will be loaded faster by the rendering pipeline.
3. Some architectures are optimized to process triangle meshes.
4. Triangle meshes are convenient for geometric transformations and level of detail optimization.

Disadvantages :

- Triangle meshes are not suitable for composing objects which are highly curved or with bumpy surfaces as more triangles are needed to model such objects.

Q.13 With the help of diagram explain parametric surface, state advantages and disadvantages of using parametric surface to compose surface of virtual objects.

Ans. : • Another way for representing shape of objects is to use **parametric surfaces.**

- A parametric spline is described by the points x(t), y(t) and z(t) as given below :

$$\left.\begin{aligned} x(t) &= a_x \cdot t^3 + b_x \cdot t^2 + c_x \cdot t + dx \\ y(t) &= a_y \cdot t^3 + b_y \cdot t^2 + c_y \cdot t + dy \\ z(t) &= a_z \cdot t^3 + b_z \cdot t^2 + c_z \cdot t + dz \end{aligned}\right\} \quad \begin{aligned} &0 < t < 1 \\ &a, b, c: \text{Constant coefficients} \end{aligned}$$

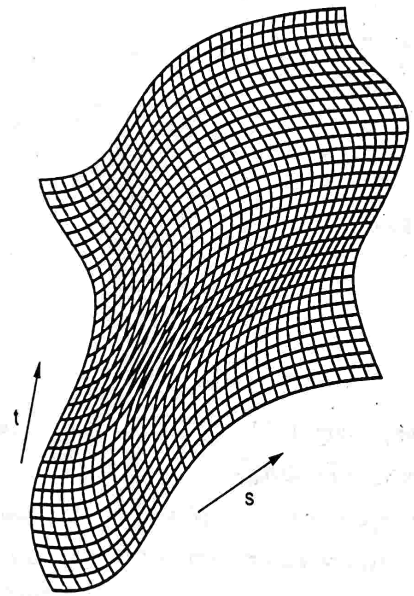


Fig. Q.13.1 Parametric surface

- Parametric surfaces are determined by introducing a second parameter s such that 0 < s < 1. Then the points on a parametric surface are given by coordinates x(s,t), y(s,t) and z(s,t).

Advantages :

- The linear functions are used for describing a polygon whereas high order functions are used for describing parametric surfaces. So parametric surfaces need comparatively less storage with improved surface smoothness.

Disadvantages :

- The shape of a polygon surface can easily be changed by direct vertex manipulation when surface is to be deformed. However, parametric surfaces do not allow the same. It is necessary to change the position of control points for changing their shape which are located outside the surface.

Q.14 State different methods to construct object surfaces.

Ans. : • The object surfaces can be constructed using different methods such as :

- Using a Toolkit Editor
- Improving CAD files
- Creating surfaces with a 3D Digitizer
- Creating surfaces with a 3D Scanner
- Using online 3D object Databases

Q.15 State different methods for modeling 3D object geometry. Give features and sources for each of them.

Ans. : • Table Q.15.1 summarizes the different methods for modeling 3D object geometry.

Method	Feature	Source
Toolkit editors	Tedious, requires skill	OpenGL, Starbase, PHIGS
CAD programs	Interactive, existing technology	AutoCAD (Autodesk), 3-D Studio, etc.

3D digitizers	Interactive, allows custom models	Autodesys Inc., Mira Imaging, Polhemus Inc., etc.
3D scanners	Fast multipoint acquisition, large objects	Polhemus Inc., Cyra Technologies, etc.
Commercial 3D databases	Vertex list, connectivity, static model, level of detail	Viewpoint Inc., etc.

Table Q.15.1 Methods for modeling 3D object geometry

Q.16 Which factors affect the appearance of an object ?

Ans. : The appearance of an object depends on

1. Type and placement of virtual light sources and object's surface reflectivity coefficient.
2. Surface texture.

Q.17 What is scene illumination ?

Ans. : Scene Illumination determines the light intensities on the surface of the object.

- It is classified as,

1. Local illumination 2. Global illumination.

Q.18 What do you mean by local and global illumination ?

Ans. : • **Local scene illumination** treats the interactions between objects and light sources in isolation irrespective of the interdependencies between objects.

- **Global scene illumination** models the interreflections between objects and shadows which results in a more realistic looking scene.

Q.19 What is texture mapping ?

Ans. : Texture Mapping improves the image realism without using additional surface polygons. It is a technique used during rasterization stage of the graphics pipeline to modify the surface properties (color, specular reflection, pixel normals etc.) of the object model.

Q.20 What is kinematic modeling ? Discuss different aspects of kinematic modeling. **[SPPU : Dec.-22, Marks 5]**

Ans. : • The kinematic modeling determines the location of 3D objects with respect to a world coordinate system and their motion in the virtual world.

- The aspects of kinematic modeling :

1. Object kinematics is governed by parent-child hierarchical relations. The motion of a parent object affects that of its child.
2. The way with which the world is viewed. i.e. the motion of a virtual camera.
3. Transformation and projection of a camera image on 2D display window for providing visual feedback to the user.

Q.21 Explain the following terms in kinematic modeling :
a. Transformation matrix b. Object position
c. Transformation invariants.

Ans. : **a. Homogeneous transformation matrices :**

- The 4×4 homogeneous transformation matrices are used to express transformations on object such as translations, rotations, scaling, etc.
- A homogeneous transformation matrix is given by,

$$T_{S \leftarrow Q} = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

where $R_{3\times3}$ is the rotation submatrix expressing the orientation of system of coordinates Q with respect to system of coordinates S and $P_{3\times1}$ is the vector expressing the position of the origin of system Q with respect to the origin of system of coordinates S.

b. Object position :

- The object surface modeling uses (x, y, z) vertex coordinates expressed in an object system of coordinates. The object remains connected to the system of coordinates, usually at its center of gravity and oriented along the object axes of symmetry.
- When the object movement takes place in the virtual world, the position and orientation of object vertices in the object system of coordinates remain unchanged regardless of object position within the scene.

c. Transformation Invariants :

- Many times one of the objects seen in the virtual scene is a 3D hand where its position and orientation are mapped to the position and orientation of the 3D tracker attached to the sensing glove worn by the user. The 3D position of tracker receiver relative to the source is given by the time dependent homogeneous transformation matrix $T_{source \leftarrow reciever}(t)$.
- Let's assume that the source is fixed, then its position in the world system of coordinates is given by the matrix $T_{w \leftarrow source}$.
- A virtual hand can be moved in the virtual world by multiplying all its vertices with a overall transformation matrix.

$$V_i^w(t) = T_{w \leftarrow source} \, T_{source \leftarrow receiver}(t) \, V_i^{hand}$$

where V_i^{hand} are the vertex co-ordinates in the hand system of co-ordinates, and i = 1, ..., n.

Q.22 What do you mean by object hierarchy ?

Ans. : • Object hierarchies define groups of objects which move together as a whole and whose parts can also move independently.

- In VR system, a hierachy implies at least two levels of virtual objects. The higher level objects are also called the **parent objects** and the lower level objects are referred to as **child objects.**
- The motion of a parent object causes all its children to move. However, a child object can move independently without affecting the position of the parent object.
- A tree graph is used to represent the segmented model hierarchy graphically. The nodes are object segments and its branches represent relationships. A node in a free graph is parent to all nodes in the lower branches of it.

Q.23 What is physical modeling ? ☞[SPPU : Dec.-22, Marks 2]

Ans. : • Physical modeling is the integration of the object's physical characteristics such as weight, inertia, surface roughness, compliance (hard or soft), deformation mode (elastic or plastic) etc.

Q.24 What is collision detection ? ☞[SPPU : Dec.-22, Marks 2]

OR Write a note on collision detection.

Ans. : • The first stage physical modeling (haptic rendering) is collision detection. It determines whether two (or more) objects are in contact with each other. The only objects that collide are processed by the haptic rendering piepline, hence this can be considered a form of haptic clipping.

- Collision detection can be classified as
  1. Approximate 2. Exact
- Approximate collision detection is also called **bouding-box collision.** It uses 3D extents (bounding boxes).
- A bounding box is a prism which encloses all the vertices of a given 3D object.
- Bounding boxes are classified as :
  1. Oriented 2. Axis-aligned
- An Oriented Bounding Box (OBB) is a prism aligned with the object's major axes and which changes orientation dynamically as the object rotates.
- An Axis-Aligned Bounding Box (AABB) is a prism aligned with the world system of coordinates. A fixed-size cube is a special case of an AABB. It can accomodate any orientation of the 3D object it encloses. Fixed size bounding boxes are computationally faster than variable - size bounding boxes, but larger than these.
- Bounding box collision detection tests whether two (or more) bounding boxes overlap in 3D space.
- Approximate collision detection is a necessary step to weed out objects that are far apart, such that their bounding boxes do not overlap. However, if the VR application needs to know exactly where the two objects are in contact, then an exact collision detection needs to be performed.
- The two types of collision detection are performed at each simulation cycle. The exact collision detection is performed only on the object pairs that are potentially in contact. AABB collision detection test is used for the same.
- Exact collision detection method uses a polygon representation for an object shape or implicit functions that algebraically define the object surface.

Q.25 What is surface deformation ? ☞[SPPU : Dec.-22, Marks 2]

Ans. : • Collision detection is followed by collision response. If the objects in contact are nonrigid, then one form of collision response is **surface deformation.** Surface deformation changes the 3D object geometry interactively and needs to be coordinated with the graphics pipeline.

- If the object is modeled using polygons, then the surface deformation is done directly, through vertex manipulation.
- If the object is modeled by parametric surfaces, then the deformation is done indirectly by modifying the position of control points surrounding the surface.
- An extreme case of surface deformation is **surface cutting.**

Q.26 Describe the following :
a) Force computation b) Force smoothing and mapping
c) Haptic texturing. ☞[SPPU : Dec.-22, Marks 2]

Ans. :

a) Force computation

- When users interact with 3D object surfaces, they should feel reaction forces. The haptics rendering pipeline is used for computing these forces. Then they are sent to the haptic display through force feedback to the user.
- Force computation takes into account :
  1. The type of surface contact
  2. The kind of surface deformation
  3. The object's physical and kinematic characteristics
- The force computation is performed according to the type of object being haptically simulated. It varies for different types of objects : Elastic virtual objects, plastic virtual objects and virtual walls (virtual objects that are neither elastic nor plastic).

b) Force smoothing and mapping

- Force shading changes the direction of the feedback force produced during interactions with polygonal surface to simulate contact with smooth curved surfaces.

- The weighted normals of the vertices of the polygon being contacted are used to compute the direction of the contact force.
- This approach affects only its direction, not the magnitude.

c) Haptic texturing :

- Haptic texturing is the last stage of the haptic rendering pipeline. Graphics textures add realism to the object appearance whereas haptic textures enhance the physical model realism of the object surface.
- Haptic textures can add new information such as slippery, cold, smooth etc. to characterize on object.
- Haptic textures can be cascaded creating new surface effects.

Q.27 Give short note on 'Behavior Modeling'. ☞[SPPU : Dec.-22, Marks 6]

Ans. : • Whenever objects interacted, it was assumed that one was controlled by the user. It is also possible to model object behavior that is independent of user's actions.

- The virtual human, also called as an agent is a 3D character that has a human behavior. Groups of such agents are called **crowds** and have crowd behaviour.
- For example : The model of a virtual cabin. The cabin has an atomatic sliding door, a wall clock, a calender, a thermometer.
- The VR engine system will update clock and calender details. The external temperature sensor interfaced to the VR engine will update the temperature display of the thermometer. When the user enters the virtual cabin; the sliding door opens automatically and the information displayed by clock, calender and thermometer will be updated. This is an example of modeling object behavior by accessing external sensors. The virtual human (agent) can also be modeled using simulation.

END...

DECEMBER - 2017 [5252]-576 Solved Paper

Course 2015

Time : 2 Hours] [Maximum Marks : 50

Instructions to the candidates :

1) Solve Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.

2) Neat diagrams must be drawn wherever necessary.

3) Figures to the right side indicate full marks.

4) Use of calculator is allowed.

5) Assume suitable data if necessary.

Q.1 a) *Derive equation for decision parameter of Bresenham's circle algorithm.* (Refer Q.24 of Chapter - 3) [6]

b) *What are the different steps for rotation about an arbitrary point in 2D ?* (Refer Q.5 of Chapter - 5) [6]

OR

Q.2 a) *Interpret Bresenham's algorithm to find which pixels are turned on for the line segment between (1, 2) and (7, 6).*
(Refer Q.12 of Chapter - 3) [6]

b) *Write pseudo code for boundary fill algorithm. Compare boundary fill algorithm with scan line algorithm.*
(Refer Q.7 of Chapter - 4) [6]

Q.3 a) *Explain with the help of suitable diagram parallel and perspective projection.* (Refer Q.8 of Chapter - 7) [6]

b) *Explain midpoint subdivision line clipping method with suitable example.* (Not in New Syllabus) [6]

OR

Q.4 a) *Explain basic transformations on 3D.*
(Refer Q.1 of Chapter - 7) [6]

b) *What is segment ? Explain the concept of segment table and display file.* (Refer Q.1 and 2 of Chapter - 8) [6]

Q.5 a) *Explain in detail Graphics Memory Pipeline.*
(Not in New Syllabus) [7]

b) *Explain pseudo C algorithm for Gouraud shading.*
(Refer Q.15 of Chapter - 9) [6]

OR

Q.6 a) *Draw and explain block diagram of i860 microprocessor.*
(Not in New Syllabus) [7]

b) *What is animation ? Explain the basic rules required for animation.* (Refer Q.1 and Q.3 of Chapter - 12) [6]

Q.7 a) *Write the properties of Bezier and B-Spline curves.*
(Refer Q.10 of Chapter - 11) [7]

b) *Why cubic Bezier curves are chosen ? Explain any Bezier curve generation method.* (Refer Q.6 and Q.9 of Chapter - 11) [6]

OR

Q.8 a) *Explain how Koch curves are generated. Also calculate the fractal dimension of Koch curve.* (Refer Q.20 of Chapter - 11) [7]

b) *Define fractals with examples. Given various categories in which fractals are classified.* (Refer Q.17 of Chapter - 11) [6]

MAY - 2018 [5352]-576 Solved Paper

Course 2015

Time : 2 Hours] [Maximum Marks : 50

Q.1 a) *What are the steps of Bresenham's circle drawing algorithm ? Explain with example.* (Refer Q.22 of Chapter - 3) [6]

b) *What is the concept of display file and display file interpreter.*
(Not in New Syllabus) [6]

OR

Q.2 a) *Discuss any two polygon filling methods.*
(Refer Q.8 and Q.9 of Chapter - 4) [6]

b) *Interpret Digital Differential Analyzer (DDA) algorithm to find which pixels are turned on for the line segment between (2, 3) and (8, 6).*
(Refer Q.8 of Chapter - 3) [6]

Q.3 a) *Explain Cohen Sutherland line clipping method with diagram.*
(Refer Q.4 of Chapter - 6) [6]

b) *Explain reflection about XY, YZ and XZ plane.*
(Refer Q.5 of Chapter - 7) [6]

OR

Q.4 a) *Explain different types of parallel and perspective projections.*
(Refer Q.9 and Q.10 of Chapter - 7) [6]

b) *Explain window and viewport in detail with example.*
(Refer Q.2 of Chapter - 6) [6]

Q.5 a) *Write a short note on animation sequence.*
(Refer Q.4 of Chapter - 12) [7]

b) *Write a short note on openGL.* (Refer Q.1 of Chapter - 2) [6]

OR

Q.6 a) *Explain block diagram of i860.* (Not in New Syllabus) [7]

b) *What is shading ? Explain different types.*
(Refer Q.11 and Q.12 of Chapter - 9) [6]

Q.7 a) *Write a note on fractal lines and surfaces.*
(Not in New Syllabus) [7]

b) *What is interpolation ? Explain any interpolation algorithm.*
(Refer Q.3 of Chapter - 11) [6]

OR

Q.8 a) *Explain Bezier curve generation using midpoint subdivision.*
(Refer Q.11 of Chapter - 11) [7]

b) *Explain techniques of smoothing curves using B Splines.*
(Refer Q.12 of Chapter - 11) [6]

DECEMBER - 2018 [5459]-206 | Solved Paper

Course 2015

Time : Two Hours] [Maximum Marks : 50

Q.1 a) *Differentiate between Raster scan and Random scan.*
(Refer Q.8 of Chapter - 1) [6]

b) *Interpret Bresemham's algorithm to find which are pixels are turned on for the line segment (1, 2) to (7, 6).*
(Refer Q.12 of Chapter - 3) [6]

OR

Q.2 a) *What are different types of Polygon ? How can we test whether the given point is inside the Polygon ?*
(Refer Q.3 and Q.4 of Chapter - 4) [6]

b) *Find the transformation of a square ABCD whose center is at (2, 2) is reduced to half of its size with center still remaining at (2, 2). The square ABCD's co-ordinates are (0, 0), (4, 0) (4, 4), (0, 4). Find the new co-ordinates.* (Refer Q.14 of Chapter - 5) [6]

Q.3 a) *Explain 3D transformation rotation about arbitrary axis.*
(Refer Q.4 of Chapter - 7) [6]

b) *In 2D clipping how are line grouped into visible, invisible and partially visible categories ?* (Refer Q.4 of Chapter - 6) [6]

OR

Q.4 a) *Explain the ways of projecting 3D object onto 2D screen in detail.* (Refer Q.8 of Chapter - 7) [6]

b) *Let ABCD be the rectangle window with A(10, 20), B(100, 20), C(100, 90), D(10, 90). Find the region codes for endpoints and use Cohen Sutherland algorithm to clip the lines P1–P2 with P1 (5, 30) and P2 (70, 110) and Q1–Q2 with Q1 (50, 70) and Q2(80, 30).*
(Refer Q.7 of Chapter - 6) [6]

Q.5 a) *Explain block diagram of i860* (Not in New Syllabus) [6]

b) *Write a short note on openGL.* (Refer Q.1 of Chapter - 2) [7]

OR

Q.6 a) *Explain pseudo C algorithm for Gourand shading.*
(Refer Q.15 of Chapter - 9) [6]

b) *Explain in detail Graphics Memory Piepline.*
(Not in New Syllabus) [7]

Q.7 a) *What is interpolation ? Explain the process of curves approximation.* (Refer Q.3 of Chapter - 11) [6]

b) *Explain features of any graphics tool you have used.*
(Not in New Syllabus) [7]

OR

Q.8 a) *Explain algorithm for fractal lines the example of generation of coastlines.* **(Not in New Syllabus)** **[7]**

b) *Write short notes on :*

i) Fractals and topological dimensions **(Refer Q.17 of Chapter - 11)**

ii) Koch curve **(Refer Q.20 of Chapter - 11)** **[6]**

MAY - 2019 [5559]-206 Solved Paper

Course 2015

Time : 2 Hours] **[Maximum Marks : 50**

Q.1 a) *Rasterize the line from (– 6, – 5) to (1, 0) using Bresenham's line drawing algorithm.* **(Refer Q.14 of Chapter - 3)** **[6]**

b) *Distinguish between Random scan and Raster scan method.* **(Refer Q.8 of Chapter - 1)** **[6]**

OR

Q.2 a) *Show that transformation matrix of reflection about a $y = x$ is equivalent to reflection relative to x-axis followed by anticlockwise rotation of 90 degree.* **(Refer Q.16 of Chapter - 5)** **[6]**

b) *Write pseudo code for boundary fill algorithm. Compare boundary fill and flood fill and fill algorithm.* **(Refer Q.6, Q.7 and Q.8 of Chapter - 4)** **[6]**

Q.3 a) *Explain parallel and perspective projection.* **(Refer Q.8 of Chapter - 7)** **[6]**

b) *Explain window to viewport transformation.* **(Refer Q.2 of Chapter - 6)** **[6]**

OR

Q.4 a) *Explain basic transformations on 3D.* **(Refer Q.1 of Chapter - 7)** **[6]**

b) *What is segment ? Explain different operations on segment with example.* **(Refer Q.1 and Q.5 of Chapter - 8)** **[6]**

Q.5 a) *Describe the steps to design animation Sequence along with concepts.* **(Refer Q.4 of Chapter - 12)** **[7]**

b) *Write down the steps for constant and Gouraud shading.* **(Refer Q.13 ns Q.14 of Chapter - 9)** **[6]**

OR

Q.6 a) *Describe in detail Graphics Memory Pipeline with block diagram.* **(Not in New Syllabus)** **[7]**

b) *Explain openGL with respect to openGL operations.* **(Refer Q.8 of Chapter - 2)** **[6]**

Q.7 a) *How midpoint subdivision can be used for Bezier curve generation.* **(Refer Q.11 of Chapter - 11)** **[6]**

b) *Write down the algorithm to draw fractal lines.* **(Not in New Syllabus)** **[7]**

OR

Q.8 a) *Write short notes on Hilbert's and Koch curve along with its topological and fractal Dimensions.* **(Refer Q.11 of Chapter - 11)** **[6]**

b) *Explain interpolation method of curve generation.* **(Refer Q.3 of Chapter - 11)** **[7]**

DECEMBER - 2019 [5668]-206 Solved Paper

Course 2015

Time : 2 Hours] **[Maximum Marks : 50**

Q.1 a) *Explain Display file and its structure with an example.* **(Not in new syllabus)** **[6]**

b) *Explain Character Generation Methods with its types.* **(Refer Q.25 of Chapter - 3)** **[6]**

OR

Q.2 a) *Explain Shearing and Scaling Transformation in detail.* **(Refer Q.17 and Q.2 of Chapter - 5)** **[6]**

b) *What is Seed point ? Explain Flood fill algorithm.* **(Refer Q.5 and Q.8 of Chapter - 4)** **[6]**

Q.3 a) *Explain different types of parallel projections.*
(Refer Q.9 of Chapter - 7) **[6]**

b) *Explain Cohen Sutherland line clipping method with suitable example.* **(Refer Q.4 of Chapter - 6)** **[6]**

OR

Q.4 a) *Explain 3D reflection about xy, yz and xz plane.*
(Refer Q.5 of Chapter - 7) **[6]**

b) *Explain the following terms :*
i) Screen co-ordinates ii) World co-ordinates
iii) Window iv) Viewport **(Refer Q.2 of Chapter - 6)** **[6]**

Q.5 a) *Draw and explain diagram of i860 along with applications.*
(Not in New Syllabus) **[7]**

b) *Enumerate and explain different shading methods in detail.*
(Refer section 9.7) **[6]**

OR

Q.6 a) *Explain in detail Graphics Memory Pipeline.*
(Not in New Syllabus) **[7]**

b) *What are the steps in design in Animation ? Describe each step briefly.* **(Refer Q.4 of Chapter - 12)** **[6]**

Q.7 a) *Explain Bezier curve generation using midpoint subdivision.*
(Refer Q.11 of Chapter - 11) **[7]**

b) *How fractals are used to generate fractal surfaces ? Give two examples of fractal surfaces.* **(Not in New Syllabus)** **[6]**

OR

Q.8 a) *Explain how Koch curves are generated. Also calculate the fractal dimesion and topological dimension of the Koch curve.*
(Refer Q.20 of Chapter - 11) **[7]**

b) *Define fractals with examples. Give various categories in which fractals are classified.* **(Refer Q.22 of Chapter - 11)** **[6]**

JUNE - 2022 [5869]-289 Solved Paper

Course 2019

Time : $2\frac{1}{2}$ Hours] **[Maximum Marks : 70**

Q.1 a) *Explain the basic transformation techniques in 3D graphics.*
i) Scaling **(Refer Q.2 of Chapter - 7)**
ii) Rotation **(Refer Q.3 of Chapter - 7)**
iii) Translation **(Refer Q.1 of Chapter - 7)** **[6]**

b) *Use the Cohen-Sutherland algorithm for clipping window having clipping window whose lower left point at (2, 1), upper right point at (7, 5) and line points are (1, 3) and (5, 6). Find the intersection points.* **(Refer Q.6 of Chapter - 6)** **[6]**

c) *Explain the following term with example :*
i) Windowing i) Clipping iii) Viewport
(Refer Q.1 and Q.2 of Chapter - 6) **[6]**

OR

Q.2 a) *Explain with diagram parallel and perspective projection.*
(Refer Q.8 of Chapter - 7)) **[6]**

b) *Explain 3D transformation rotation about arbitrary axis.*
((Refer Q.4 of Chapter - 7) **[6]**

c) *Using Sutherland-Hodgeman method, clip polygon ABCDE against window PQRS. The coordinators of polygon are A(80, 200), B(220, 120), C(150, 100), D(100, 30), E(10, 120). Coordnates of the window are P(200, 50), Q(50, 150), R(200, 150), S(50, 50).*
(Refer Q.11 of Chapter - 6) **[6]**

Q.3 a) *What is segment ? Explain different operations on segment with example.* **(Refer Q.1 of Chapter - 8 and section 8.3)** **[6]**

b) *Explain RGB, HSV and HLS color models.*
(Refer Q.3 and Q.6 of Chapter - 10) **[6]**

c) *Explain with diagram Gouraud shading algorithm in detail.* (Refer Q.14 of Chapter - 9) [5]

OR

Q.4 a) *Explain the concept of segment table and display file.* (Refer Q.2 of Chapter - 8) [6]

b) *Explain with diagram Phong shading algorithm in detail.* (Refer Q.16 of Chapter - 9) [6]

c) *Define color gamut. Explain with diagram CIE chromaticity diagram.* (Refer Q.1 and Q.2 of Chapter - 10) [5]

Q.5 a) *Differentiate between Bezier curve and B-spline curve.* (Refer Q.16 of Chapter - 11) [6]

b) *Write a short note on interpolation and approximation.* (Refer Q.4 and Q.5 of Chapter - 11) [6]

c) *Explain various types of animation languages.* (Refer Q.5 of Chapter - 12) [6]

OR

Q.6 a) *Explain Bezier curve. List its properties.* (Refer Q.9 and Q.10 of Chapter - 11) [6]

b) *Write short notes on :*

i) Koch curve (Refer Q.20 of Chapter - 11)

ii) Frame-by-frame animation techniques

(Refer Q.14 of Chapter - 12) [6]

c) *What is fractal ? Explain Hilbert curve in detail.* (Refer Q.17 and Q.19 of Chapter - 11) [6]

Q.7 a) *What is the different usage of virtual reality ? Explain in detail.* (Refer Q.4 of Chapter - 13) [6]

b) *What is haptics rendering pipeline modeling in virtual reality ?* (Refer Q.5 of Chapter - 15) [6]

c) *Differentiate HMD and CAVE in virtual reality.* (Refer Q.21 and Q.23 of Chapter - 14) [5]

OR

Q.8 a) *Explain the graphics rendering pipeline.* (Refer Q.1 of Chapter - 15) [6]

b) *Explain the applications of virtual reality systems.* (Refer Q.4 of Chapter - 13) [6]

c) *Explain 3D position trackers.* (Refer Q.1 of Chapter - 14) [5]

DECEMBER - 2022 [5925]-272 Solved Paper

Course 2019

Time : $2\frac{1}{2}$ Hours] [Maximum Marks : 70

Q.1 a) *Explain with diagram Cohen Sutherland line clipping algorithm.* (Refer Q.4 of Chapter - 6) [6]

b) *Compare homogeneous co-ordinate system and normalized co-ordinate system.* (Refer Q.4 of Chapter - 5) [6]

c) *Show that the transformation matrix of reflection about line $y = x$ is equivalent to reflection relative to x-axis followed by anticlockwise rotation of 90 degree.* (Refer Q.16 of Chapter - 5) [6]

OR

Q.2 a) *What is the concept of vanishing point in perspective projection ? Explain with diagram.* (Refer Q.10 of Chapter - 7) [6]

b) *Let ABCD be a rectangle window with A(20,20), B(90,20), C(90,70), D(20,70). Find the region codes for the end points and use Cohen Sutherland line clipping algorithm to clip the following line Q_1Q_2 with $Q_1(10,10)$ and $Q_2(70,60)$.* (Refer similar Q.7 of Chapter - 6) [6]

c) *Explain 3D reflection about XY, YZ and XZ plane.* (Refer Q.5 of Chapter - 7) [6]

Q.3 a) *What is shading. Explain with diagram constant intensity shading method.* **(Refer Q.11 and Q.13 of Chapter - 9)** [6]

b) *Explain CMY and HSV color models.*
(Refer Q.4 and Q.6 of Chapter - 10) [6]

c) *What is a segment ? How do we create it ? Why do we need segments ?* **(Refer Q.1 and Q.5 of Chapter - 8)** [5]

OR

Q.4 a) *Compare Gourand and Phong method of shading.*
(Refer Q.17 of Chapter - 9) [6]

b) *What is segment ? Explain the concept of segment table and display file.* **(Refer Q.1 and Q.2 of Chapter - 8)** [6]

c) *Explain CIE chromaticity diagram; also explain how RGB to CMY conversion is done.* **(Refer Q.2 of Chapter - 10)** [5]

Q.5 a) *Explain Koch curve and its application in detail.*
(Refer Q.20 of Chapter - 11) [6]

b) *Write short notes on*
i) Morphing **(Refer Q.6 and Q.7 of Chapter - 12)**
ii) Design of animation sequence **(Refer Q.4 of Chapter - 12)** [6]

c) *What is fractal ? Explain Hilbert curve in detail.*
(Refer Q.17 and Q.19 of Chapter - 11) [6]

OR

Q.6 a) *Write short notes on*
i) B-spline curve **(Refer Q.12 of Chapter - 11)**
ii) Blending function of Bezier curve **(Refer Q.9 of Chapter - 11)** [6]

b) *What are the methods of controlling animation ?*
(Refer Q.10 of Chapter - 12) [6]

c) *Explain various types of animation languages.*
(Refer Q.5 of Chapter - 12) [6]

Q.7 a) *Explain the physical modeling in Virtual Reality.*
(Refer Q.23, Q.24, Q.25 and Q.26 of Chapter - 15) [6]

b) *Explain haptic feedback in Virtual Reality system.*
(Refer Q.25 of Chapter - 14) [6]

c) *What is navigation and manipulation interfaces in virtual reality system ?* **(Refer Q.12 of Chapter - 14)** [5]

OR

Q.8 a) *Explain the behavioral modeling in virtual reality.*
(Refer Q.27 of Chapter - 15) [6]

b) *What are sound displays in virtual reality ?*
(Refer Q.24 of Chapter - 14) [6]

c) *Explain kinematic modeling in virtual reality.*
(Refer Q.20 of Chapter - 15) [5]

END...