0021848

01

QP Code:___________

[Time: Three Hours] [Marks: 80]

N.B: (1) Question No.1 is compulsory
(2) Attempt any three of remaining five questions
(3) Assume any suitable data if necessary and justify the same

Note :Answer Key intends to cover the important points expected in solution and doesn't necessarily provide a full solution.

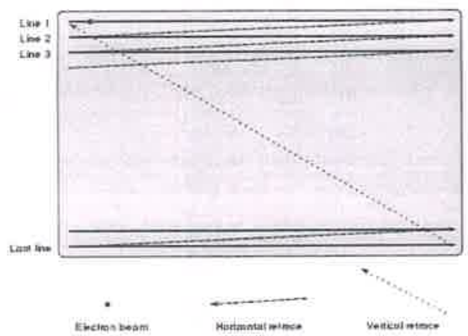
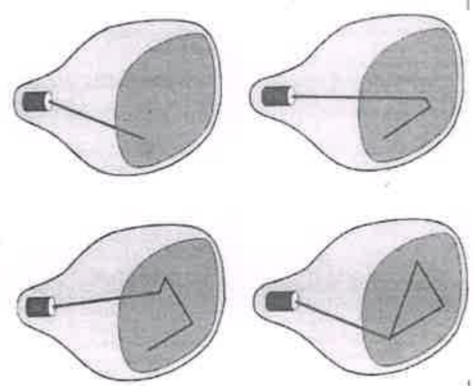
Q 1	**a)**	**Explain CSG method for solid modeling.** **Answer :** Constructive Solid geometry method attempts to represent a solid object by combining the volume occupied by overlapping 3D objects and by applying set operations like union, difference and intersection. Typical standard primitives are: cone, cylinder, sphere, torus, block, closed spline surface, right angular wedge. Operations are union, intersection and difference. 1.> Select the primitives 2.> Overlap them as per the objective 3.> Apply set operations as per the objective. eg: 	5
	b)	**What is aliasing and Explain any one antialiasing method.** **Answer :** Aliasing: Aliasing is a phenomenon that occurs while sampling a continuous function with insufficient resolution or when the sampling frequency doesn't satisfy the nyquist criterion. It typically creates a staircase like effect. Antialiasing: Antialiasing is the application of techniques that reduce or eliminate aliasing effect. Antialiasing methods were developed to combat the effects of aliasing. The two major categories of antialiasing techniques are prefiltering and postfiltering. Postfiltering, also known as supersampling, is the more popular approach to antialiasing. For each displayed pixel, a postfiltering method takes several samples from the scene and computes an average of the samples to determine the pixel's color.	5

02

The two steps in the postfiltering process are: 1. Sample the scene at n times the display resolution. For example, suppose the display resolution is 512x512. Sampling at three times the width and three times the height of the display resolution would yield 1536x1536 samples.
2. The color of each pixel in the rendered image will be an average of several samples. A filter provides the weights used to compute the average.

c) Compare Raster Scan and Random Scan displays. (5)

Answer :

Raster Scan	Random Scan
It is a refresh type of display	It is a Refresh type of display
It is of point Plotting Category and a picture is represented by a 2D array (Grid) of pixels	It is of line drawing (Calligraphic / Vector) category and a picture is represented by a collection of short straight line segments (strokes)
The electron beam is deflected in a predetermined manner , starting from top left , scanning one scan line at a time and after reaching to the end of current scan line the electron beam is deflected so as to position it to the beginning of next immediate scan line (non interlaced) or next to next scan line(interlaced) , the mechanism is called as horizontal retrace , and then the process is repeated till the end of last scan line . There after the electron beam is deflected back to the top left of the screen (vertical retrace) and the activity is repeated specified no. of times/ sec. depending on the refresh rate. 	The electron beam is deflected in a random manner so as to only scan those straight line paths which constitutes a picture and the activity is repeated specified no. of times/ sec. depending on the refresh rate. 
During the scanning the beam velocity / intensity is modulated so as to coincide with the picture definition.	During the scanning the beam velocity / intensity is modulated so as to coincide with the picture definition.
Picture definition is recorded in refresh buffer / Raster / Frame buffer in form of intensity values for every pixel.	Picture definition is recorded in refresh buffer / Raster / Frame buffer in form of display file , which contains command to draw the line , to move the electron beam and to decide the line attributes.
Suitable for applications which involves	Suitable for representing wire frame

		dynamic motions.	models.	

d) Prove that two successive rotations are additive i.e. $R1(\theta_1) * R2(\theta_2) = R(\theta_1 + \theta_2)$

Answer :

$R1(\theta_1) * R2(\theta_2) =$

$$\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1)\cos(\theta_2)-\sin(\theta_1)\sin(\theta_2) & -\cos(\theta_1)\sin(\theta_2)-\sin(\theta_1)\cos(\theta_2) & 0 \\ \sin(\theta_1)\cos(\theta_2)+\cos(\theta_1)\sin(\theta_2) & -\sin(\theta_1)\sin(\theta_2)+\cos(\theta_1)\cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$= R(\theta_1 + \theta_2)$

5

Q 2 a) Explain Bresenham line drawing algorithm with proper mathematical analysis and identify the pixel positions along a line between A(10,10) and B(18,16) using it.

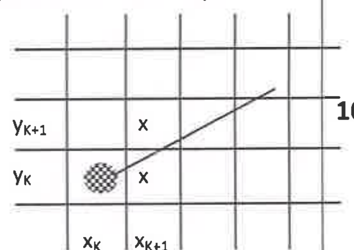
Answer :

Bresenham's line algorithm: (involves only integer calculations)

(i) In Bresenham's line algorithm, the pixel positions along a line path are obtained by determining the pixel that is nearer to the true line path at each step.

(ii) Consider a line segment AB with starting point at A (x_1, y_1) and endpoint B (x_2, y_2). Consider the case for lines with $m \leq 1$.

Let, (x_K, y_K) be the selected position to be closest to the line path in the K^{th} step of algorithm. Then the next two candidate pixels are:



(x_{K+1}, y_K) or $(x_K + 1, y_K + 1)$

i.e. $(x_{K+1}, y_{K+1}) = (x_K + 1, y_K)$ or $(x_K + 1, y_K + 1)$

here K+1 = next

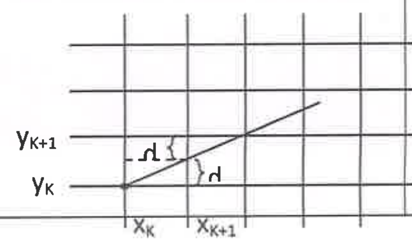
The equation of line is

$y = mx + h$

At, $x = x_{K+1}$, $y = m(x_{K+1}) + h$

10

(iii) To obtain the decision parameter to select one of the candidate pixel position, the algorithm compares the distance of true line value i.e. y from y_K and y_{K+1}.



From figure:

If $d_1 < d_2$, $y_K + 1$ is close to actual line path

$\therefore \quad y_{K+1} = y_K + 1$

If $d_2 < d_1$, y_K is close to actual line path

$\therefore \quad y_{K+1} = y_K$

The distance d_1 of the true line path from y_{K+1} can be determined as,

$d_1 = (y_K + 1) - m(x_K + 1) - h$

and d_2 as distance of y from y_K can be computed as,

$d_2 = y - y_K$

$\therefore \quad d_2 = m(x_K + 1) + h - y_K$

The decision parameter p_K is based on the distances d_1 and d_2 and can be obtained as

$p_K = d_2 - d_1$

If $p_K < 0$, then $d_2 < d_1$. Hence, the next pixel will be $(x_K + 1, y_K)$ else $(x_K + 1, y_K + 1)$

(iv) Since d_1 and d_2 involves 'm' (real); to remove the real computations, the decision parameter p_K is changed as,

$p_K = \Delta x\,(d_2 - d_1)$ which would not effect the sign of p_K as Δx is +ve (in this case)

$\therefore \quad p_K = \Delta x\,[m(x_K + 1) + h - y_K] - [(y_K + 1) - m(x_K + 1) - h)]$

$= \Delta x\,[m(x_K + 1) + h - y_K - y_K - 1 + m(x_K + 1) + h]$

$= \Delta x\,[2m(x_K + 1) + 2h - 2y_K - 1]$

$= 2\Delta y x_K + 2\Delta y + 2\Delta x h - 2\Delta x y k - \Delta x$

as $m = \Delta y / \Delta x$

$p_K = 2\Delta y x_K - 2\Delta x y_K + 2\Delta y + 2\Delta x h - \Delta x$

The above computation of p_K involves constant $2\Delta y + 2\Delta x h - \Delta x$ which can be eliminated by making decision parameter incremental. Therefore,

$p_{K+1} = 2\Delta y\, x_{K+1} - 2\Delta x y_{K+1} + 2\Delta y + 2\Delta x h - \Delta x$

$\therefore \quad p_{K+1} - p_K = 2\Delta y\, x_{K+1} - 2\Delta y x_K - (2\Delta x y_{K+1} - 2\Delta x y_K)$

But $x_{K+1} = x_K + 1$

$\therefore \quad p_{K+1} - p_K = 2\Delta y - 2\Delta x\,(y_{K+1} - y_K)$

$\therefore \quad \mathbf{p_{K+1} = p_K + 2\Delta y - 2\Delta x\,(y_{K+1} - y_K)}$

The initial value of decision parameter is

$\mathbf{p_0 = 2\Delta y - \Delta x}$

If $p_K < 0$, i.e. $d_2 < d_1$ then

$y_{K+1} = y_K$

$\therefore \quad \mathbf{p_{K+1} = p_K + 2\Delta y}$

If $p_K > 0$, i.e. $d_1 < d_2$, then

$y_{K+1} = y_K + 1$

$\therefore \quad \mathbf{p_{K+1} = p_K + 2\Delta y - 2\Delta x}$

- **Algorithm:**

1.Accept the end point co-ordinates of the line segment. At A (x_1, y_1) and B (x_2, y_2).

2.Display the first point (x_1, y_1) put pixel (x_1, y_1).

3.Calculate, $\Delta y = y_2 - y_1$

$\Delta x = x_2 - x_1$

Also calculate constants $2\Delta y$ and $2\Delta y - 2\Delta x$.

05

4.Calculate the initial value of decision parameter

$p_0 = 2\Delta y - \Delta x$

5.At each x_K position, starting at K = 0, perform the following test,

If $p_K < 0$, then

$x_{next} = x_{K+1} = x_K + 1$

$y_{next} = y_{K+1} = y_K$

and $p_{K+1} = p_K + 2\Delta y$

otherwise,

$x_{next} = x_{K+1} = x_K + 1$

$y_{next} = y_{K+1} = y_K + 1$

and $p_{K+1} = p_K + 2\Delta y - 2\Delta x$

6.Display the point i.e. put pixel (x_{K+1}, y_{K+1})

7.Repeat steps 5 and 6 until

$x_{K+1} = x_2, y_{K+1} = y_2$

8.Stop.

Given : x1=10 , y1=10 , x2 = 18 , y2 = 16

dx=8 , dy=6 ➔ 2(dy-dx) = -4 and 2dy=12

Initial Parameter p= 2dy-dx = 4

x	y	Pixel_Position (x,y)	P	Next x	Next y	p=p+2dy (if p<0) p=p+2(dy-dx)
10	10	(10,10)	4	11	11	0
11	11	(11,11)	0	12	12	-4
12	12	(12,12)	-4	13	12	8
13	12	(13,12)	8	14	13	4
14	13	(14,13)	4	15	14	0
15	14	(15,14)	0	16	15	-4
16	15	(16,15)	-4	17	15	8
17	15	(17,15)	8	18	16	4
18	16	(18,16)	4	19	17	0

b) **Explain the steps for 2D anticlockwise rotation about arbitrary point and provide a composite transformation for the same.** 10

Answer :

Let the arbitrary point be pa(xa,ya) and angle of rotation is θ

Steps :

1.> Translation (Tx = -xa , Ty= -ya)
2.> Rotation (θ)
3.> Translation (Tx=xa , Ty=ya)

Note : The single composite Matrix representing the concatenation of 3 transformation involved is expected , diagram showing the effect of each individual transformation expected.

Q 3 a) **Explain Liang Barsky line clipping algorithm. Apply the algorithm to clip the line with coordinates (30,60) and (60,20) against window(xmin,ymin)=(10,10) and (xmax,ymax)=(50,50).** (10)

Answer :

LIANG – BARSKY LINE CLIPPING ALGORITHM

(i) Consider a line segment with endpoints (x_1, y_1) and (x_2, y_2).

Equation of line is determined by,

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = u \quad ...(i)$$

$$x - x_1 = u\,(x_2 - x_1) \quad \text{and} \quad y - y_1 = u\,(y_2 - y_1)$$

$$x = x_1 + u\,(x_2 - x_1) = x_1 + u\,(\Delta x) \quad ...(ii)$$

$$y = y_1 + u\,(y_2 - y_1) = y_1 + u\,(\Delta y) \quad ...(iii)$$

(ii) Assuming that the clip window is a rectangle in standard position, any point (x, y) is visible if the following inequalities are satisfied.

$$x\omega_{min} \leq x \leq x\omega_{max}$$

$$y\omega_{min} \leq y \leq y\omega_{max}$$

where $(x\omega_{min}, x\omega_{max}, y\omega_{min}, y\omega_{max})$ are the window boundaries.

Each of these four inequalities can be expressed as:

$$u\,p_K \leq q_K \quad \text{for} \quad K = 1, 2, 3, 4.$$

where K = 1, 2, 3 and 4 correspond to the left, right, bottom and top boundaries.

$$p_1 = -\Delta x$$

$$q_1 = x_1 - x\omega_{min}.$$

$$p_2 = \Delta x$$

$$q_2 = x\omega_{max} - x_1.$$

$$p_3 = -\Delta y$$

$$q_3 = y_1 - y\omega_{min}.$$

$$p_4 = \Delta y$$

$$q_4 = y\omega_{max} - y_1.$$

Algorithm :

Step 1 Accept the line segment co-ordinates $\{x_1, y_1, x_2, y_2\}$ and window boundaries $\{x\omega_{min}, y\omega_{min}, x\omega_{max}, y\omega_{max}\}$

Step 2 Calculate p_K and q_K for K = 1, 2, 3, 4 such that:

p_1	=	$-\Delta x$	q_1	=	$x_1 - x\omega_{min}$
p_2	=	Δx	q_2	=	$x\omega_{max} - x_1$
p_3	=	$-\Delta y$	q_3	=	$y_1 - y\omega_{min}$
p_4	=	Δy	q_4	=	$y\omega_{max} - y_1$

Step 3 If $p_K = 0$, then

{ the line is parallel to K^{th} boundary.

If $q_K < 0$, then

{ line is outside the boundary. Discard the line segment STOP

}

If $q_K = 0$, then

{ line is inside the parallel boundary.

}

}

Step 4 Calculate $r_K = \frac{q_K}{r_K}$ for K = 1, 2, 3., 4.

Step 5 Determine u_1 for all $p_K < 0$ from the set consisting $\{r_K, 0\}$

Select r_K for all $p_K < 0$.

Then, $u_1 = \{r_K, 0\}_{max}$

Step 6 Determine u_2 for all $p_K > 0$ from the set consisting $\{r_K, 1\}$

Select r_K for all $p_K > 0$

Then, u_2 $\{r_K, 1\}_{min}$

Step 7 If $u_1 > u_2$ then

{the line is completely outside the boundary. Discard the line segment

STOP

}

Step 8 Calculate endpoints of the clipped line.

$\because$ $x' = x_1 + u_1 \Delta x$ $\therefore$ $I_1 (x', y')$

$\because$ $y' = y_1 + u_1 \Delta y$

$\because$ $x'' = x_1 + u_2 \Delta x$ $\therefore$ $I_2 (x'', y'')$

$\because$ $y'' = y_1 + u_2 \Delta y$

Step 9 Display the line segment $I_1 I_2$

STOP

Given : x1=30 , y1=60 , x2=60 , y2=20 $x\omega_{min}$=10 $y\omega_{min}$=10 $x\omega max$=50 $y\omega_{max}$=50

⇨ $\Delta x = 30$ $\Delta y = -40$

08

k	p_k	q_k	$r_k=q_k / p_k$
1	-30	20	-2/3
2	30	20	2/3
3	40	50	5/4
4	-40	-10	1/4

$u_1=\{0,-2/3,1/4\}_{max} = 1/4$ $u_1=\{1,2/3,1\}_{min} = 2/3$ +5/4

Since $u_1 <= u_2$

→

$x' = x_1 + u_1 \Delta x = 30 + 1/4 (30) = 37.5$

$y' = y_1 + u_1 \Delta y = 60 + 1/4 (-40) = 50$

$x'' = x_1 + u_2 \Delta x = 30 + 2/3 (30) = 50$

$y'' = y_1 + u_2 \Delta y = 60 + 2/3(-40) = 33.34$

∴ $I_1 (x', y') = (37.5 ,50)$

∴ $I_2 (x'', y'') = (50,33.34)$

Display the line between $I_1 I_2$

b) Explain Sutherland Hodgman polygon clipping algorithm with suitable example and comment on its shortcoming. **10**

Answer

Points to be covered :

1.> Diagram representing a pipeline of Left – Right –Top – Bottom Clipper
2.> 4 CASES to be handled by every clipper with appropriate diagram
3.> Example for illustration

Q 4 a) What is window and viewport? Derive the window to viewport transformation and also identify the geometric transformation involved. **10**

Answer

Points to be covered :

1.> Definition of Window , Viewport
2.> Diagram to represent Window and Viewport extents
3.> Derivation of equation for the purpose
4.> Identification of the geometric transformation involved.

b) Explain what is meant by Bezier curve? State the various properties of Bezier curve. **10**

Answer

Points to be covered :

1.> Parametric equation for n+1 control points and explanation of each of its component.
2.> Expressing equation in terms of x(u) and y(u)
3.> Explanation about how curve is plotted by gradually changing u from 0 to 1
4.> Properties of Bezier Curve

Q 5 a) What is meant by parallel and perspective projection? Derive matrix for oblique **10**

		projection. **Answer :** Points to be covered : 1.> Comparison between Parallel and Perspective projection with suitable diagram. 2.> Perspective Projection Anomalies 3.> Explanation about Oblique projection with the help of diagram. **4.>** Derivation of Matrix using the geometry.	
	b)	**Explain Z Buffer algorithm for hidden surface removal.** **Answer :** Points to be covered : 1.> Use of Depth buffer and refresh buffer. **2.>** Z buffer algorithm with suitable diagram.	**10**
Q 6		**Write short notes on(any two)**	**20**
	a)	**Koch curve**	
	b)	**Sweep representation and Octree representation**	
	c)	**Gouraud and phong shading**	
	d)	**Halftoning and Dithering.**	