APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION

Model Question Paper - SOLUTION

Course Code: CST 304

Course Name: Computer Graphics and Image Processing

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. **Justify the approach of using integer arithmetic in Bresenham's line drawing algorithm.**

Answer:

It is an efficient method because it involves only integer addition, subtractions, and multiplication operations. These operations can be performed very rapidly so lines can be generated quickly.

ANS)

(*) An incremental Scan conversion Algorithm.

(*) It scans the co-ordinates and instead of rounding them off, it takes the incremental value in account by adding or subtracting.

(*) Bresenhams algorithm uses ~~pts~~ fixed points ie, integer arithmetic and Bresenhams algorithm uses only subtraction and addition. So, it runs significantly faster.

2. Consider a raster system with a resolution of 1024*1024. What is the size of the raster needed to store 4 bits per pixel? How much storage is needed if 8 bits per pixel are to be stored?

Ans:) For 1 pixel needs 4 bits.

→ Resolution = 1024×1024

→ Total pixels = 1,049,576 pixels.

→ To represent 1048576 pixels we need = 1048576 × 4 bits

= 4,194304 bits

1 Byte = 8 bit

∴ 4194304 bits = $\frac{4194304}{8}$ = 524288 bytes

1 KB = 1024 Byte

∴ 524288 bytes = 512 KB

For 1 pixel needs 8 bits per

→ Resolution = 1024×1024

→ Total pixels = 1048576 pixels.

→ To represent 1048576 pixels we need = 8388608 bits.

1 Byte = 8 bit

∴ 8388608 bits = 1048576 bytes

1 KB = 1024 Byte

∴ 1048576 bytes = 1024 KB

For 8 bits per pixel ~~1024 KB~~ Size of 1024 KB is needed to storage.

Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin

Ans) Single rotation = 360°

Reflection 1

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_1' = \text{Reflection}_1 * P$$

Reflection 2

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_2' = \text{Reflection}_2 * P$$

$P = P_1 * P_2$

$$\text{LHS} \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation ($\theta = 360^\circ$)

$P' = R(\theta) \cdot P$

$$\text{RHS} \Rightarrow = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos 360 & -\sin 360 & 0 \\ \sin 360 & \cos 360 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

LHS = RHS Hence proved.

4. Determine a sequence of basic transformations that are equivalent to the x-direction shearing matrix.

A shear transformation A can be interpreted as a rotation, followed by a nonuniform scaling, followed by another rotation. In fact, you can see this immediately if you consider the Singular Value Decomposition (see "Matrix Computations" by Golub and Van Loan), $A = USV^T$ where both U and V are orthogonal (rotation) matrices, and S is a diagonal scaling matrix.

It is sufficient to consider 2D shear (3D is analogous), and for the case of a shear along a particular axis. A particularly nice identity for shear in the y-direction (x-direction is just the transpose) factorized into rotation*scaling*rotation is

$$\begin{bmatrix} 1 & 0 \\ a - \frac{1}{a} & 1 \end{bmatrix} = \left(\frac{1}{\sqrt{1+a^2}} \begin{bmatrix} 1 & -a \\ a & 1 \end{bmatrix} \right) \begin{bmatrix} a & 0 \\ 0 & \frac{1}{a} \end{bmatrix} \left(\begin{bmatrix} a & 1 \\ -1 & a \end{bmatrix} \frac{1}{\sqrt{1+a^2}} \right).$$

This can be interpreted as follows. If we consider an angle α such that $\tan\alpha = \frac{1}{a}$, then (using $\sin\alpha = \frac{1}{\sqrt{1+s^2}}$ and $\cos\alpha = \frac{s}{\sqrt{1+s^2}}$) the rotation matrix

$$R(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} = \frac{1}{\sqrt{1+a^2}} \begin{bmatrix} a & -1 \\ 1 & a \end{bmatrix}$$

can be used to observe that the last rotation matrix is just $R(-\alpha)$. We can also observe that

$$R(\frac{\pi}{2} - \alpha) = \begin{bmatrix} \sin\alpha & -\cos\alpha \\ \cos\alpha & \sin\alpha \end{bmatrix} = \frac{1}{\sqrt{1+a^2}} \begin{bmatrix} 1 & -a \\ a & 1 \end{bmatrix}.$$

Therefore the shear factorization is

$$R(-\alpha) \begin{bmatrix} a & 0 \\ 0 & \frac{1}{a} \end{bmatrix} R(\frac{\pi}{2} - \alpha).$$

Note that unless $\alpha = \frac{\pi}{4} \Leftrightarrow a = 1$, and hence the shear is simply the identity, the second rotation is never simply the inverse of the first.

5. Find the window to viewport normalization transformation with window lower left corner at (1,1) and upper right corner at (2,6).

Solution:-

Matrix representation

$$M_{vw} = T_{(u_{min}, v_{min})} * S_{(S_x, S_y)} * T_{(-x_{min}, -y_{min})}$$

$$S_x = \frac{u_{max} - u_{min}}{x_{max} - x_{min}}$$

$$S_y = \frac{v_{max} - v_{min}}{y_{max} - y_{min}}$$

The view coordinates are not given. So take the range as (0,0) to (1,1)

Lower: (1,1)
Upper: (2,6)

$X_{min} = 1$ $\quad Y_{min} = 1$ $\quad S_x = \frac{1-0}{2-1} = \underline{\underline{1}}$

$X_{max} = 2$ $\quad Y_{max} = 6$

$U_{min} = 0$ $\quad V_{min} = 0$ $\quad S_y = \frac{1-0}{6-1} = \underline{\underline{\frac{1}{5}}}$

$U_{max} = 1$ $\quad V_{max} = 1$

$$M_{vw} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{T(0,0)}^{T(u_{min}, v_{min})} * \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/5 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{S(1, 1/5)}^{S(S_x, S_y)} * \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}}_{T(-1,-1)}^{T(-x_{min}, -y_{min})}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} = \underline{\underline{\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1/5 & -1/5 \\ 0 & 0 & 1 \end{bmatrix}}}$$

6. Find the orthographic projection of a unit cube onto the x=0, y=0 and z=0 plane.

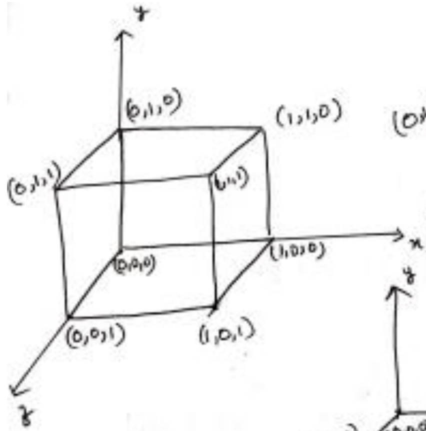
<u>x=0 plane</u>

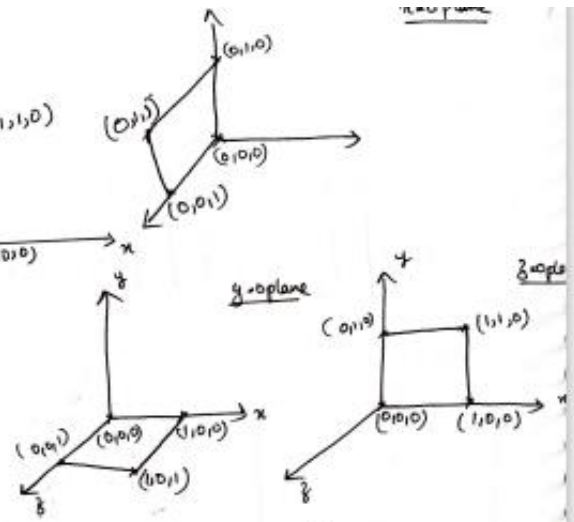
$P' = P_{x=0} \cdot P$

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

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

$= (0,0,0)\ (0,0,0)\ (0,1,0)\ (0,1,0)\ (0,1,1)\ (0,1,1)$
$(0,0,1)\ (0,0,1)$





<u>y = 0 plane</u>

$P' = P_{y=0} \cdot P$

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

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{matrix} (0,0,0) & (1,0,0) & (0,0,1) & (1,0,1) \\ (0,0,0) & (1,0,0) & (0,0,1) & (1,0,1) \end{matrix}$$

<u>z = 0 plane</u>

$P' = P_{z=0} \cdot P$

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

(0,0,0) (0,0,0) (1,0,0) (1,0,0)
(1,1,0) (1,1,0) (0,1,0) (0,1,0)

7. Define Sampling and Quantization of an image.

In order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude.

A frame grabber or digitizer gets the analogue video signal and to create an image which is digital, we need to covert continuous data into digital form.

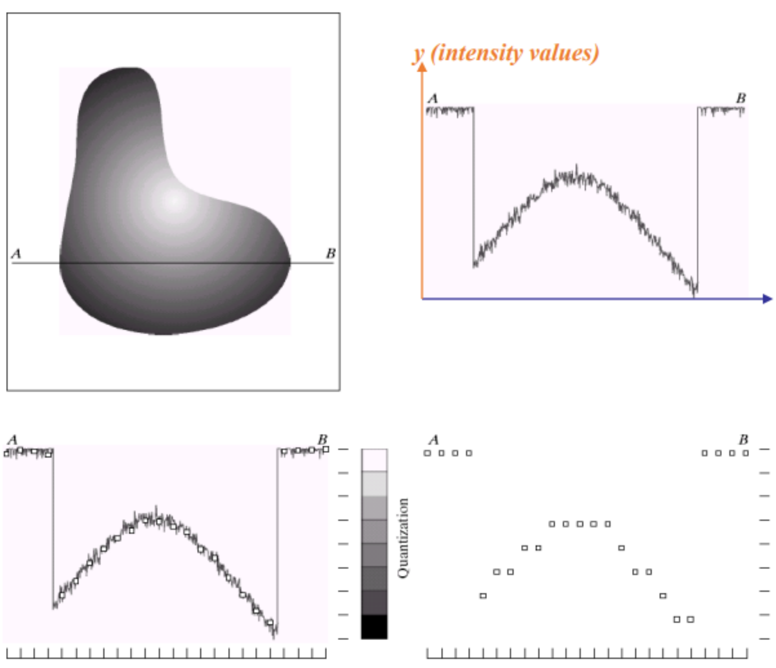
There are two steps in which it is done:

- Sampling
- Quantization

  - **The sampling rate determines the spatial resolution of the digitized image**
  - **The quantization level determines the number of grey levels in the digitized image.**

A magnitude of the sampled image is expressed as a digital value in image processing.

The transition between continuous values of the image function and its digital equivalent is called quantization.



a b
c d

Generating a digital image. (a) Continuous image. (b) A scaling line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) sampling and quantization. (d) Digital scan line.

8. Give any three applications of digital image processing

1) Image sharpening and restoration
It refers to the process in which we can modify the look and feel of an image. It basically manipulates the images and achieves the desired output. It includes conversion, sharpening, blurring, detecting edges, retrieval, and recognition of images.

2) Medical Field
There are several applications under medical field which depends on the functioning of digital image processing.
- Gamma-ray imaging
- PET scan
- X-Ray Imaging
- Medical CT scan
- UV imaging

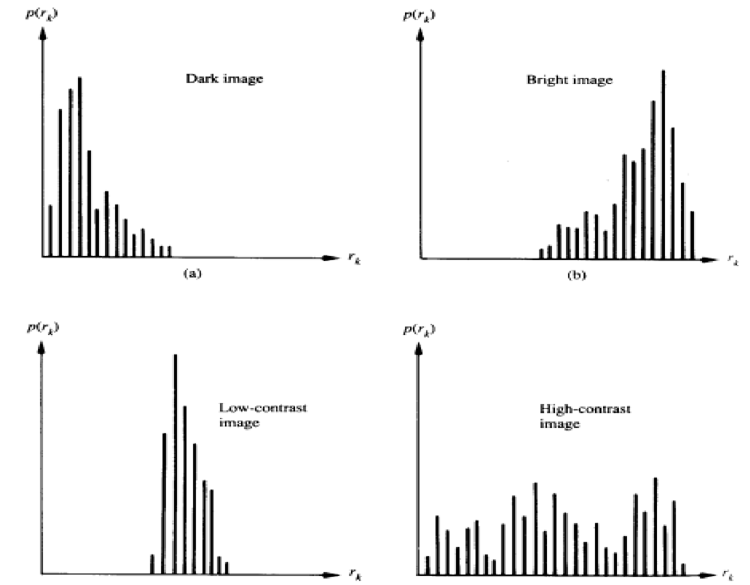
3) Robot vision
There are several robotic machines which work on the digital image processing. Through image processing technique robot finds their ways, for example, hurdle detection root and line follower robot.

4) Pattern recognition
It involves the study of image processing, it is also combined with artificial intelligence such that computer-aided diagnosis, handwriting recognition and images recognition can be easily implemented. Now a days, image processing is used for pattern recognition.

5) Video processing
It is also one of the applications of digital image processing. A collection of frames or pictures are arranged in such a way that it makes the fast movement of pictures. It involves frame rate conversion, motion detection, reduction of noise and colour space conversion etc.

9. A captured image appears very dark because of wrong lens aperture setting. Describe an enhancement technique which is appropriate to enhance such an image.

- **Histogram equalization (HE)** is a simple and effective contrast enhancement technique for enhancing an image.
- HE spreads the intensities of an image pixels based on the whole image information.
- The shape of the histogram of an image gives us useful information about the possibility for contrast enhancement.



- Aim is to transform the first 3 histograms into the 4th type. That is try to increase the dynamic range of the image . This is called **Histogram Processing**
- In **dark Image** the components of the histogram are concentrated on the low (dark) side of gray scale
- The components of the histogram of the **bright image** are biased toward the high side of gray scale.
- A **low contrast image** histogram will be narrow and centered towards the middle of the gray scale.
- Components of the histogram in the **high-contrast image** cover a broad range of the gray scale and, further, that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others.

10. Suggest an approach of thresholding that should be used in case of uniform illumination.

Global Thresholding

- The simplest of all thresholding techniques is to partition the image histogram by using a single global threshold, T.
- Segmentation is then accomplished by scanning the image pixel by pixel
- Labeling each pixel as object or back-ground, depending on whether the gray level of that pixel is greater or less than the value of T

- Binary Thresholding

THRESH_BINARY

$$\mathtt{dst}(x,y) = \begin{cases} \mathtt{maxval} & \text{if } \mathtt{src}(x,y) > \mathtt{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Input Image

3	0	1	5
7	6	0	4
2	7	0	6
1	3	5	5

4 X 4 Image Threshold =4

Output image matrix=

0 0 0 7
7 7 0 0
0 7 0 7
0 0 7 7

Part B

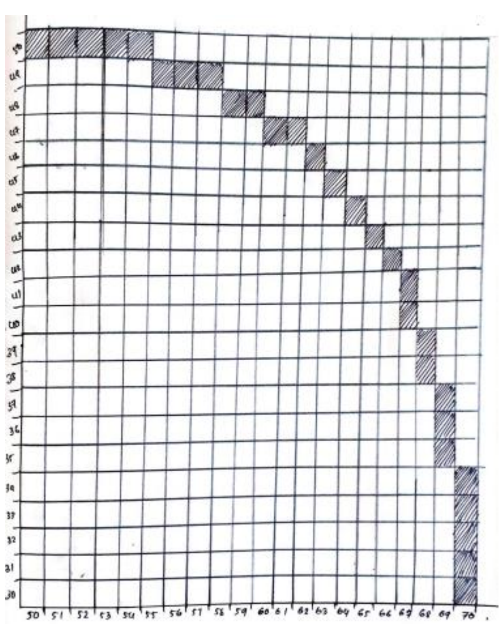
(Answer any one question from each module. Each question carries 14 Marks)

11 (a) Write Midpoint circle drawing algorithm and use it to plot a circle withradius=20 and center is (50,30).

Algorithm

1. Begin
2. input the radius r and center of the circle and obtain the initial point as (0,r) for a circle centered at origin.
3. calculate the initial decision parameter,
   $P_0 = 1 - r$
4. if $P_k < 0$
   (a) Next point is (x_{k+1}, y_k) → (N)
   (b) $P_{k+1} = P_k + 2x_k + 3$
5. Else
   (a) Next point is (x_{k+1}, y_{k-1}) → (S)
   (b) $P_{k+1} = P_k + 2(x_k - y_k) + 5$
6. Determine the symmetry points in the other 7 octants.
7. move each calculated position (x,y) on to a circular path centered at (x_c, y_c) & plot the coordinate value $x = x + x_c$ and $y = y + y_c$
8. Repeat the steps from 4 until $x \ge y$.

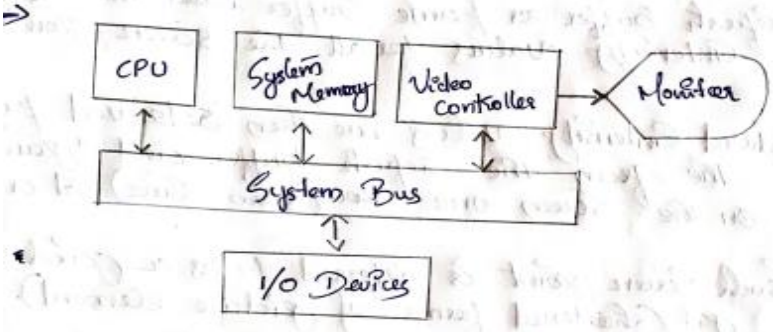
K	P_K	N/S	pixel
0	$P_0 = 1-r = 1-20 = -19<0$	N (x_{k+1}, y_k)	(0,20) (1,20)
1	$P_1 = P_0 + 2x_k + 3 = -19+(2\times0)+3 = -16<0$	N (x_{k+1}, y_k)	(2,20)
2	$P_2 = P_1 + 2x_k + 3 = -16+(2\times1)+3 = -11<0$	N (x_{k+1}, y_k)	(3,20)
3	$P_3 = P_2 + 2x_k + 3 = -11+(2\times2)+3 = -4<0$	N (x_{k+1}, y_k)	(4,20)
4	$P_4 = P_3 + 2x_k + 3 = -4+(2\times3)+3 = 5>0$	S (x_{k+1}, y_{k-1})	(5,19)
5	$P_5 = P_4 + 2(x_k - y_k) + 5 = 5+2(4-20)+5 = -22<0$	N (x_{k+1}, y_k)	(6,19)
6	$P_6 = P_5 + 2x_k + 3 = -22+2(5)+3 = -9<0$	N (x_{k+1}, y_k)	(7,19)
7	$P_7 = P_6 + 2x_k + 3 = -9+(2\times6)+3 = 6>0$	S (x_{k+1}, y_{k-1})	(8,18)
8	$P_8 = P_7 + (2(x_k - y_k)) + 5 = 6+2(7-19)+5 = -13<0$	N (x_{k+1}, y_k)	(9,18)
9	$P_9 = P_8 + 2x_k + 3 = -13+(2\times8)+3 = 6>0$	S (x_{k+1}, y_{k-1})	(10,17)
10	$P_{10} = P_9 + 2(x_k - y_k) + 5 = 6+2(9-18)+5 = -7<0$	N (x_{k+1}, y_k)	(11,17)
11	$P_{11} = P_{10} + 2x_k + 3 = -7+(2\times10)+3 = 16>0$	S (x_{k+1}, y_{k-1})	(12,16)
12	$P_{12} = P_{11} + 2(x_k - y_k) + 5 = 16+2(11-17)+5 = 9>0$	S (x_{k+1}, y_{k-1})	(13,15)
13	$P_{13} = P_{12} + 2(x_k - y_k) + 5 = 9+2(12-16)+5 = 6>0$	S (x_{k+1}, y_{k-1})	(14,14)



octant 1	octant 2
(0,20) + (50,30) = (50,50)	(20,0) + (50,30) = (70,30)
(1,20) + (50,30) = (51,50)	(20,1) + (50,30) = (70,31)
(2,20) + (50,30) = (52,50)	(20,2) + (50,30) = (70,32)
(3,20) + (50,30) = (53,50)	(20,3) + (50,30) = (70,33)
(4,20) + (50,30) = (54,50)	(20,4) + (50,30) = (70,34)
(5,19) + (50,30) = (55,49)	(19,5) + (50,30) = (69,35)
(6,19) + (50,30) = (56,49)	(19,6) + (50,30) = (69,36)
(7,19) + (50,30) = (57,49)	(19,7) + (50,30) = (69,37)
(8,18) + (50,30) = (58,48)	(18,8) + (50,30) = (68,38)
(9,18) + (50,30) = (59,48)	(18,9) + (50,30) = (68,39)
(10,17) + (50,30) = (60,47)	(17,10) + (50,30) = (67,40)
(11,17) + (50,30) = (61,47)	(17,11) + (50,30) = (67,41)
(12,16) + (50,30) = (62,46)	(16,12) + (50,30) = (66,42)
(13,15) + (50,30) = (63,45)	(15,13) + (50,30) = (65,43)
(14,14) + (50,30) = (64,44)	(14,14) + (50,30) = (64,44)

b) Draw the architecture of raster scan display systems and explain its workingprinciple.**(4)**

⇒



Processing Units :

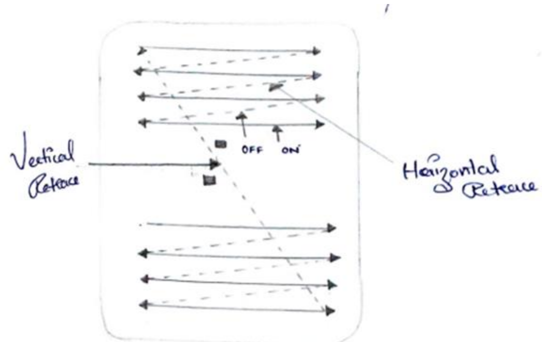
- CPU
- Video Controller or Display Controller – Special purpose processor.
  - * To control the operation of the display device.

Frame Buffer

- The frame buffer can be anywhere in the system memory.
- The video controller accesses the frame buffer to refresh the screen.

- The electron beam is swept across the screen, one row at a time from top to bottom.
- As the electron beam moves the beam intensity is turned on and off to create a pattern of illuminated spots.
- Refresh buffer or frame buffer holds the set of intensity values for all the screen points.
- Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row (scan line) at a time.
- Each screen point is referred to as a pixel or pel (shortened form of picture element)
- Refreshing is carried out at the rate of 60 to 80 frames per second.
- When beam is moved from right to left after one row, it is off. This process is called HORIZONTAL RETRACE.

When beam reaches the bottom of the screen, it is made off and rapidly retraced back to the left top corner and the process is continued. This process is called VERTICAL RETRACE.



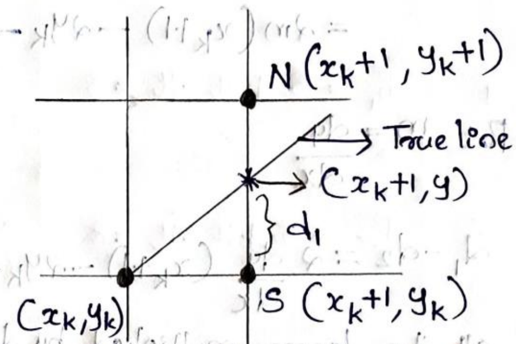
Raster Scan CRT.

OR

12 (a) Derive the initial decision parameter of Bresenham's line drawing algorithmand use the algorithm to rasterize a line with endpoints (2,2) and (10,10).

12a) Derive the initial decision parameter of Bresenham's line drawing algorithm and use the algorithm to rasterize a line with endpoints (2,2) and (10,10).

Bresenham's Line drawing derivation



Slope intercept form

$$y = mx + c \quad \text{---(1)}$$

$$y = m(x_k+1) + c \quad \text{---(2)}$$

slope intercept form of the true line

$$d_1 = y - y_k \quad \text{---(3)}$$

$$d_2 = y_k + 1 - y \quad \text{---(4)}$$

Put eq^n (2) in eq^n (3) and (4)

$$d_1 = m(x_k+1) + c - y_k \quad \text{---(5)}$$

$$d_2 = y_k + 1 - [m(x_k+1) + c]$$

$$= y_k + 1 - m(x_k+1) - c \quad \text{---(6)}$$

(10)

$eq^n(5) - eq^n(6)$

$$d_1 - d_2 = m(x_k+1) + c - y_k - (y_k+1 - m(x_k+1) - c)$$

$$= m(x_k+1) + c - y_k - y_k - 1 + m(x_k+1) + c$$

$$= 2m(x_k+1) - 2y_k - 1 + 2c$$

Put $m = \frac{dy}{dx}$

$$d_1 - d_2 = 2\frac{dy}{dx}(x_k+1) - 2y_k + 2c - 1$$

all the terms multiplied by dx

$$\boxed{dx(d_1 - d_2) = 2dy(x_k+1) - 2y_k dx + 2c dx - dx} \quad \text{(7)}$$

Initial decision parameter

At (x_0, y_0)

$y_0 = mx_0 + c$

$c = y_0 - mx_0$ $\qquad m = \frac{dy}{dx}$

$c = y_0 - \frac{dy}{dx}x_0$ —— (8)

Put $eq^n(8)$ in $eq^n(7)$

$x_k = x_0$ and $y_k = y_0$

$$P_0 = 2dy(x_0+1) - 2y_0 dx + 2\left(y_0 - \frac{dy}{dx} \cdot x_0\right)dx - dx$$

$$= \cancel{2dy\,x_0} + 2dy - \cancel{2y_0 dx} + \cancel{2y_0 dx} - \cancel{2\frac{dy}{dx} x_0 dx} - dx$$

$$\boxed{P_0 = 2dy - dx}$$

Rasterize the line with the endpoints (2,2) and (10,10)

$dx = 10 - 2 = 8 \qquad 2dx = 16$

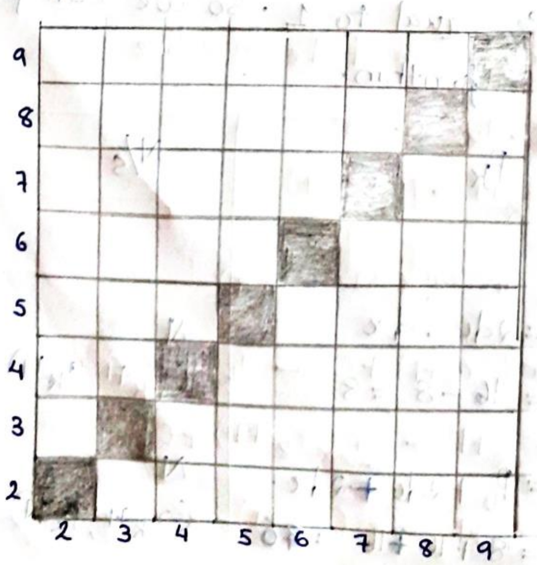
$dy = 10 - 2 = 8 \qquad 2dy = 16$

$\text{Slope} = \frac{dy}{dx} = \frac{8}{8} = 1$

Since it is equal to 1. So we can proceed with this algorithm.

k	P_k	N/S	Pixel
			(2,2)
0	$P_0 = 2dy - dx$ $= 16 - 8 = 8$	N (x_k+1, y_k+1)	(3,3)
1	$P_1 = P_0 + 2dy - 2dx$ $= 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(4,4)

2	$P_2 = P_1 + 2dy - 2dx$ $= 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(5,5)
3	$P_3 = P_2 + 2dy - 2dx$ $= 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(6,6)
4	$P_4 = 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(7,7)
5	$P_5 = 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(8,8)
6	$P_6 = 8 + 16 - 16 = 8$	N (x_k+1, y_k+1)	(9,9)
7	$P_7 = 8 + 16 - 16$	N (x_k+1, y_k+1)	(10,10)



b) Explain the working principle of color CRT monitors with suitableillustrations.

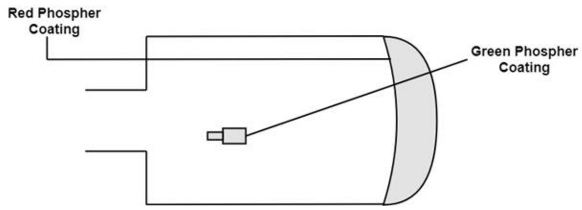
Color CRT Monitors:

The CRT Monitor display by using a combination of phosphors. The phosphors are different colors. There are two popular approaches for producing color displays with a CRT are:

1. Beam Penetration Method
2. Shadow-Mask Method

1. Beam Penetration Method:

The Beam-Penetration method has been used with random-scan monitors. In this method, the CRT screen is coated with two layers of phosphor, red and green and the displayed color depends on how far the electron beam penetrates the phosphor layers. This method produces four colors only, red, green, orange and yellow. A beam of slow electrons excites the outer red layer only; hence screen shows red color only. A beam of high-speed electrons excites the inner green layer. Thus screen shows a green color.



2. Shadow-Mask Method:

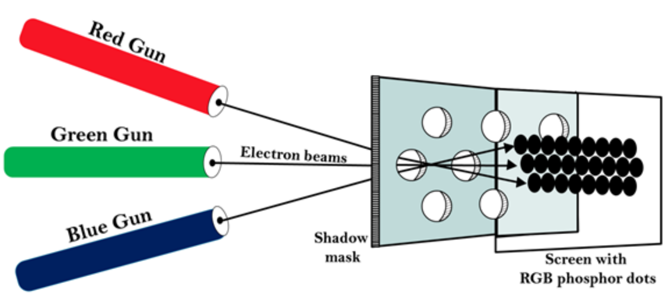
- Shadow Mask Method is commonly used in Raster-Scan System because they produce a much wider range of colors than the beam-penetration method.

Construction: A shadow mask CRT has 3 phosphor color dots at each pixel position.

- One phosphor dot emits: red light
- Another emits: green light
- Third emits: blue light

This type of CRT has 3 electron guns, one for each color dot and a shadow mask grid just behind the phosphor coated screen.

Shadow mask grid is pierced with small round holes in a triangular pattern.
Figure shows the delta-delta shadow mask method commonly used in color CRT system.



- **Working:** Triad arrangement of red, green, and blue guns.
- The deflection system of the CRT operates on all 3 electron beams simultaneously; the 3 electron beams are deflected and focused as a group onto the shadow mask, which contains a sequence of holes aligned with the phosphor- dot patterns.
- When the three beams pass through a hole in the shadow mask, they activate a dotted triangle, which occurs as a small color spot on the screen.
- The phosphor dots in the triangles are organized so that each electron beam can activate only its corresponding color dot when it passes through the shadow mask.
- **Inline arrangement:** Another configuration for the 3 electron guns is an Inline arrangement in which the 3 electron guns and the corresponding red-green-blue color dots on the screen, are aligned along one scan line rather of in a triangular pattern.
- This inline arrangement of electron guns in easier to keep in alignment and is commonly used in high-resolution color CRT's.

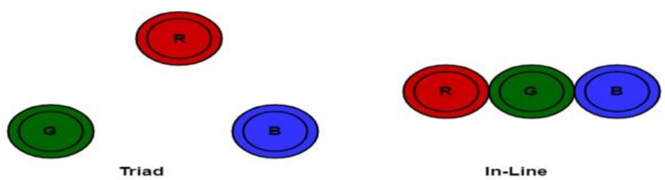


Fig: Triad-and -in-line arrangements of red, green and blue electron guns of CRT for color monitors.

13 (a) Compare boundary fill algorithm and flood fill algorithm. (5)

Flood-fill algorithm	Boundary-fill algorithm
1. It can process the image containing more than one boundary colours.	1. It can only process the image containing single boundary colour.
2. Flood-fill algorithm is comparatively slower than the boundary-fill algorithm.	2. Boundary-fill algorithm is faster than the flood-fill algorithm.
3. In this algorithm a random colour can be used to paint the interior portion then the old one is replaced with a new one.	3. In this algorithm interior points are painted by continuously searching for the boundary colour.
4. It requires huge amount of memory.	4. It requires low amount of memory.
5. Simple and efficient	5. Complexity is high.

b) Reflect a triangle ABC about the line 3x-4y+8=0. The position vector of the coordinate ABC is given as A(4,1), B(5,2) and C(4,3). (9)

$$3x - 4y + 8 = 0$$
$$4y = 3x + 8$$
$$y = \frac{3}{4}x + 2$$

Comparing with $y = mx + c$,

$$m = \frac{3}{4} \text{ and } c = 2$$

$y = mx + c$
$y = 3x + 10$
when $x = 0$
$y = 10$
$\therefore$ y-intercept, $c = 10$
Gradient $= 3$

By the equation,

$$\tan\theta = \frac{\Delta y}{\Delta x} = m$$
$$\theta = \tan^{-1}(m)$$
$$\theta = \tan^{-1}(3/4) = 36.87^\circ$$

(To bring the line to x axis clockwise rotation must be considered)

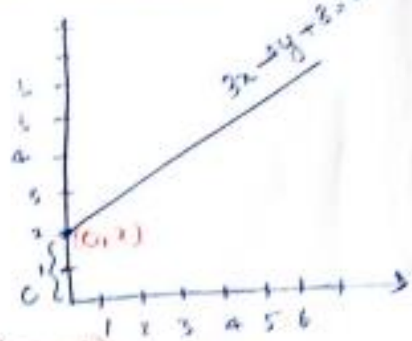
Step 1: Bring the line to origin - $T(0,-2)$

Step 2: Rotate the line so that it merges with the x axis $R(36.87^\circ)$

Step 3: Reflection w.r.t x axis R_x

Step 4: Reverse rotation ( ) $R(-36.87)$

Step 5: Bring the line back to (0,2) $T(0,2)$



$$P' = T_{(0,2)} * R^{T}_{(36.87)} * R_x * R_{(36.87)} * T_{(0,-2)} * P$$

$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & +2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(36.87) & \sin(36.87) & 0 \\ -\sin(36.87) & \cos(36.87) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(36.87) & -\sin(36.87) & 0 \\ \sin(36.87) & \cos(36.87) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} P$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 & 0.12 & 0 \\ -0.12 & 0.8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 & -0.12 & 0 \\ 0.12 & 0.8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} * P$$

$$= \begin{bmatrix} 0.63 & -0.19 & 0.38 \\ -0.19 & -0.63 & 3.25 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 5 & 4 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 2.69 & 3.13 & 2.31 \\ 1.86 & 1.04 & 0.61 \\ 1 & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 3 & 3 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

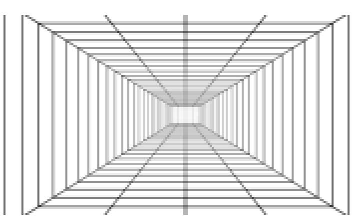
The new coordinates are $A'(3,2)$
$B'(3,1)$
$C'(2,1)$

OR

(a) Explain the need of using vanishing points in projections. **(4)**

A vanishing point is a point on the image plane of a perspective drawing where the two-dimensional perspective projections of mutually parallel lines in three-dimensional space appear to converge.

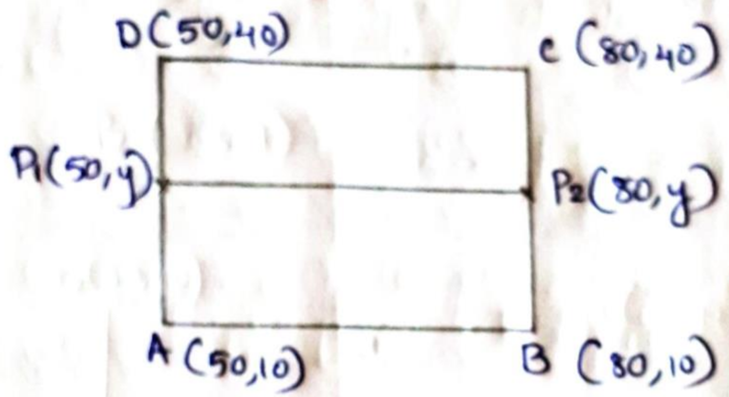
The vanishing point may also be referred to as the “direction point”.



Vanishing Point Perspective is used in Graphic editing and 3D video games. It can be used to render 3D shapes (3D Buildings and objects), add perspective to a background scene (road, train track) or add shadow effects.

The above image demonstrates the use of vanishing points in computer graphics.

Explain Cohen-Sutherland line clipping algorithm. Use the algorithm to clip line P1(70, 20) and P2(100,10) against a window lower left hand corner (50,10) and upper right hand corner (80,40). (10)



$y_T = 40$ $x_R = 80$

$y_B = 10$ $x_L = 50$

$P_1(70,20)$			$P_2(100,10)$		
T –	20-40	0	T –	10-40	0
B –	10-20	0	B –	10-10	0
R –	70-80	0	R –	100-80	1
L –	50-70	0	L –	50-100	0

P_1	P_2	Bitwise OR	Bitwise AND
0	0	0	0
0	0	0	0
0	1	1	0
0	0	0	0

Bitwise OR yields a non zero Value. ∴ The line is not Completely inside.

Bitwise AND yields all zero values. ∴ It is not Completely outside.

The line PQ is partially inside and partially outside. we have to find the intersection using the standard formula.

Left Intersection

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

$$= -0.33(50-70) + 20$$

$$= (-0.33 \times -20) + 20$$

$$= \underline{\underline{26.67}} \approx 27$$

$m = dy/dx$

$dx = 100 - 70 = 30$

$dy = 10 - 20 = -10$

$m = -10/30 = \underline{\underline{-0.33}}$

The point 27 is between the range of y_T and y_B. Therefore the Point (50,27) is accepted.

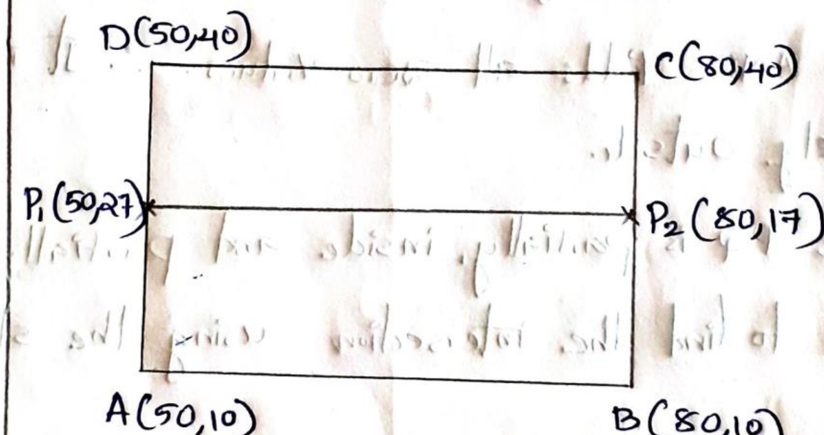
Right Intersection

$$y = m(x_R - x_1) + y_1$$
$$= -0.33\,(80-70) + 20$$
$$= -0.33 \times 10 + 20$$
$$= 16.67 \approx 17$$

The point 17 is between the range of y_T and y_B.

$\therefore$ The point (80,17) is accepted.

$P_2\,(80,17)$



Algorithm

1. Begin
2. Given a line Segment with end points (x_1,y_1) and (x_2,y_2)
3. Compute TBRL Codes for each point.
4. Perform bitwise OR operation on the outcode

a) If it yields 0000, it is Completely inside the window
b) pass the endpoints to the draw routine

5. Perform Bitwise AND
   a) If it doesnot yields 0000, the line is outside the window.
   b) It can be Trivially Rejected.

6. If it doesnot belong to any of these 2 categories above the line must be clip at the window edge using the following intersection equations.

1. Left : (x_L, y)
   where
   $y = m(x_L - x_1) + y_1$ where $(m \neq \alpha)$

2. Right : (x_R, y)
   where
   $y = m(x_R - x_1) + y_1$ where $(m \neq \alpha)$

3. Top : (x, y_T)
   $x = 1/m\,(y_T - y_1) + x_1$ where $(m \neq 0)$

4. Bottom : (x, y_B)
   $x = 1/m\,(y_B - y_1) + x_1$ where $(m \neq 0)$

(a) Describe Sutherland Hodegman polygon clipping algorithm and what are its limitations (7)

Sutherland-Hodgeman Polygon Clipping Algorithm

1. Begin
2. Read coordinates of all vertices of the Polygon.
3. Read coordinates of the clipping window
4. Consider the left edge of the window
5. Compare the vertices of each edge of the polygon, individually with the clipping plane.
6. Save the resulting intersections and vertices in the new list of vertices according to four possible relationships between the edge and the clipping boundary.
7. Repeat the steps 5 and 6 for remaining edges or the clipping window. Each time the resultant list of vertices is successively passed to process the next edge of the clipping window.
8. End
9.

Table 1: Rules for Sutherland-Hodgman clipping.

Case	1st vertex	2nd vertex	output
1	inside	inside	2nd vertex
2	inside	outside	intersection
3	outside	outside	none
4	outside	inside	2nd and intersection

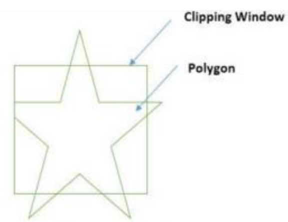


Figure: Clipping Left Edge

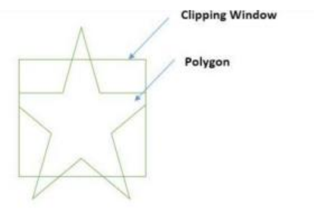


Figure: Clipping Right Edge

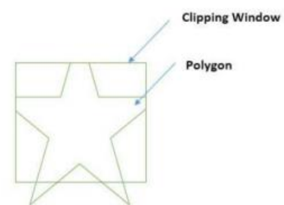


Figure: Clipping Top Edge

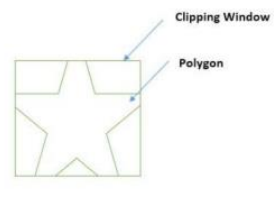


Figure: Clipping Bottom Edge

Limitations

- Convex polygons are correctly clipped by the Sutherland-Hodgeman Algorithm. But, concave polygons cannot be clipped correctly.
- It may be displayed with extraneous lines. Example shown in the following figure.

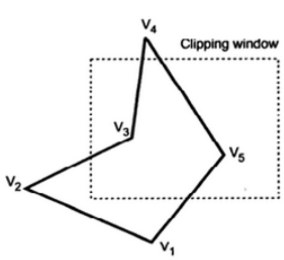
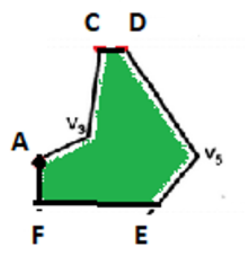


Fig. Before Clipping



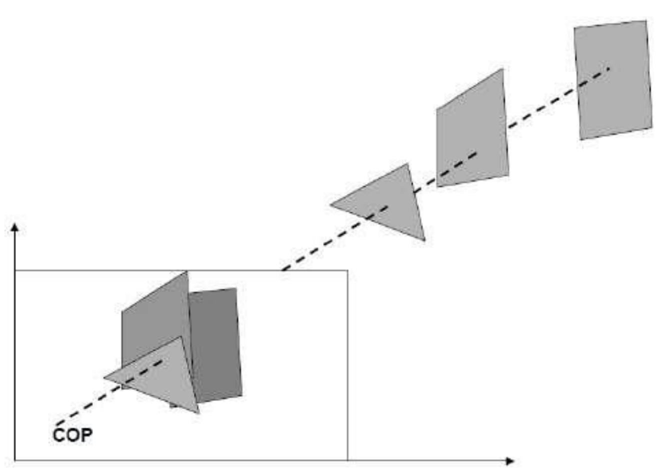
- **Here Edges AF,EF are extra edges**
- **The algorithm clips convex polygons correctly**
- **But for concave polygons it display extraneous edges**
- **This drawback is overcome in Weiler Atherton Clipping Algorithm**

(b) How visible surfaces can be detected using depth buffer algorithm. **(7)**

Depth Buffer (Z-Buffer Method)

- This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest visible surface.
- In this method each surface is processed separately one pixel position at a time across the surface.
- The depth values for a pixel are compared and the closest smallest z surface determines the color to be displayed in the frame buffer.
- It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order.
- To override the closer polygons from the far ones, two buffers named **frame buffer** and **depth buffer,** are used.
- **Depth buffer** is used to store depth values for x,y position, as surfaces are processed $0 \le depth \le 1$.
- The **frame buffer** is used to store the intensity value of color value at each position x,y
- The z-coordinates are usually normalized to the range [0, 1].
- The 0 value for z-coordinate indicates back clipping pane and 1 value for z-coordinates indicates front clipping pane.
- The algorithm:

1. Begin
2. Set the buffer values –
   - Depthbuffer (x, y) = 0
   - Framebuffer (x, y) = background color
3. Process each polygon (One at a time)
   1. For each projected (x, y) pixel position of a polygon, calculate depth Z.
   2. If Z > depthbuffer (x, y)
      - Compute surface color,
      - set depthbuffer (x, y) = z,
      - framebuffer (x, y) = surfacecolor (x, y)
5. End



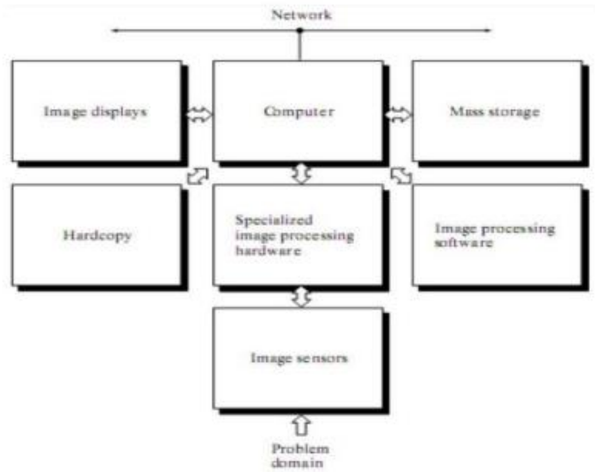
17. (a) Explain the components of an image processing system with suitable diagram **(9)**

- Image Processing System is the combination of the different elements involved in the digital image processing.

- Digital image processing is the processing of an image by means of a digital computer.

- Digital image processing uses different computer algorithms to perform image processing on the digital images.

It consists of following components:-



- **Image Sensors:**
  Image sensors senses the intensity, amplitude, co-ordinates and other features of the images and passes the result to the image processing hardware. It includes the problem domain.
- **Image Processing Hardware:**
  Image processing hardware is the dedicated hardware that is used to process the instructions obtained from the image sensors. It passes the result to general purpose computer.
- **Computer:**
  Computer used in the image processing system is the general purpose computer that is used by us in our daily life.
- **Image Processing Software:**
  Image processing software is the software that includes all the mechanisms and algorithms that are used in image processing system.
- **Mass Storage:**
  Mass storage stores the pixels of the images during the processing.
- **Hard Copy Device:**
  Once the image is processed then it is stored in the hard copy device. It can be a pen drive or any external ROM device.
- **Image Display**:

It includes the monitor or display screen that displays the processed images.

- **Network:**

Network is the connection of all the above elements of the image processing system.

(b) Define Resolution of an image. Explain the spatial and gray level resolutionof an image with an example. **(5)**

Resolution

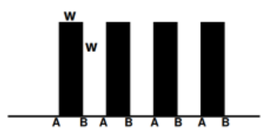
Image resolution is typically described in PPI, which refers to how many pixels are displayed per inch of an image.

Higher resolutions mean that there more pixels per inch (PPI), resulting in more pixel information and creating a high-quality, crisp image.

Images with lower resolutions have fewer pixels, and if those few pixels are too large (usually when an image is stretched), they can become visible

Spatial resolution

- Spatial resolution is the smallest distinguishable detail in an image.
- It depends on sampling.



Line pairs : AA and BB

Distance between the line pairs = 2w

No. of lines per unit length = $\frac{1}{2w}$

Spatial resolution = no. of distinguishable lines/length

$= \frac{1}{2W}$

Gray Level Resolution

- Refers to the smallest distinguishable change in the gray level.
- Gray level resolution is highly subjective and it depends on the hardware utilized to capture the image
- In short gray level resolution is equal to the number of bits per pixel.
- The number of different colors in an image is depends on the depth of color or bits per pixel.

The mathematical relation that can be established between gray level resolution and bits per pixel can be given as.

$$L = 2^k$$

In this equation L refers to number of gray levels. k refers to bpp or bits per pixel.
For example: Consider an image with 8 bits per pixel or 8bpp.
Now if were to calculate the gray level resolution:

$L = 2^k$

Where k = 8

$L = 2^8$

L = 256.

It means it gray level resolution is 256. Or in other way we can say that this image has 256 different shades of gray.
The more is the bits per pixel of an image, the more is its gray level resolution.

OR

18. (a) Define 4-adjacency, 8 adjacency and m-adjacency. Consider the image segment shown. **(7)**

4 2 3 2 (q)
3 3 1 3
2 3 2 2
(p)2 1 2 3

Let V={1,2} and compute the length of the shortest 4- ,8- and m- path between p and q. If a particular path does not exist between these two points, explain why?

Three types of adjacency:

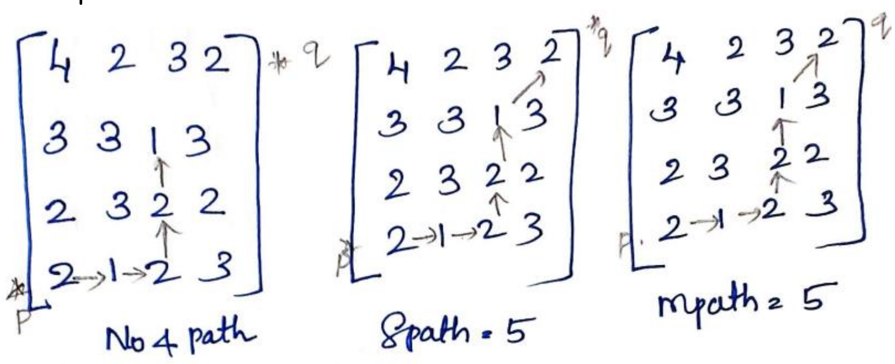
a) 4-adjacency: Two pixels p and q with values from V are 4-adjacent if q is in the set N4(p).

b) 8-adjacency: Two pixels p and q with values from V are 8-adjacent if q is in the set N8(p).

c) m-adjacency(mixed adjacency): Two pixels p and q with values from V are m-adjacent if

1. q is in N4(p), or
2. 2) q is in ND(p) and the set N4(p)∩N4(q) has no pixels whose values are from V.

Example:



Only horizontal & Vertical path is allowed.

(b) Using any one application, explain the steps involved in image processing. (7)

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D

Fundamental Image Processing Steps

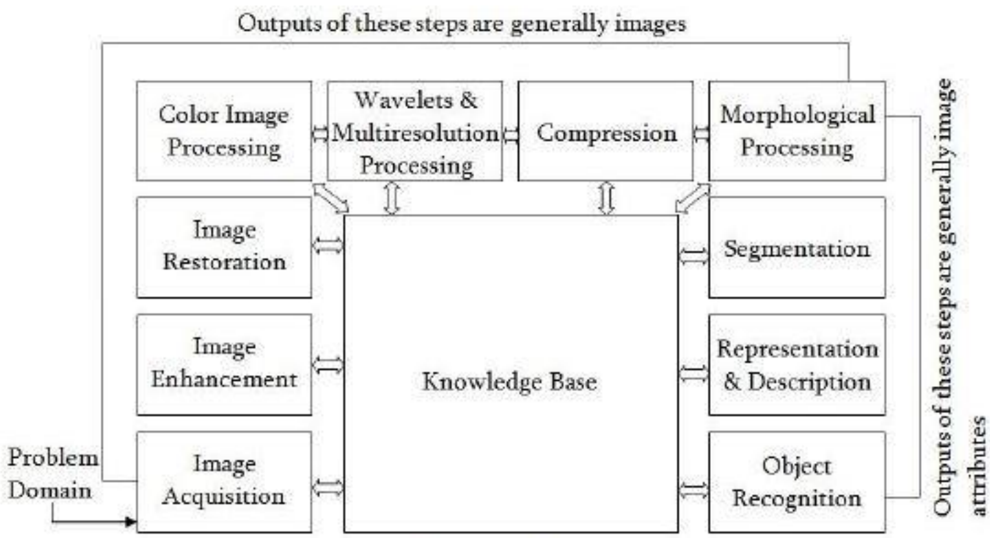


Figure 1

- **Image Acquisition**
- Image acquisition is the first step in image processing. This step is also known as preprocessing in image processing. It involves retrieving the image from a source, usually a hardware-based source.
- **Image Enhancement**
- Image enhancement is the process of bringing out and highlighting certain features of interest in an image that has been obscured. This can involve changing the brightness, contrast, etc.
- **Image Restoration**
- Image restoration is the process of improving the appearance of an image. However, unlike image enhancement, image restoration is done using certain mathematical or probabilistic models.
- **Color Image Processing**
- Color image processing includes a number of color modeling

techniques in a digital domain. This step has gained prominence due to the significant use of digital images over the internet.

- **Wavelets and Multiresolution Processing**
- Wavelets are used to represent images in various degrees of resolution. The images are subdivided into wavelets or smaller regions for data compression and for pyramidal representation.
- **Compression**
- Compression is a process used to reduce the storage required to save an image or the bandwidth required to transmit it. This is done particularly when the image is for use on the Internet.
- **Morphological Processing**
- Morphological processing is a set of processing operations for morphing images based on their shapes.
- **Segmentation**
- Segmentation is one of the most difficult steps of image processing. It involves partitioning an image into its constituent parts or objects.
- **Representation and Description**
- After an image is segmented into regions in the segmentation process, each region is represented and described in a form suitable for further computer processing. Representation deals with the image's characteristics and regional properties. Description deals with extracting quantitative information that helps differentiate one class of objects from the other.
- **Recognition**
- Recognition assigns a label to an object based on its description.
- **Knowledge Base:**

Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information.

The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

19. (a) A 5x5 image patch is shown below. Compute the value of the marked pixel ifit is smoothened by a 3x3 average filter and median filter. **(4)**

$$f(m,n)=\begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

Average filter:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{bmatrix} \xrightarrow[\text{filter}]{\text{mean}} \begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & \boxed{6} & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{bmatrix}$$

Value of 2 after applying mean filter is $\boxed{6}$

Median filter

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{bmatrix}$$

6, 7, 8, 3, 2, 1, 7, 6, 5

→ sort

1, 2, 3, 5, 6, 6, 7, 7, 8

Median = 6

Replace 2 with 6

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & \boxed{6} & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{bmatrix}$$

(b) Define Image segmentation and describe in detail method of edge and regionbased segmentation technique. **(10)**

- A method of extracting and representing information from an image is to group pixels together into regions of similarity.
- This process is commonly called as **segmentation**.

Region Splitting & Merging

The basic idea of region splitting is to break the image into a set of disjoint regions which are coherent within themselves:

1. Initially take the image as a whole to be the area of interest.
2. Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.

a) If **TRUE** then the area of interest corresponds to a region in the image.
b) If **FALSE** split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.
c) This process continues until no further splitting occurs.

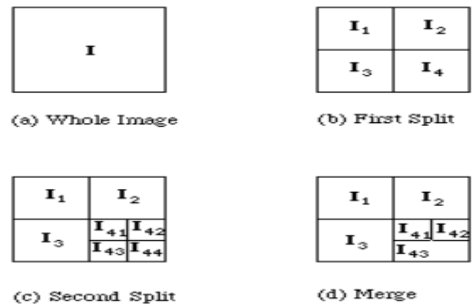
In the worst case this happens when the areas are just one pixel in size.

This is a ***divide and conquer*** or ***top down*** method.

- If only a splitting schedule is used then the final segmentation would probably contain many neighboring regions that have identical or similar properties.
- Thus, a ***merging*** process is used after each split which compares adjacent regions and merges them if necessary.
- Algorithms of this nature are called ***split and merge*** algorithms.

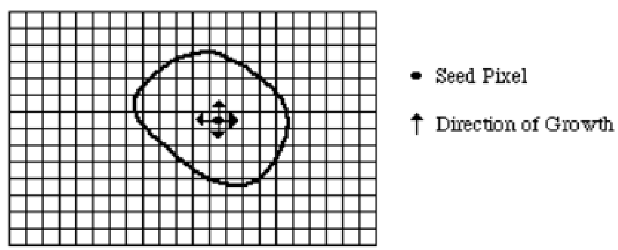
Example:

- Let I denote the whole image shown in Fig(a).
- Not all the pixels in I are similar so the region is split as in Fig (b).
- Assume that all pixels within regions I1, I2 & I3 are similar but those in I4 are not.
- Therefore I4 is split next as in Fig(c).
- Now assume that all pixels within each region are similar with respect to that region, and that after comparing the split regions, regions I43 & I44 are found to be identical.
- These are thus merged together as in Fig (d).

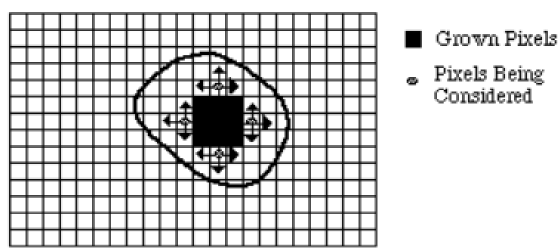


Region growing

- **Region growing** approach is the opposite of the split and merge approach
- An initial set of small areas are iteratively merged according to similarity constraints.
- Start by choosing an arbitrary ***seed pixel*** and compare it with neighbouring pixels
- Region is ***grown*** from the seed pixel by adding in neighbouring pixels that are similar, increasing the size of the region.
- When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again.
- This whole process is continued until all pixels belong to some region.
- A ***bottom up*** method.



(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

Egde Based Method

Steps:

1. Smoothing:
   - Suppress as much noise as possible, without destroying the true edges.
2. Enhancement:
   - Apply a filter to enhance the quality of the edges in the image (sharpening).
3. Detection:
   - Determine which edge pixels should be discarded as noise and which should be retained (usually, thresholding provides the criterion used for detection).
4. Localization:
   - Determine the exact location of an edge

OR

20. (a) Distinguish between smoothing and sharpening filters in terms of **(10)**
Functionality
Types
Applications
Mask Coefficients

		Smoothing Filter	Sharpening Filter
1	**Functionality**	➢ Low pass Filter ➢ Useful for reducing noise and removing small details. ➢ The elements of the mask must be positive. ➢ Mask elements sum to 1 assuming normalized weights (i.e., divide each weight by the sum of weights).	➢ High Pass Filter ➢ Useful for emphasizing fine ➢ details. ➢ The elements of the mask contain both **positive** and **negative** weights. ➢ Mask elements sum to 0.
2	**Types**	• Averaging (linear) • Gaussian (linear) • Median filtering (non-linear)	• Unsharp masking • High Boost filtering • Gradient (1^{st} derivative) • Laplacian (2^{nd} derivative)
3	**Applications**	➢ These filters can effectively reduce noise	➢ Sharpening filters are used to enhance the edges of objects and adjust the contrast and the shade characteristics. ➢ In combination with threshold they can be used as edge detectors. ➢ Sharpening or high-pass filters let high frequencies pass and reduce the lower frequencies and are extremely sensitive to shut noise.
4	**Mask Coefficients**	7 × 7 Gaussian mask $\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix}$	Laplacian of Gaussian Filters The Laplacian of Gaussian filter (LoG) is a combination of a Laplacian and Gaussian filter where its characteristic is determined by the σparameter and the kernel size as shown in the mathematical expression of the kernel: $LoG(i,j) = \frac{1}{\pi\sigma^4}\left(1 - \frac{i^2+j^2}{2\sigma^2}\right)e^{-(i^2+j^2)/2\sigma^2}$

Describe how an image is segmented using split and merge technique inassociation with the region adjacency graph.

(b) The basic idea of region splitting is to break the image into a set of disjoint regions which are coherent within themselves:

3. Initially take the image as a whole to be the area of interest.
4. Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.

d) If **TRUE** then the area of interest corresponds to a region in the image.

e) If **FALSE** split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.

f) This process continues until no further splitting occurs.

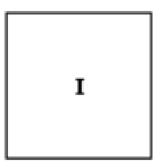
In the worst case this happens when the areas are just one pixel in size.

This is a ***divide and conquer*** or ***top down*** method.

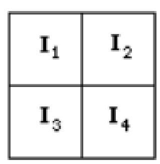
- If only a splitting schedule is used then the final segmentation would probably contain many neighboring regions that have identical or similar properties.
- Thus, a ***merging*** process is used after each split which compares adjacent regions and merges them if necessary.
- Algorithms of this nature are called ***split and merge*** algorithms.

Example:

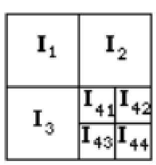
- Let I denote the whole image shown in Fig(a).
- Not all the pixels in I are similar so the region is split as in Fig (b).
- Assume that all pixels within regions I1, I2 & I3 are similar but those in I4 are not.
- Therefore I4 is split next as in Fig(c).
- Now assume that all pixels within each region are similar with respect to that region, and that after comparing the split regions, regions I43 & I44 are found to be identical.
- These are thus merged together as in Fig (d).



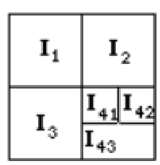
(a) Whole Image



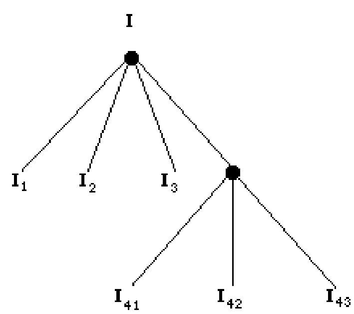
(b) First Split



(c) Second Split



(d) Merge



****