"Computer Graphics"

Segment - 1

Computer Graphics: It's a branch of Computer science that deal with the theory and technology for Computerized image synthesis.

Application of CG:
(i) Computer Aided Design (CAD)
(ii) Computer simulation
(iii) Graphics Design
(iv) Computational Physics
(v) Scientific Visualization
(vi) Special effect for cinemas.

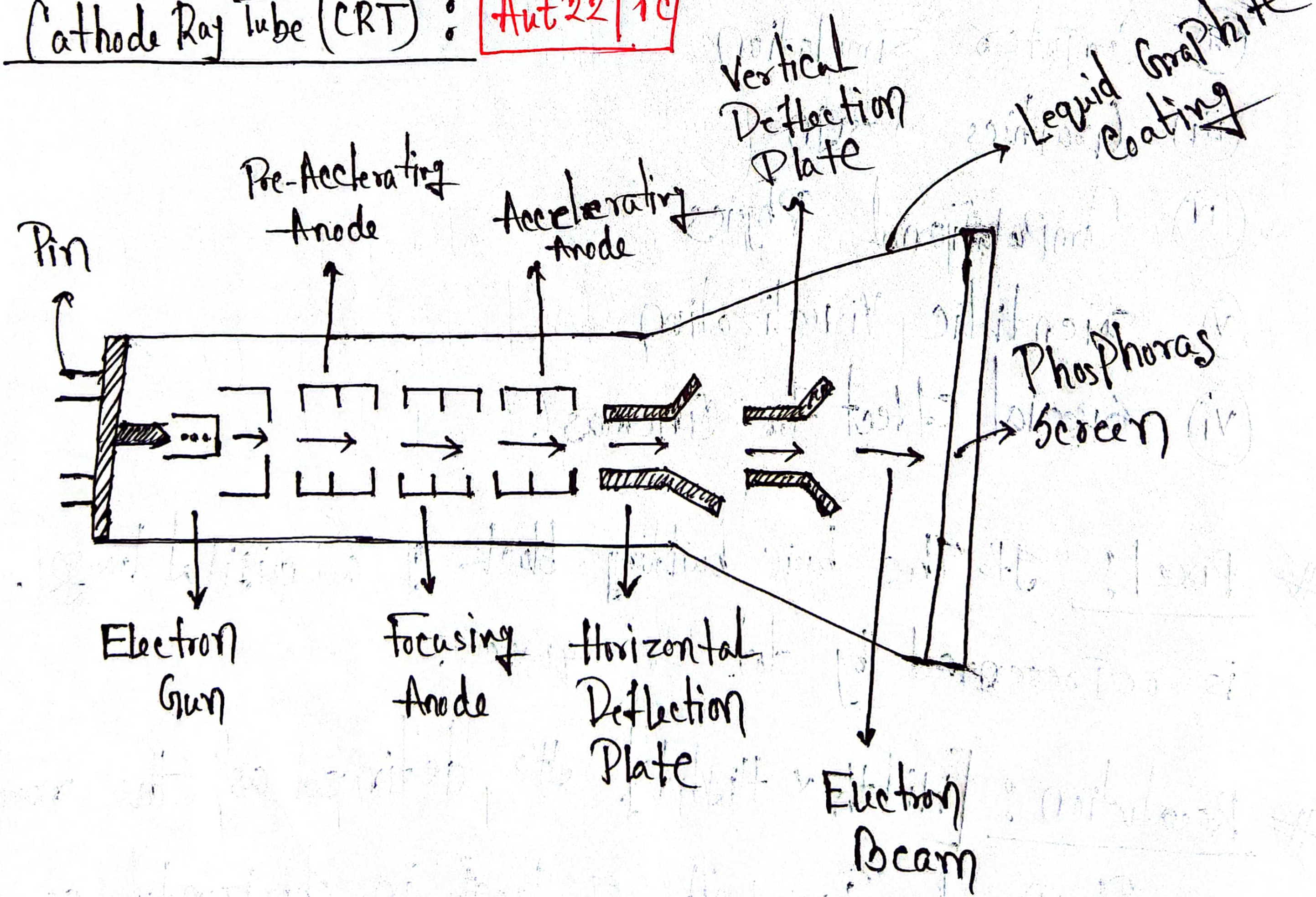
Pixel: It's the basic building block of a digital Image. Pixel is represented by dot or square.

Resolution: [Width x Hight]. It's defined as the number of ~~Pi~~ Pixel Per unit of length in horizontal as well as vertical direction.

Aspect Ration : [Width : Height]. The aspect ratio of an image describes the proportional relationship between it's width and it's height. Aspect ratio maintains a balance between the appearance of an image on the screen.

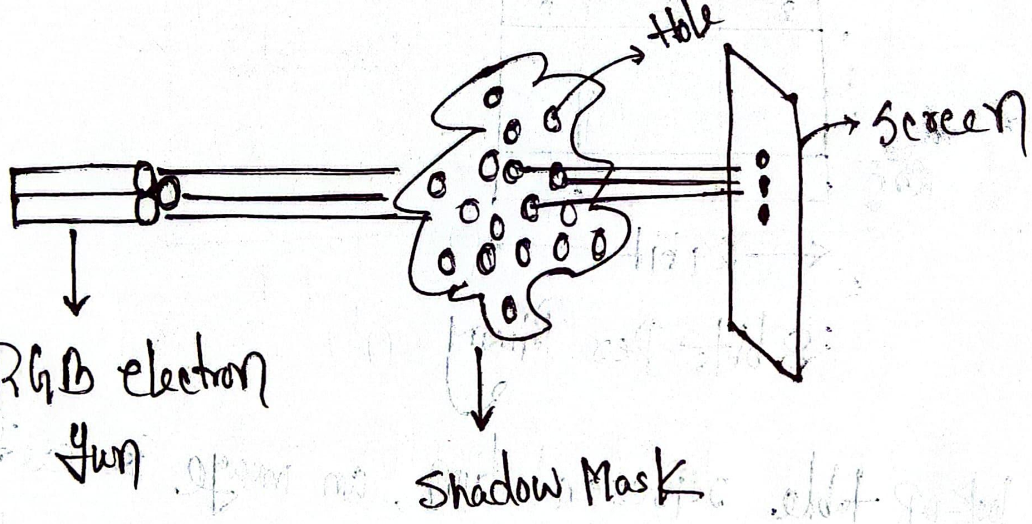
MegaPixel : Resolution / 1 million

Cathode Ray Tube (CRT) : Aut'22 | 1c



Shadow Mask Method :

- Color Display
- 3 Phosphor Color → RGB
- 3 electron gun with shadow mask grid
- Activate dot Triangle.
- Can Produce 2^{24} types of Colors



CMY (Cyan, Magenta, Yellow) :

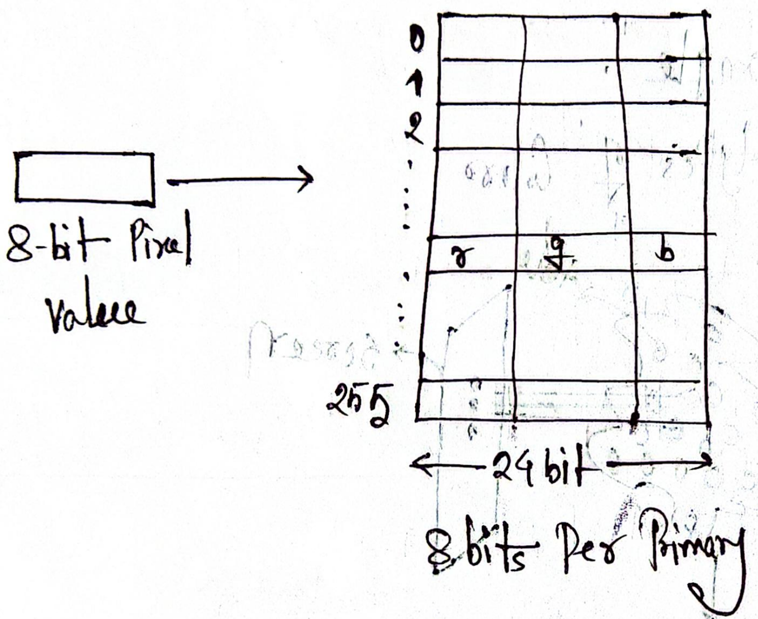
→ Defines Colors using substractive Process.

→ Matches the working Principle with Printer.

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

Lookup Table :

It's a method of Compressing an image that enables 8 bit Per Pixel to look almost as good as 24 bits Per Pixel.



Using the lookup table representation, an image is defined not only by its Pixel values but also by the Color values in the Corresponding lookup table.

Direct Coding :

R	G	B	Color
0	0	0	black
1	1	1	White
1	0	0	red
0	1	1	Cyan
1	1	0	Yellow

Image File : $\boxed{SP'22 \mid 1d}$

Format/version identification Image width and height in Pixels Image type Image data format Compression type etc.	Header
Color map (if any) Pixel values	Image Data

Segment - 2

DDA Line Drawing Algo:

Step 1: $m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$

Step 2: $\Delta x = \frac{\Delta y}{m} = \Delta y \cdot \frac{x_2 - x}{y_2 - y_1} = x_2 - x_1$

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

Step 3: Case 1: $|\Delta x| \geq |\Delta y|$, Assign, $\Delta x = 1$

$$x_{i+1} = x_i + \Delta x = x_i + 1$$

$$y_{i+1} = y_i + \Delta y = y_i + m \cdot \Delta x = y_i + m$$

Case 2: $|\Delta x| < |\Delta y|$, Assign, $\Delta y = 1$

$$x_{i+1} = x_i + \Delta x = x_i + \frac{\Delta y}{m} = x_i + \frac{1}{m}$$

$$y_{i+1} = y_i + \Delta y = y_i + 1$$

~~Problem:~~ Exercise :

$A(1,1)$ $B(4,3)$

$$m = \frac{3-1}{4-1} = \frac{2}{3} = .67$$

$\Delta x = 3$

$\Delta y = 2$

As $|\Delta x| \geq |\Delta y|$ so, $\Delta x = 1$

$x_{i+1} = x_i + 1 = 1+1 = 2$

$y_{i+1} = y_i + \Delta y = y_i + m.\Delta x$

$= 1 + .67 \times 1$

$= 1.67$

x_i	y_i	x_{i+1}	y_{i+1}
1	1	2	1.67
2	1.67	3	2.34
3	2.34	4	3.01

$\Rightarrow (1,1)\ (2,1.67), (3, 2.34)\ (4,3)$

After round off $\Rightarrow (1,1)\ (2,2)\ (3,2)\ (4,3)$ Ans

Scan Conversion : It's used to convert geometric primitives of object space into a set of pixels of image space.

DisAdv. of DDA Algo :

(i) Floating Point
(ii) Round off

} Need to .

Line Drawing (Bresenham's Algo):

Step 1 : $\Delta x = x_2 - x_1$

$\Delta y = y_2 - y_1$

Step 2 : $P_k = 2\Delta y - \Delta x$

Step 3 : Case $(P<0)$

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i$

$P_{k+1} = P_k + 2\Delta y$

Case $(P \geq 0)$

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i + 1$

$P_{k+1} = P_k + 2\Delta y - 2\Delta x$

Repeat step 3 until find (x_i, y_i)

$\boxed{SP'22|2b}$ #

$A(9,18)$ & $B(14,22)$

Step 1 : $\Delta x = 14 - 9 = 5$

$\Delta y = 22 - 18 = 4$

Step 2 : $P_k = 2\Delta y - \Delta x = 2 \times 4 - 5 = 8 - 5 = 3$

Step 3 : As $P \geq 0$,

$x_{i+1} = x_i + 1 = 9 + 1 = 10$

$y_{i+1} = y_i + 1 = 18 + 1 = 19$

$P_{i+1} = P_i + 2\Delta y - 2\Delta x = 3 - 2 = 1$

P_i	P_{i+1}	x_{i+1}	y_{i+1}
3	1	10	19
1	-1	11	20
-1	7	12	20
7	5	13	21
5	3	14	22

$$\boxed{\begin{aligned} 2\Delta y - 2\Delta x &= -2 \\ 2\Delta y &= 8 \end{aligned}}$$

Ans : (9,18) (10,19) (11,20) (12,20) (13,21) (14,22)

DisAdv. : The output line isn't smooth. Can overcome using midPoint Algorithm.

Circle Drawing Algo. (Bresenham's) :

Step 1 : $x_i = 0$, $y_i = R$

Step 2 : $P_i = 3 - 2R$

Step 3 : Case $(P_i < 0)$

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i$

$P_{i+1} = P_i + 4 * x_{i+1} + 6$

Case $(P_i \geq 0)$

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i - 1$

$P_{i+1} = P_i + 4(x_{i+1} - y_{i+1}) + 10$

Step ~~Case~~ 4 : (x_0, y_0) [if Center isn't (0,0)]

Center

$$x_{plot} = x_c + x_0$$

$$y_{plot} = y_c + y_0$$

Step 5 : If $(x_{plot} \geq y_{plot})$ then stop

Step 6 : Generate other Point using 8-way symmetry Property.

Aut'22 | 2C

Radius = 10

Center = (100, 100)

Soln :

$x_i = 0$

$y_i = 10$ } Step 1

Step 2 : $P_i = 3 - 2R = 3 - 20 = -17$

Step 3 : Case $(P_i < 0)$

$$x_{i+1} = x_i + 1 = 1$$

$$y_{i+1} = y_i = 10$$

$$P_{i+1} = P_i + 4 \times x_{i+1} + 6 = -17 + 4 \times 1 + 6 = -7$$

Step 4 : $(x_0, y_0) = (100, 100)$:

$x_{plot} = x_c + x_0 = 1 + 100 = 101$

$y_{plot} = y_c + y_0 = 10 + 100 = 110$

P_i	P_{i+1}	(x_{i+1}, y_{i+1})	(x_{plot}, y_{plot})
	-7	1, 10	101, 110
-7	7	2, 10	102, 110
7	-7	3, 9	103, 109
-7	15	4, 9	104, 109
15	13	5, 8	105, 108
13	19	6, 7	106, 107
19		7, 6 x	

Quadrant 1

Octant 1	Octan 2
100, 110	107, 106
101, 110	108, 105
102, 110	109, 104
103, 109	109, 103
104, 109	110, 102
105, 108	110, 101
106, 107	110, 100

[And find the All 4 Quadrant Point - - -]

Circle drawing (Mid Point Algo) : SP22 | 2a

Step 1 : $x_i = 0$

$y_i = R$

Step 2 : $P_i = 1 - R$

Step 3 :

Case ($P_i < 0$)

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i$

$P_{i+1} = P_i + 2 * x_{i+1} + 1$

Case ($P_i \geq 0$)

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i - 1$

$P_{i+1} = P_i + 2(x_{i+1} - y_{i+1}) + 1$

Step 4 : $(x_0, y_0) \rightarrow$ Center [if Center $\neq (0,0)$]

$x_{plot} = x_c + x_0$

$y_{plot} = y_c + y_0$

Step 5 : Continue until ($x_{plot} \geq y_{plot}$)

Step 6 : Thus find all 8 Quadrant.

Arbitrarily Centered Circle :

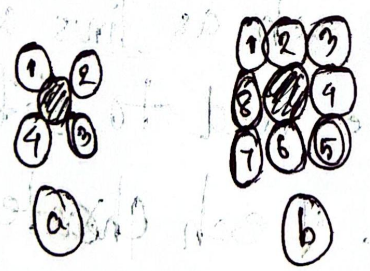
(x_0, y_0) Given Center

New Center is $\begin{bmatrix} x' = x_c + x_0 \\ y' = y_c + y_0 \end{bmatrix}$

Region Filling Algo : $\boxed{SP22|2d}$

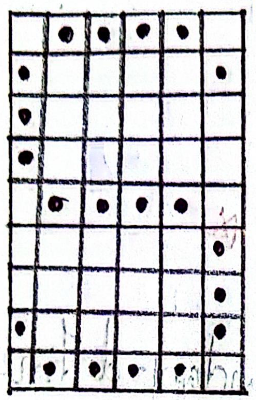
(i) Boundary fill-Algorithm / Boundary-defined region
(ii) Flood fill -Algorithm / Inner defined region

4-Connected Vs 8-Connected :



Scan-Converting a Character : SP23 | 2C

① Bitmap Font :



The size of bitmap font depend on the image resulation.

② Outline Font :

In this method, graphical Primitive such as lines and arcs are used to define the Outline of each character.

Anti-Aliasing : Its a technique that is used to reduce the aliasing effects or distortion. SP'22 | 2C

3 Anti-aliasing technique :

SP'22	2 C
Aut'21	2b

(i) Increase the resulation.

(ii) Area Sampling : It's a Pre-filtering technique in which we super-impose a Pixel grid Pattern onto the Continuous object definition.

(iii) Super Sampling : In this approch, we divide each Pixel into sub-Pixel and check the Position of each sub-Pixel in relation to the object to be scan Converted.

Segment - 3

Transformation : It's a Process of modifying and re-Positioning the existing graphics/object.

Technique :

(i) Translation
(ii) Scaling
(iii) Rotation
(iv) Reflection
(v) Shear

Translation $\longrightarrow$ Position Change

Scaling $\longrightarrow$ Size Change

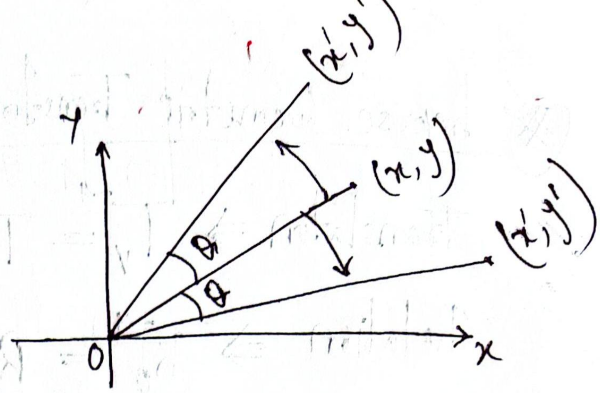
Shearing $\longrightarrow$ Shape Change

Translation $\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} x_t \\ y_t \end{bmatrix}}_{\text{Given}}$

Scaling $\Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} * \begin{bmatrix} x_t \\ y_t \end{bmatrix}$

Rotation :

For Anti-Clockwise :



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

For Clockwise :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

Shearing : (Change shape)

Given

In x-axis :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & x_s \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

In Y-axis :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ y_s & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

1 → 4 Types of Transformation

Inverse Geometric Transformation : *

Translation ⇒ $T_V^{-1} = T_{-V}$

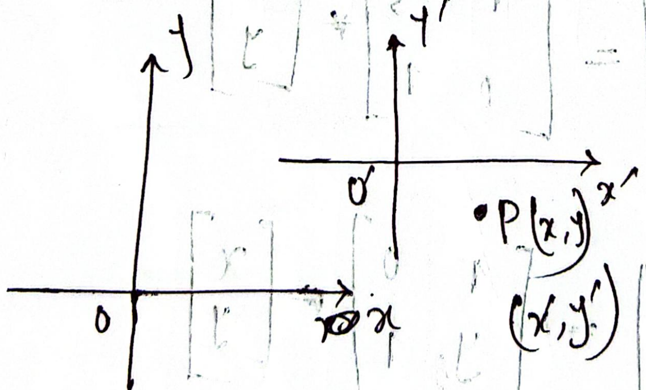
Rotation ⇒ $R_\theta^{-1} = R_{-\theta}$

Scaling ⇒ $S_{s_x,s_y}^{-1} = S_{1/s_x,\, 1/s_y}$

Mirroring ⇒ $M_x^{-1} = M_x$ and $M_y^{-1} = M_y$

2

Coordinate Transformation : Tranformate an object ~~folloing~~ ~~followi~~ following given Condition as well as Change the Coordinates.



3

Composite Transformation : Two or more tranformation technique applied on an object at a time.

4

Instance Tranformation : Join two or more object with each other on a Perticular Coordinate is called Instance tranformation. SP'23 | 3b *

4.4

$$\boxed{R_{\theta,0} = T_v \cdot R_\theta \cdot T_{-v}}$$

$$= \begin{pmatrix} 1 & 0 & h \\ 0 & 1 & K \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -h \\ 0 & 1 & -K \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation θ, about (h,k) Point.

4.7 Scaling respect to $P(h,K)$ → fixed Point

$$S = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & K \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \boxed{T_v \cdot S_{a,b} \cdot T_{-v}}$$

FAHIM REZA | C211046

4.8//

$$\text{Triangle}(A,B,C) = \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

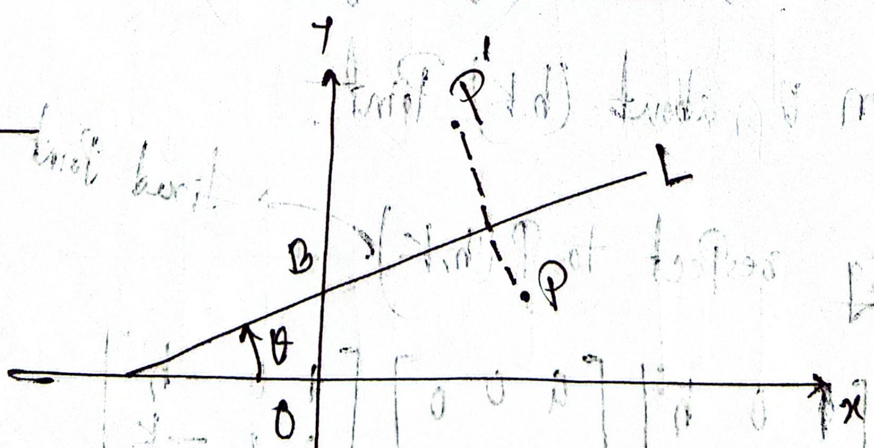
Scaling with fixed Point $C(5,2)$

$$S = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S.(ABC) = S * \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

* 4.9// Describe the transformation M_L which reflects an object about a line L. SP 23 | 3C

Sol^n:



(1) Translate the intersection Point B to the origin

(2) Rotate by $-\theta°$ so that line L align with x-axis.

(3) Mirror reflect about x-axis.

(4) Rotate back to $\theta°$

(5) Translate B back.

Steps of Mirror Reflection

SP22	3(b)
Aut21	3b

$$\boxed{M_L = T_V \cdot R_\theta \cdot M_x \cdot R_{-\theta} \cdot T_{-V}}$$

Aut 22 | 2a — Compute the sz of the image that would measure 1.6 inches by 1.2 inches at 240 Pixel/inches.

Soln: Sz in Pixel = Sz in Inches × Pixel Per Inches

Weidth in Pixel = 1.6×240 = 384 Pixel

~~Hig~~ Height in Pixel = 1.2×240 = 288 Pixel.

$\boxed{\text{Aut 22} \mid \text{3a}}$ Procedure to fill Polygon using Flood Fill algo.

Sol^n: Flood Fill (x, y, Fill_Color, Old_Color)

```
{
  if (getPixel(x,y) = old_Color)
  {
    setPixel (x, y, Fill_Color)
      Fill (x+1, y, Fill_Color)
      Fill (x-1, y,      "     )
      Fill (x, y+1,      "     )
      Fill (x, y-1,      "     )
  }
}
```

Or, Aliasing effect of scan-Conversion.

Sol^n:

(i) Staircase effect

(ii) Unequal "

(iii) Picket-Fence "

SP'22 | 1 b

(i) Direct Coding method Possible Colors each Pixel for 3 bit

$\Rightarrow$ $2^3 = 8$ Possible value

$8 \times 8 \times 8 = 512$ different Color for each Pixel

(ii) How many entries in lookup table for 3 bit?

$\Rightarrow$ Total entries = 8 (red) × 8 (Green) × 8 (Blue)

= 512 different entries.

FAHIM REZA | C211046
fahim.bdctg51@gmail.com