

## Line Drawing Algorithm

Actually, pixels get selected and turned on.

\* The more pixels  $\approx$  The smoother the picture gets.

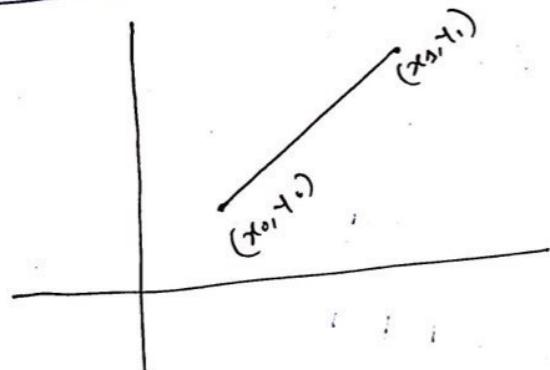
\* pixel can't be fractional.

1. Simple Solution

2. DDA

3. Midpoint Line Algorithm.

Simple Solution



General Eq:  $y = mx + c$

Slope      Intercept  
 $m$                    $c$

$$m = \frac{y_1 - y_0}{x_1 - x_0} \quad \begin{cases} c = y - mx \\ c = y_0 - mx_0 \end{cases}$$

[We can use any point  
to get the value of  
 $c$ ]

# Using SS, find out the intermediate pixels from

$(2, 2)$  to  $(7, 5)$  to convert linear interpolation to pixel A  
 $(x_0, y_0)$  to  $(x_1, y_1)$  using soft vertical quantization = 2 bits per pixel

Soln:

$$m = \frac{5-2}{7-2} = \frac{3}{5} = 0.6$$
 to convert to 2 bits per pixel

$$c = y_0 - mx_0$$

$$= 2 - 0.6 \times 2$$

$$= 2 - 1.2 = 0.8$$

$$y = 0.6x + 0.8$$

$$\Rightarrow y(3) = (0.6 \times 3) + 0.8 = 2.6 \approx 3$$

$$y(4) = (0.6 \times 4) + 0.8 = 3.2 \approx 3$$

$$y(5) = (0.6 \times 5) + 0.8 = 3.8 \approx 4$$

$$y(6) = (0.6 \times 6) + 0.8 = 4.4 \approx 4$$

To figure out the pixels from  $(2, 2)$  to  $(7, 5)$ , we will input value of  $x$  [in this case 3 to 6] to get the value of  $y$

\* As we can't get fractional values for pixels, we will round off  $y$  to the nearest integers.

Final Answer:

- Pixels → (3,3) To standardize with row wise & column wise path  
(4,3)  
(5,4)  
(6,4) (8,8,8) (8,8,8) (2,8,8) (3,8,8)

Drawbacks of Simple Sol<sup>n</sup>:

- Multiplication is costly operation → Time Consuming  
Round off → Expensive.

## Digital Differential Analyzer

\*\* Overcomes the drawbacks of Simple Solution.

(2,2) (3,2.6) (4,(3,2)) (5,3.8) (6,4.4)

$\Delta x \rightarrow 1$

$y \rightarrow 0.6$

$$x_{\text{new}} = x_{\text{prev}} + 1$$

$$y_{\text{new}} = y_{\text{prev}} + m(\text{slope})$$

→ for

$$-1 \leq m \leq 1$$

\*\* if the slope is really high,

$$x_{\text{new}} = x_{\text{prev}} + \frac{1}{m}$$

$$y_{\text{new}} = y_{\text{prev}} + 1$$

→ for all the other cases.

Using DDA, find out the intermediate pixels from (2,2) to (5,9)

Soln:

$$\text{Slope, } m = \frac{5-2}{5-2} = \frac{3}{3} = 1.67$$

$m > 1.67$ , falls under other cases.

$$x_{\text{new}} = x_{\text{prev}} + \frac{1}{m}$$

$$y_{\text{new}} = y_{\text{prev}} + 1$$

$x (+\frac{1}{m})$	$y (+1)$	Pixels
2	2	(2, 2)
2.6 $\approx$ 3	3	(3, 3)
3.2 $\approx$ 3	4	(3, 4)
3.8 $\approx$ 4	5	(4, 5)
4.4 $\approx$ 4	6	(4, 6)

$$(\text{new}) x = 6$$

(D)

Scanned with CamScanner

Scanned with CamScanner

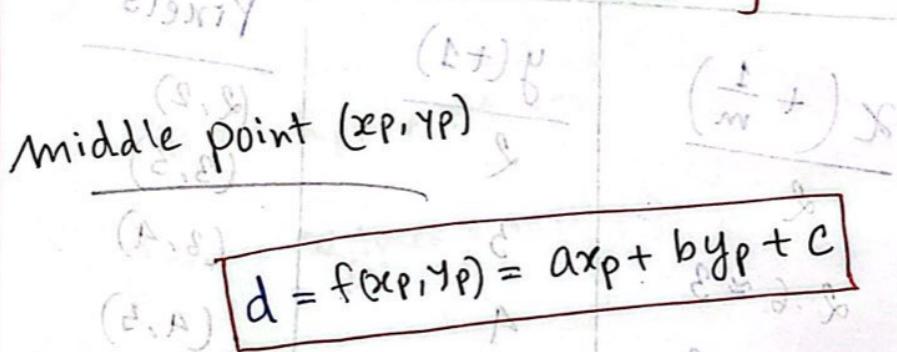
## Midpoint Line Algorithm

Different forms of Line eqn:

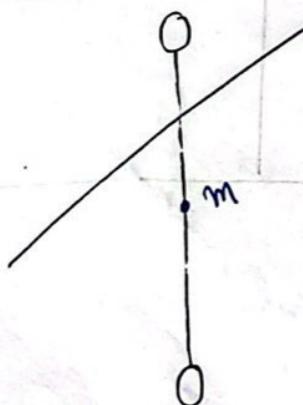
$$1) y = mx + c$$

$$2) x/a + y/b = 1 \quad P_2 \leq m$$

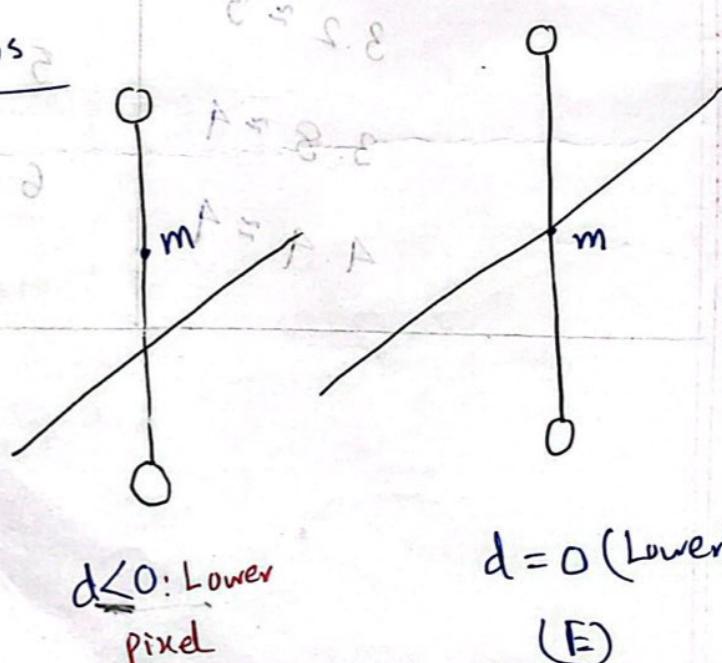
$$3) ax + by + c = 0 \quad ] \text{Convenient one for this algorithm.}$$



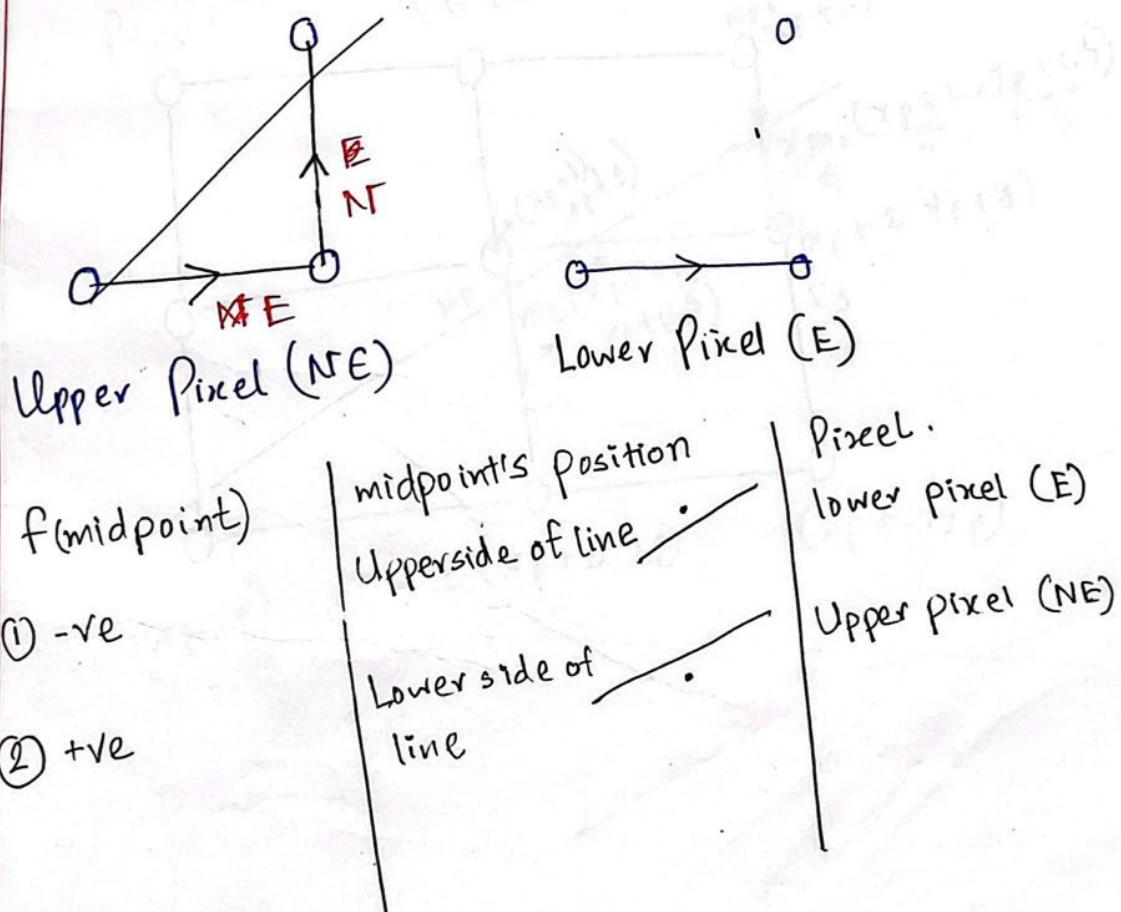
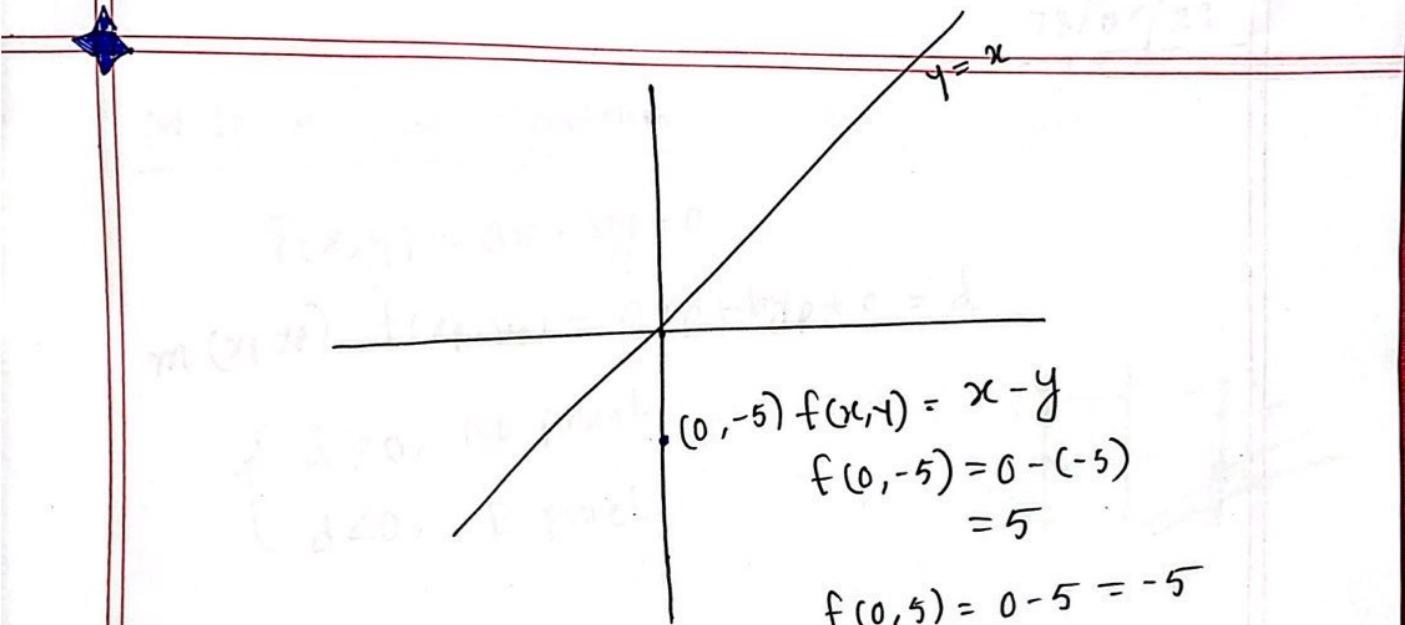
3 type of Scenarios



$d > 0$ : Upper (NE) pixel



$d = 0$  (Lower)  
(E)



Scanned with CamScanner

Scanned with CamScanner

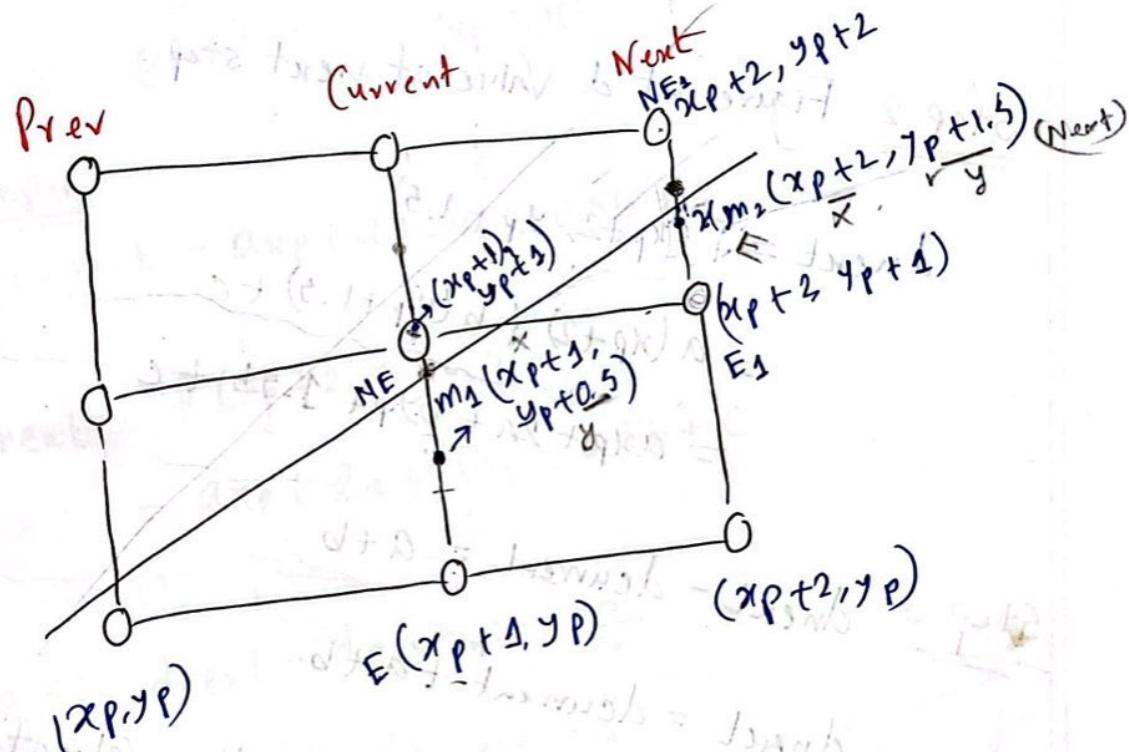
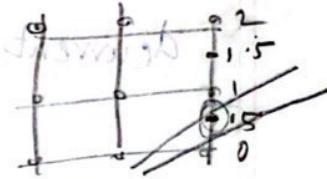
03/05/23

## Midpoint Line Algorithm

$$f(x, y) = ax + by + c$$

$$m(x_p, y_p) \quad f(x_p, y_p) = ax_p + by_p + c = d$$

$$\begin{cases} d > 0, & \text{NE pixel} \\ d < 0, & \text{E pixel.} \end{cases}$$



Scanned with CamScanner

Scanned with CamScanner

Step 1: Figure out the value of current step in grid

$$\begin{aligned} f(x_1) &= ax + by + c \quad 9+15+20 = (f, x)_1 \\ d_{\text{current}} &= f(x_p+1, y_p+0.5) \quad 9+15+9 \times 0 = (g, y)_1 \\ &= a(x_p+1) + b(y_p+0.5) + c \quad 0.5b \\ &= ax_p + a + by_p + 0.5b + c \end{aligned}$$

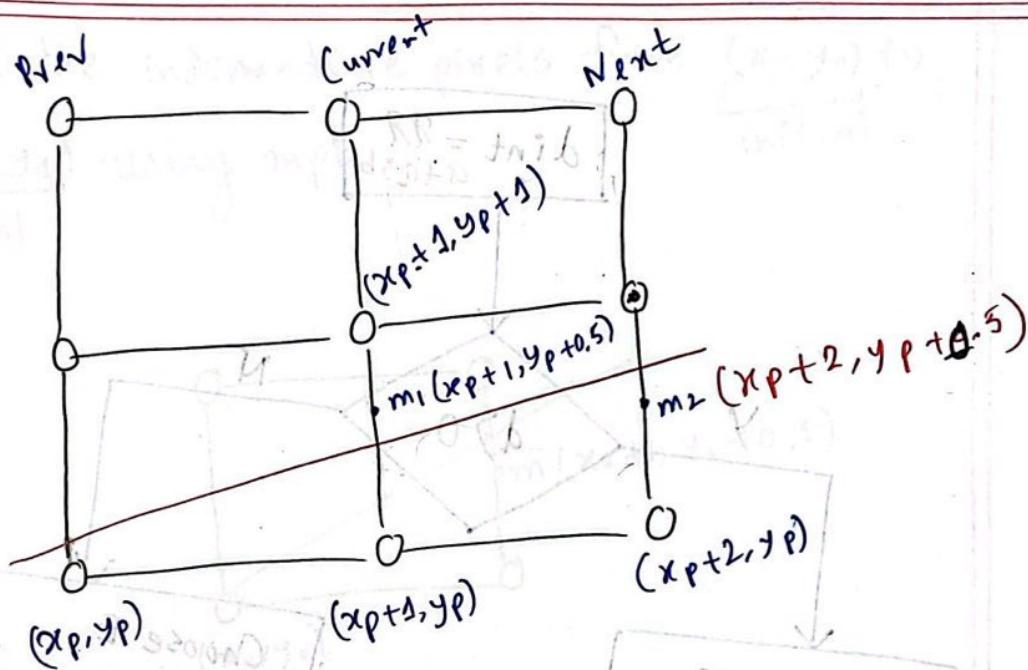
Step 2: Figure out the value of next step.

$$\begin{aligned} d_{\text{next}} &= f(x_p+2, y_p+1.5) \\ &= a(x_p+2) + b(y_p+1.5) + c \\ &= ax_p + 2a + by_p + 1.5b + c \end{aligned}$$

Step 3:  $d_{\text{next}} - d_{\text{current}} = a+b$

$$d_{\text{next}} = d_{\text{current}} + a+b$$

\* if my current state is NE, then the value of  
next  $d$  will be  $d = d + a+b$



Step 4:

$$d_{current} = ax_p + a + by_p + 0.5b + c$$

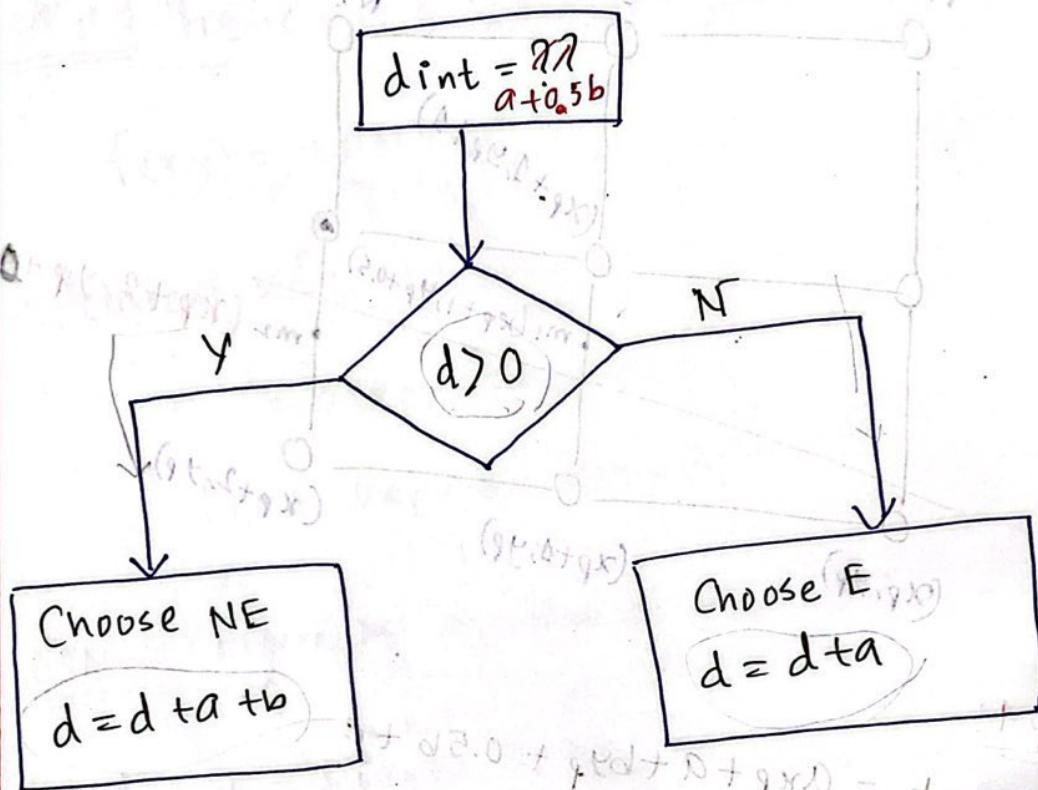
$$d_{next} = f(x_p+2, y_p+0.5)$$

$$= ax_p + 2a + by_p + 0.5b + c$$

Step 5:  $d_{next} - d_{current} = a$

$$d_{next} = d_{current} + a$$

if my current step is E, then the value of next d  
will be  $d = d + a$



$$(e_{x1} + g_{x1} + r_{x1})/7 = f_{x1}$$

$$g_{x2} + r_{x2} + e_{x2}/7 + n_2 + g_{n2} =$$

$$D = f_{x1} - f_{x2}$$

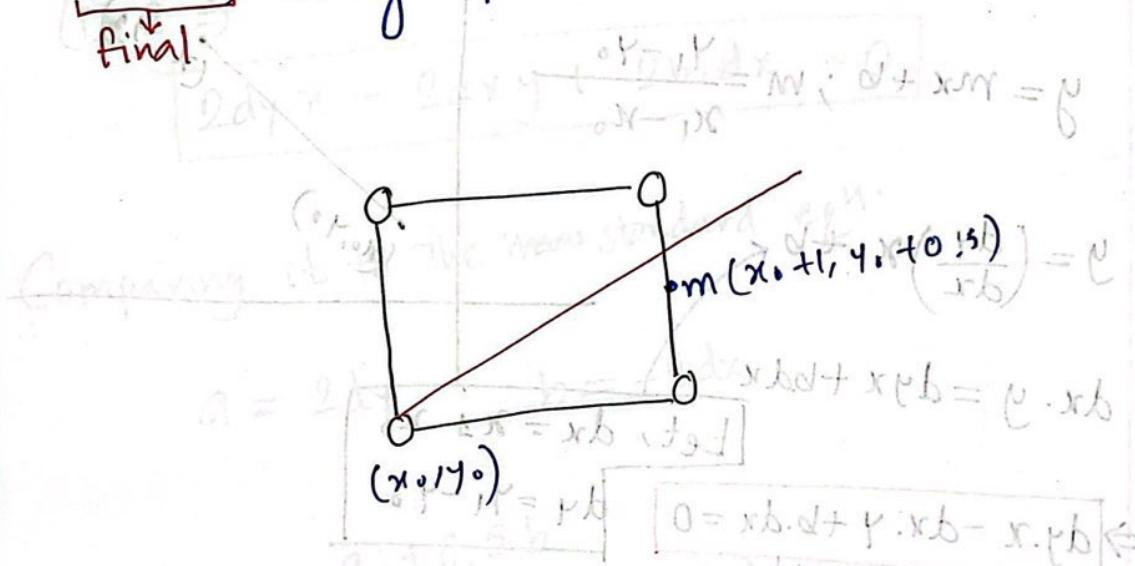
$D = f_{x1} - f_{x2}$

Scanned with CamScanner

Scanned with CamScanner

# Find the intermediate pixels from  $(x_0, y_0)$  to  $(x_1, y_1)$  using m<sub>l</sub>.

initial  
final



$$f(x, y) = ax + by + c$$

$f(x_0, y_0) = ax_0 + by_0 + c = 0$  [as the point  $(x_0, y_0)$  is on the line, so that verifies the eqn, so the value is 0]

$$\begin{aligned} d_{int} &= f(x_0+1, y_0+0.5) \\ &= a(x_0+1) + b(y_0+0.5) + c \\ &= ax_0 + a + by_0 + 0.5b + c \\ &= \boxed{ax_0 + by_0 + c} + a + 0.5b \\ &= 0 + a + 0.5b \end{aligned}$$

# Find out value of  $a, b$  from  $(x_0, y_0)$  and  $(x_1, y_1)$

$$y = mx + b ; m = \frac{y_1 - y_0}{x_1 - x_0}$$

$$y = \left(\frac{dy}{dx}\right)x + b$$

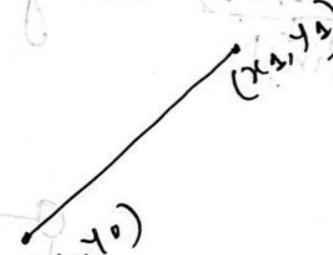
$$dx \cdot y = dyx + bd x$$

$$\text{Let, } dx = x_1 - x_0$$

$$\Rightarrow dy \cdot x - dx \cdot y + b \cdot dx = 0 \quad dy = y_1 - y_0$$

(i)

Eqn given  $(x_0, y_0)$



Comparing (i) with the standard line eq<sup>n</sup>  $ax + by + c = 0$

$$a = dy \quad b = -dx$$

$$dint = dy - 0.5 dx$$

But we can't have fraction here.

To solve the fractional problem, transform it

(i)  $\times 2 \Rightarrow$

$$2dy \cdot x + 2dx \cdot y + 2b \cdot dx = 0$$

Comparing it to the standard eqn:

$$a = 2dy$$

$$b = -2dx$$

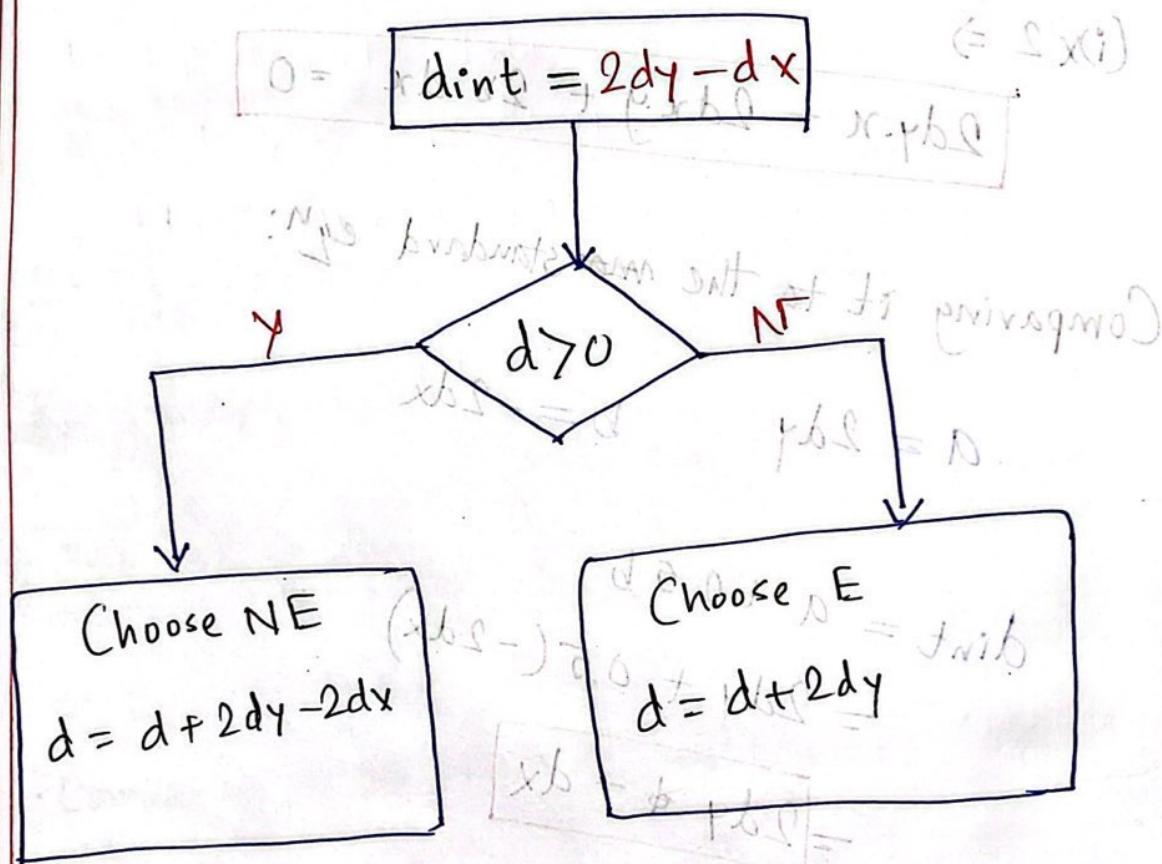
$$\begin{aligned} d\text{int} &= a + 0.5b \\ &= 2dy + 0.5(-2dx) \\ &= 2dy - dx \end{aligned}$$

$$\begin{aligned} &x^2 - y^2 + b = 0 \\ &x^2 - y^2 + b = 0 \end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

## Final Flowchart



Scanned with CamScanner

Scanned with CamScanner

# Find out 7 pixels of a line from (0,2) to (70, 52) using mpe.

Step 1

$$x_0 = 0, y_0 = 2$$

$$x_1 = 70 \quad y_1 = 52$$

Step 2

$$dy = \frac{52 - 2}{70} = 0.52$$

$$dx = 70 - 0 = 70$$

$$2dy - 2dx = (2 \times 50) - (2 \times 70) = -40$$

$$2dy = 2 \times 50 = 100 \quad \text{dint} = 2dy - dx \\ = 100 - 70 = 30$$

essential  
values.

x	y	d	E/NE	update	pixels
0	2	30	NE	$30 + (-40) = -10$	(0, 2)
1	3	-10	E	$-10 + 100 = 90$	(1, 3)
2	3	90	NE	<del><math>90 + 100 =</math></del> $90 + (-40) = 50$	(2, 3)
3	4	50			

Scanned with CamScanner

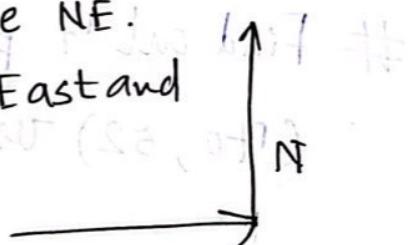
Scanned with CamScanner

\* For the first step, we have gone NE.

For NF, we go 1 step towards East and  
1 step towards North.

So next value of  $(x, y)$

would be **NE(+1, +1)**



$$Q = Q^P + 0 = Q^R$$

$$Q^R = Q^E \quad Q^E = 1, 0$$

$$Q^S = Q^E - Q^R = 1 - 1 = 0$$

$$Q^F = Q^S + Q^R = 0 + 1 = 1$$

$$Q^L = (Q^F \times 2) - (Q^R \times 2) = 2 - 2 = 0$$

$$Q^B = Q^L - Q^R = 0 - 1 = -1$$

$$Q^D = Q^B - Q^R = -1 - 1 = -2$$

Index	Output	NE	E	R	S
(0, 0)	$Q^L = (Q^P) + Q^R$	1	0	1	0
(1, 0)	$Q^E = Q^L + Q^R$	1	1	0	1
(1, 1)	$Q^R = (Q^E) + Q^S$ $Q^S = (Q^L) + Q^R$	1	0	1	1

$$y_0 - mx_0 = b$$

Example:

Suppose a line segment starts at co-ordinate  $(A, B)$  and ends at  $(A+C, B+10)$

a) Find the slope ( $m$ ) and intercept ( $b$ ) of this line

b) Simulate the DDA line drawing algorithm to find the first three intermediary pixels of this line after  $(A, B)$

Sol<sup>n</sup>:  $A = 21, B = 30, C = 15, D = 80$

$$\text{Slope} = \frac{A+C-A}{B+10-B} = \frac{15}{10} = 1.5$$

$$\text{intercept } b = \frac{30}{15} - mx_0$$

$$y = mx + b$$
$$\Rightarrow b = y - mx$$

$$b = 1 + 780 - 1.5(21)$$

$$b = -1.5$$

$$m = 1.5 \therefore x_{\text{new}} = x_{\text{prev}} + \frac{1}{m}$$

$$y_{\text{new}} = y_{\text{prev}} + 1$$

$$d = \alpha m - \beta$$

$x_{\text{prev}} + \frac{1}{m} = x_{\text{new}}$	$y_{\text{prev}} + 1 = y_{\text{new}}$	(pixels)
21	30	(21, 30)
$(21 + \frac{1}{1.5}) = 21.67 \approx 22$	$30 + 1 = 31$	(22, 31) ✓
$22 + \frac{1}{1.5} = 22.67 \approx 23$	$31 + 1 = 32$	(23, 32) ✓
$23 + \frac{1}{1.5} = 23.67 \approx 24$	$32 + 1 = 33$	(24, 33) ✓

$x_{\text{new}} = x_{\text{prev}} + \frac{1}{m}$	$x_{\text{round}}$	$y_{\text{new}} = y_{\text{prev}} + 1$	(pixels)
21	21	30	(21, 30)
$21 + \frac{1}{1.5} = 21.67$	22	$30 + 1 = 31$	(22, 31)
$21.67 + \frac{1}{1.5} = 22.34$	22	$31 + 1 = 32$	(22, 32)
$22.34 + \frac{1}{1.5} = 23.01$	23	$32 + 1 = 33$	(23, 33)

Scanned with CamScanner

Scanned with CamScanner

## DDA Example

(-1, 7) (2, 2)

$$m = \frac{2 - 7}{2 - (-1)} = \frac{-5}{3} = -1.67 \quad (0.7)(0.8) \underline{(0)}$$

$$\frac{1}{m} = \frac{1}{-1.67} = -\frac{3}{5} = 0.6$$

error = d

$$(x_{\text{prev}} + \frac{1}{m}) - 0.6 = x_{\text{new}} - p = d$$

$$y_{\text{new}} = y_{\text{prev}} + 1$$

$x_{\text{new}}$	$x_{\text{round}}$	$y_{\text{new}}$	(pixels)
2	2	2 (0.2, 2)	(2, 2)
$2 + (0.6) = 1.4$	1	3	(1, 3)
$1 + (-0.6) = 0.4$	0.1	4	(0, 4)

Slide 25 of 32

sign masx ADD

Exercise

(l, l) (E, E)

(a)  $(20, 35)(9, 50)$

$$m = \frac{50 - 35}{9 - 20} = -\frac{15}{11}$$

$$b = y_0 - mx_0 = 20 - \left(-\frac{15}{11} \times 35\right)$$
$$= 67.1727$$

$$b = y_0 - mx_0$$

$$= 35 - \left(-\frac{15}{11} \times 20\right)$$
$$= 62$$

(b)  $(-5, 50) (-5, 0)$

$$m = \frac{0 - 50}{-5 + 5} = 0$$

$$b = 50 - (-5 \times 0)$$
$$= 50$$

(c)  $(-10, 10), (48, 24)$

$$m = \frac{24 - 10}{48 + 10} = \frac{14}{58} = \frac{7}{29}$$

$$b = 10 - \frac{7}{29}(-10)$$

$$= 12.41$$

~~2~~

~~$\alpha(x_0, y_0)(x_1, y_1)$~~

$$m = \frac{50 - 5}{19 - 20} = -15$$

$$\frac{1}{m} = -\frac{1}{15}$$

$x_{new} =$ $x_{prev} + \frac{1}{m}$	$y_{new} =$ $y_{prev} + 1$	$x_{round}$	(pixels)
20	5	20	(20, 5)
$20 + (-\frac{1}{15}) =$ 19.97	$5 + 1 = 6$	20	(20, 6)
$19.97 + (-\frac{1}{15}) =$ 19.64	$6 + 1 = 7$	20	(20, 7)
$19.64 + (-\frac{1}{15}) =$ 19.41	$7 + 1 = 8$	19	(19, 8)
$19.41 + (-\frac{1}{15}) =$ 19.39	$8 + 1 = 9$	19	(19, 9)
$19.39 + (-\frac{1}{15}) =$ 19.36	$9 + 1 = 10$	19	(19, 10)

Scanned with CamScanner

Scanned with CamScanner

$$(b) (-5, 50), (-15, 0)$$

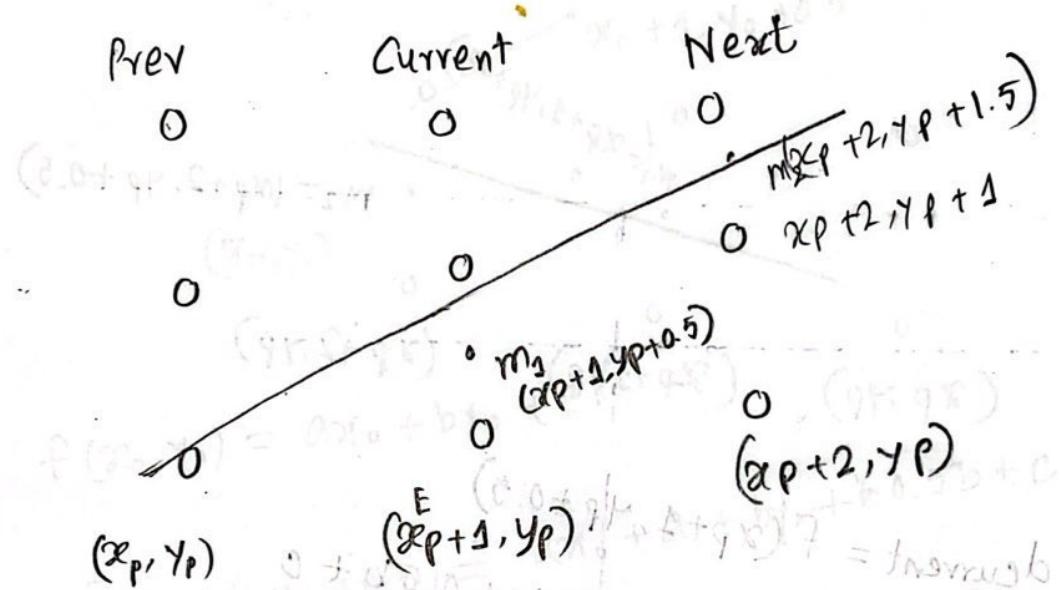
$$m = \frac{0 - 50}{-15 + 5} = \frac{-50}{-10} = 5$$

$x_{\text{new}} = x_{\text{prev}} + \frac{1}{m}$	$y_{\text{new}} = y_{\text{prev}} + 1$	$x_{\text{round}}$	(pixel)
$-5 + \frac{1}{5} = -4.8$	$50 + 1 = 51$	-5	(-5, 51)
$-4.8 + \frac{1}{5} = -4.6$	$51 + 1 = 52$	-5	(-5, 52)
$-4.6 + \frac{1}{5} = -4.4$	$52 + 1 = 53$	-4	(-4, 53)
$-4.4 + \frac{1}{5} = -4.2$	$53 + 1 = 54$	-4	(-4, 54)
$-4.2 + \frac{1}{5} = -4$	$54 + 1 = 55$	-4	(-4, 55)

Scanned with CamScanner

Scanned with CamScanner

## # Proof the Midpoint Line Algorithm



$$f(x_p, y_p) = ax_p + by_p + c$$

$$d_{\text{current}} = f(x_p + 1, y_p + 0.5)$$

$$= ax_p + a + by_p + 0.5b + c$$

$$d_{\text{next}} = f(x_p + 2, y_p + 1.5)$$

$$= ax_p + 2a + by_p + 1.5b + c$$

$$d_{\text{next}} - d_{\text{current}} = a + b$$

$\therefore$  if next step is NE, then

$$\therefore d_{\text{next}} = d_{\text{current}} + a + b$$

$$d_{\text{next}} = d_{\text{current}} + a + b$$

The diagram illustrates a straight line segment. It starts at a point labeled  $m_1 = (x_p + 1, y_p + 0.5)$  and ends at a point labeled  $m_2 = (x_p + 2, y_p + 0.5)$ . The line also passes through two other points:  $(x_p, y_p)$  at the far left and  $(x_p + 1, y_p)$  in the middle. The points are arranged horizontally along the line, with  $m_1$  above  $m_2$ .

$$\begin{aligned} \text{dCurrent} &= f(x_{p+1}, y_{p+0.5}) \\ &= ax_p + a + by_p + 0.5^b + c \end{aligned}$$

$$d_{\text{next}} = f(x_p + 2, y_p + 0.5) \\ = ax_p + 2a + by_p + 0.5b + c$$

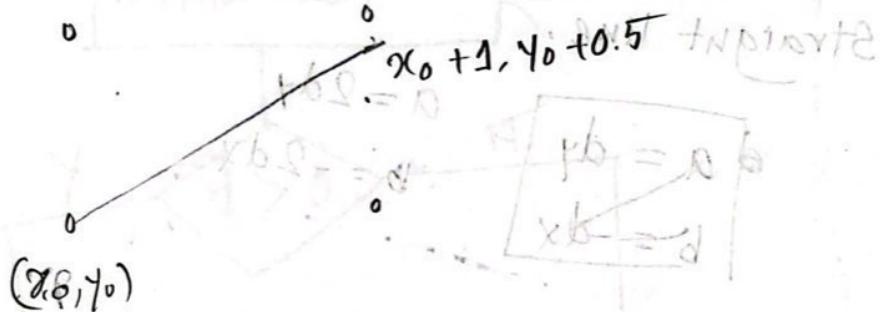
$$d_{\text{next}} - d_{\text{current}} = a$$

$d_{\text{next}} = d_{\text{current}} \oplus t_a$ .  
If the next step is E, then

If the next step is E, then

$$d = d + \alpha$$

To find the value of  $f(x_0 + 1, y_0 + 0.5)$  using finite difference



$$f(x_0, y_0) = ax_0 + by_0 + c$$

$$\begin{aligned} \therefore f(x_0 + 1, y_0 + 0.5) &= ax_0 + a + by_0 + b(0.5b) + c \\ &= ax_0 + by_0 + c + a + 0.5b. \end{aligned}$$

but at  $(x_0, y_0)$ ,  $f(x_0, y_0) = 0$ .

$$\text{initial } = f(x_0 + 1, y_0 + 0.5) = a + 0.5b.$$

$$\text{Now, } y_0 = mx + b,$$

$$m = \frac{x_1 - x_0}{y_1 - y_0}$$

$$y = \frac{dy}{dx}x + b$$

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

$$\begin{aligned} \Rightarrow y dx &= x dy + b dx \\ \Rightarrow x dy - y dx + b dx &= 0 \quad (1) \\ (1) \times 2 &= 2x dy - 2y dx + 2b dx = 0 \end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

Equating eq(i) with standard eq<sup>n</sup> of  
Straight line.

$$\begin{cases} a = dy \\ b = -dx \end{cases}$$

$$a = 2dy$$

$$b = -2dx$$

$$\boxed{d_{\text{initial}} = a + -0.5dx}$$

$$\therefore d_{\text{initial}} = 2a - b$$

$$\boxed{d_{\text{initial}} = 2a - b}$$

$$d > 0$$

Y

$$\begin{cases} \text{Choose NE} \\ d = d + a + b \end{cases}$$

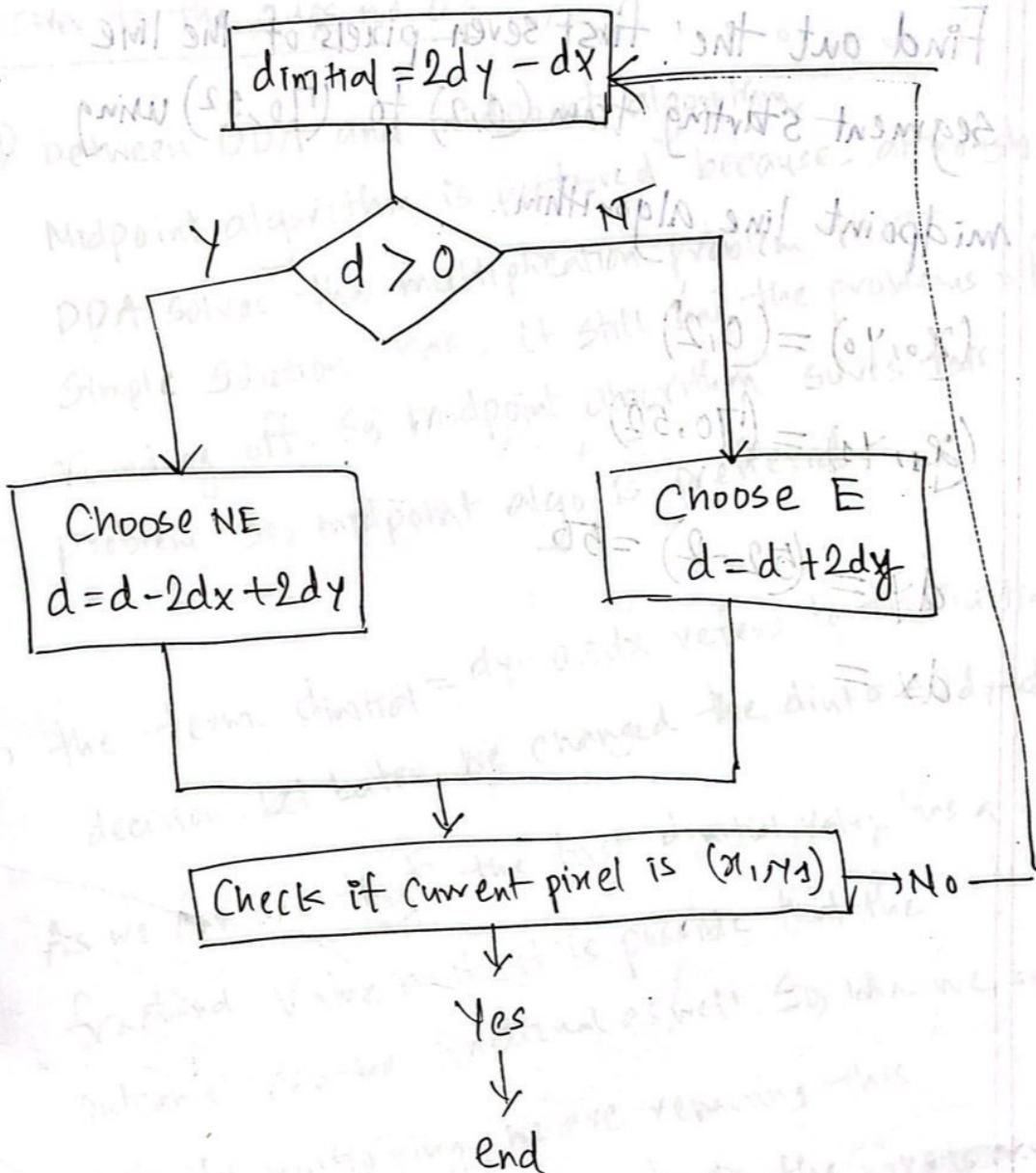
N

$$\begin{cases} \text{Choose E} \\ d = d + a \end{cases}$$

$$\boxed{\text{Check if } (x_p, y_p) = (a, b)}$$

Scanned with CamScanner

Scanned with CamScanner



Scanned with CamScanner

Scanned with CamScanner

Answer to the ques no. 01

(a) between DDA and midpoint algorithm,  
Midpoint algorithm is preferred because, although  
DDA solves the multiplication problem that  
Simple Solution was, it still has the problems of  
Rounding off. So, midpoint algorithm solves this  
problem. So, midpoint algo is preferred.

(b) the term  $d_{initial} = dy - 0.5dx$  refers to the initial  
decision. Later, we changed the  $d_{initial} = 2dy - dx$ .

As we can see that the first  $d_{initial}$  value has a  
fractional value in it, it is possible that the  
outcome can be fractional as well. So, when we are  
actually multiplying, we are removing this  
fractional part as the output or the pixels can  
not be pixels.

No, this result won't affect the output of  
the algorithm because, as we

P.T.O

figure out the value for ~~the slope of gradient~~

$$dyx - dxy + b dx = 0$$

then we multiply it with 2,

$$2dyx - 2dxy + 2dx \cdot b = 0$$

The value on the right hand side isn't changed

as well. So it gives a valid answer.

(C) (3,7) (6,13)

x	y	d initial	E/NE	d update	pixel
3	7	9	NE	15	(3,7)
4	8	15	NE	21	(4,8)
5	9	21	NE	27	(5,9)
6	10	27	NE	33	(6,10)

$$\begin{aligned} d_{\text{initial}} &= 2dy - dx \\ &= 2(y_1 - y_0) - (x_1 - x_0) \\ &= 2(13 - 7) - (6 - 3) = 12 - 3 = 9 \end{aligned}$$

$$\begin{aligned}d &= d + 2dy - 2dx \\&= 9 + 12 - 6 \\&= 21 - 6 = 15\end{aligned}$$

$$\begin{aligned}d &= d + 2dy - 2dx \\&= 9 + 2(13-7) - 2(6-3) \\&= 15\end{aligned}$$

~~$$\begin{aligned}d &= d + 2dy - 2dx \\&= 15 +\end{aligned}$$~~

$$\begin{aligned}d &= d + 2dy - 2dx \\&= 15 + 2 \times 6 - 2 \times 3\end{aligned}$$

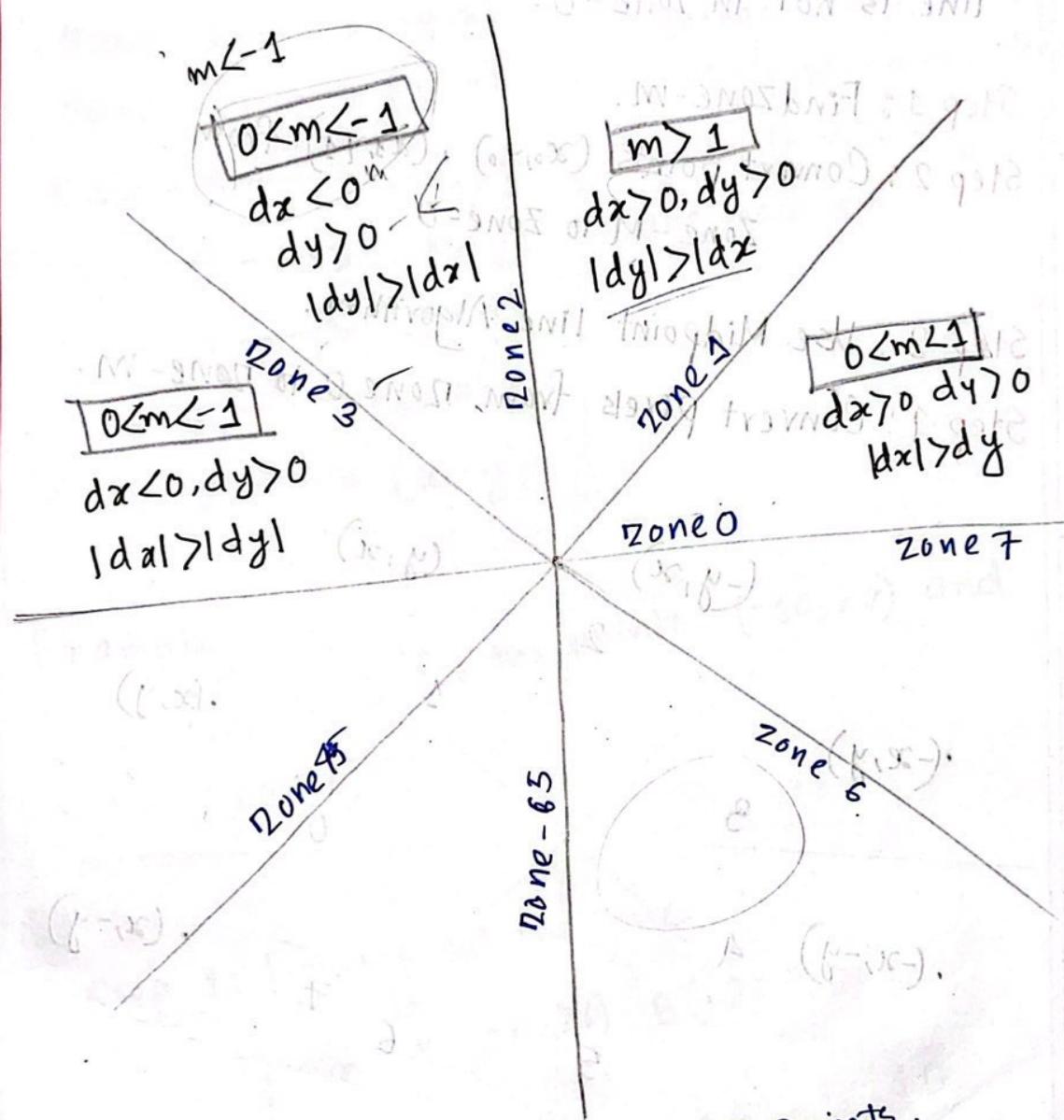
$$\begin{aligned}d &= 21 + 2 \times 6 - 2 \times 3 \\&= 27\end{aligned}$$

$$\begin{aligned}d &= 27 + 2 \times 6 - 2 \times 3 \\&= 33\end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

## Eight Way Symmetry



\* How to find out zone from given points.

A  $(x_0, y_0)$  B  $(x_1, y_1)$

$$\begin{aligned} dx &= x_1 - x_0 \\ dy &= y_1 - y_0 \end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

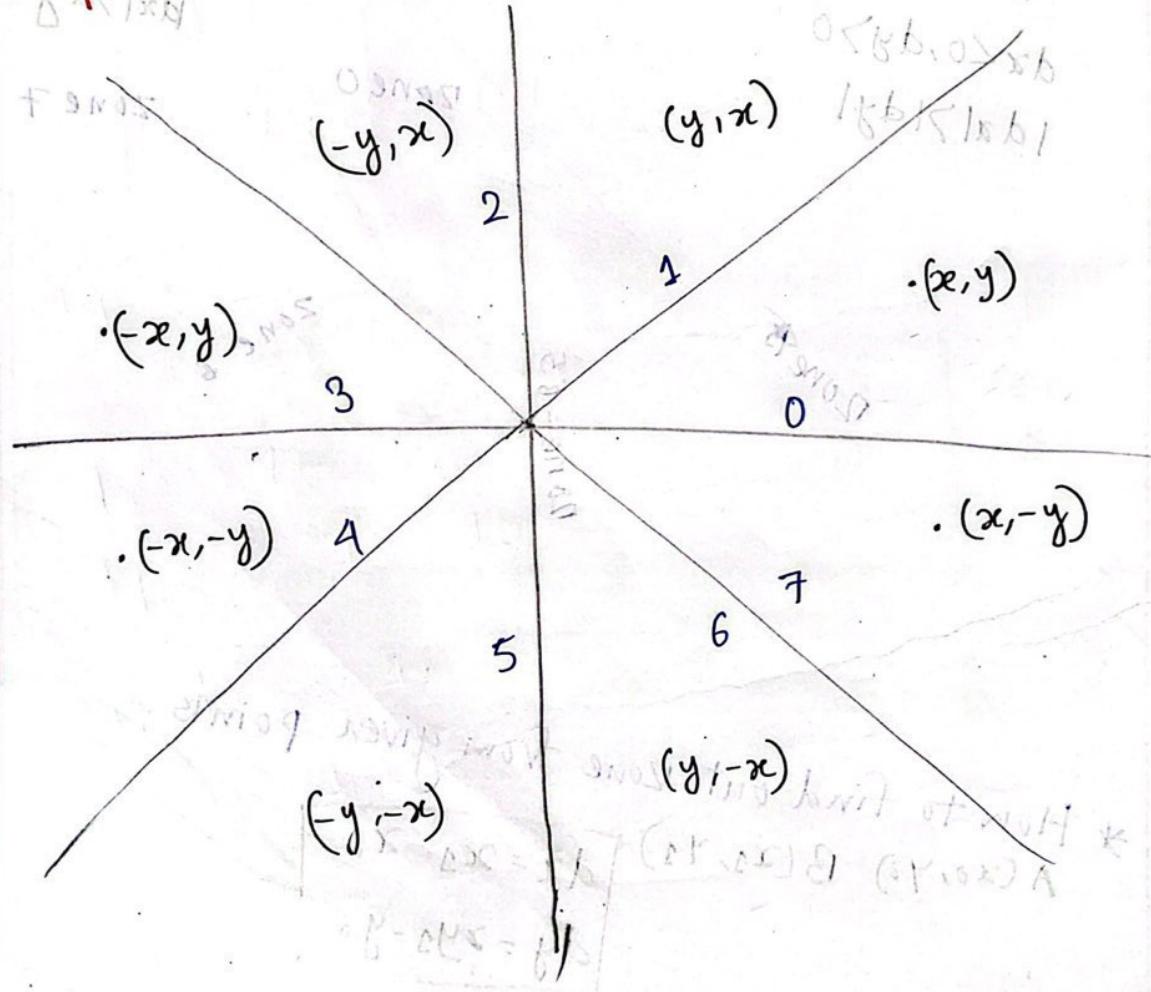
How to use Midpoint Line Algorithm when the line is not in zone-0.

Step 1: Find zone-M.

Step 2: Convert points  $(x_0, y_0)$ ,  $(x_1, y_1)$  from Zone-M to Zone-0

Step 3: Use Midpoint line Algorithm.

Step 4: Convert pixels from Zone 0 to Zone-M.



Scanned with CamScanner

Scanned with CamScanner

- Zone - 0  $\rightarrow (x, y)$
- Zone - 1  $\rightarrow (y, x)$
- Zone - 2  $\rightarrow (-y, x)$
- Zone - 3  $\rightarrow (-x, y)$
- Zone - 4  $\rightarrow (-x, -y)$
- Zone - 5  $\rightarrow (y, -x)$
- Zone - 6  $\rightarrow (x, -y)$

To change zone,  
we will just change the  
co-ordinate according to our  
zone movement.

Example  
Use MPL for the points  $(-10, 20)$  and  $(-20, 10)$

Sol<sup>n</sup>:

Step 1: Figuring out the zone.

A  $(-10, 20)$  B  $(-20, 10)$

$$dx = -20 - (-10) = -20 + 10 = -10$$

$$dy = 10 - 20 = 50$$

$dx < 0, dy > 0, |dy| > |dx|$

Zone 2

## Step 2

Convert the points from Zone-m to Zone 0

$$(x_0, y_0) = (-10, 20)$$

$$(x_1, y_1) = (-20, 70)$$

Zone - 2	to	Zone - 0
(-10, 20)	$\rightarrow$	(-20, 10)
(-y, x)		(x, y)
(-20, 70)	$\rightarrow$	(70, 20)

$$(-y, x) \equiv (x, y)$$

$$x = -y$$

$$y = -x$$

## Step 3

$$(-20, 10) \rightarrow (70, 20)$$

$$dx = -70 - (-20) = 90$$

$$dy = 20 - 10 = 10$$

$$2dy - dx = (2 \times 10) - 90 = -70$$

$$2dy = (2 \times 10) = 20$$

$$2dy - 2dx = 20 - (2 \times 90) = -160$$

$$\begin{aligned}
 d_{init} &= 2dy - dx \\
 &= (2 \times 10) - 90 \\
 &= 70
 \end{aligned}$$

x	y	d	E/NE	duupdate	Pixels
-20	10	-70	E	-50	(-20, 10)
-19	10	-50	E	-30	(-19, 10)
-18	10	-30	F	-10	(-18, 10)
-17	10	-10	E	10	(-17, 10)
-16	10	10	NE	-150	(-16, 10)

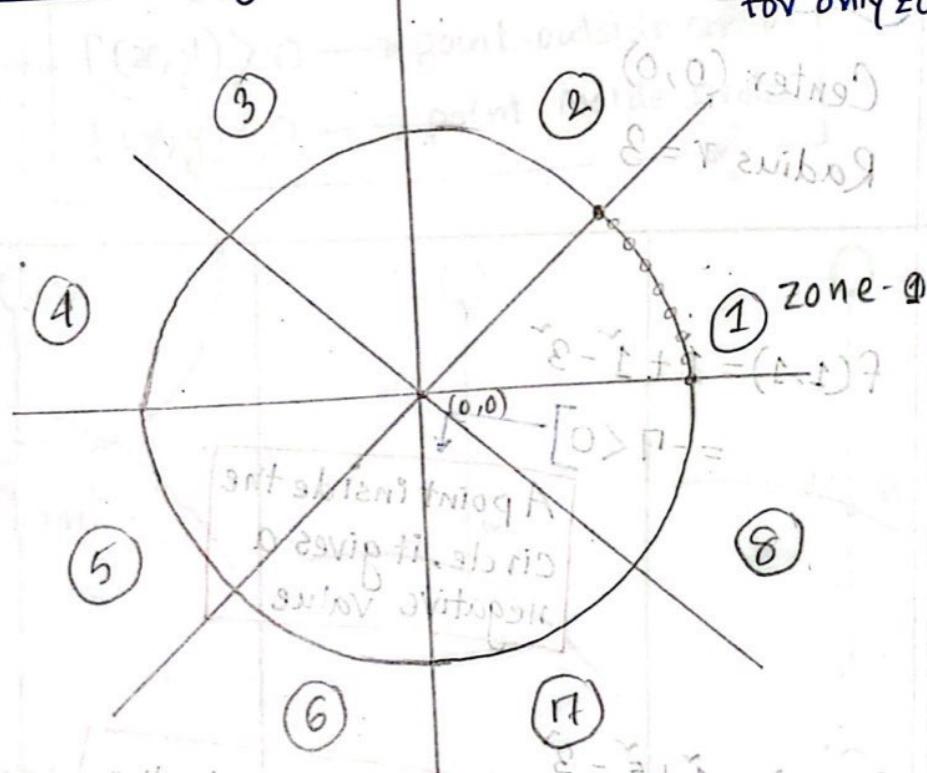
#### Step 4

Convert them back to Zone-m

$$\begin{aligned}
 (x, y) &\rightarrow (-y, x) \\
 (-20, 10) &\rightarrow (-10, -20) \\
 (-19, 10) &\rightarrow (-10, -19) \\
 (-18, 10) &\rightarrow (-10, -18) \\
 (-17, 10) &\rightarrow (-10, -17) \\
 (-16, 10) &\rightarrow (-10, -16)
 \end{aligned}$$

## Midpoint Circle Algorithm

We draw  
for only zone-1



Center  $(0, 0)$

radius  $r$

$$x^2 + y^2 - r^2 = 0$$

$$f(x, y) = x^2 + y^2 - r^2$$

Eq<sup>n</sup> for Circle.

Scanned with CamScanner

Scanned with CamScanner

Example

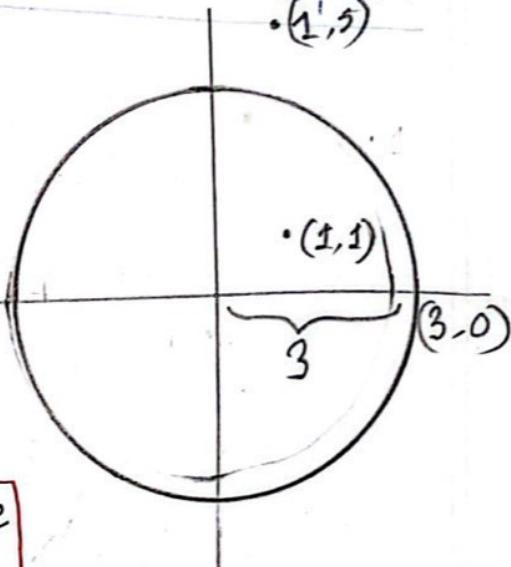
Center  $(0,0)$

Radius  $r = 3$

$$f(1,1) = \sqrt{1} + \sqrt{1} - 3^2$$

$$= -7 < 0$$

A point inside the circle, it gives a negative value



$$f(1,5) = \sqrt{1} + \sqrt{5} - 3^2$$

$$= 17 > 0$$

A point outside the circle, gives positive value

$$f(3,0) = \sqrt{3} + \sqrt{0} - 3^2$$

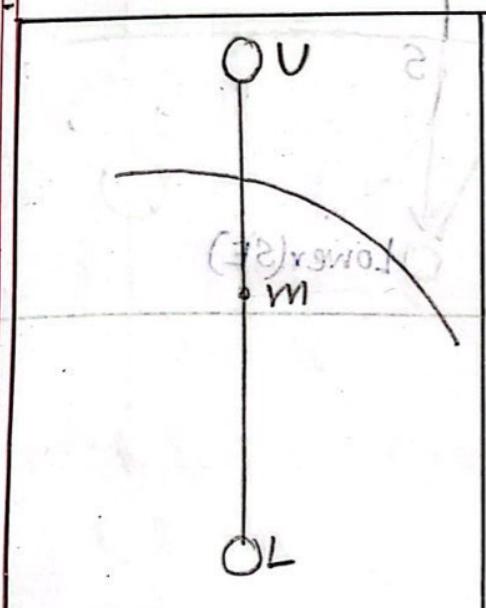
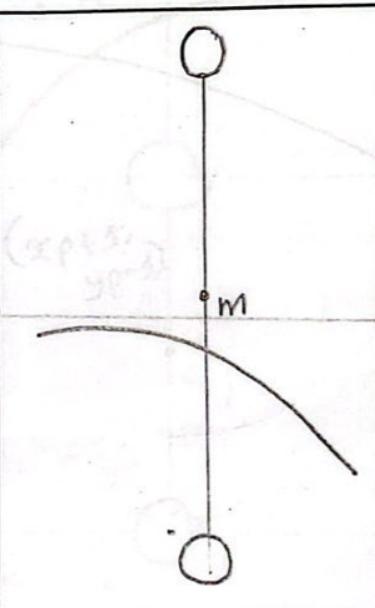
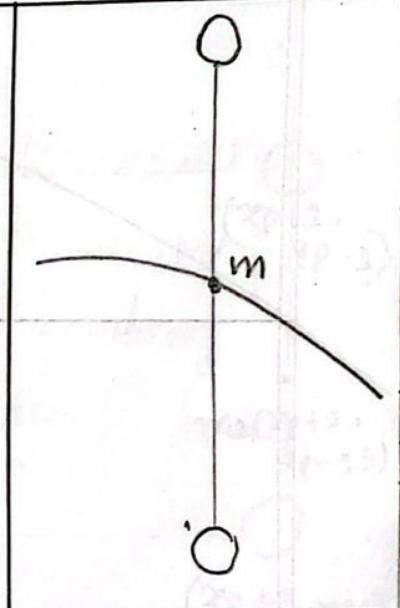
$$= 0$$

A point on the circle gives out 0 as output

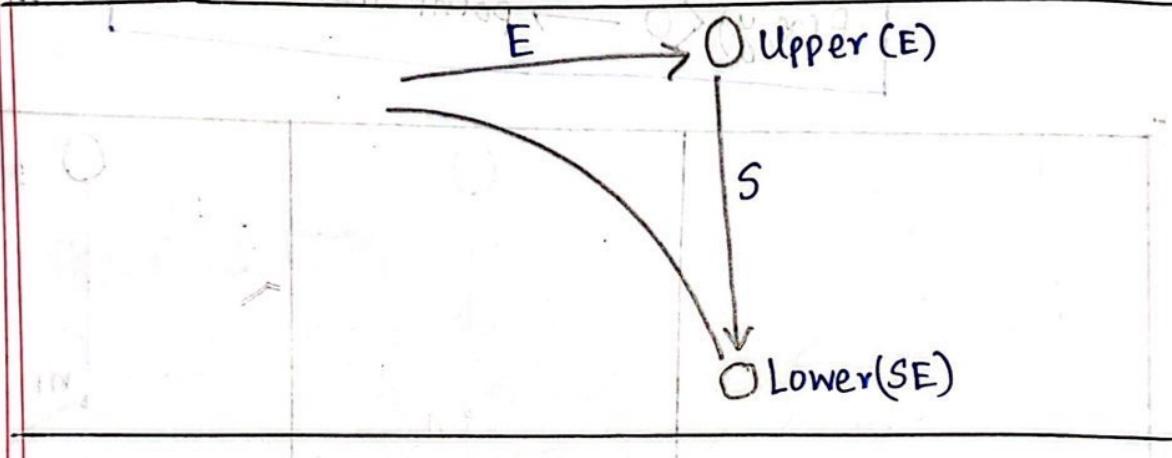
$f(x, y) = 0 \rightarrow$  point on circle

$f(x, y) > 0 \rightarrow$  point outside circle

$f(x, y) < 0 \rightarrow$  point inside circle

		
<ul style="list-style-type: none"><li>point on circle <math>f(m) &lt; 0</math></li><li>Upper Pixel</li></ul>	<ul style="list-style-type: none"><li>point outside circle <math>f(m) &gt; 0</math></li><li>Lower Pixel</li></ul>	<ul style="list-style-type: none"><li>point on surface of circle <math>f(m) = 0</math></li><li>Lower Pixel</li><li>We can choose UP as well.</li></ul>

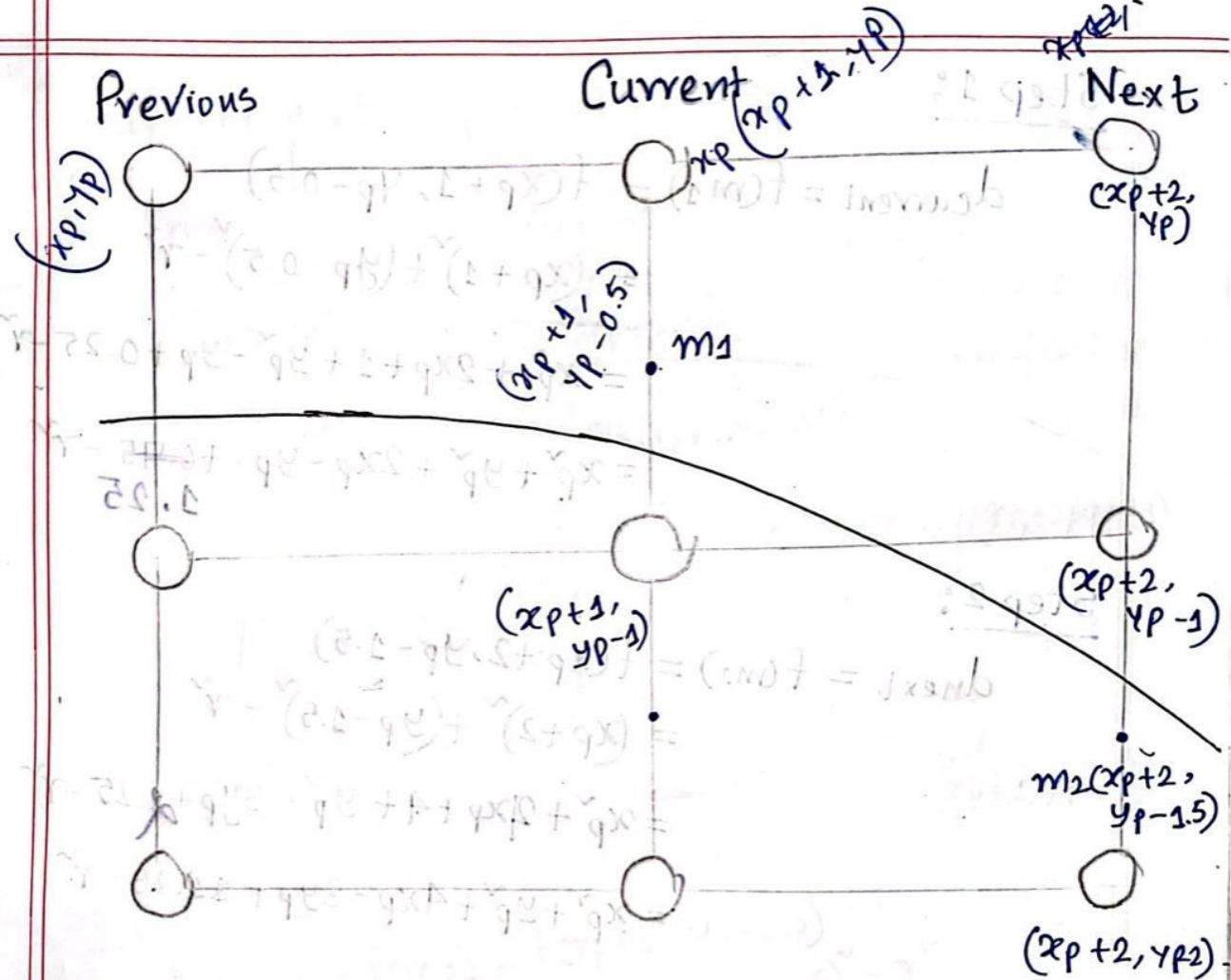
- For Upper, we will choose E
- For Lower, we will choose SE



Scanned with CamScanner

Scanned with CamScanner

Midpoint Circle  
Algo.



- If the current step pixel is SE, then what is the value of  $d$  in next step?

$$f(x, y) = x^2 + y^2 - r^2$$

P.T.O

Scanned with CamScanner

Scanned with CamScanner

### Step 1:

$$\begin{aligned}
 d_{\text{current}} &= f(m_1) = f(x_p + 1, y_p - 0.5) \\
 &= (x_p + 1)^{\tilde{v}} + (y_p - 0.5)^{\tilde{v}} - r^{\tilde{v}} \\
 &= x_p^{\tilde{v}} + 2x_p + 1 + y_p^{\tilde{v}} - y_p + 0.25 - r^{\tilde{v}} \\
 &= x_p^{\tilde{v}} + y_p^{\tilde{v}} + 2x_p - y_p + \frac{0.25 - r^{\tilde{v}}}{1.25}
 \end{aligned}$$

### Step 2:

$$\begin{aligned}
 d_{\text{next}} &= f(m_2) = f(x_p + 2, y_p - 1.5) \\
 &= (x_p + 2)^{\tilde{v}} + (y_p - 1.5)^{\tilde{v}} - r^{\tilde{v}} \\
 &= x_p^{\tilde{v}} + 2x_p + 4 + y_p^{\tilde{v}} - 3y_p + 2.25 - r^{\tilde{v}} \\
 &= x_p^{\tilde{v}} + y_p^{\tilde{v}} + 4x_p - 3y_p + \frac{2.25 - r^{\tilde{v}}}{1.25}
 \end{aligned}$$

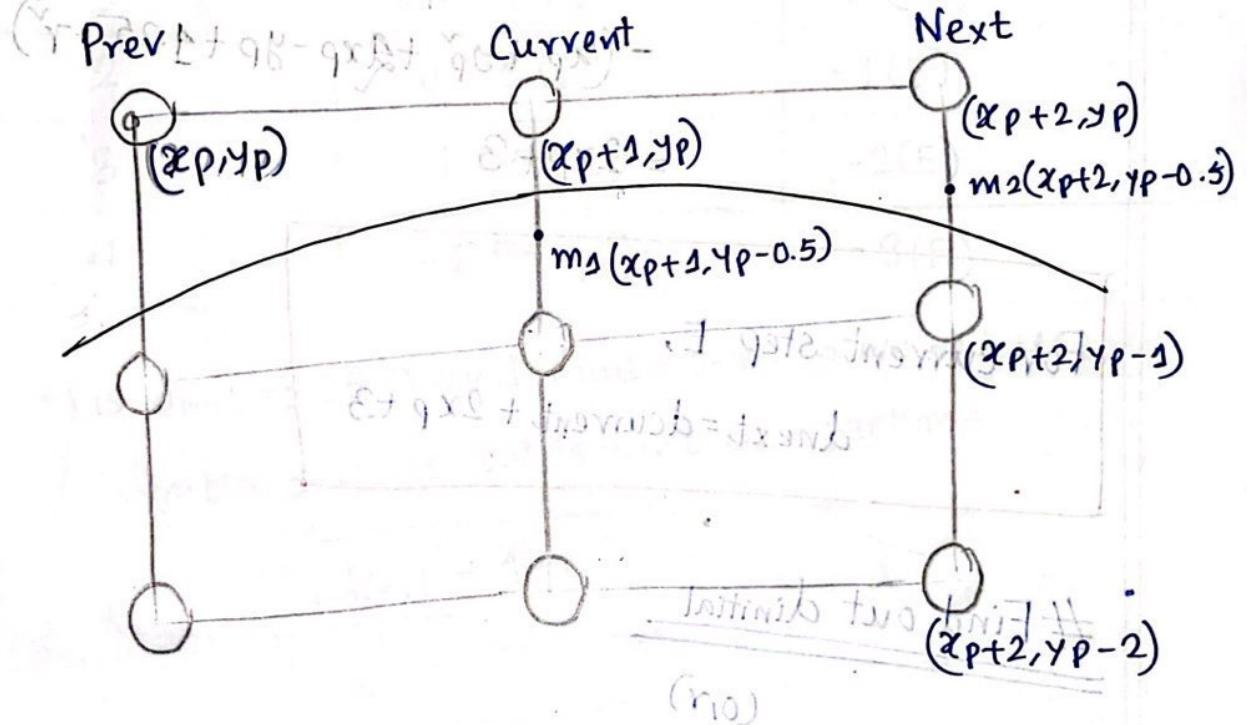
### Step 3:

$$d_{\text{next}} - d_{\text{current}} = 2x_p + 3 - 2y_p + 2$$

$$\therefore \boxed{d_{\text{update}} = 2x_p + 3 - 2y_p + 5}$$

if the next step is SE,  $d_{\text{next}} = \boxed{2x_p - 2y_p + 5}$

- If the next step pixel is E  $\Rightarrow$



Step 01:

$$\begin{aligned}
 d_{\text{current}} &= f(m_3) = f(x_p + 1, y_p - 0.5) \\
 &= (x_p + 1)^2 + (y_p - 0.5)^2 - r^2 \\
 &= x_p^2 + 2x_p + 1 + y_p^2 - 2y_p + 0.25 - r^2 \\
 &= x_p^2 + y_p^2 + 2x_p - y_p + 1.25 - r^2
 \end{aligned}$$

Step 02:

$$\begin{aligned}
 d_{\text{next}} &= f(m_2) = f(x_p + 2, y_p - 0.5) \\
 &= (x_p + 2)^2 + (y_p - 0.5)^2 - r^2 \\
 &= x_p^2 + 4x_p + 4 + y_p^2 - 2y_p + 0.25 - r^2 \\
 &= x_p^2 + y_p^2 + 4x_p - y_p + 4.25 - r^2
 \end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

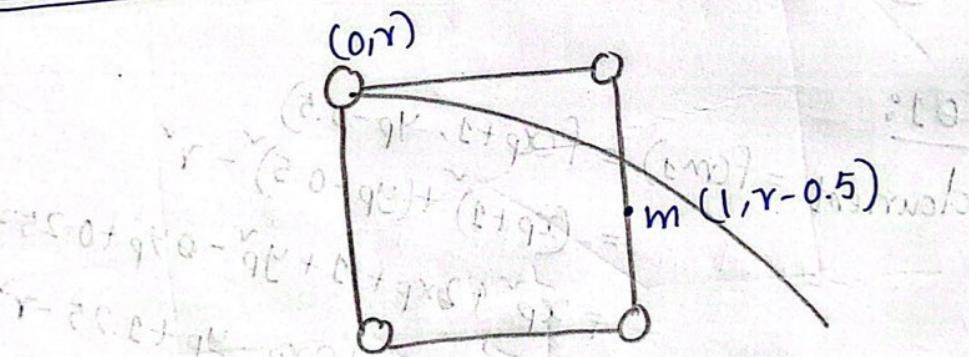
Step 03 :

$$d_{\text{next}} - d_{\text{current}} = (x_p + y_p + 2xp - y_p + 4.25 - r) - (x_p + y_p + 2xp - y_p + 1.25 - r)$$

(E) For current step E,

$$d_{\text{next}} = d_{\text{current}} + 2xp + 3$$

# Find out  $d_{\text{initial}}$



$$\begin{aligned} f(m_0) &= f(1, r-0.5) = 1 + (r-0.5)^2 - r^2 \\ d_{\text{initial}} &= f(1, r-0.5) = 1 + r^2 - r + 0.25 - r^2 \\ &= 1.25 - r \end{aligned}$$

$r$	$d_{init} = 1.25 - r$	$d_{init} = 1 - r$
1	0.25 (SE)	0 (SE)
2	-0.75 (E)	-1 (E)
3	-1.75 (E)	-2 (E)
4	-2.75 (E)	-3 (E)

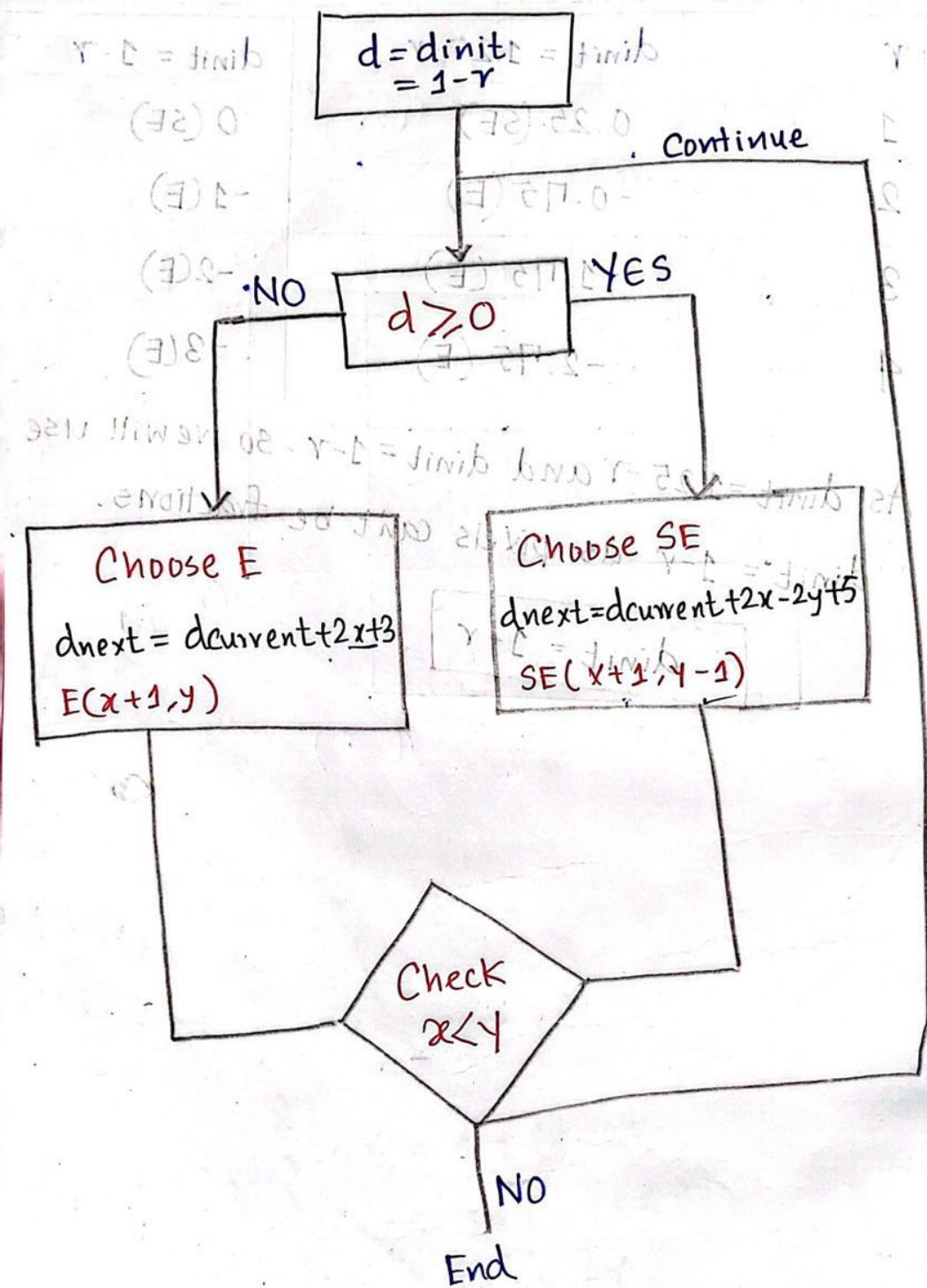
- As  $d_{init} = 1.25 - r$  and  $d_{init} = 1 - r$ . So we will use

$d_{init} = 1 - r$  as pixels can't be fractions.

$$\therefore d_{init} = 1 - r$$

on

line



Scanned with CamScanner

Scanned with CamScanner



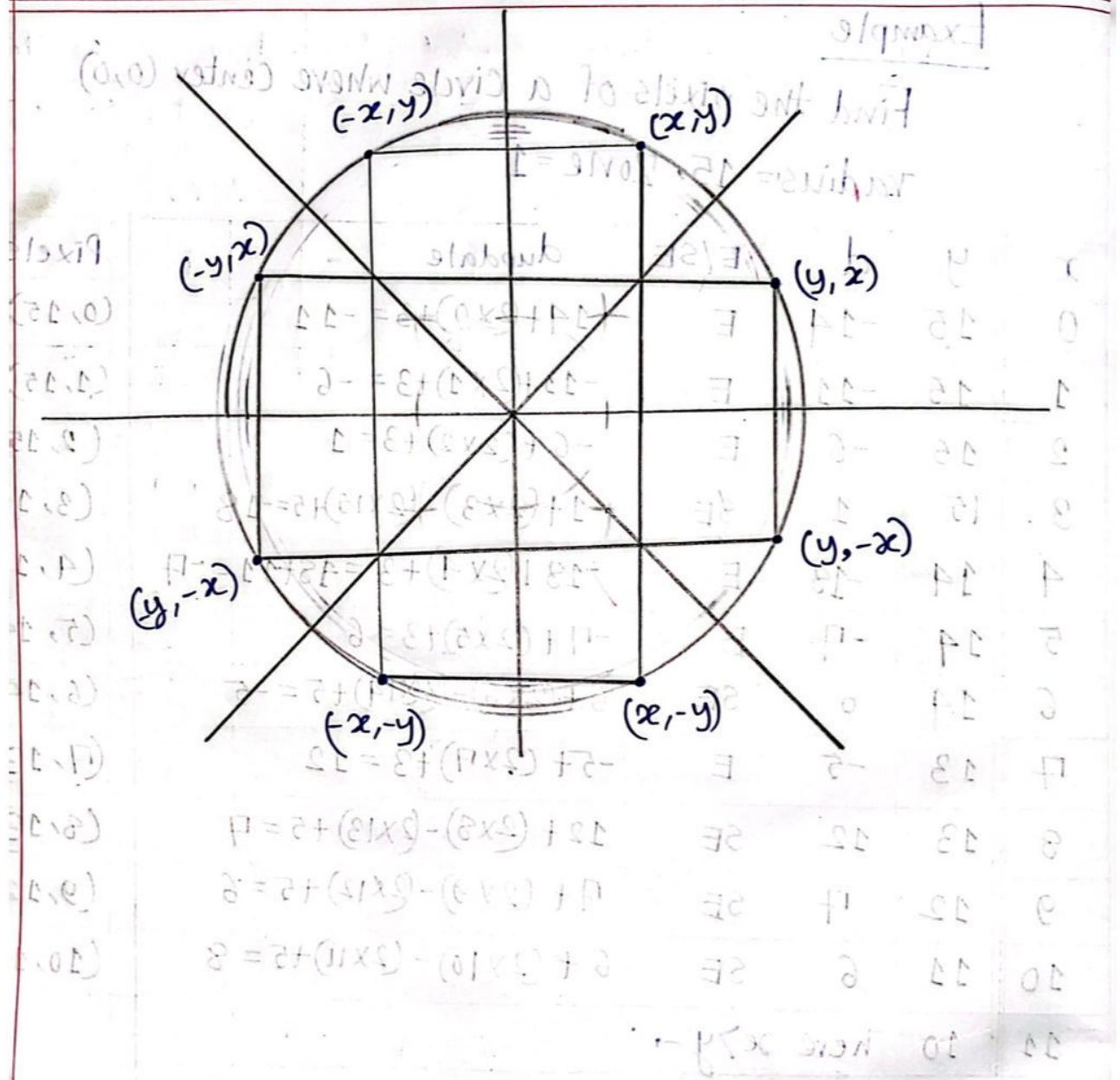
## Example

Find the pixels of a circle where center (0,0)

radius = 15, Zone = 1

x	y	d	E/SE	update	Pixels
0	15	-14	E	$-14 + (2 \times 0) + 3 = -11$	(0, 15)
1	15	-11	E	$-11 + (2 \times 1) + 3 = -6$	(1, 15)
2	15	-6	E	$-6 + (2 \times 2) + 3 = 1$	(2, 15)
3	15	1	SE	$1 + (2 \times 3) - (2 \times 15) + 5 = -18$	(3, 15)
4	14	-18	E	$-18 + (2 \times 4) + 3 = 18 + 1 = -17$	(4, 14)
5	14	-7	E	$-7 + (2 \times 5) + 3 = 6$	(5, 14)
6	14	6	SE	$6 + (2 \times 6) - (2 \times 14) + 5 = -5$	(6, 14)
7	13	-5	E	$-5 + (2 \times 7) + 3 = 12$	(7, 13)
8	13	12	SE	$12 + (2 \times 8) - (2 \times 13) + 5 = 7$	(8, 13)
9	12	7	SE	$7 + (2 \times 9) - (2 \times 12) + 5 = 6$	(9, 12)
10	11	6	SE	$6 + (2 \times 10) - (2 \times 11) + 5 = 8$	(10, 11)
11	10	here $x > y \rightarrow$			

$$\begin{aligned}d_{init} &= 1 - r \\&= 1 - 15 = -14\end{aligned}$$



$$Y = E \times \text{Im}(\phi)$$

1960-1961

Scanned with CamScanner

Scanned with CamScanner

Change Zone

	Zone-1	Zone-0	Zone-2	Zone-3	Zone-4	Zone-5	Zone-6
x	y	y	-x	y	-y	x	-y
0	15	15	0	0	15	-15	0
1	15	15	1	-1	15	-15	-1
2	15	15	2	-2	15	-15	2
3	15	15	3	-3	15	-15	3
4	14	14	4	-4	14	-14	4
5	14	14	5	-5	14	-14	5
6	14	14	6	-6	14	-14	6
7	13	13	7	-7	13	-13	7
8	13	13	8	-8	13	-13	8
9	12	12	9	-9	12	-12	9
10	11	11	10	-10	11	-11	10

Scanned with CamScanner

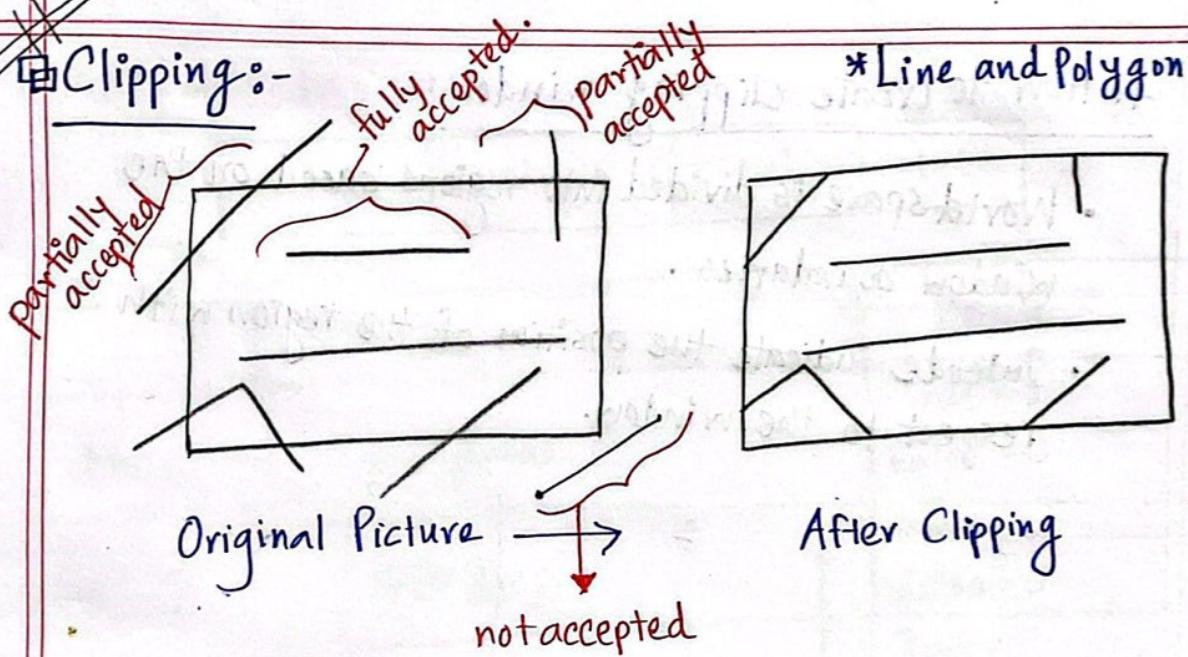
Scanned with CamScanner

Circles Left									
Zone F					Zone G				
Row 1		Row 2		Row 3	Row 4		Row 5		Row 6
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200

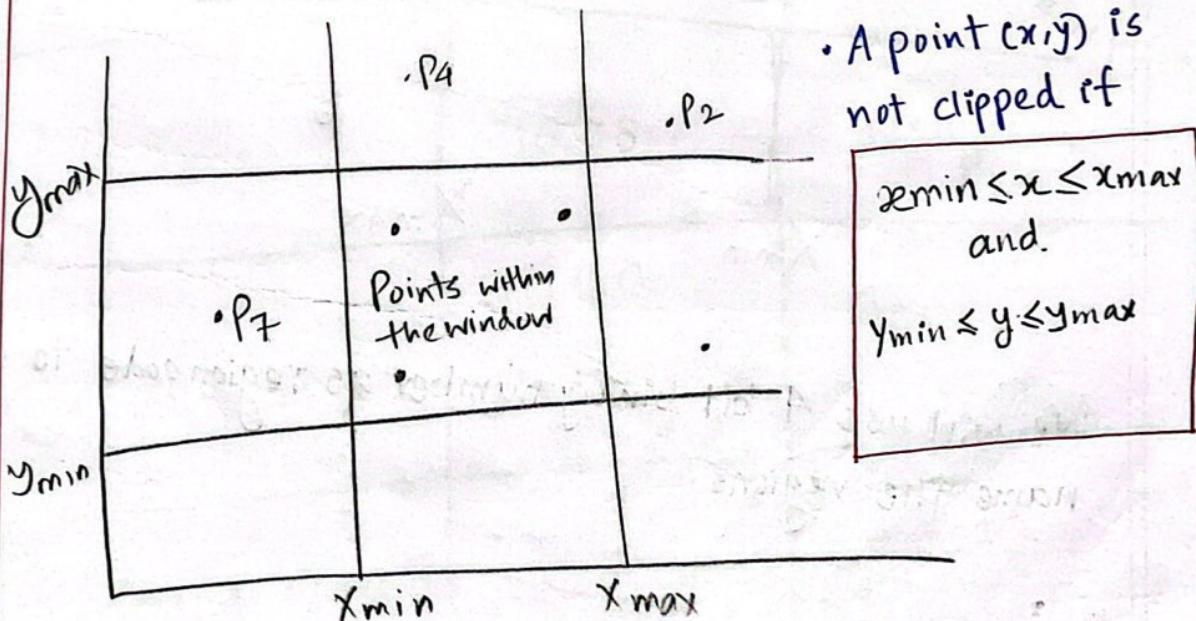
Scanned with CamScanner

Scanned with CamScanner

## ~~Clipping :-~~



- 2 Algorithm →
  - i) Cohen Sutherland Algorithm
  - ii) Cyrus beck Algorithm

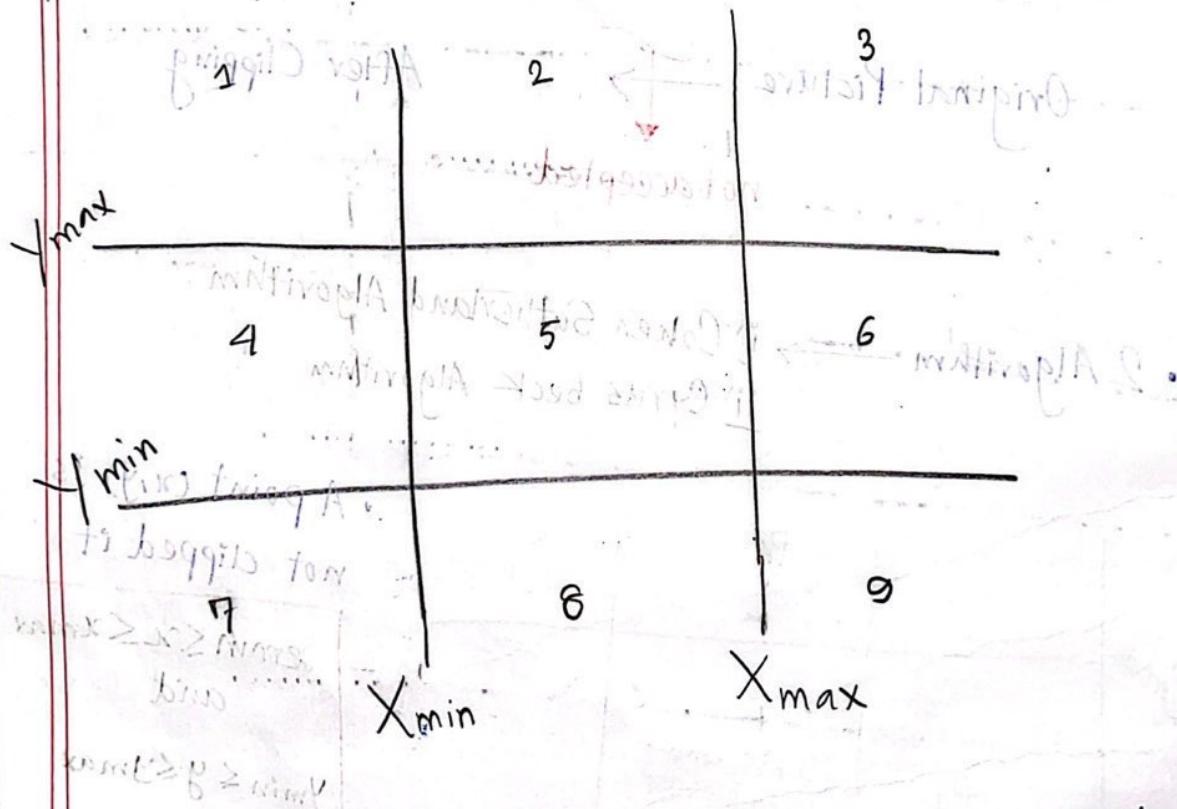


Scanned with CamScanner

Scanned with CamScanner

## # How to create clipping window:

- World space is divided into regions based on the window boundaries.
- Outcode indicate the position of the region with respect to the window.



We will use 4 bit binary number as region code to name the regions.

## Region Code:

above	below	right	left
-------	-------	-------	------

	Check	Result
Above	$y > y_{max}$	True = 1 False = 0
Below	$y < y_{min}$	True = 1 False = 0
Right	$x > x_{max}$	True = 1 False = 0
Left	$x < x_{min}$	True = 1 False = 0
$y_{max}$	1001	1010
	1000	0010
$y_{min}$	0000	0110
	0001	0100
Xmin		
Xmax		

Scanned with CamScanner

Scanned with CamScanner

# Note: If the point is on axis, suppose between 1010 and 0010, choose the lower one 0010.

• Where the number of "1" is less, select that.

### Pseudo Code:

Cohen-Sutherland( $x_1, y_1, x_2, y_2$ ):

OC1 = calculate\_outcode( $x_1, y_1$ )

OC2 = calculate\_outcode( $x_2, y_2$ )

while True:

if ( $OC1 = OC2 = 0000$ )

fully accepted

break;

elseif ( $OC1 \text{ and } OCT \neq 0000$ )

fully rejected

break

else { if  $OCT1 \neq 0000$  {

( $x_1, y_1$ ) = find intersection point of line and  
the boundary corresponding to non-zero bit  
of  $OC1$

$OC1 = \text{calculate\_outcode}(x_1, y_1) \rightarrow \text{update } OC1$

}

```

else {
     $(x_2, y_2)$  = find intersection point of line and the
    boundary corresponding to non-zero oct bit
    of OC2.
    OC2 = calculate_outcode( $x_2, y_2$ ) → update OC2
}
Continue
}

```

### Rejection Process :-

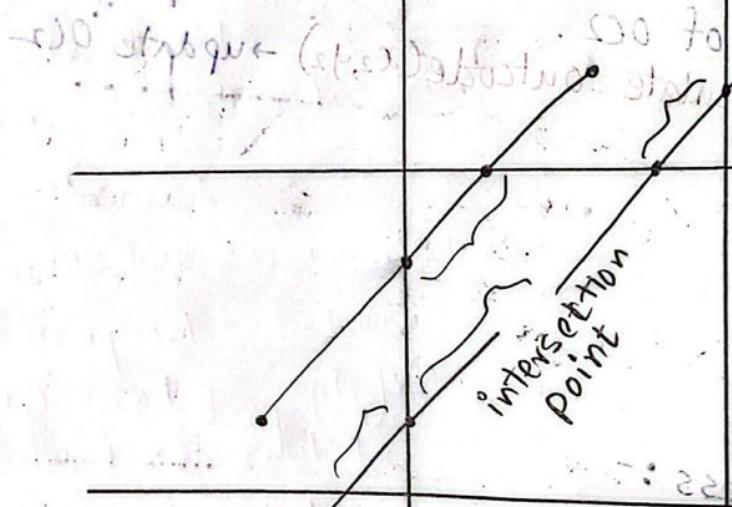
Suppose,

$$\begin{array}{r} 1000 \\ 1010 \\ \hline 1010 \end{array}$$

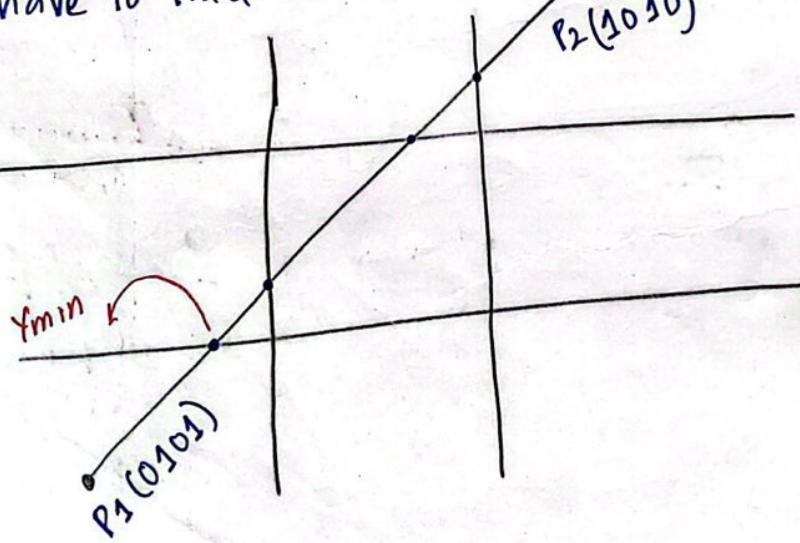
minimum 1, at least get a "1" then fully rejected.

# What if partially accepted?

Find the position of pricing curve



We have to find the intersection point.



Scanned with CamScanner

Scanned with CamScanner

This intersection point shares region 4 and 7.

To select the intersection point, we will select the region with least 1's.  
∴ Region 4 is selected.

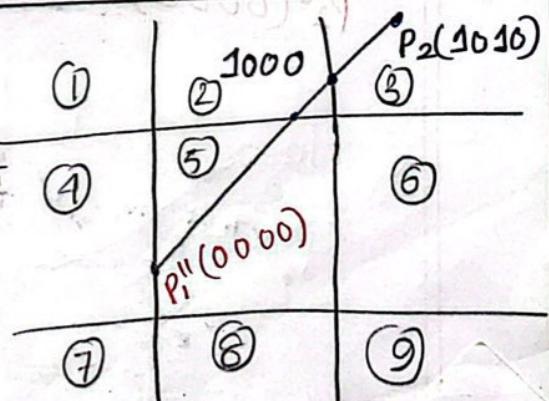
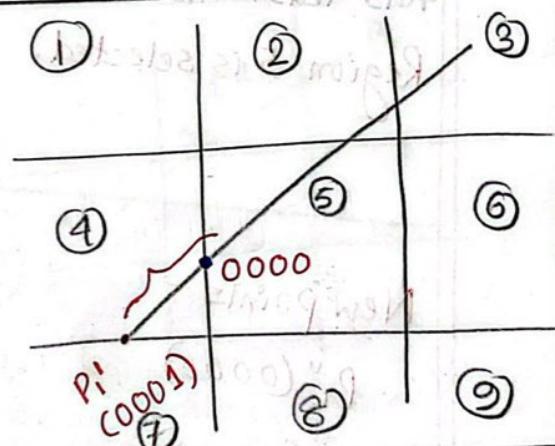
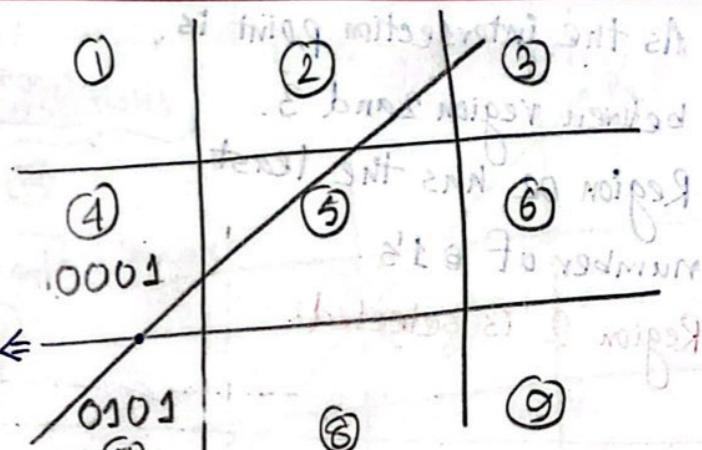
Now, the new point shares region with region 4 and 5.

Number of 1's less in Region 5.

∴ Region 5 is selected.

So,  $P_1''$  is on the window so it is accepted.

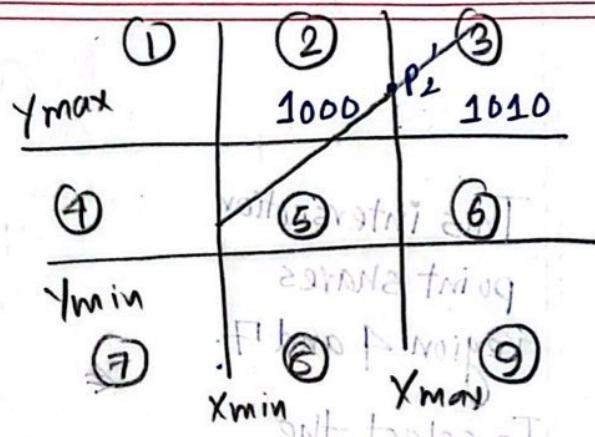
Now,  $P_2$  is on 3rd region. And its not on the window. So, we would look for an intersection point.



As the intersection point is between region 2 and 3.

Region 02 has the least number of 01's.

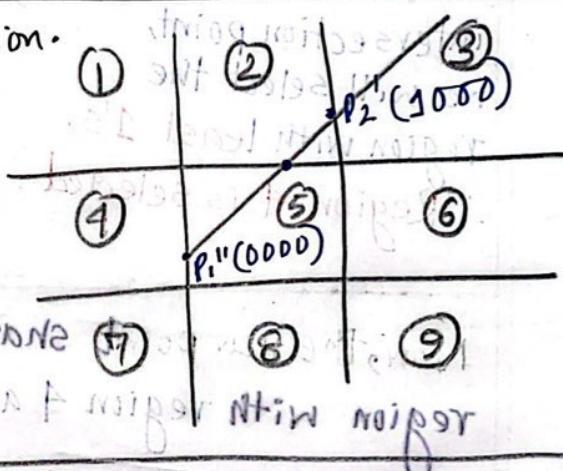
Region 2 is selected.



Now, the line has a new intersection.

The new point is between region 2 and 5. So, region 5 has least number of 1's.

Region 5 is selected.

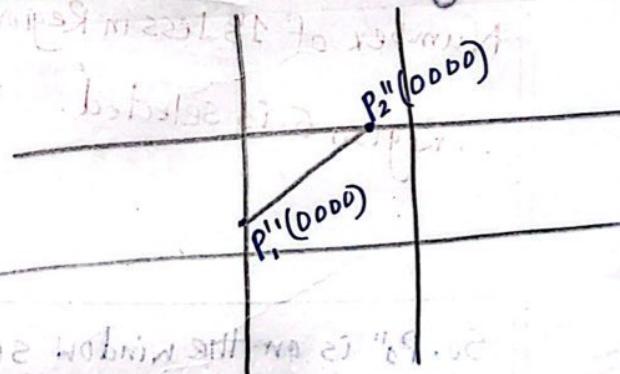


New points -

$P_1''(0000)$

$P_2''(0000)$

(0000) 1000



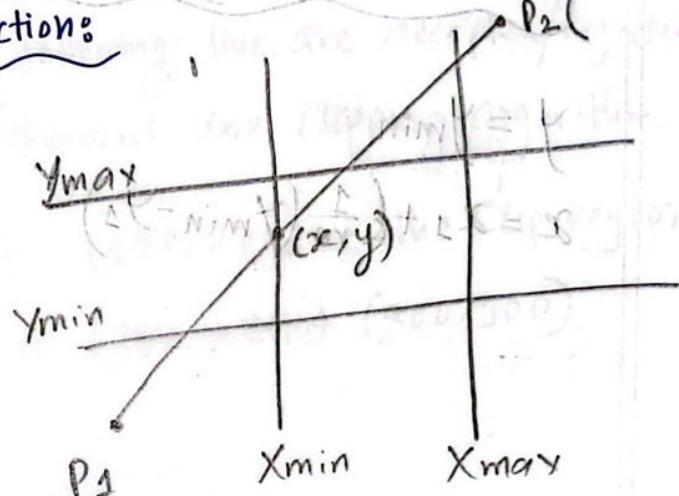
for 2nd bnf. major bre no 21 5 9. now  
1001 below 0002. below 0001. INT no  
. find 9. major bre no 10

## Formula

### • Left Boundary Intersection:

$$X = X_{\min}$$

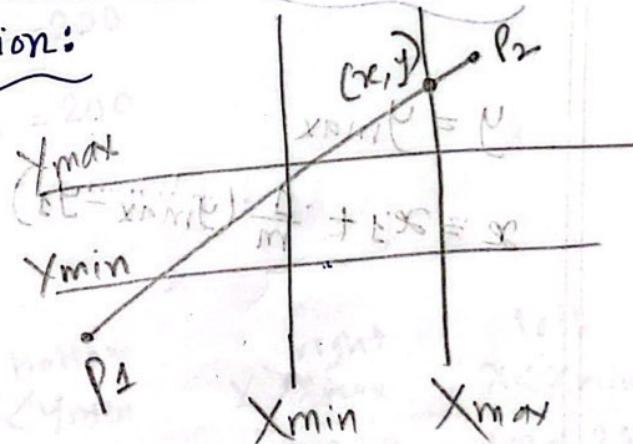
$$y = y_1 + (X_{\min} - X_1)$$



### • Right Boundary Intersection:

$$X = X_{\max}$$

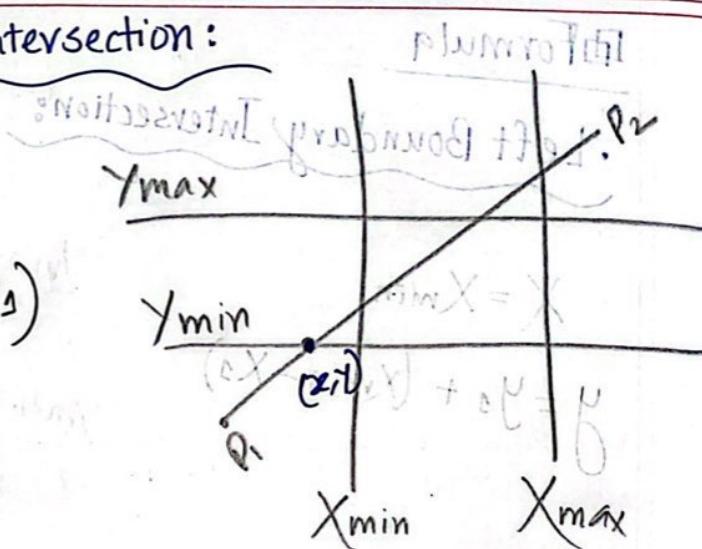
$$y = y_1 + m(X_{\max} - X_1)$$



### Bottom Boundary Intersection:

$$y = y_{\min}$$

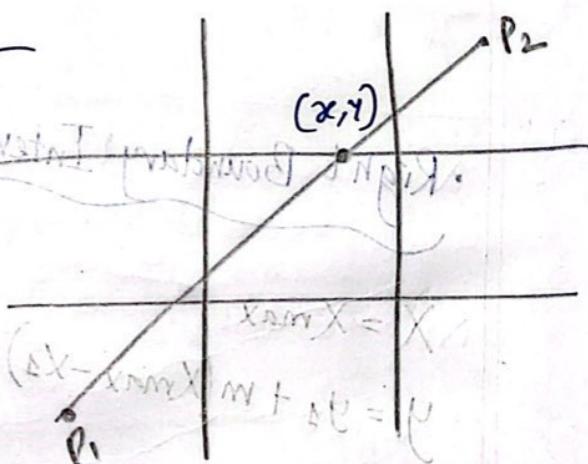
$$x = x_1 + \frac{1}{m} (y_{\min} - y_1)$$



### Top Boundary Intersection:

$$y = y_{\max}$$

$$x = x_1 + \frac{1}{m} (y_{\max} - y_1)$$



\* There is nothing like partially, actually if its following partial then it's perfect.

## Example

Determine whether the following line are accepted/rejected/partial using Cohen Sutherland line clipping algorithm.

a) Given  $(-250, -200)$  to  $(250, 200)$  be the Clip region.

i)  $(-100, -220)$  to  $(300, 100)$ .

Sol<sup>n</sup>:

Step 01: Check Condition and Make Opcode.

$$X_{\min} = -250 \quad Y_{\min} = -200$$

$$X_{\max} = 250 \quad Y_{\max} = 200$$

$$P_1(100, 50) \rightarrow OC1.$$

$y > y_{\max}$	$y < y_{\min}$	$x > x_{\max}$	$x < x_{\min}$
$50 > 200$	$50 < -200$	$100 > 250$	$100 < -250$
0	0	0	0
$100 > 200$	$100 < -200$	$300 > 250$	$300 < -250$
0	0	1	0

Scanned with CamScanner

Scanned with CamScanner

## Step 02: Check Loop Conditions

#First if Condition:

$$OC1 = OC2 = 0000$$

→ X

#Second if Condition:

$$OC1 \text{ and } OC2 \neq 0000$$

→ X

#3rd if Condition:

$$OC1 \neq 0000$$

Find Non Zero Bit intersection

#4th if Condition:

$$OC2 \neq 0000$$

Find Intersection

## Step 03:

Intersect P<sub>2</sub> → X<sub>max</sub>

$$\text{Formula: } X = X_{\max} = 250$$

$$y = y_1 + m(X_{\max} - x_1)$$

$$= 100 + m(250 - 330)$$

$$= 100 + \frac{100 - 50}{300 - 150} (-50)$$

$$= 100 + (0.25)(-50)$$

$$= 100 - 12.5$$

$$= 87.5$$

$\therefore y = 87.5 \rightarrow$  Don't Roundup Now.

$\therefore$  Intersection Point  $P_2'(250, 87.5)$

$\downarrow$   
 $x$

$\downarrow$   
 $y$  (New Value)

Check Conditions Again:

$$x = 250 \quad y = 87.5$$

$$y > y_{\max} \rightarrow 87.5 > 200 = 0 \quad [0 \text{ because there is no condition matching}]$$

$$y < y_{\min} \rightarrow 87.5 < -200 = 0$$

$$x > x_{\max} \rightarrow 250 > 250 = 0$$

$$x < x_{\min} \rightarrow 250 < 250 = 0$$

$\therefore$  Updated Values (0000)

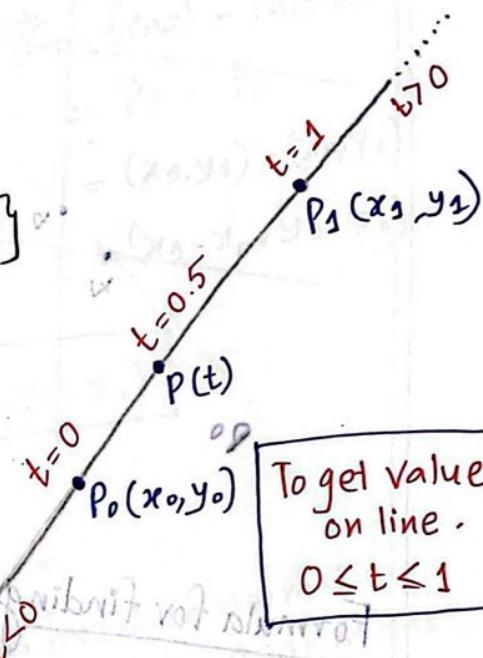
Again Start from Condition 1 from Loop.

## Cyrus Beck Algorithm

Step 01: Parametric Eq<sup>n</sup> of a Line.

$$\begin{aligned}P(t) &= P_0 + t(P_1 - P_0) \\&= (x_0, y_0) + t \{ (x_1, y_1) - (x_0, y_0) \} \\&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0)\end{aligned}$$

We can use different values of  $t$  and get values.

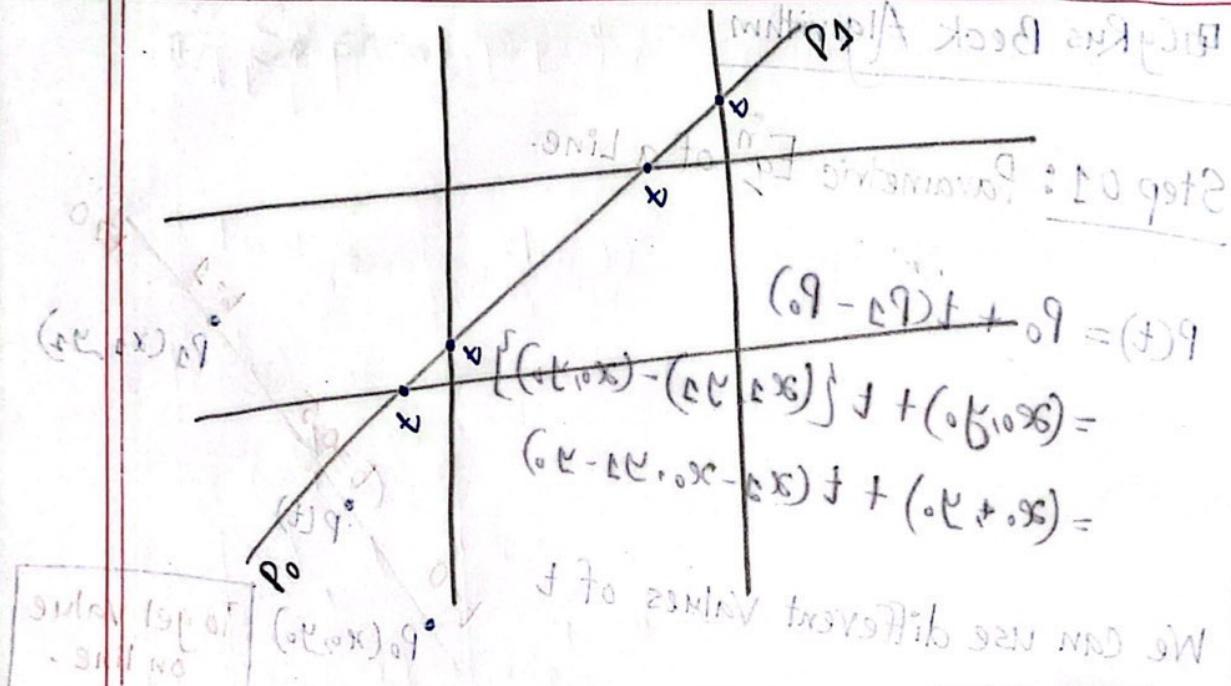


$$\begin{aligned}t=0, \quad P(0) &= (x_0, y_0) \rightarrow \text{initial point} \\P(0) &= (x_0, y_0) + 0(x_1 - x_0, y_1 - y_0) \\P(1) &= (x_0, y_0) + 1(x_1 - x_0, y_1 - y_0) \\&= (x_0 + x_1 - x_0, y_0 + y_1 - y_0) \\&= (x_1, y_1)\end{aligned}$$

P.T.O

Scanned with CamScanner

Scanned with CamScanner



Formula for finding out  $t$ :

$$t = \frac{|N| \cdot (P_0 - P_E)}{|N| \cdot D - N \cdot (P_0 - P_E)}$$

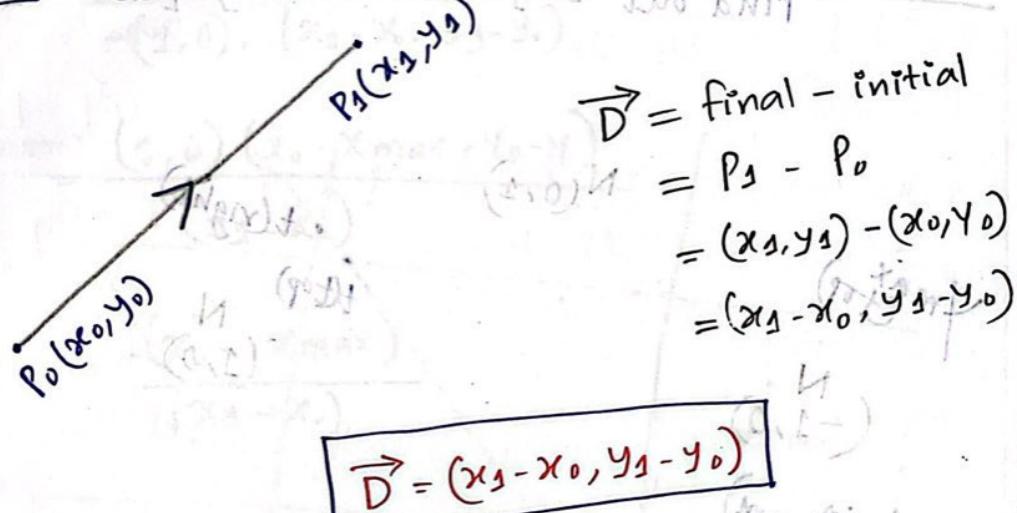
→ Normal Vectors  
 → Dot Product  
 → Initial Point  $(x_0, y_0)$   
 → Random point  $(x_1, y_1)$  on Boundary  
 → Dot Product  
 → Normal Vector

$(x_1 - x_0, y_1 - y_0) = (x_2 - x_0, y_2 - y_0) =$   
 $(x_2 - x_0, y_2 - y_0) = (x_1 - x_0, y_1 - y_0) =$

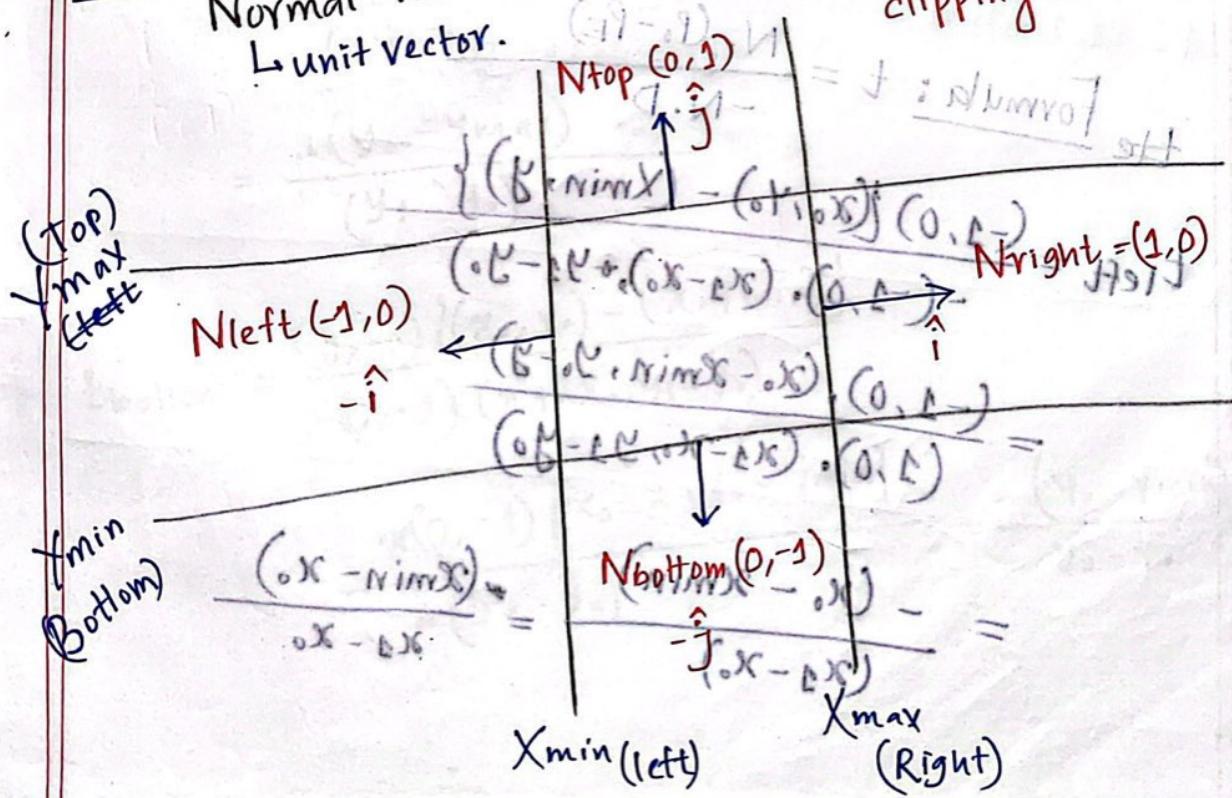
Scanned with CamScanner

Scanned with CamScanner

## Step 02: Vector Representation of a Line.



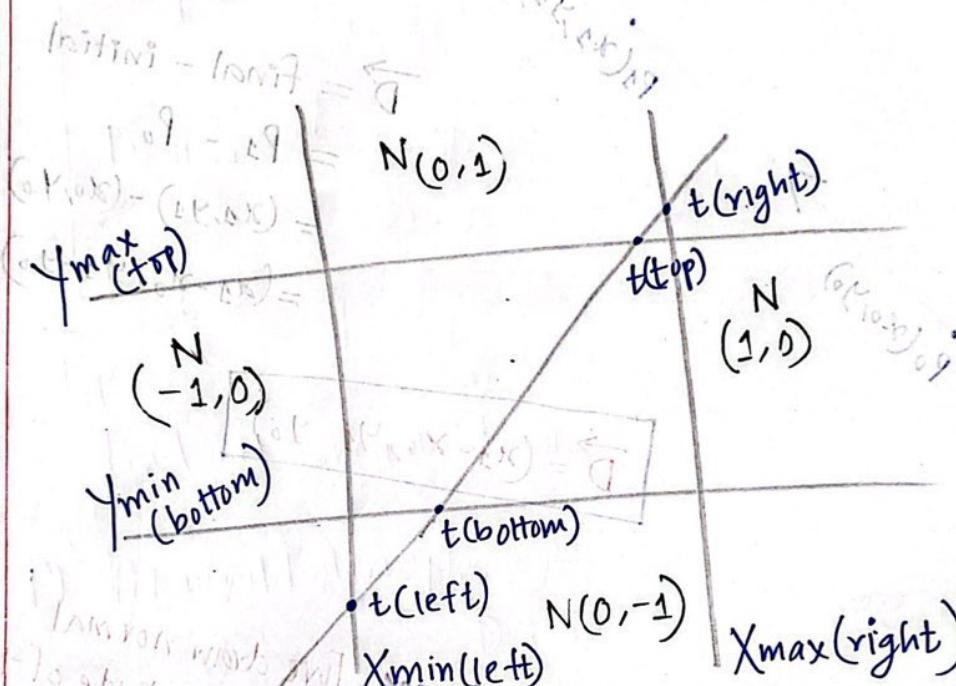
## Step 03: Normal Vectors of Boundaries. [We draw normal vectors outside of the clipping window]



Scanned with CamScanner

Scanned with CamScanner

Step 04: Find out  $t$  for the boundary [4 times].



The formula:  $t = \frac{N \cdot (P_0 - P_E)}{-N \cdot D}$

$$t_{left} = \frac{(-1, 0) \cdot ((x_0, y_0) - (x_{min}, y))}{-(-1, 0) \cdot (x_1 - x_0, y_1 - y_0)}$$

$$= \frac{(-1, 0) \cdot (x_0 - x_{min}, y_0 - y)}{(1, 0) \cdot (x_1 - x_0, y_1 - y_0)}$$

$$= \frac{-(x_0 - x_{min})}{(x_1 - x_0)} = \frac{(x_{min} - x_0)}{x_1 - x_0}$$

Scanned with CamScanner

Scanned with CamScanner

$$t_{right} = \frac{(1, 0) \cdot \{(x_0, y_0) + (x_{max}, y)\}}{(1, 0) \cdot (x_1 - x_0, y_1 - y_0)} = \frac{y - y_0}{x_1 - x_0} = \text{right}$$

$$\frac{(x_{max})(1, 0)(x_0 - x_{max}, y_0 - y)}{(x_1 - x_0)(1, 0)} = \text{right}$$

$$= \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{(0, 1) \cdot ((x_0, y_0) - (x_1, y_{max}))}{(0, 1) \cdot (x_1 - x_0, y_1 - y_0)} = \text{top}$$

$$t_{top} = \frac{(0, 1) \cdot (x_0 - x_1, y_0 - y_{max})}{(0, 1) \cdot (y_1 - y_0)} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{(0, -1) \cdot ((x_0, y_0) - (x_1, y_{min}))}{(0, -1) \cdot (x_1 - x_0, y_1 - y_0)}$$

$$= \frac{x(0, -1) \cdot (x_0 - x_1, y_0 - y_{min})}{x(y_1 - y_0)} = \frac{-(y_0 - y_{min})}{y_1 - y_0}$$

Scanned with CamScanner

Scanned with CamScanner

$$t_{top} = \frac{-(y_0 - y_{max})}{y_1 - y_0} \cdot \frac{(x_{min} - x_0) / (x_1 - x_0)}{x_{min} - x_0} = \frac{x_{min} - x_0}{x_1 - x_0}$$

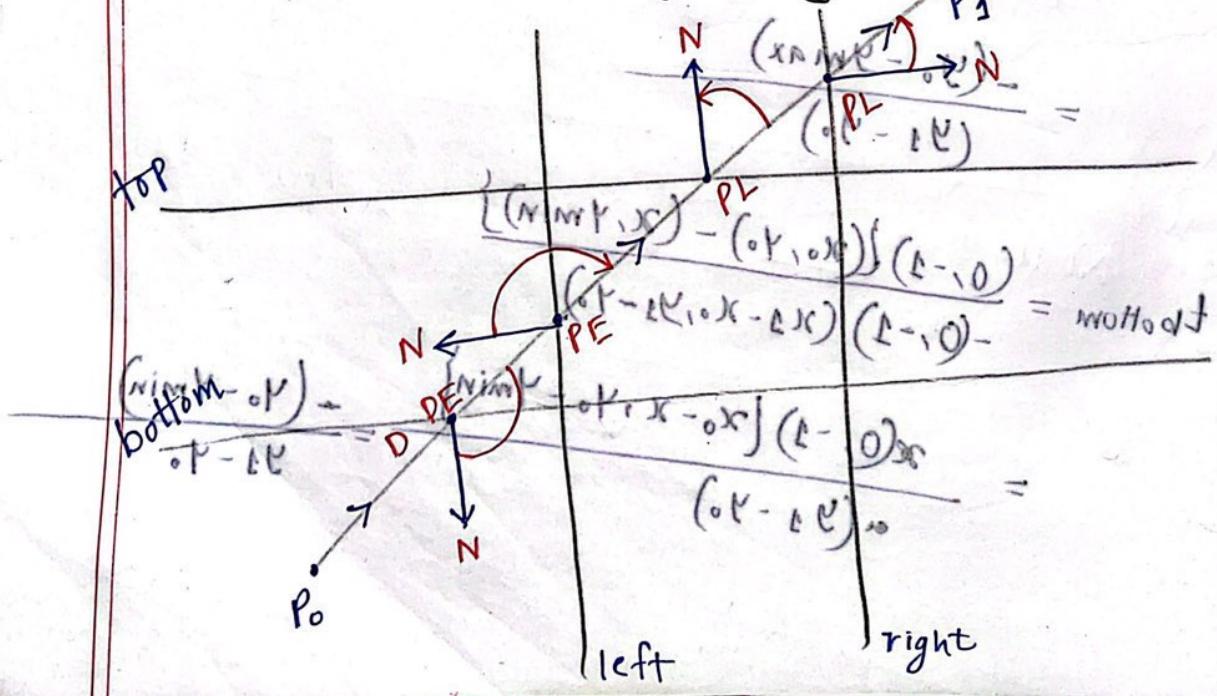
$$t_{bottom} = \frac{-(y_0 - y_{min})}{y_1 - y_0} \cdot \frac{x_{max} - x_0}{x_{max} - x_0} = \frac{x_{max} - x_0}{x_1 - x_0}$$

Step 05:

Find out when to draw and when to discard.

PE (Potential Entering): Angle between  $N$  (Normal) and  $D$  (Line) is greater than  $90^\circ$  just

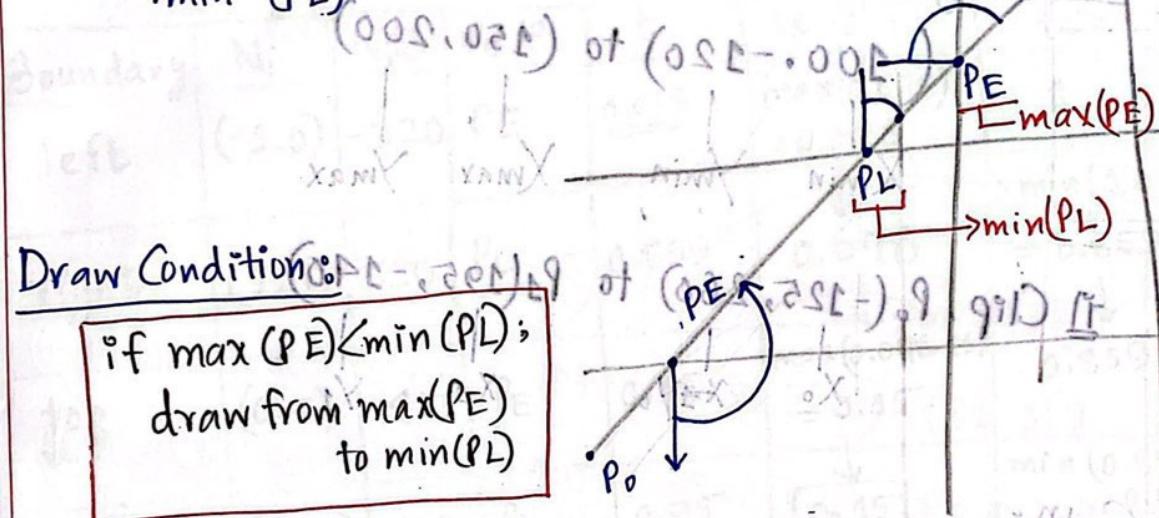
PL (Potential Leaving): Angle between  $N$  (Normal) and  $D$  (Line) is greater less than  $90^\circ$ .



Scanned with CamScanner

Scanned with CamScanner

Take max(PE)  
min(PL)



Draw Condition:

if  $\max(PE) < \min(PL)$ ;  
draw from  $\max(PE)$   
to  $\min(PL)$

if  $\max(PE) > \min(PL)$ ;  
discard

PE  $\rightarrow N.D < 0$   
PL  $\rightarrow N.D > 0$

### Clip Region:

(-100, -120) to (150, 200)

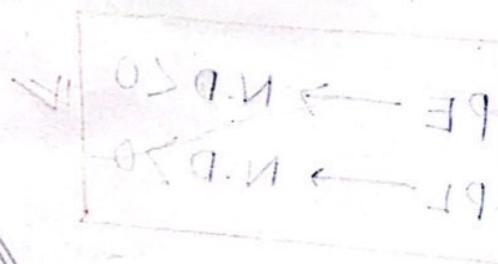
X <sub>min</sub>	Y <sub>min</sub>	X <sub>max</sub>	Y <sub>max</sub>
------------------	------------------	------------------	------------------

Clip, P<sub>0</sub>(-125, 260) to P<sub>1</sub>(195, -140)

X <sub>0</sub>	X <sub>1</sub>	Y <sub>0</sub>
----------------	----------------	----------------

Soln:

$$\begin{aligned} \text{Line Vector} &= (x_1 - x_0, y_1 - y_0) \\ &= (195 - (-125), -140 - 260) \\ &= (320, -400) \end{aligned}$$



N.D < 0 → PE  
N.D > 0 → PL

initially  $t_E = 0, t_L = 1$ ; D = (320, -400)

Boundary	$N_i$	$N_i.D$	PE/PL	$t$	$t_E = 0$	$t_L = 1$
left	(-1, 0)	-320	PE	<u>0.078</u>	$\max(0, 0.078) = 0.078$	1
right	(1, 0)	320	PL	0.859	$\min(1, 0.859) = 0.859$	
top	(0, 1)	-400	PE	0.15	$\max(0.078, 0.15) = 0.15$	0.859
bottom	(0, -1)	400	PL	0.95	$\min(0.859, 0.95) = 0.859$	$t_E$

$$t_{left} = \frac{(1, 0) \cdot (-125 + 100)}{195 + 125} = 0.078$$

$$t_{right} = \frac{-(-125 + 150)}{195 + 125} = 0.859$$

$$t_{top} = \frac{-(y_1 - y_{max})}{y_2 - y_1} = \frac{-(260 - 200)}{-140 - 260} = 0.15$$

$$t_{bottom} = \frac{-(y_1 - y_{min})}{y_2 - y_1} = \frac{-(260 + 120)}{-140 - 260} = 0.95$$

if  $t_E > t_L$   
line segment  
outside window

$t_E < t_L \rightarrow$  draw  $L = \{J, O = f\}$   $\text{y} \parallel \text{nitini}$

$L = \{J\}$   $0.15 < 0.859 \rightarrow$  we can draw.

$L = \{J, O = f\}$   $(\vec{P}(0,0) \times \vec{N}) \cdot \vec{N} = 0.15 < 0.859$

Step 01

$$\vec{P}(t) = (-125, 260) + t(320, -400) \quad (0, L) \quad \text{Inpir}$$

$$\vec{P}(0.15) = (-125, 260) + 0.15(320, -400) = (-77, 200)$$

$$\vec{P}(0.859) = (-125, 260) + 0.15(320, -400) = (150, -84)$$

$\vec{P}(0.859) = (-125, 260) + 0.15(320, -400) = (150, -84)$  method

$$\vec{P}(0,0) = \frac{(001 + 251) - f(0, L)}{251 + 251} = f_{11}$$

$$\vec{P}(0) = \frac{(001 + 251) - }{251 + 251} = f_{11}$$

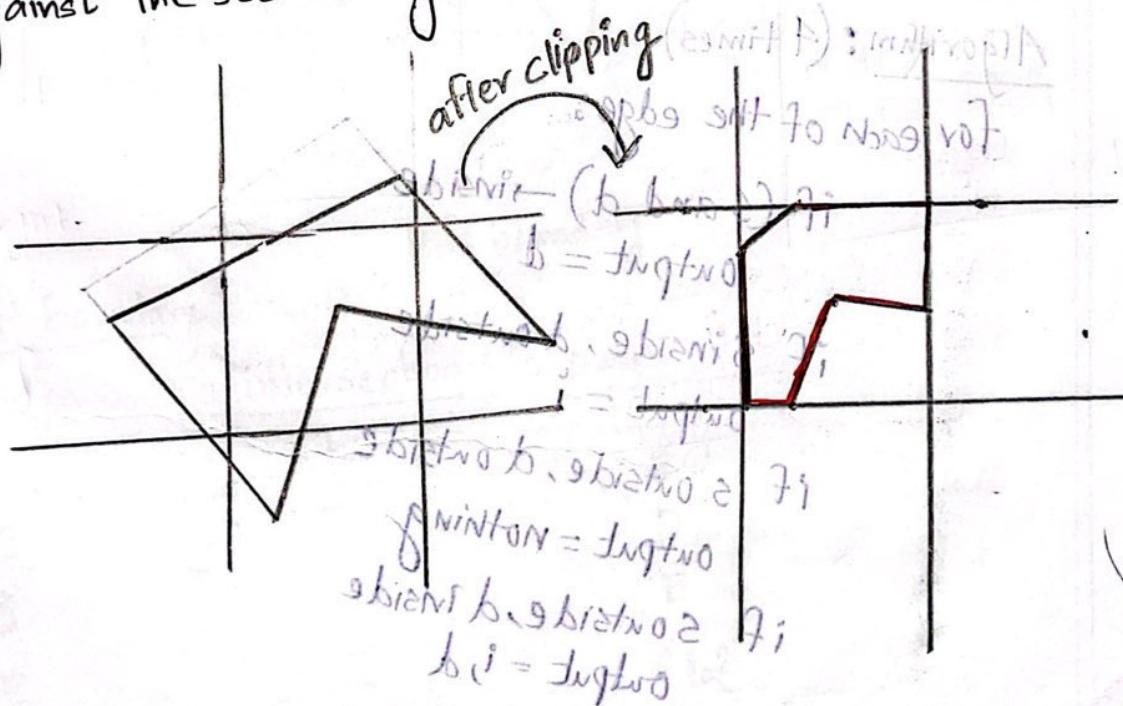
$$\vec{P}(0) = \frac{(001 - 251) - }{251 - 251} = \frac{(x_{\text{NIP}} - 0, L) - }{0, L - 0, L} = g_{11}$$

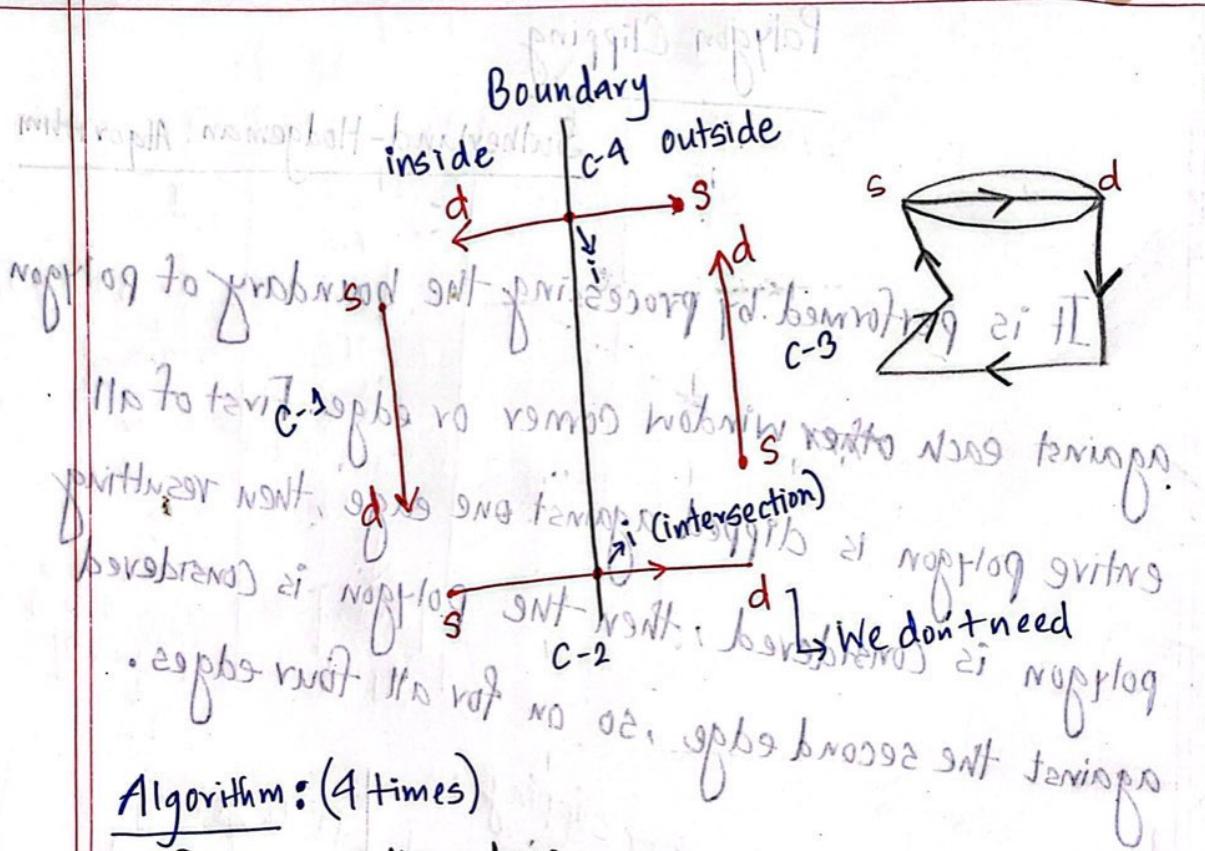
$$\vec{P}(0) = \frac{(001 + 251) - }{251 - 0, L} = \frac{(x_{\text{NIP}} - 0, L) - }{0, L - 0, L} = m_{11}$$

## Polygon Clipping

### Sutherland-Hodgeman Algorithm

It is performed by processing the boundary of polygon against each other window corner or edge. First of all entire polygon is clipped against one edge, then resulting polygon is considered, then the polygon is considered against the second edge, so on for all four edges.





Algorithm: (4 times)

for each of the edges:

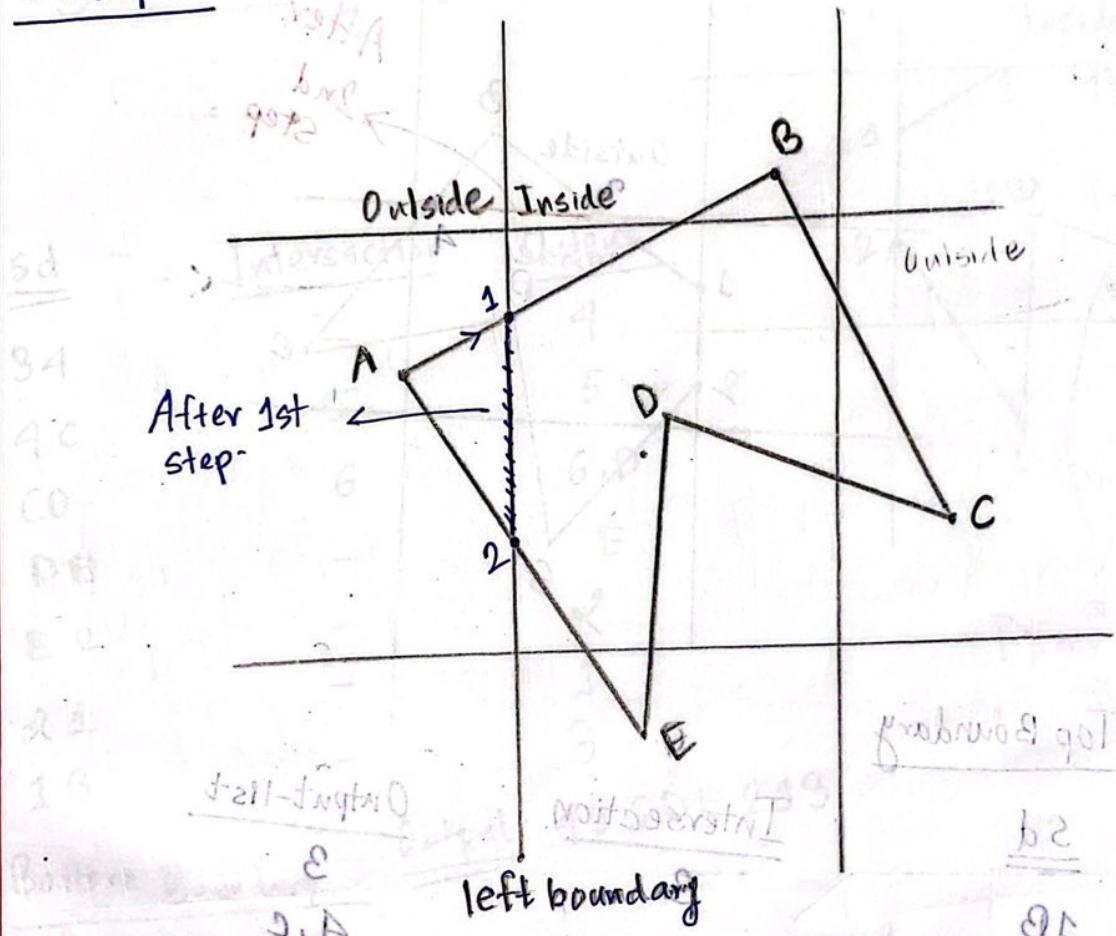
if (S and d) → inside  
output = d

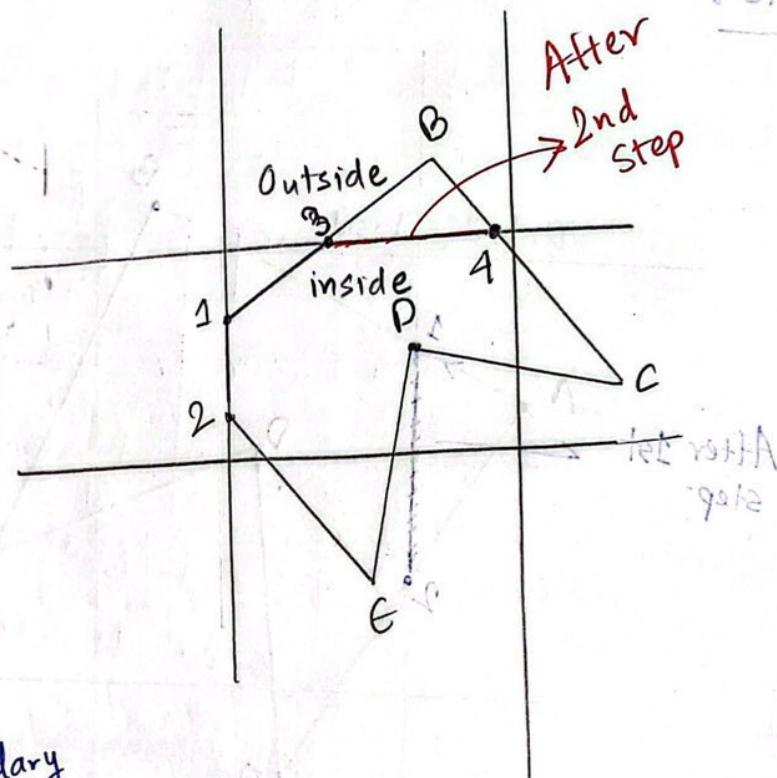
if s inside, d outside  
output = i

if s outside, d outside  
output = nothing

if s outside, d inside  
output = i, d

Example:





### Top Boundary

<u>Sd</u>	<u>Intersection</u>	<u>Output-list</u>
1B	B	3
BC	A	4, C
CD	D, E	P
DE	E	E
E2	D	2
21	E	1

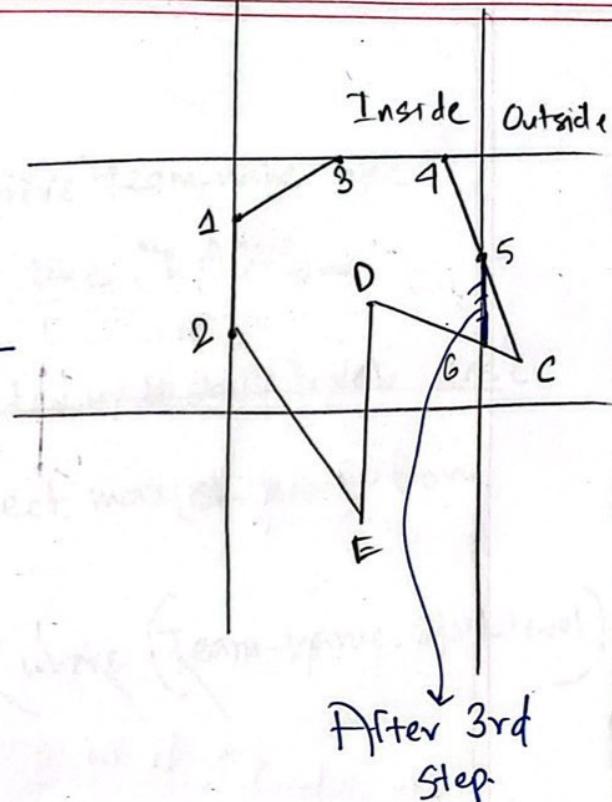
Output-Post = 3, 4, C, D, E, 2, 1

B, E, D, G, I = left boundary

## Right Boundary

<u>sd</u>	<u>Intersection</u>	<u>Output</u>
34	-	4
4C	5	5
CD	6	6, D
DE	-	E
E2	-	2
21	-	1
13	-	3

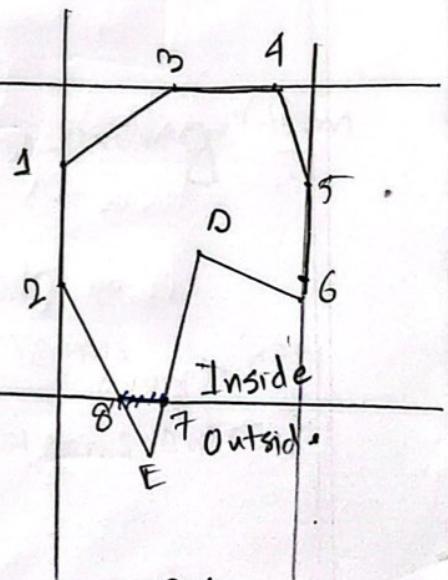
Output: 4 5 6 D E 2 1 3



## Bottom Boundary

<u>sd</u>	<u>Intersection</u>	<u>Output</u>
45	-	5
56	-	6
6D	-	D
DE	7	7
E2	8	8, 2
21	-	1
13	-	3
34	-	4

Output: 5 6 D 7 8 2 1 3 4



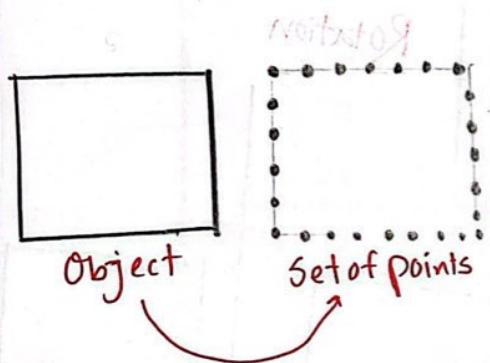
Scanned with CamScanner

Scanned with CamScanner

## Transformation

Transformations are operations applied to geometrical description of an object to change its position, orientation or size.

Reposition and resizing of an object.

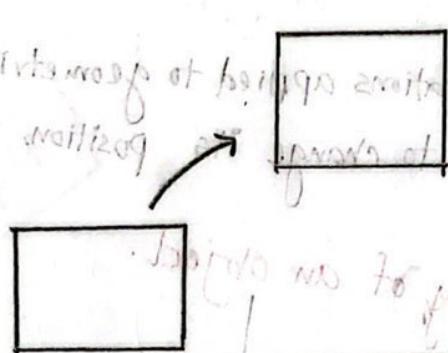


Object is nothing but set of points.

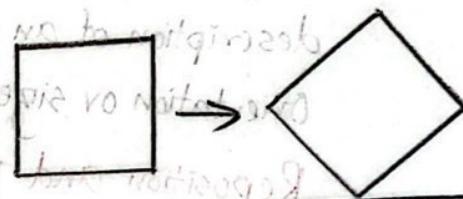
5 type of Transformation

1. Translation : Changing position of an object
2. Rotation : Changing angle of an object
3. Scaling : Shrink or resize it into bigger or smaller size by keeping the main object unchanged.
4. Reflection : We make reflection by considering any axis or a line a mirror.
5. Shraving : We put force to change shape from any side.

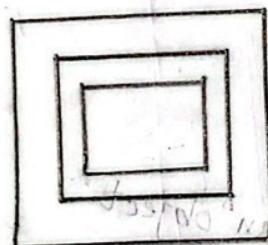
Instancing of having multiple objects in environment



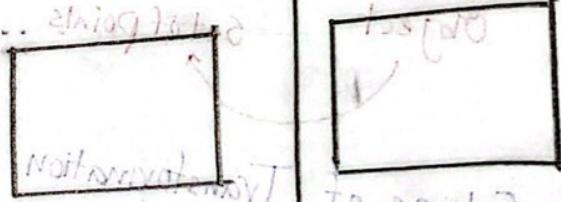
Without rotation



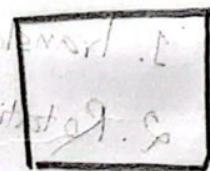
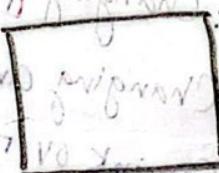
Translation



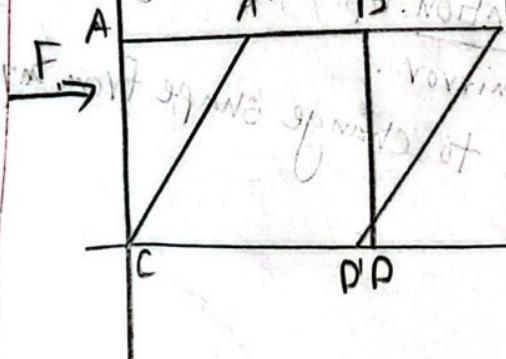
Rotation



Scaling



Reflection



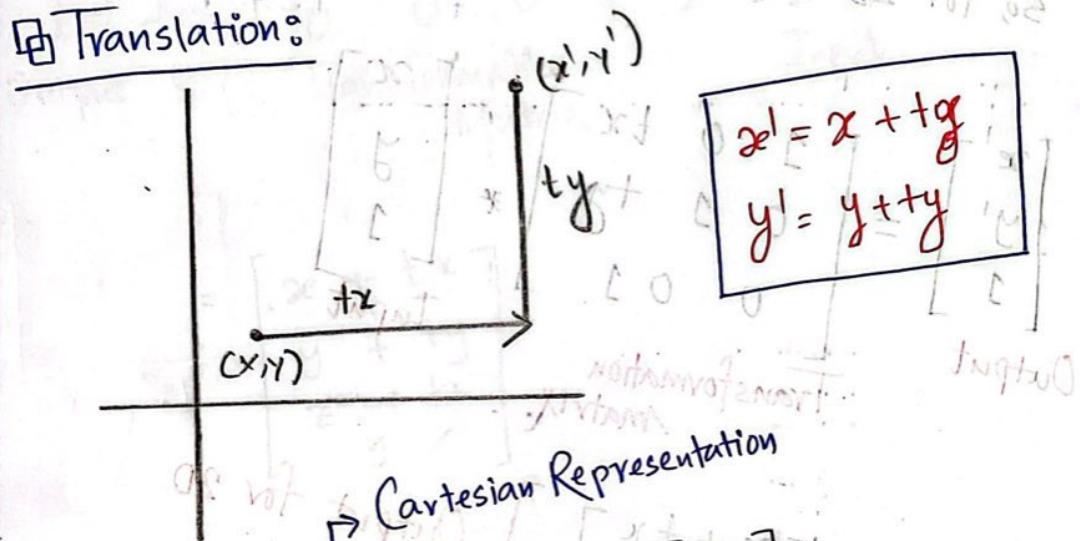
Sharing

Scanned with CamScanner

Scanned with CamScanner

$$\begin{bmatrix} \text{Output} \\ \text{Matrix} \end{bmatrix} = \begin{bmatrix} \text{Transformation} \\ \text{Matrix} \end{bmatrix} \times \begin{bmatrix} \text{Input} \\ \text{Matrix} \end{bmatrix}$$

Translation:



$$\begin{aligned} x' &= x + tx \\ y' &= y + ty \end{aligned}$$

Cartesian Representation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}.$$

Output      Transformation Matrix      Input

$$= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

Output

## Homogeneous Representation:

For n-dimension

we use  $(n+1) \times (n+1)$  Matrix.

~~2D~~ So, for 2D - we use  $3 \times 3$  Matrix

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

*Transformation Matrix*

Input

$$= \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix}$$

Output for 2D

we need  $(4 \times 4)$  Matrix

~~3D~~

$$x' = x + tx$$

$$y' = y + ty$$

$$z' = z + tz$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Output

Transformation Matrix

Input

$$= \begin{bmatrix} x + tx \\ y + ty \\ z + tz \\ 1 \end{bmatrix}$$

Note :

Transformation Matrix for Translation:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & tx \\ 0 & 1 & 0 & \dots & ty \\ 0 & 0 & 1 & \dots & tz \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Scanned with CamScanner

Scanned with CamScanner

Example

Translate point  $(3, 2, 5)$ , 6 step in x-axis.  
 $(-5)$  step in y-axis, 7 step in z-axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 5 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 9 \\ -3 \\ 12 \\ 1 \end{bmatrix}$$

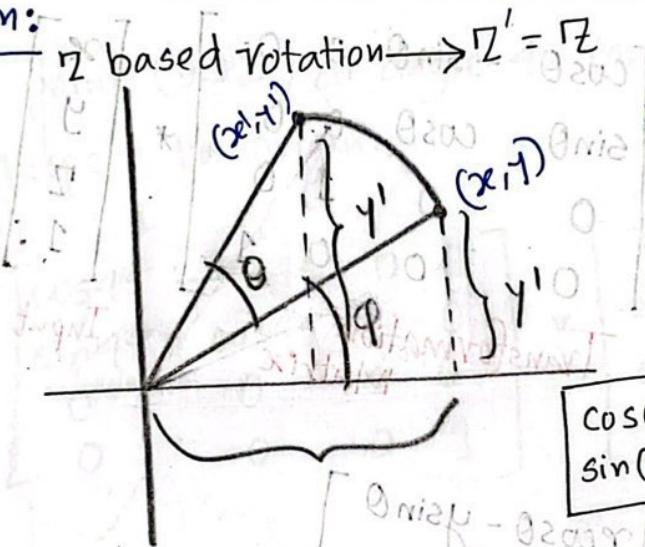
$$\therefore \text{point } (9, -3, 12)$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

Scanned with CamScanner

## Rotation:



We rotate it  $\theta$  amount anti-clockwise.

$$\begin{aligned}\cos(A+B) &= \cos A \cos B - \sin A \sin B \\ \sin(A+B) &= \sin A \cos B + \cos A \sin B\end{aligned}$$

$$\cos \phi = \frac{x}{r} \Rightarrow x = r \cos \phi$$

$$\sin \phi = \frac{y}{r} \Rightarrow y = r \sin \phi$$

$$\cos(\theta + \phi) = \frac{x'}{r} \Rightarrow x' = r \cos(\theta + \phi) = r \cos \theta \cos \phi - r \sin \theta \sin \phi \\ = x \cos \theta - y \sin \theta$$

$$\sin(\theta + \phi) = \frac{y'}{r} \Rightarrow y' = r \sin(\theta + \phi) = r \sin \theta \cos \phi + r \cos \theta \sin \phi \\ = x \sin \theta + y \cos \theta$$

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = r$$

as it is being rotated  $r$  based  
so  $r$  is unchanged.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Output      Transformation Matrix      Input

$$= \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ z \\ 1 \end{bmatrix}$$

$$\Phi_{200317} - \Phi_{200317} = (\theta + \phi) 200$$

$$\Phi_{200317} = \frac{\theta}{\pi} = \frac{\theta}{\pi} = \Phi_{200317}$$

$$\Phi_{200317} - \Phi_{200317} = (\theta + \phi) 200$$

$$\Phi_{200317} + \Phi_{200317} = (\theta + \phi) 200$$

$$\theta_{max} + \theta_{min} =$$

$$\theta_{max} - \theta_{min} = 60$$

$$\theta_{max} + \theta_{min} = 120$$

$$120 = 180$$

Scanned with CamScanner

Scanned with CamScanner

Example

Rotate point  $(9, 2, 5)$  at angle  $45^\circ$  (anti-clockwise)  
with respect to (origin).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 45 & -\sin 45 & 0 & 0 \\ \sin 45 & \cos 45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 9 \\ 2 \\ 5 \\ 1 \end{bmatrix}$$

*Matrix multiplication*

$$= \begin{bmatrix} 4.95 \\ 7.78 \\ 5 \\ 1 \end{bmatrix}$$

*Final answer*

Answer:

$$\begin{bmatrix} 4.95 \\ 7.78 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 4.95 \\ 7.78 \\ 5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4.95 \\ 7.78 \\ 5 \\ 1 \end{bmatrix} =$$

P.T.O

*(2, 3) - Final rotated point*

Scanned with CamScanner

Scanned with CamScanner

~~Example~~

Rotate Point  $(6, 2)$  at an angle  $90^\circ$  anti-clockwise  
with respect to point  $(2, 2)$ ?

Note: When we are given a point to rotate  
with respect to another point, we need  
to translate the point to origin

Sequence: Translate to origin  $\rightarrow$  Rotation  $\rightarrow$  Translate to origin

Output =  $T(2, 2) * R(90) * T(-2, -2) * \text{Input}$

$$= \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos 90 & -\sin 90 \\ \sin 90 & \cos 90 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & +2 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 6 \\ 2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 6 \\ 1 \end{bmatrix}$$

$\therefore$  Rotation Point  $(2, 6)$

Note:

- Output for Rotation with respect to another point:

$$\text{Output} = T(x_2, y_2) * R(\theta) * T(x_1, y_1) * \text{Input}$$

Output =

R Translation  
to  
Original  
Rotation

Translation to  
origin

- For clockwise, we have to use  $(-\theta)$  with the given angle.

$$\begin{bmatrix} x_2, y_2 \\ x_1, y_1 \end{bmatrix} = \begin{bmatrix} s\theta & c\theta \\ c\theta & -s\theta \end{bmatrix} \begin{bmatrix} 0 & 0 & x_2 \\ 0 & 1 & y_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s\theta & c\theta \\ c\theta & -s\theta \end{bmatrix}$$

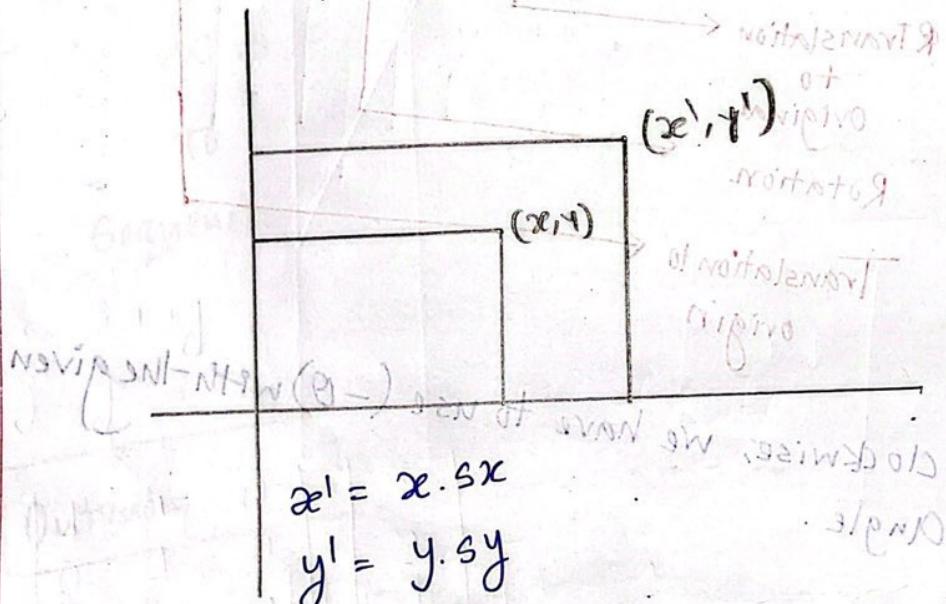
rotation  
matrix

## Scaling:

### Cases:

Scale  $s_x$  units in  $x$  with respect to origin.

Scale  $s_y$  units in  $y$  with respect to origin.



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \\ 1 \end{bmatrix}$$

Scaling Matrix

Example

Scale point  $(3, 5)$   $\begin{cases} 5 \text{ units in } x\text{-axis} \\ 8 \text{ units in } y\text{-axis} \end{cases}$  with respect to origin

Soln:

$$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 15 \\ 40 \\ 1 \end{bmatrix} \quad \text{Ans: } (15, 40)$$

Example

Scale point  $(3, 5)$   $\begin{cases} 5 \text{ units } \rightarrow x \\ 8 \text{ units } \rightarrow y \end{cases}$  with respect to  $(8, 6)$

$$\begin{aligned} \text{Soln: Output} &= T(8, 6) * \text{Scale}(5, 8) * T(-8, -6) * \text{Input} \\ &= \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 5 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -8 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} \end{aligned}$$

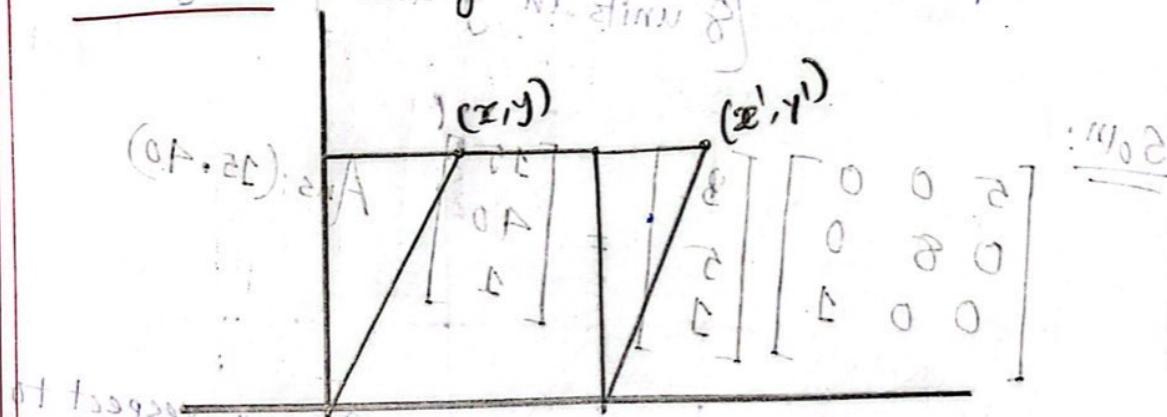
=

Scanned with CamScanner

Scanned with CamScanner

Shearing :-

Case 01 :- Shearing with x-axis (a amount)



$$x' = x + ay$$

$$y' = y$$

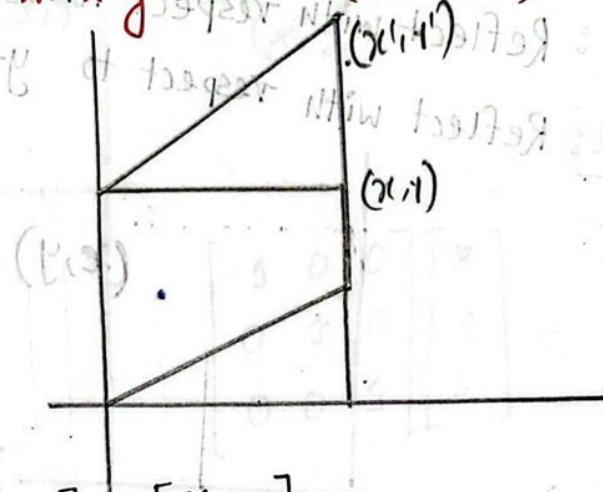
$$\begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+a y \\ y \\ 1 \end{bmatrix}$$

Scanned with CamScanner

Scanned with CamScanner

### Case 02:

Shearing with y-axis. (b amount)



$$x' = x$$

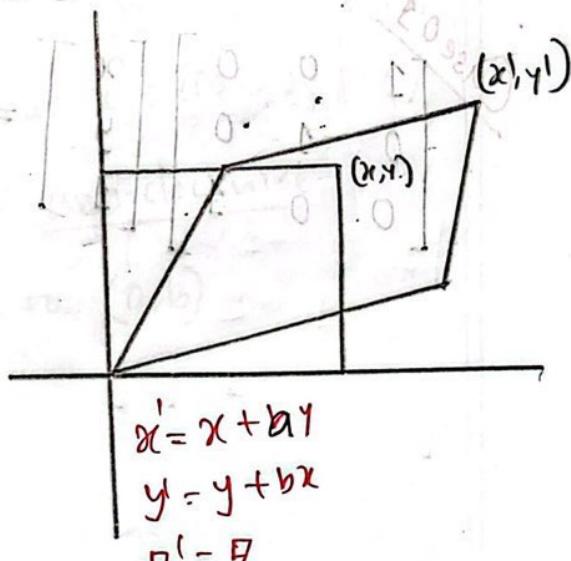
$$y' = y + bx$$

$$\begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y + bx \\ 1 \end{bmatrix}$$

### Case 03:

Shearing with both x and y axis.

$$\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x + ay \\ y + bx \\ z \end{bmatrix}$$



## Reflection:

Case 01: Reflect with respect to x-axis.

Case 02: Reflect with respect to y-axis.

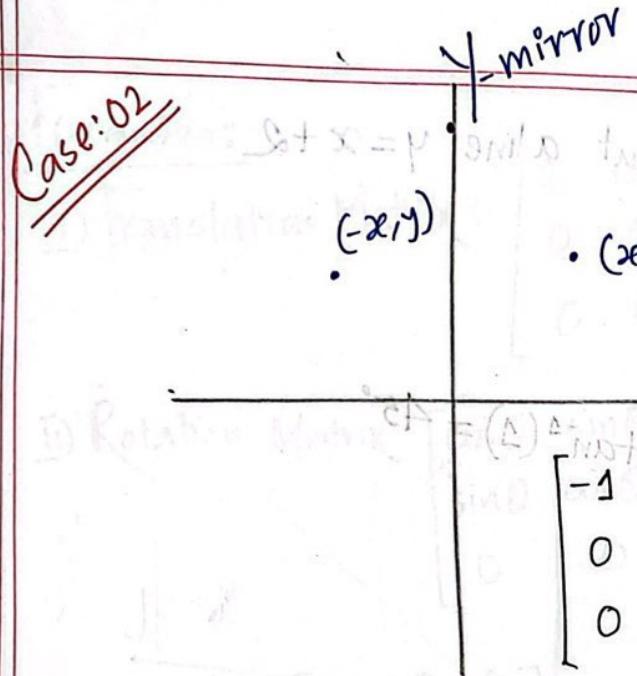
$$\begin{bmatrix} x \\ xd + p \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ p \\ 1 \end{bmatrix} \begin{bmatrix} 0 & d & 0 \\ d & 0 & 0 \end{bmatrix}$$

Case 03

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

$$\begin{bmatrix} x \\ xd + p \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ p \\ 1 \end{bmatrix} \begin{bmatrix} 0 & d & 0 \\ d & 0 & 0 \end{bmatrix}$$

Case: 02



$(x, y)$

$(-x, y)$

$(x, y)$

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

# Case 03: Reflection with respect to a line.

Step 01: Translate  $(a, b)$  to origin.

Step 02: Rotate  $\theta$ -degrees clockwise.

Step 03: Mirror reflect about  $x$ -axis.

Step 04: Rotate  $\theta$ -degree anti-clockwise.

Step 05: Translate Origin to  $(a, b)$ .

Scanned with CamScanner

Scanned with CamScanner

\* Reflect (10, 5) about a line  $y = x + 2$

Soln:

$$y = mx + c$$

$$\begin{bmatrix} y \\ x \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \tan^{-1}(1) = 45^\circ$$

$$m = 1$$

$$c = 2$$

(i)  $T(0, 2)$

(ii)  $R(-45^\circ)$

(iii) Ref(x)

(iv)  $R(45^\circ)$

(v)  $T(0, 12)$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-45) & -\sin(-45) & 0 \\ \sin(-45) & \cos(-45) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$T(0, 2)$                      $R(\theta)$                     Ref(x)                     $R(-45^\circ)$

$$= \begin{bmatrix} 3 \\ 12 \\ 1 \end{bmatrix} \Rightarrow (3, 12)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix}$$

$T(0, -2)$

Scanned with CamScanner

Scanned with CamScanner

## Formulas:

i) Translation Matrix:

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

ii) Rotation Matrix

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

iii) Scaling

$$\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

iv) Shearing

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

x-axis

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

y-axis

$$\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

both axis

v) Reflection:

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

x-axis

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

y-axis

$T(0, b)$

$Re(\theta)$

$Ref$

$Re(\theta)$

$T(0, -b)$

## Color Models in Computer Graphics

Color is the aspect of things that is caused by differentiating qualities of light being reflected or emitted by them.

Ques: Write the differences between Additive and Subtractive color model.

Additive Color Model	Subtractive Color Model.
RGB, HSV, HSL	CMY, CMYK
Generates color by adding multiple colors	Generates color by subtracting different colors.
Active Display → TV, Mobile, PC	Device that deposit color → printer, copiers.
No data → Black	No data → White
Increase Brightness	Decrease Brightness.

Scanned with CamScanner

Scanned with CamScanner

## Primary color of light (RGB)

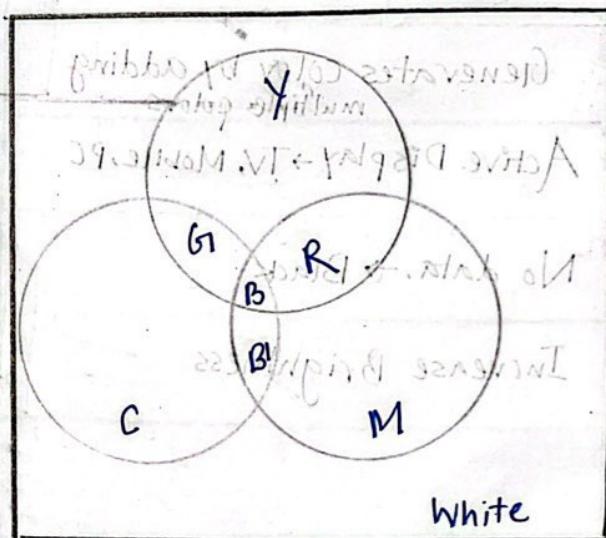
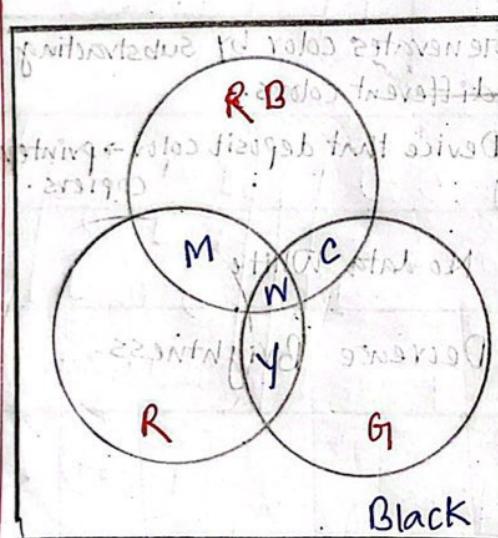
Red (R), Green (G), Blue (B)

## Secondary Colors of light (CMY)

Cyan (C) → Green + Blue

Magenta → Red + Blue

Yellow → Red + Green

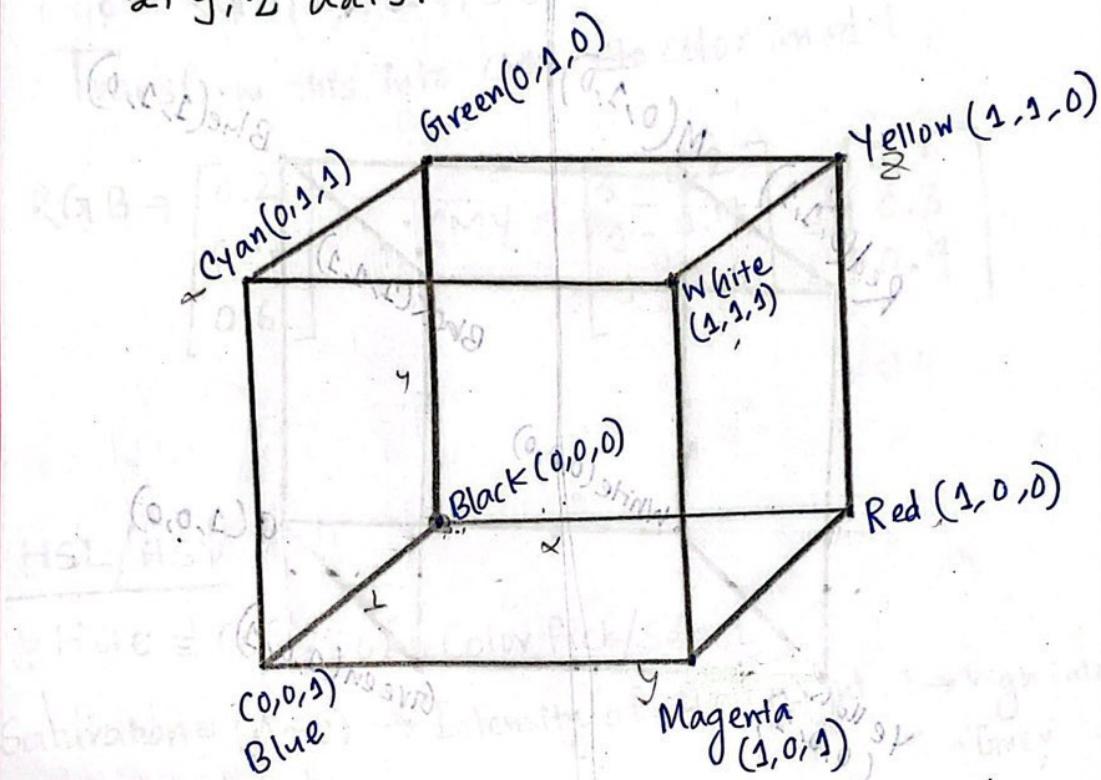


RGB (1, 0, 0) → Red Range (0 - 1)

RGB (1, 1, 0) → Yellow

## Color Cube

x, y, z axis.



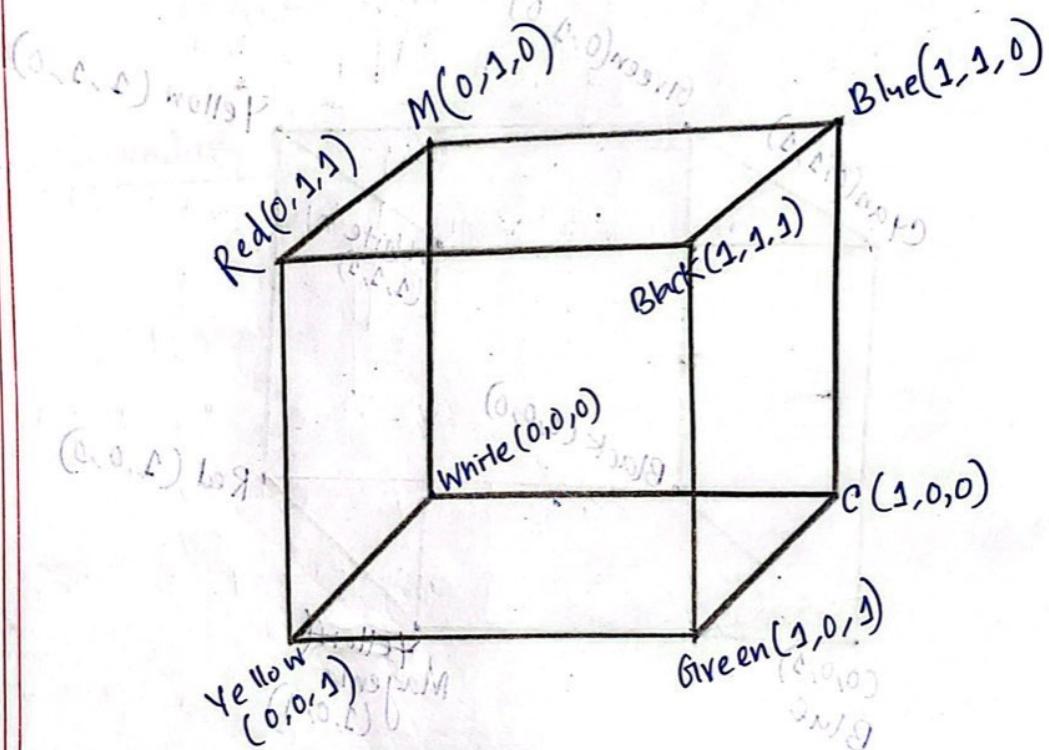
Total Combination that can be generated =  $2^8 \times 2^8 \times 2^8$

= 16 million.

Scanned with CamScanner

Scanned with CamScanner

## CMY Color Cube



$\text{RGB} \rightarrow \text{CMY}$  (Switching between RGB and CMY is just switching into T)

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

\* Switching between RGB and CMY

## Example

Let RGB  $(0.2, 0.7, 0.6)$

Transform this into CMY color model.

$$\text{RGB} = \begin{bmatrix} 0.2 \\ 0.7 \\ 0.6 \end{bmatrix} \quad \therefore \text{CMY} = \begin{bmatrix} 1 - 0.2 \\ 1 - 0.7 \\ 1 - 0.6 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.3 \\ 0.4 \end{bmatrix}$$

HSL/HSV

Hue  $\equiv (0 - 360)^\circ$  → Color Pick>Select.

Saturation  $\equiv (0 \sim 1)$  → Intensity of Color/Light  $1 \rightarrow$  high intensity  
 $0 \rightarrow$  Grey

Lightness  $(0 \sim 1) \rightarrow$  Background Light  $1 \rightarrow$  White  
 $0 \rightarrow$  Black

Hue  $\equiv (0 \sim 360^\circ)$

Saturation  $\equiv (0 \sim 1)$

Value  $\equiv (0 \sim 1) \rightarrow$  Brightness.

\* The human eye can see 128 different hues, 130 different saturations, and number values between 16 (blue) and 23 (yellow)

Red  $\rightarrow (0 - 60^\circ)$

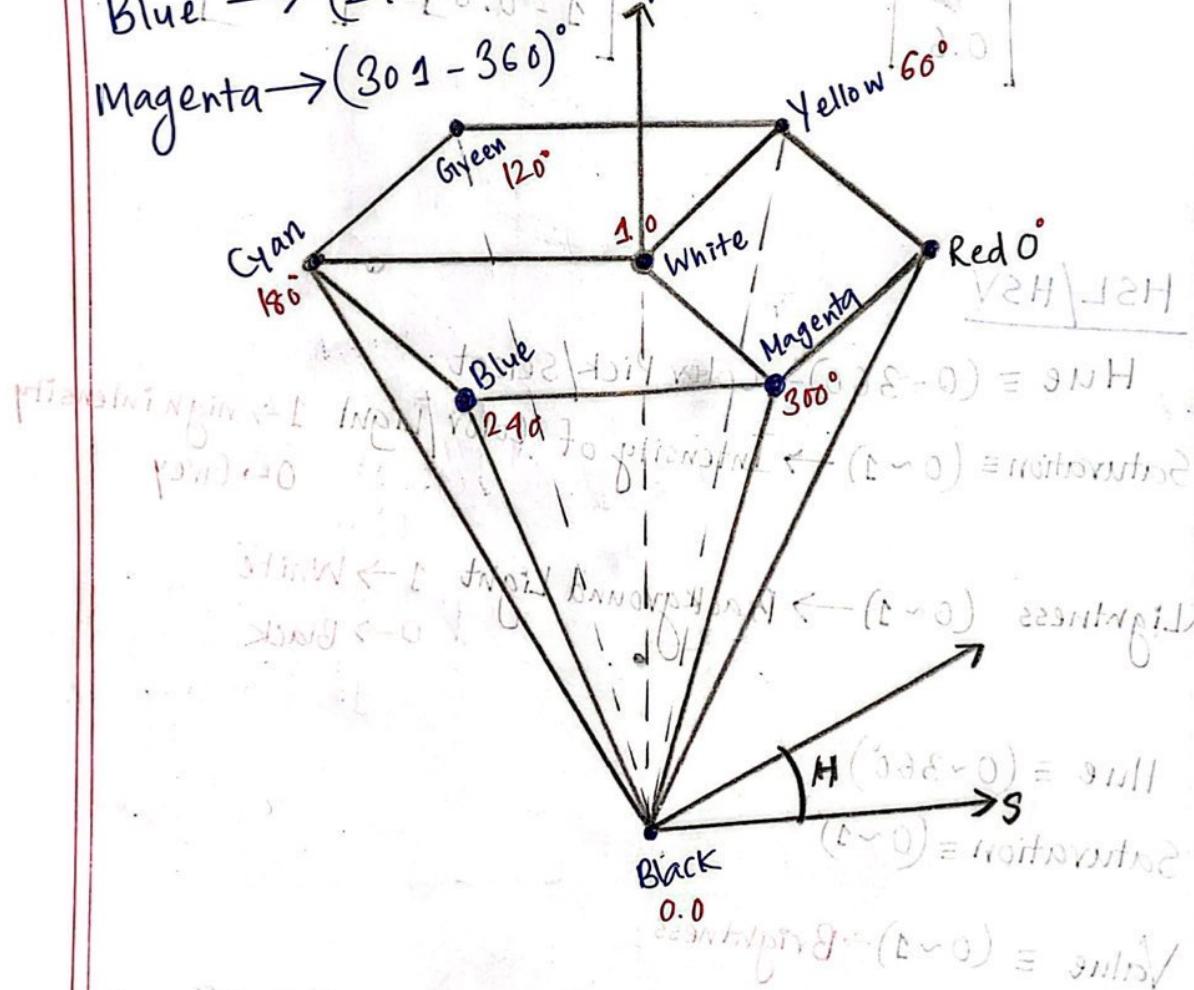
Yellow  $\rightarrow (60 - 120^\circ)$

Green  $\rightarrow (120 - 180^\circ)$

Cyan  $\rightarrow (180 - 240^\circ)$

Blue  $\rightarrow (240 - 300^\circ)$

Magenta  $\rightarrow (300 - 360^\circ)$



Scanned with CamScanner

Scanned with CamScanner

## RGB to HSL/HSV

80%

### Step 01:

Find out Maximum and Minimum Value from RGB

Values:  $\text{Max} = \max(R, G, B)$

$$\text{Max} = \max(R, G, B)$$

$$\text{Min} = \min(R, G, B)$$

$$l = \frac{\text{Max} - \text{Min}}{\text{Max}}$$

### Step 02:

Find out Lightness and Saturation for HSL

& Find out Value and Saturation for HSV.

$\text{HSL} \rightarrow \text{Lightness: } \frac{\text{Max} + \text{Min}}{2}$ $\text{HSL} \rightarrow \text{Saturation: } \frac{l}{1 -  L - 1 } \text{ here } L = \text{Lightness}$
$\text{HSV} \rightarrow \text{Saturation: } \frac{l}{\text{Max}}$ $\text{HSV} \rightarrow \text{Value: Max (Max Lightness)}$

### Step 03:

Calculate Hue

if  $R = \text{Max}$ :

$$H = \left( \frac{G - B}{l} \right) * 60^\circ \quad \begin{cases} \text{if } H < 0^\circ \text{ then} \\ \quad H = H + 360^\circ \end{cases}$$

if  $G = \text{Max}$ :

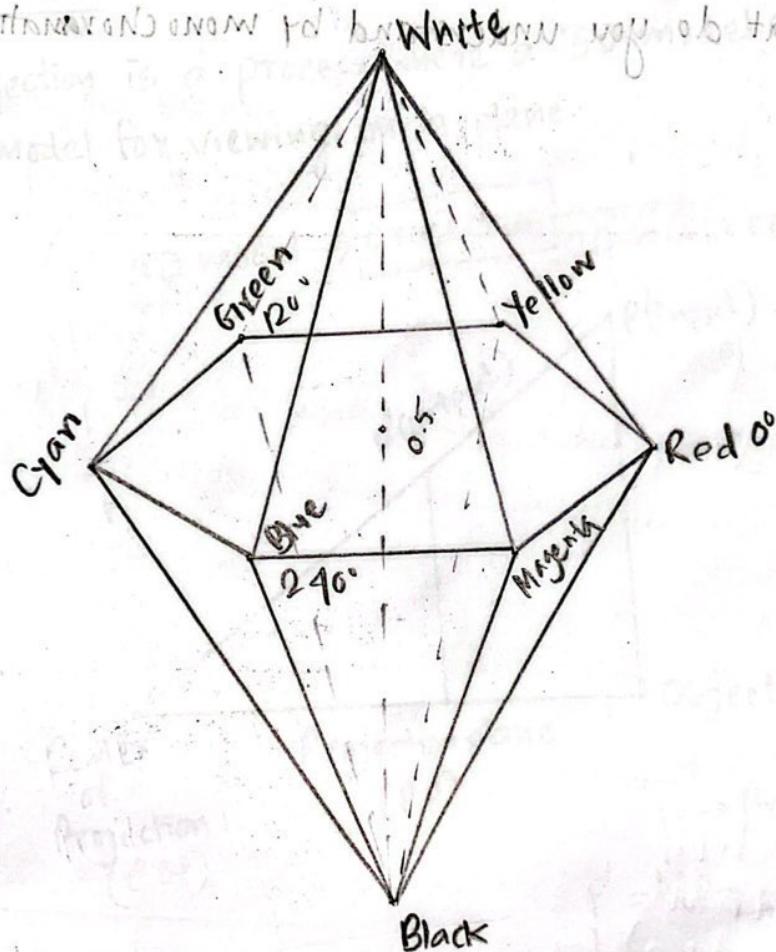
$$H = \left( \frac{B - R}{l} \right) * 60^\circ + 120^\circ \quad \begin{cases} \text{By default} \\ \text{add } 120^\circ \end{cases}$$

if  $B = \text{Max}$ :

$$H = \left( \frac{R - G}{l} \right) * 60^\circ + 240^\circ$$

## HSL Color Model

10/10/2019



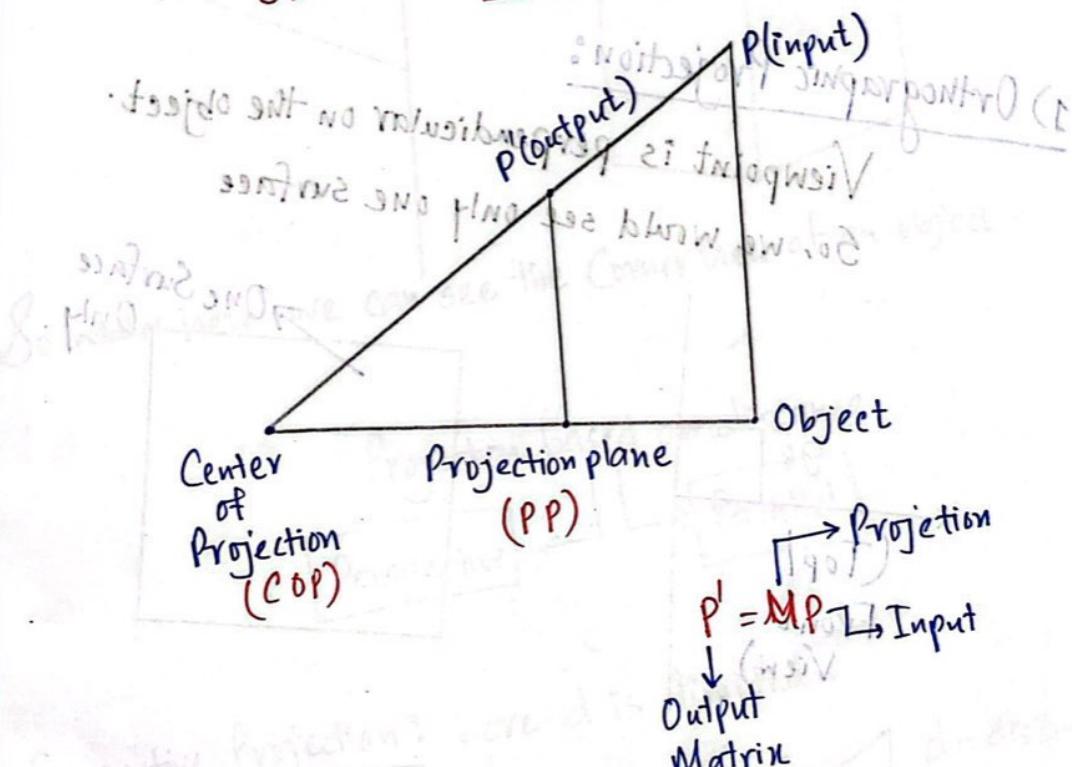
Scanned with CamScanner

Scanned with CamScanner

## Projection

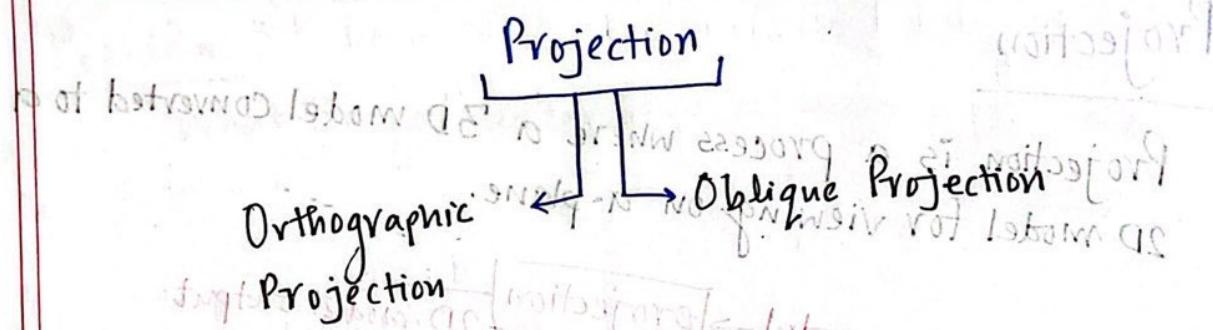
Projection is a process where a 3D model converted to a 2D model for viewing on a plane.

3D model  $\rightarrow$  projection  $\rightarrow$  2D model output.



Scanned with CamScanner

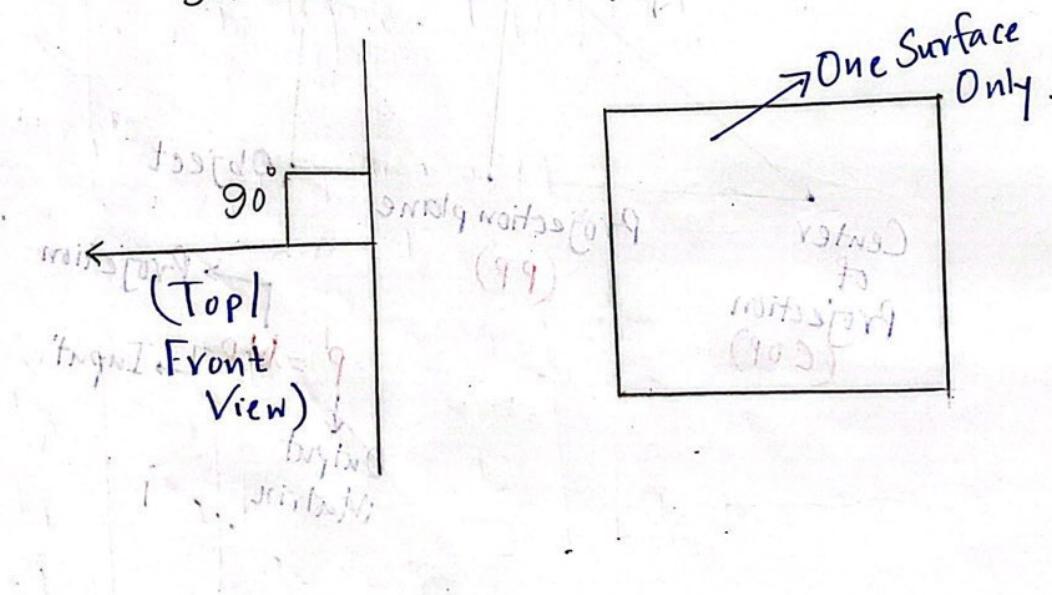
Scanned with CamScanner



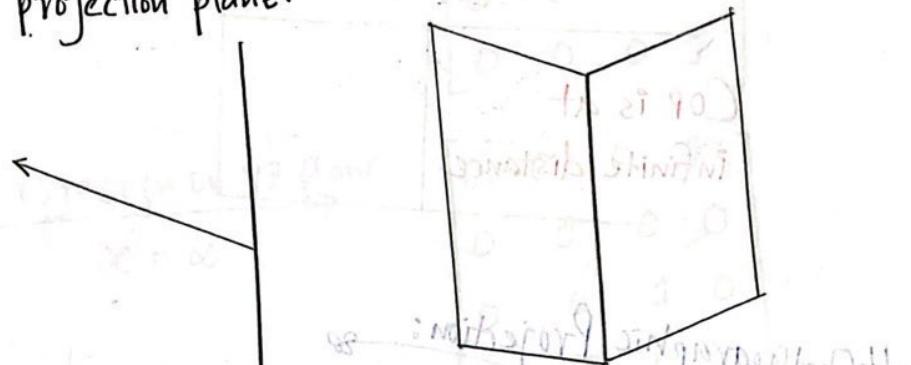
### 1) Orthographic Projection:

Viewpoint is perpendicular on the object.

So, we would see only one surface



2) Oblique Projection: An oblique projection is a parallel projection in which the lines of sight are not perpendicular to the projection plane.



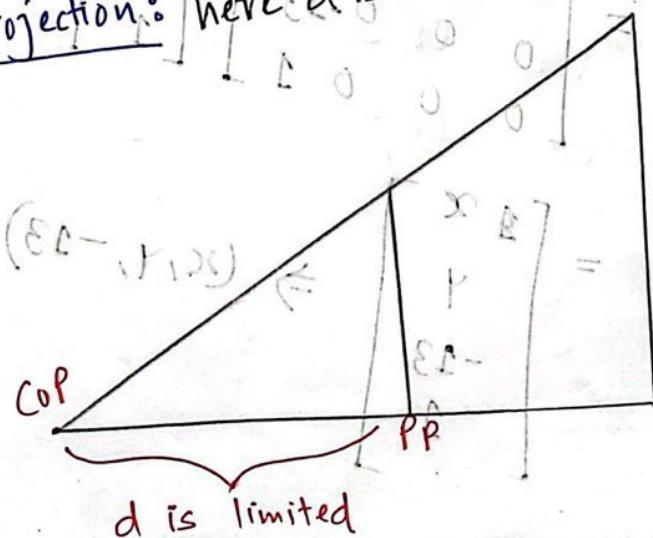
So here we can see the corner view of an object.

(Eg - Pictorial drawing)  
Projection based on distance

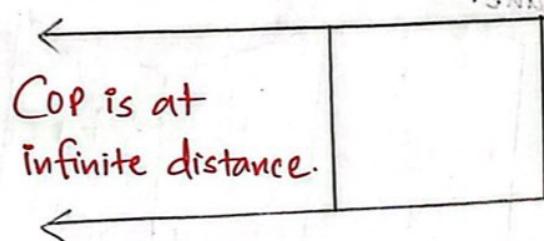
Perspective      Parallel

3) Perspective Projection: here  $d$  is limited.

$d$  = distance between C.O.P and P.P



ii) Parallel Projection: d is infinite. Example:



COP is at  
infinite distance.

# Orthographic Projection:

We have to figure out projection matrix.

Example

$$(x, y, z) \xrightarrow{\text{projection on XY plane}} (x, y, -13)$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -13 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x \\ y \\ -13 \\ 1 \end{bmatrix} \Rightarrow (x, y, -13)$$

Final

Scanned with CamScanner

Scanned with CamScanner

Formula:

$(x, y, z)$  projection on xy plane  
 $z = \alpha$

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

$$qM = q$$

$(x, y, z)$  projection on yz plane  
 $x \neq \alpha$

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

$$L = M$$

$(x, y, z)$  projection on zx plane  
 $y = \alpha$

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

$$x \cdot 99 - x \cdot 900 = xb$$

$$b \cdot 99 - b \cdot 900 = cb$$

$$5 \cdot 99 - 5 \cdot 900 = sb$$

## General Projection Matrix

for zey plane

$$p' = mp$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{dx}{dz} & \left(\frac{dx}{dz}\right)pp.z \\ -\frac{dy}{dz} & \left(\frac{dy}{dz}\right)pp.z \\ -\frac{pp.z}{dz} & pp.z\left(1 + \frac{pp.z}{dz}\right) \\ -\frac{1}{dz} & 1 + \frac{pp.z}{dz} \end{bmatrix}$$

$$dx = cop.x - pp.x$$

$$dy = cop.y - pp.y$$

$$dz = cop.z - pp.z$$

$cop.x$  = center of projection  
at x axis

$pp.x$  = projection plane  
at x coordinate.

# Given COP(50, 40, 100) and PP(0, 0, -200) Find out  
the projected point for the given input (30, 50, -250)

$$dx = COP_x - PP_x = 50 - 0 = 50$$

$$dy = COP_y - PP_y = 40 - 0 = 40$$

$$dz = COP_z - PP_z = 100 - (-200) = 300$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{50}{300} & \frac{50}{300}(-250) \\ 0 & 1 & -\frac{40}{300} & \frac{40}{300}(-250) \\ 0 & 0 & -\frac{(-200)}{300} & -200\left(1 + \frac{-200}{300}\right) \\ 0 & 0 & \frac{-1}{300} & 1 + \frac{-200}{300} \end{bmatrix} \begin{bmatrix} 30 \\ 50 \\ -250 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 115/3 \\ 170/3 \\ 750/3 \\ 1.167 \end{bmatrix} \xrightarrow{\text{Normalized}} \begin{bmatrix} 32.85 \\ 46.5 \\ -250 \\ 1 \end{bmatrix}$$

As this result is  
not in 1, so we  
have to normalize  
it

Case 03

Find out the matrix where COP is at origin and Projection Plane is at  $d$  distance from COP. ( $x-y$ -plane)

COP  $(0, 0, 0)$

PP  $(0, 0, d)$

$$\begin{aligned} dx &= 0 \\ dy &= 0 \\ dz &= 0 - d = -d \end{aligned}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projection Matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{z}{d} \\ 1 \end{bmatrix}$$

$\Rightarrow d(1 + \frac{z}{d}) = d(x(1 - 1)) = 0$

$1 + \frac{z}{d} = 1 - 1 = 0$

Case 02:

Find out the matrix where PP is at origin

and COP is at d-distance from PP

$$COP(0, 0, d)$$

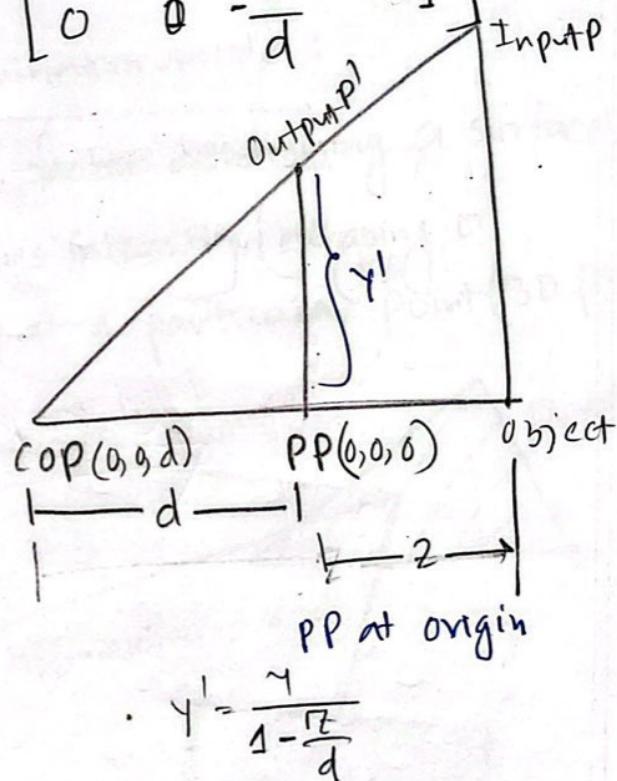
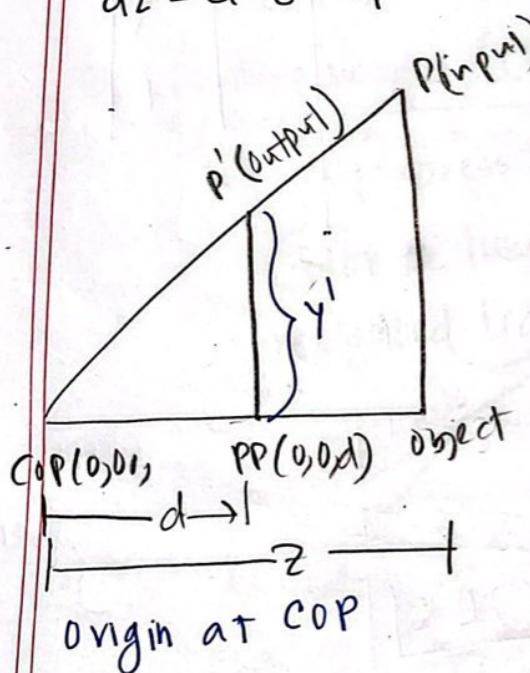
$$PP(0, 0, 0)$$

$$dx = 0$$

$$dy = 0$$

$$dz = d - 0 = d$$

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



$$y' = \frac{y}{1 - \frac{dz}{d}}$$

# Lighting / Illumination / Reflection Model

## ④ Illumination:

The transport of energy from light sources to surfaces & points.

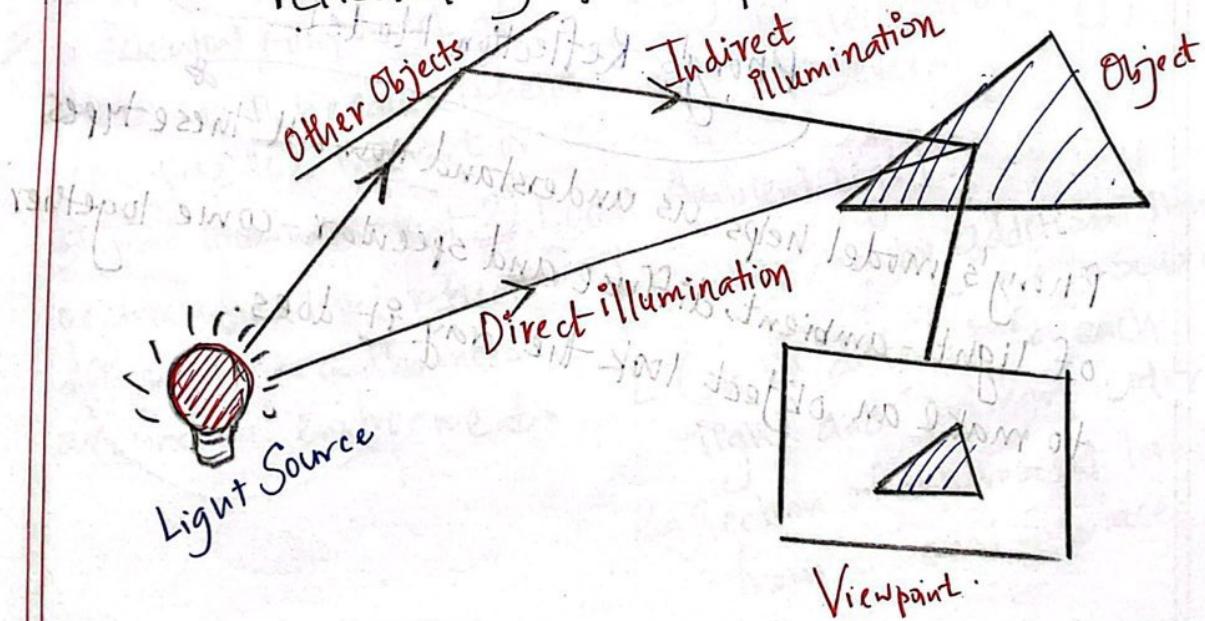
→ Local illumination

→ Global illumination

## ⑤ Lighting model / Illumination model:

Express the factors determining a surface's

Color or luminous intensity (outgoing or reflected light) at a particular point (3D point)



## Components of Illumination:

1. Light source

2. Surface Properties

$I = \text{luminance}/\text{Intensity}$ . Each color is described separately.

$$I = I_r I_g I_b$$

## Type of Light Source:

1. Ambient Light. [Environment Light]

2. Diffuse Light [Main Light]

3. Spot Light: [ViewPoint - Object shines when light is upon it]

Phong's Reflection Model.

Phong's model helps us understand how all these types of light - ambient, diffuse and specular - come together to make an object look the way it does.

$$I = I_a + I_d + I_s$$

↓      ↓      ↓  
 Final      Ambient      Diffuse      Specular

## Ambient Light

- No identifiable Source or direction
- Product of multiple reflections of light from the many surfaces present in the environment

### Categories:-

- 1) Global Ambient Light
  - Independent of Light Source
  - Lights entire scene.
  - Reflection of Light from Several
- 2) Local Ambient Light
  - Contributed by additional light source
  - Can be different for each light and primary color.
  - Reflection of fluorescent lamps from several surfaces.

## Ambient Reflection

\* \* the reflection of light that doesn't come directly from a light source.

\* \* As phong is a direct lighting model, we assume that ambient light falls ~~uniformly~~ ~~equally~~ on all objects.

$$A = I_a k_a$$

here,  $I_a$  = Intensity of the ambient light

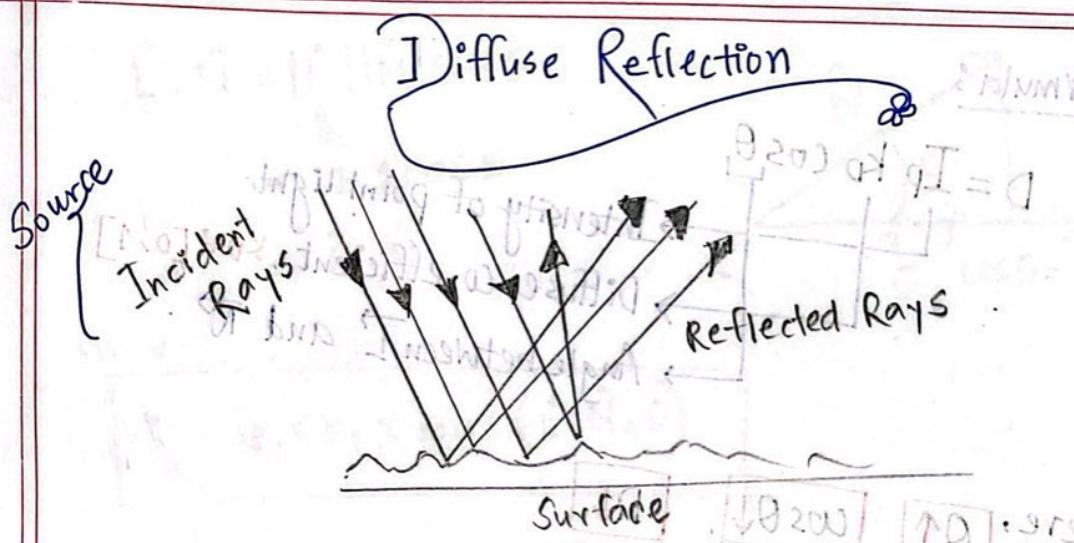
\* Ambient Reflection Co-efficient:-  $k_a$  = Ambient co-efficient.

Moving from left to right, the  
Ambient.

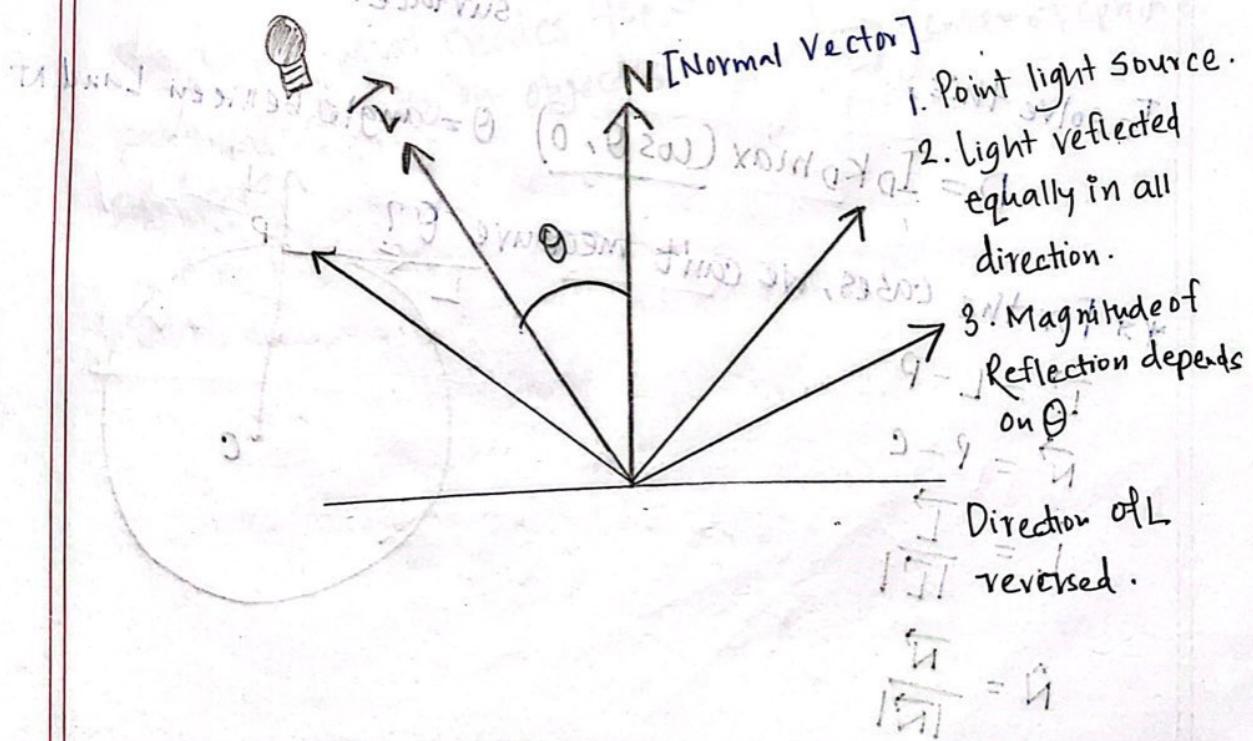
$$k_a \in [0, 1]$$

Nightime scene Bright scene

Ambient light $\downarrow$	Shadows too deep and harsh.
Ambient Light $\uparrow$	Picture looks washed out and blend.



- \* Parallel rays are from light source.
- \* Reflected rays are scattered in many angles.
- \* Diffuse reflection is calculated based on the structure of the surface itself.



Scanned with CamScanner

Scanned with CamScanner

Formula:

$$D = I_p k_d \cos \theta$$

→ Intensity of point light

→ Diffuse co-efficient.  $k_d \in [0, 1]$

→ Angle between  $\vec{L}'$  and  $\vec{N}$

here,

$$\Theta \uparrow$$

$$\cos \theta \downarrow$$

$$D \downarrow$$

$$\theta = 90^\circ \quad \cos \theta = 0 \quad D = 0$$

$$\text{So, } \theta > 90^\circ, \cos \theta < 0 \therefore \cos \theta = -\text{ve} \quad D = -\text{ve}$$

We need to solve this as  
no light is reflected if the  
light source is behind the  
surface.

To solve this,

$$D = I_p k_d \max(\cos \theta, 0) \quad \theta = \text{angle between } \vec{L} \text{ and } \vec{N}$$

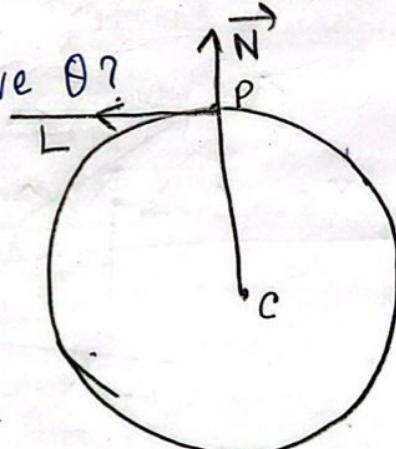
\* For the cases, we can't measure  $\theta$ ?

$$\vec{L}' = \vec{L} - \vec{P}$$

$$\vec{N} = \vec{P} - \vec{C}$$

$$\vec{L}' = \frac{\vec{L}}{|\vec{L}'|}$$

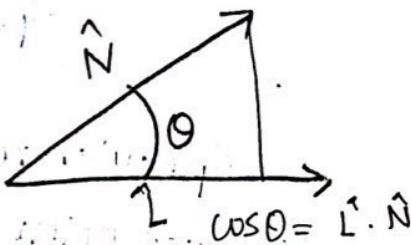
$$\hat{N} = \frac{\vec{N}}{|\vec{N}|}$$



$$\hat{L} \cdot \hat{N} = |\hat{L}| |\hat{N}| \cos \theta$$

$$= 1 \cdot \cos \theta$$

$$= \hat{L} \cdot \hat{N}$$



$$D = I_p \times K_p \times \max(\hat{L} \cdot \hat{N}, 0)$$

\* Diffuse light is a component that simulates how a surface reflects light in a scattered or diffuse manner. Diffuse reflection is what creates the basic shading of an object's surface.

$$\vec{L} = \vec{L} - \vec{P} \rightarrow \text{illumination point}$$

↳ light source

$$\vec{N} = \vec{P} - \vec{C}$$

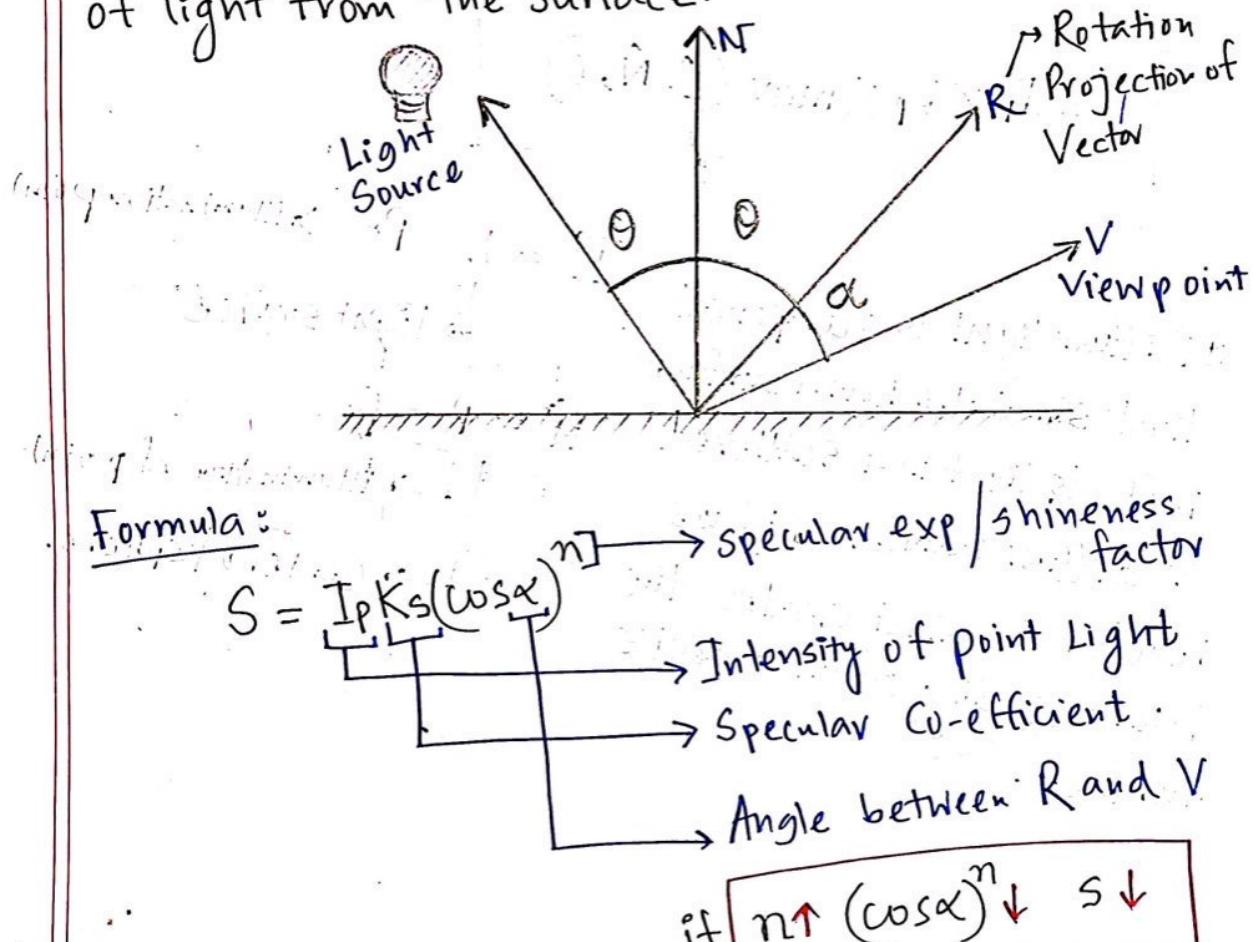
↳ illumination ref point

↳ Center of sphere.

## Specular Reflection

Shineness - Dependent  
Viewpoint.

Specular reflection is a type of surface reflectance often described as a mirror-like reflection of light from the surface.



Formula:

$$S = I_p K_s (\cos \alpha)^n$$

→ Specular exp / shininess factor  
 → Intensity of point Light.  
 → Specular Co-efficient.  
 → Angle between R and V  
 if  $n \uparrow (\cos \alpha)^n \downarrow S \downarrow$

Modified Formula:

$$\begin{aligned} S &= I_p K_s (\hat{R} \cdot \hat{V})^n \\ &= I_p K_s \{\max(\hat{R} \cdot \hat{V}, 0)\}^n \end{aligned}$$

## Final Formula for Phong's Reflection Model:

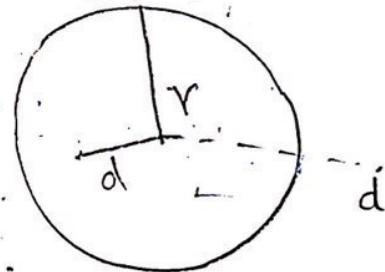
$$I = I_a k_a + I_p k_s \max(L \cdot N, 0) + I_p k_s \left\{ \max(R \cdot V, 0) \right\}^n$$

### Attenuation

Attenuation is the loss of light energy over space.

- \* Attenuation is accounted for by the variable  $f_{att}$  and applied to diffuse and specular components.
- \* Follows inverse square law.

$$f_{att} = \frac{1}{d^2} \quad d = \text{distance of object}$$



As it removes too much light,  
Phong's model uses,

$$f_{att} = 1 - \left( \frac{d}{r} \right)^2$$

if point d is outside the radius,

$$\begin{aligned} d &> r \\ \Rightarrow \frac{d}{r} &> 1 \quad \therefore \left( \frac{d}{r} \right)^2 > 1 \end{aligned}$$

Final Formula:

$$f_{att} = \text{Max}\left(1 - \left(\frac{d}{r}\right)^n, 0\right)$$

Updated Formula for Phong's Reflection Model:

$$I = I_{ak} + I_p f_{att} \left( K_d \max(\hat{L} \cdot \hat{N}, 0), K_s \left( \max(\hat{V} \cdot \hat{R}, 0) \right)^n \right)$$

For Multiple Light Sources.

$$I = I_{ak} + \sum_{i=1}^m I_p f_{att} \left( K_d \max(\hat{L}_i \cdot \hat{N}, 0) + K_s \left( \max(\hat{V} \cdot \hat{R}_i, 0) \right)^n \right)$$

For each light source-

$I_p$  is,  $L, R$  is multiple

Answer to the ques no. 01 is given in Q&A page

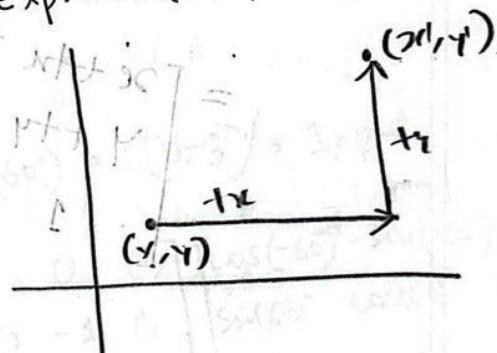
1(a)

Homogeneous co-ordinates is better fit for use in translation instead of Cartesian co-ordinates.

Initially, translation is expressed as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} +x \\ +y \end{bmatrix}$$

$$= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} +x \\ +y \end{bmatrix}$$



$$t_{xy} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x + +x \\ y + +y \end{bmatrix}$$

Composition is difficult to express since translation is not expressed as a matrix multiplication. But homogeneous co-ordinates allow all three to be expressed homogeneously using multiplication by  $3 \times 3$  matrices and  $W$  is 1 for affine transformation in graphics.

P.T.O

Suppose, for 2D, we use a  $3 \times 3$  matrix.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} tx \\ ty \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} tx \\ ty \\ 0 \end{bmatrix}$$

1(b) Input points =  $(5, 6)$

Rotation Angle  $60^\circ$  & Clockwise.

Rotation with respect to  $(6, 6)$

$$\text{Output} = T(6, 6) * R(-60^\circ) * T(-6, -6) * \text{Input}$$

$$= \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-60^\circ) & \sin(-60^\circ) & 0 \\ \sin(-60^\circ) & \cos(-60^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -6 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 5.5 & 6.866 \\ 6.866 & 1 \end{bmatrix}$$

Answer:  $(5.5, 6.866)$

1(c)

Lab M value

Given line, travel distance between origin & point A is 8

$$y = \sqrt{3}x + 3 \text{ has slope } m = \sqrt{3}, \theta = \tan^{-1}(\sqrt{3}) = 60^\circ$$

slope = 3.

Output:

$$\begin{aligned} & T(0, 3) * R(-60^\circ) * \text{Ref}(x) * R(60^\circ) * T(0, -3) * \text{Input} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ & 0 \\ \sin 60^\circ & \cos 60^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(-60^\circ) & -\sin(-60^\circ) & 0 \\ \sin(-60^\circ) & \cos(-60^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 5.5 \\ 6.5 \\ 1 \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} 4.6961 \\ 8.598 \\ 1 \end{bmatrix}$$

(Ansatz)  $x_{AB} = 5.5V$

(Ansatz)  $y_{AB} = 6.5V$

$x_{AB} - y_{AB} = 1$

$x_{AB} = 5.5V$

Scanned with CamScanner

Scanned with CamScanner

## Color Model

3(a) Leonardo Da Vinci used different colors to complete his picture. So, for the final picture in this scenario, he used RGB model.

Michael Angelo snapped a picture on his phone to see it on his screen. For this, RGB model was used as well.

3(b) Algorithm to convert RGB to HSV

Step 01: Find Min and Max value from RGB

Values.

$$\text{Max} = \max(R, G, B)$$

$$\text{Min} = \min(R, G, B)$$

$$l = \text{Max} - \text{Min}$$

Step 02: Saturation =  $\frac{l}{\text{Max}}$

$$\text{Value} = \text{Max}$$

$$\underline{\text{Hue}}: \text{H.S.O} = \frac{S.O}{O.O} = \frac{l}{N.M} = \text{without } \checkmark$$

if  $R == \text{Max}$ :

$$H = \left( \frac{G - B}{l} \right) * 60^\circ \quad \begin{bmatrix} \text{if } H < 0^\circ \\ H = H + 360^\circ \end{bmatrix}$$

if  $G == \text{Max}$ :

$$H = \left( \frac{B - R}{l} \right) * 60^\circ + 120^\circ$$

if  $B == \text{Max}$ :

$$H = \left( \frac{R - G}{l} \right) * 60^\circ + 240^\circ$$

3(c)

$$C = 0.4$$

$$M = 0.2$$

$$Y = 0.3$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 - 0.4 \\ 1 - 0.2 \\ 1 - 0.3 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0.7 \end{bmatrix}$$

$$\text{Max} = G = 0.8$$

$$\text{Min} = R = 0.6$$

$$l = (0.8 - 0.6) = 0.2$$

$$\text{Saturation} = \frac{l}{\text{Max}} = \frac{0.2}{0.6} = 0.25 \quad \underline{\underline{\text{Value}}}$$

$$\text{Value} = \text{Max} = 0.8 \quad \left[ \begin{array}{l} l > H \Rightarrow \\ d = \frac{(D-H)}{2} = H \end{array} \right]$$

Hue:

$$\text{here, } b_1 = \max \left( 0.8, 0.2 + \frac{(D-H)}{2} \right) = H$$

$$\therefore \text{Hue} = \left( \frac{0.7 - 0.6}{0.2} \right) * 60 + 120$$

$$= 150$$

$$0.8 = X_{NM}$$

$$0.8 = R = X_{NM}$$

$$0.8 - 0.01 = L$$

$$\begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.0 - 0.0 \\ 1.0 - 0.0 \\ 1.0 - 0.0 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

## Projection:

3(a)

Parallel Projection	Perspective Projection
Parallel Projection represents object in a different way.	Represents in 3D way.
No different effects.	Objects that are far appear smaller, objects that are near appear bigger.
distance of object from COP is infinite	distance of object from COP is finite.
Gives accurate projection of object.	Cannot give accurate projection.
Cannot give Realistic View	Realistic view of object

3(b)

Mr. Joy Could not take Jaguar's full picture because of ~~parallel~~ projection.

~~Perspective~~

As the jaguar was nearer, it filled up more of frames and as there was no proper distance between the jaguar and the camera so the full picture could not be clicked.

3(c)

$$\text{PP}(0, 0, 0)$$

$$\text{COP}(0, 0, 175)$$

$$\text{Given points } (35, 60, -350)$$

$$dx = \frac{\text{COP.x} - \text{PP.x}}{\text{PP.z}}$$

$$dy = \frac{\text{COP.y} - \text{PP.y}}{\text{PP.z}}$$

$$dz = \frac{\text{COP.z} - \text{PP.z}}{\text{PP.z}}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 175 \end{bmatrix} \begin{bmatrix} 35 \\ 60 \\ -350 \\ 175 \end{bmatrix}$$

$$= \begin{bmatrix} 35 \\ 60 \\ 0 \\ -517 \end{bmatrix} \xrightarrow{\text{Normalize}} \begin{bmatrix} -49 \\ -84 \\ 0 \\ 11 \end{bmatrix}$$

label principal  
(a)

Planned to find initial principal

vector of final value

initial value is  $(0, (r^k)-1) \times 1 = \hat{h}_1$

softmax function is stronger b  
softmax function is more complex  
softmax function is to find maximum

0.95

$$\begin{aligned} I - A(I, D) &= \hat{A} \\ (\hat{A}, A, \hat{A}) &= 1 \\ 2, A, 8 &= 1 \\ \hat{A} &= 1 \end{aligned}$$

Scanned with CamScanner

Scanned with CamScanner

# Final Formula Sheet for CSE423

## Transformation

### 1) Translation:

$$x' = x + tx$$

$$y' = y + ty$$

$$z' = z + tz$$

Matrix:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & tx \\ 0 & 1 & 0 & \dots & ty \\ 0 & 0 & 1 & \dots & tz \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

### 2) Rotation:

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

Matrix:

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \text{for 3D.}$$

$\theta$  will be (-ve) for clockwise

### 3) Scaling:

$$x' = s_x \cdot x$$

$$y' = s_y \cdot y$$

Matrix:

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

for xaxis

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

### 4) Shearing:

no rotation

#### \* Case 01: shear with x-axis

$$x' = x + ay$$

$$y' = y$$

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

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

rotation (1)

$b = c$

$d = f$

$e = g$

$h = i$

#### \* Case 02: shear with y-axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{shear}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & b & 1 \end{bmatrix}$$

$$x' = x + by$$

$$y' = y$$

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

rotation (2)

$a = d$

$c = f$

#### Case 03: shear with x-axis and y-axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{shear}} \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x + ay$$

$$y' = y + bx$$

$$\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

rotation (3)

$a = d$

$c = f$

## 5) Reflection:

Case 01: Reflect with respect to x-axis.

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

Case 02: Reflect with respect to y-axis.

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

Case 03: Reflection with respect to a line.

1) figure out m, c and θ from m.

2) Translate (a, b) to origin

3) Rotate clockwise

4) Mirror Reflect

5) Rotate anti-clockwise

6) Translate origin back to (a, b)

Output =  $T(0, b) * R(\theta) * \text{Ref}(x/y) + R(-\theta) * T(0, -b)$ , Input.

## Color Models

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

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Primary: R, G, B

Secondary: Cyan  $\rightarrow$  Red + Green + Blue

Magenta  $\rightarrow$  Red + Blue

Yellow  $\rightarrow$  Red + Green

RGB to HSL / HSV

$$\text{Max} = \max(R, G, B)$$

$$\text{Min} = \min(R, G, B)$$

$$L = \frac{\text{Max} - \text{Min}}{\text{Max}}$$

$$\text{HSL: Saturation} = \frac{\text{Max} + \text{Min}}{2} \quad \text{Lightness} = \frac{1}{1 + 2L - 1}$$

Scanned with CamScanner

Scanned with CamScanner

### HSV:

$$\text{Saturation} = \frac{l}{\text{Max}}$$

$$\text{Value} = \text{Max}$$

### Hue:

if  $R == \text{Max}$ :

$$\text{Hue} = \left( \frac{G - B}{l} \right) * 60$$

if  $G == \text{Max}$ :

$$\text{Hue} = \left( \frac{B - R}{l} \right) * 1260 + 120$$

if  $B == \text{Max}$ :

$$\text{Hue} = \left( \frac{R - G}{l} \right) * 60 + 240$$

## Projection:

### Orthographic Projection:

$(x, y, z)$   $\xrightarrow[\text{z}=\infty]{\text{Projection on XY-plane}}$

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

$\text{VNH} = \text{WVH}$

$(x, y, z)$   $\xrightarrow[\text{z}=\infty]{\text{Projection on YZ-plane}}$

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

$\text{VNM} = \text{WVH}$

$(x, y, z)$   $\xrightarrow[y=\infty]{\text{Projection on XZ-plane}}$

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

## General Projection Matrix:

$$dx = C_0 P_x - PP_x \quad (x \rightarrow \text{plane})$$

$$dy = C_0 P_y - PP_y$$

$$dz = C_0 P_z - PP_z$$

$$P = \begin{bmatrix} 1 & 0 & -\frac{dx}{dz} & \frac{dx}{dt}(PP_z) \\ 0 & 1 & -\frac{dy}{dz} & \frac{dy}{dt}(PP_z) \\ 0 & 0 & -\frac{PP_z^2}{dz} & PP_z \left(1 + \frac{PP_z}{dz}\right) \\ 0 & 0 & -\frac{1}{dz} & \left(1 + \frac{PP_z}{dz}\right) \end{bmatrix}$$

Cases 01:  $C_0 P(0, 0, 0)$   
 $PP(0, 0, d)$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$

Case 02:

$$COP(0, 0, d)$$

$$PP(0, 0, 0)$$

$$dx = 0$$

$$dy = 0$$

$$dz = d \Rightarrow -x/b = xb$$

$$m + 4900 = xb$$

$$499 - 49a = xb$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \left(\frac{-x}{b}\right) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \left(\frac{-x}{b}\right) - \frac{1}{d} & 1 \end{bmatrix}$$

$\therefore = 9$

$$(0, 0, 0) 900$$

$$(0, 0, 0) 99$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{b} & 0 \end{bmatrix} = 9$$

Scanned with CamScanner

Scanned with CamScanner

## Lighting | Projection, Illumination

\* Phong's Reflection Model  $\Rightarrow I = I_a + I_d + I_s$

I<sub>a</sub> (ambient)  $\rightarrow$  Ambient  
 I<sub>d</sub> (diffuse)  $\rightarrow$  Diffuse  
 I<sub>s</sub> (specular)  $\rightarrow$  Specular

Ambient:  $I_a = I_a K_a$

$$I_d = I_p K_d (\max(\hat{L} \cdot \hat{N}, 0))$$

Diffuse:  $I_d = I_p K_d (\max(\hat{L} \cdot \hat{N}, 0))$

Specular:  $I_s = I_p K_s (\max(\hat{R} \cdot \hat{V}, 0))$

Formula Without Attenuation:

$$I = I_a K_a + I_p K_d \max(\hat{L} \cdot \hat{N}, 0) + I_p K_s \max(\hat{R} \cdot \hat{V}, 0)$$

Formula with Attenuation:

$$I = I_a K_a + I_p f_{att} (K_d \max(\hat{L} \cdot \hat{N}, 0) + K_s \max(\hat{R} \cdot \hat{V}, 0))$$

$$f_{att} = \max\left(1 - \left(\frac{d}{r}\right)^n, 0\right)$$

Formula for Reflection  $\hat{E} [R = \alpha(\hat{L} \cdot \hat{N})\hat{N} - \hat{L}]$

$I_L + I_D + nI_S = E$  (Global lighting equation)

$I$  = Intensity of the light.

$I_a$  = Intensity of Ambient light

$K_a$  = ambient reflection Co-efficient

$I_p$  = Intensity of a the point light

$K_d$  = diffuse reflection Co-efficient

$\hat{L}$  = Directional vector of the incoming light from a point source

$\hat{N}$  = Surface normal.

$K_s$  = Specular reflection Co-efficient

$\hat{R}$  = Reflection direction Vector.

$\hat{V}$  = Viewing direction Vector.

$n$  = shininess exponent