Line Drawing Algorithm

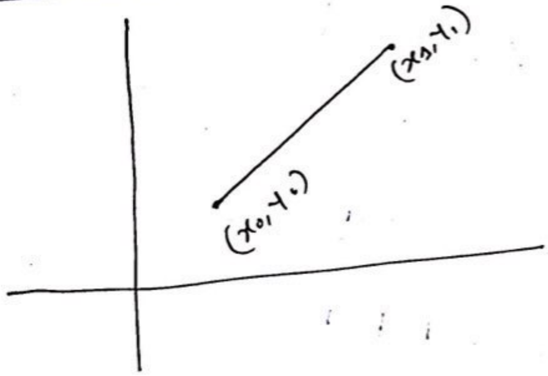
Actually, pixels get selected and turned on.

* The more pixels ≈ 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 (m), Intercept (c)

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

$c = y - mx$

$c = y_0 - mx_0$

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

Scanned with CamScanner

Using SS, find out the intermediate pixels from

$\underbrace{(2,2)}_{(x_0,y_0)}$ to $\underbrace{(7,5)}_{(x_1,y_1)}$

Soln:

$$m = \frac{5-2}{7-2} = \frac{3}{5} = 0.6$$

$$c = y_0 - mx_0$$
$$= 2 - 0.6 \times 2$$
$$= 2 - 1.2 = 0.8$$

$$y = 0.6x + 0.8$$

$$\Rightarrow y(3) = (0.6 \times 3) + 0.8 = 2.6 \approx 3$$
$$y(4) = (0.6 \times 4) + 0.8 = 3.2 \approx 3$$
$$y(5) = (0.6 \times 5) + 0.8 = 3.8 \approx 4$$
$$y(6) = (0.6 \times 6) + 0.8 = 4.4 \approx 4$$

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 ~~up~~ off to the nearest integers.

Scanned with CamScanner

Final Answer:

Pixels → (3,3)

(4,3)

(5,4)

(6,4)

Drawbacks of Simple Solⁿ:

- Multiplication is Costly operation → Time Consuming, → Expensive.
- Round off

Scanned with CamScanner

Digital Differential Analyzer

** Overcomes the drawbacks of Simple Solution.

(2,2) (3,2.6) (4,3.2) (5,3.8) (6,4.4)

$x \rightarrow 1$

$y \rightarrow 0.6$

$$\boxed{\begin{aligned} x_{new} &= x_{prev} + 1 \\ y_{new} &= y_{prev} + m \text{ (slope)} \end{aligned}}$$

→ for, $-1 \leq m \leq 1$

** if the slope is really high,

$$\boxed{\begin{aligned} x_{new} &= x_{prev} + \frac{1}{m} \\ y_{new} &= y_{prev} + 1 \end{aligned}}$$

→ for all the other cases.

Scanned with CamScanner

□ Using DDA, find out the intermediate pixels from (2,2) to (5,7)

Soln:

$$\text{Slope}, m = \frac{\cancel{5}7-2}{5-2} = \frac{5}{3} = 1.67$$

$m > 1.67$, falls under other cases.

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

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

$x\left(+\frac{1}{m}\right)$	$y(+1)$	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)

Scanned with CamScanner

Midpoint Line Algorithm

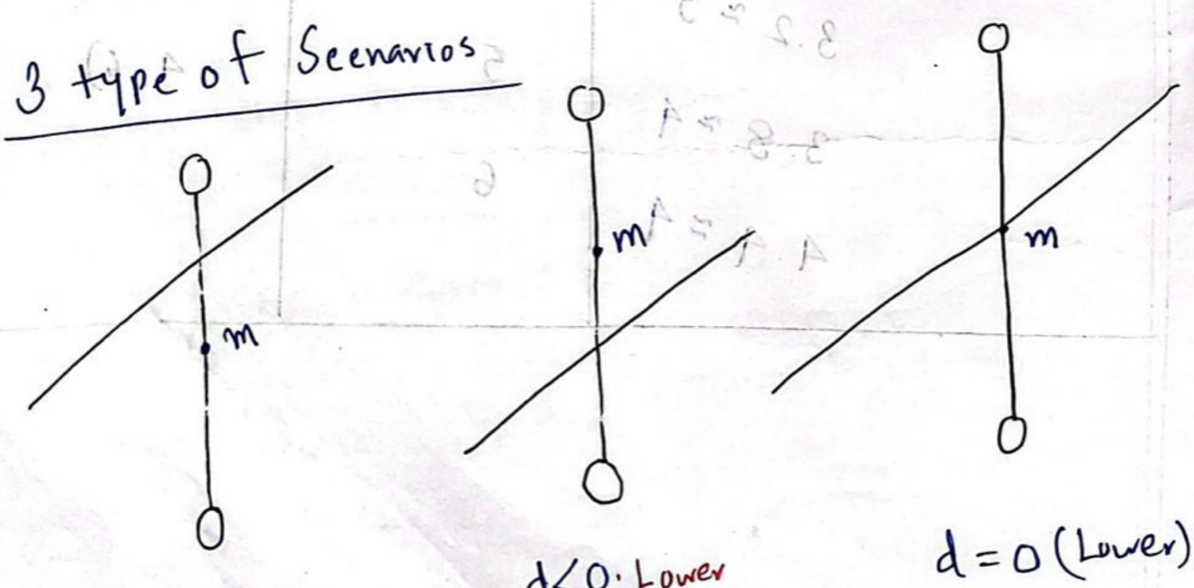
- Diffenent forms of Line eq^n:

1) $y = mx + c$
2) $x/a + y/b = 1$
3) $ax + by + c = 0$] Convenient One for this algorithm.

Middle point (x_p, y_p)

$$\boxed{d = f(x_p, y_p) = ax_p + by_p + c}$$

3 type of Scenarios

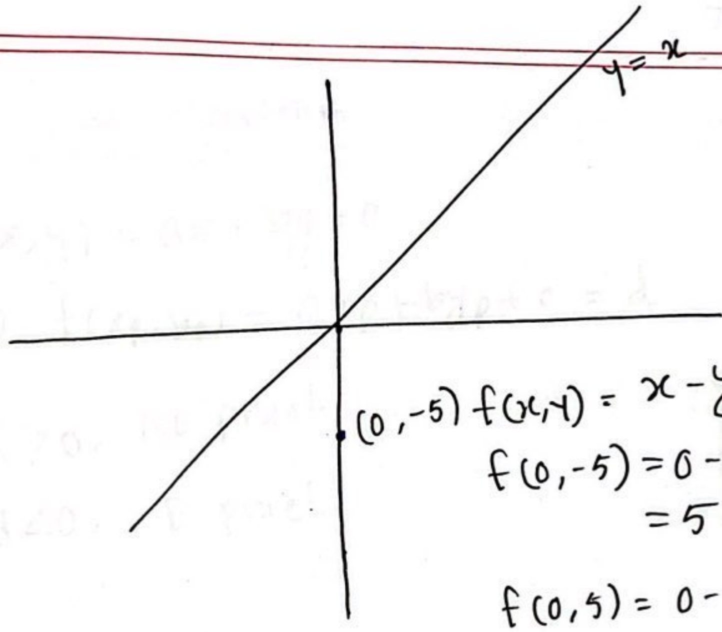


$d > 0$: Upper pixel (NE)

$d < 0$: Lower pixel

$d = 0$ (Lower) (E)

Scanned with CamScanner

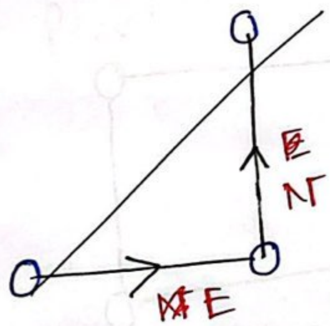


$f(x,y) = x - y$

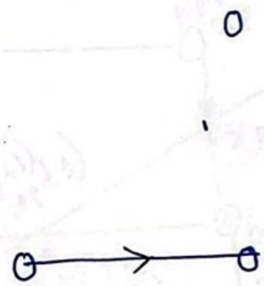
$f(0,-5) = 0 - (-5)$

$= 5$

$f(0,5) = 0 - 5 = -5$



Upper Pixel (NE)



Lower Pixel (E)

f(midpoint)	midpoint's position	Pixel.
① -ve	Upperside of line	lower pixel (E)
② +ve	Lower side of line	Upper pixel (NE)

Scanned with CamScanner

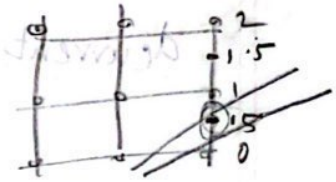
03/05/23

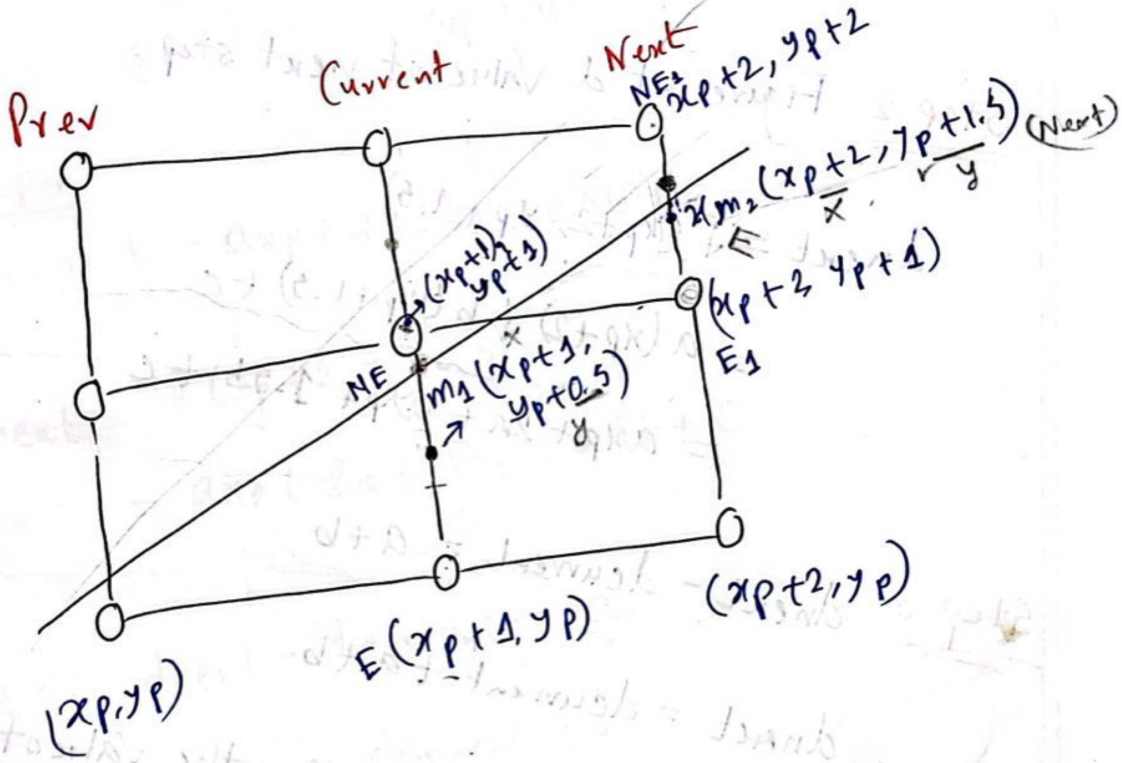
Midpoint Line Algorithm

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

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

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





Scanned with CamScanner

Step 1: Figure out d value of current step.

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

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

$= a(x_p + 1) + b(y_p + 0.5) + c$

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

Step 2 Figure out d value of next step.

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

$= a(x_p + 2) + b(y_p + 1.5) + c$

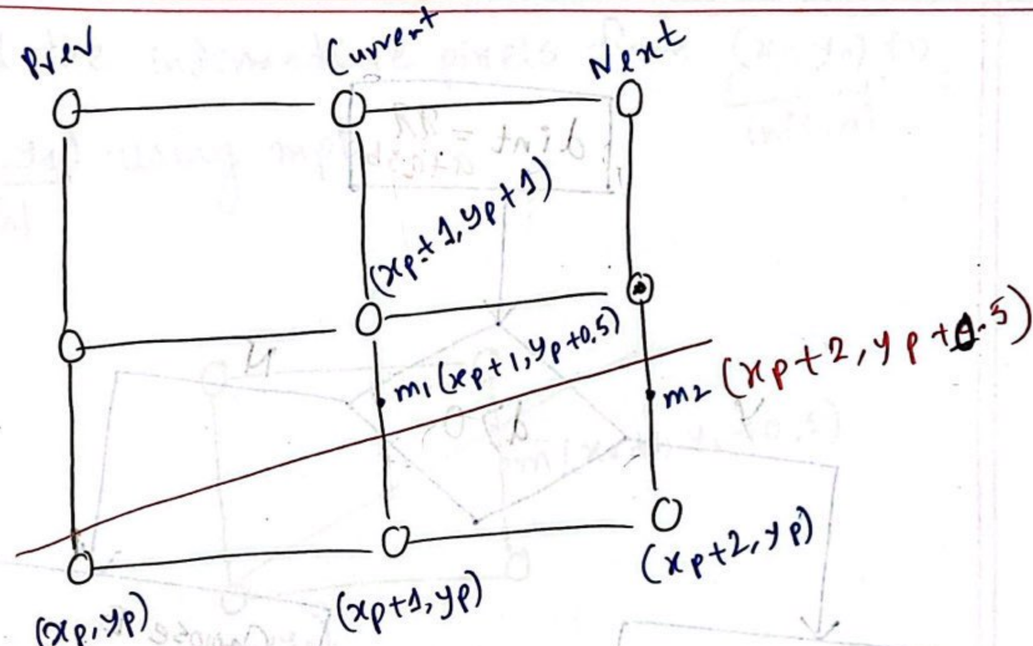
$= ax_p + 2a + by_p + 1.5b + c$

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

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

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

Scanned with CamScanner



Step 4:

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

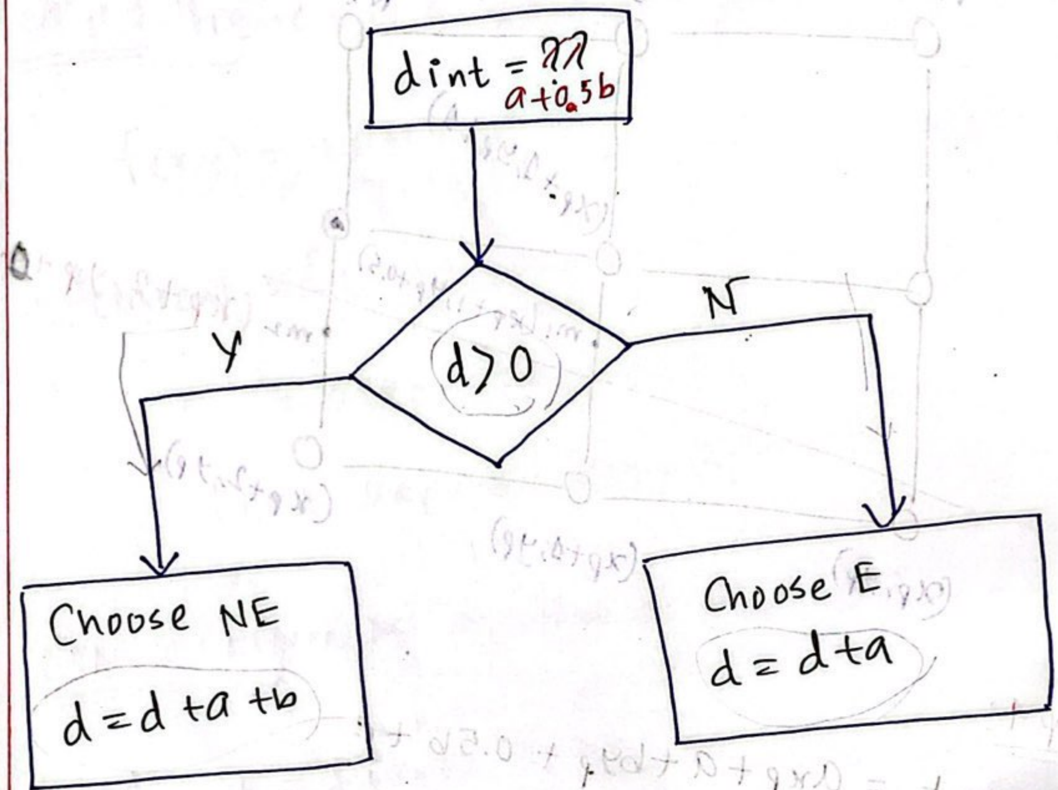
$$d_{next} = f(x_p+2, y_p+0.5)$$
$$= ax_p + 2a + by_p + 0.5b + c$$

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

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

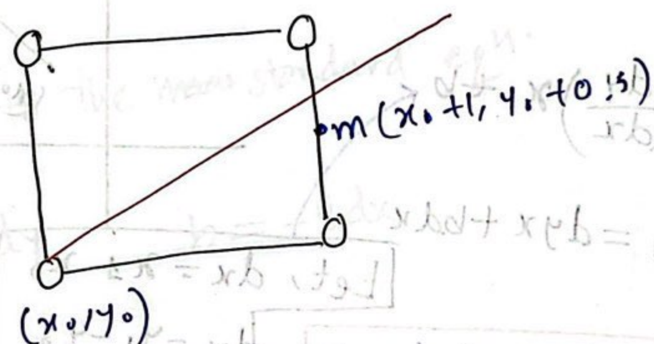
if my current step is E, than the value of next d will be $\boxed{d = d + a}$

Scanned with CamScanner



Scanned with CamScanner

Find the intermediate pixels from $\underbrace{(x_0, y_0)}_{\text{initial}}$ to $\underbrace{(x_1, y_1)}_{\text{final}}$ using mpl.



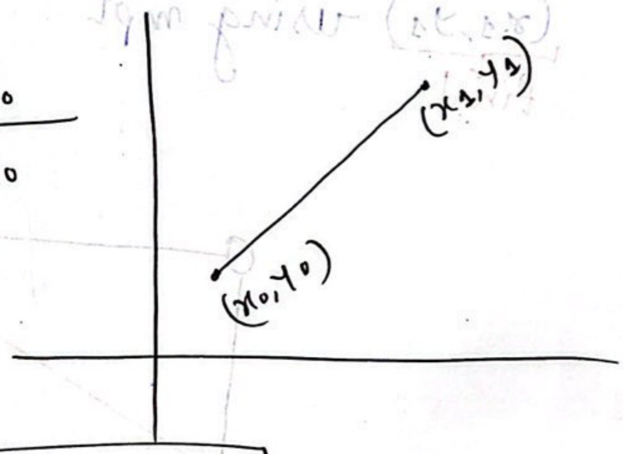
$$f(x,y) = ax + by + c$$

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

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

Scanned with CamScanner

Find out Value of a,b from (x_0, y_0) and (x_1, y_1)



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

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

$$dx \cdot y = dyx + bdx$$

$$\boxed{\text{Let, } dx = x_1 - x_0 \quad dy = y_1 - y_0}$$

$$\Rightarrow \boxed{dy \cdot x - dx \cdot y + b \cdot dx = 0} \; \text{(i)}$$

• Comparing (i) with the standard line eq^n $ax + by + c = 0$
$\Rightarrow$

$$\boxed{a = dy \quad b = -dx}$$

$$\boxed{d_{int} = dy - 0.5dx}$$

But we can't have fraction here.

Scanned with CamScanner

To solve the fractional problem,

(i)x 2 ⇒

$$\boxed{2dy.x - 2dx.y + 2b.dx = 0}$$

Comparing it to the ~~ma~~ standard eq^n:

$a = 2dy$ $b = -2dx$

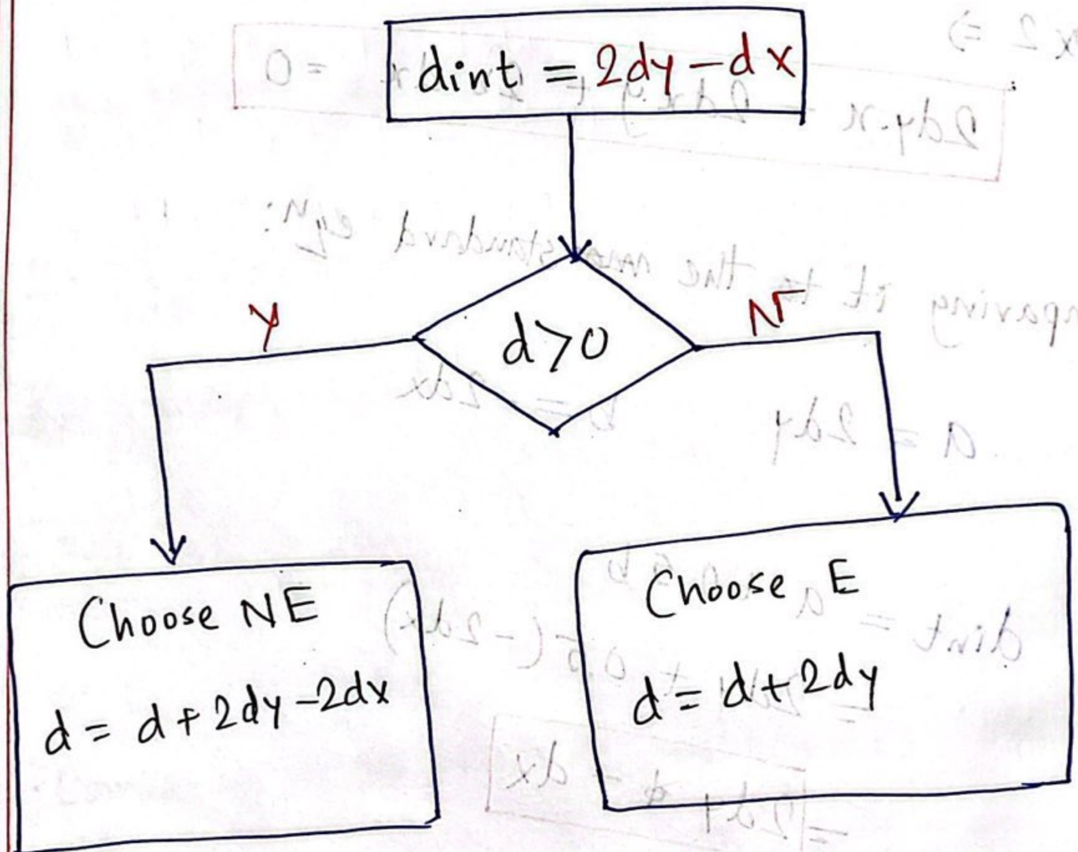
$$d_{int} = a + 0.5b$$

$$= 2dy + 0.5(-2dx)$$

$$= \boxed{2dy \cancel{+} - dx}$$

Scanned with CamScanner

Final Flowchart



Scanned with CamScanner

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

Step 1

$x_0 = 0,\ y_0 = 2$

$x_1 = 70 \quad y_1 = 52$

Step 2

$dy = 52 - 2 = 0 52$

$dx = 70 - 0 = 70$

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

$2dy = 2 \times 50 = 100 \quad d_{int} = 2dy - dx$
$= 100 - 70 = 30$

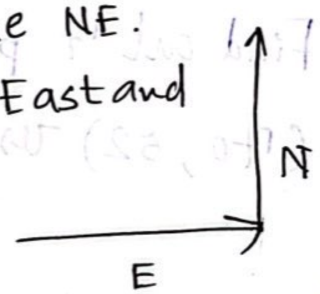
essential Values.

x	y	d	E/NE	dupdate	pixelc
0	2	30	NE	30+(-40)=-10	(0,2)
1	3	-10	E	-10+100=90	(1,3)
2	3	90	NE	~~90+100~~ = 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$(+1, +1)$

Scanned with CamScanner

$y_0 - mx_0 = b$

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)

$C = 15 \quad D = 80$

Soln: $A = 21, \; B = 30$

a) Slope $= \dfrac{A+C-A}{B+10-B} = \dfrac{15}{10} = \boxed{1.5}$

$y = mx + b$

$\Rightarrow b = y - mx$

intercept $b = \overset{30}{y_0} - mx_0$

$= 30 - 1.5(21)$

$= -1.5$

$m = 1.5 \therefore \; x_{new} = x_{prev} + \dfrac{1}{m}$

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

Scanned with CamScanner

$x_{prev} + \frac{1}{m} = x_{new}$	$y_{prev} + 1 = y_{new}$	(pixels)
21	30	(21, 30)
$\left(21 + \frac{1}{1.5}\right) = 21.67 \approx 22$	$30 + 1 = 31$	(22, 31)
$22 + \frac{1}{1.5} = 22.67 \approx 23$	$31 + 1 = 32$	(23, 32)
$23 + \frac{1}{1.5} = 23.64 \approx 24$	$32 + 1 = 33$	(24, 33)

$x_{new} = x_{prev} + \frac{1}{m}$	x_{round}	$y_{new} = y_{prev} + 1$	(pixels)
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

DDA Example

$(-1, 7)\ (2, 2)$

$$m = \frac{7-2}{\cancel{2-(-1)}} = -\frac{5}{3} = -1.67$$

$-1-2$

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

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

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

x_{new}	x_{round}	y_{new}	(pixels)
2	2	2	(2,2)
$2+(-0.6)=1.4$	1	3	(1,3)
$1.4+(-0.6)=0.8$	1	4	(1,4)

Scanned with CamScanner

Scanned with CamScanner

Slide 25 of 32

Exercise

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

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

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

$$= 67.7727$$

$$b = y_0 - mx_0$$
$$= 35 - \left(-\frac{15}{11} \times 20\right)$$
$$= 62$$

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

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

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

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

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

Scanned with CamScanner

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

$= 12.41$

2 ~~#~~

a) (20, 5) (19, 50)

$m = \frac{50-5}{19-20} = -45$

$\frac{1}{m} = -\frac{1}{45}$

$x_{new} = x_{prev} + \frac{1}{m}$	$y_{new} = y_{prev} + 1$	x_{round}	(pixels)
20	5	20	(20, 5)
$20 + (-\frac{1}{45}) = 19.97$	$5+1=6$	20	(20, 6)
$19.67 + (-\frac{1}{45}) = 19.64$	$6+1=7$	20	(20, 7)
	$7+1=8$	~~20~~ 19	(~~20~~ 19, 8)
$19.64 + (-\frac{1}{45}) = 19.42$	$8+1=9$	19	(19, 9)
$19.42 + (-\frac{1}{43}) = 19.39$		19	(19, 10)
$19.39 + (-\frac{1}{45}) = 19.36$	$9+1=10$		

Scanned with CamScanner

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

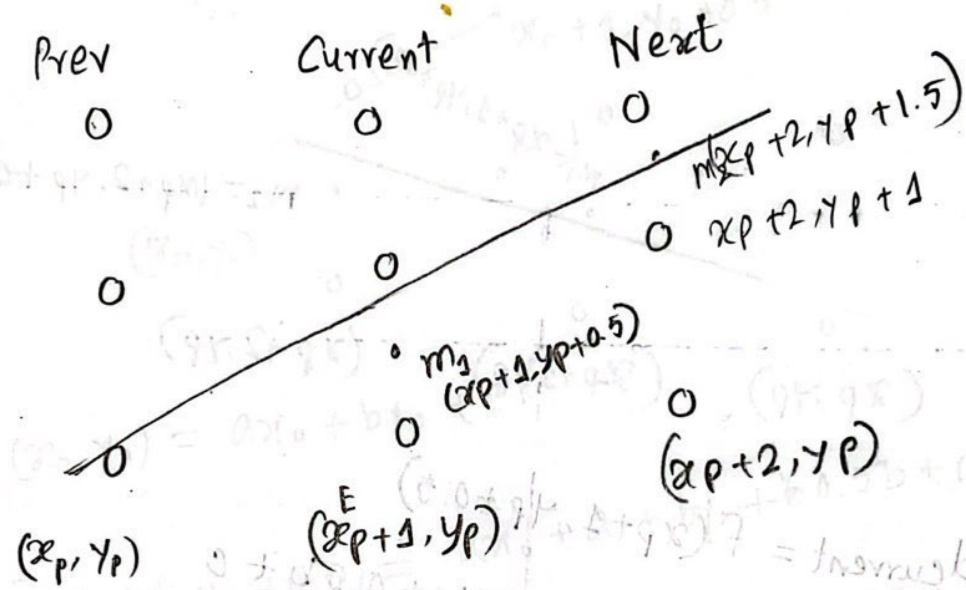
$$m = \frac{0-50}{-15+5} = \frac{-50}{-10} = 5$$

$x_{new} = x_{prev} + \frac{1}{m}$	$y_{new} = y_{prev} + 1$	x round	(pixel)
			(-5, 50)
$-5 + \frac{1}{5} = -4.8$	$50+1=51$	-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.



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

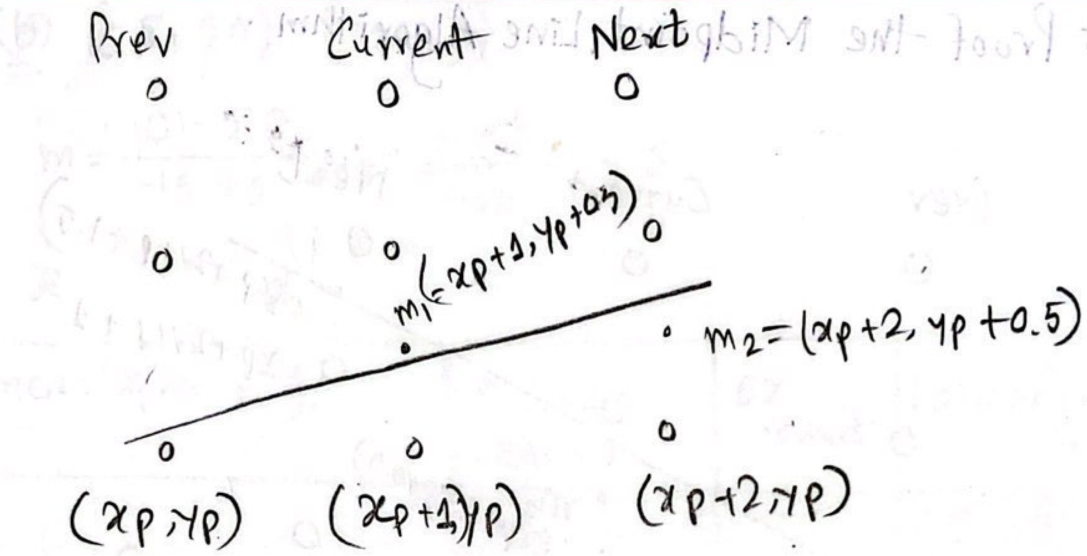
$d_{current} = f(x_p+1, y_p+0.5)$
$= ax_p + a + by_p + 0.5b + c$

$d_{next} = f(x_p+2, y_p+1.5)$
$= ax_p + 2a + by_p + 1.5b + c$

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

$\therefore d_{next} = d_{current} + a + b$ $\therefore$ if next step is NE, then
$d_{next} = d_{current} + a + b$.

Scanned with CamScanner



$$d_{current} = f(xp+1, yp+0.5)$$
$$= axp + a + byp + 0.5b + c$$

$$d_{next} = f(xp+2, yp+0.5)$$
$$= axp + 2a + byp + 0.5b + c$$

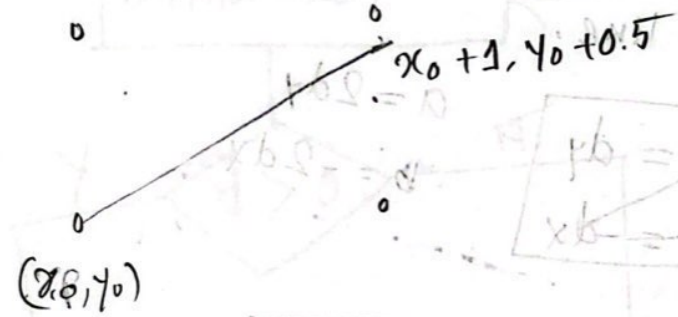
$$d_{next} - d_{current} = a$$

$\therefore d_{next} = d_{current} + a$.

$\therefore$ If the next step is E, then

$$d = d + a$$

Scanned with CamScanner



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

$$\therefore f(x_0+1, y_0+0.5) = ax_0 + a + by_0 + 0.5b + c$$

$$= ax_0 + by_0 + c + a + 0.5b.$$

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

$$f(x_0+1, y_0+0.5) = a + 0.5b.$$

$$\therefore d_{initial} = f(x_0+1, y_0+0.5)$$

Now, $y_0 = mx + b$ $\qquad m = \frac{x_1 - x_0}{y_1 - y_0}$

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

$$\Rightarrow ydx = xdy + bdx$$

$$\Rightarrow xdy - ydx + bdx = 0 \quad \text{---(1)}$$

$$(1) \times 2 = 2xdy - 2ydx + 2bdx = 0$$

Scanned with CamScanner

Scanned with CamScanner

Equating eq(i) with standard eqⁿ of straight line,

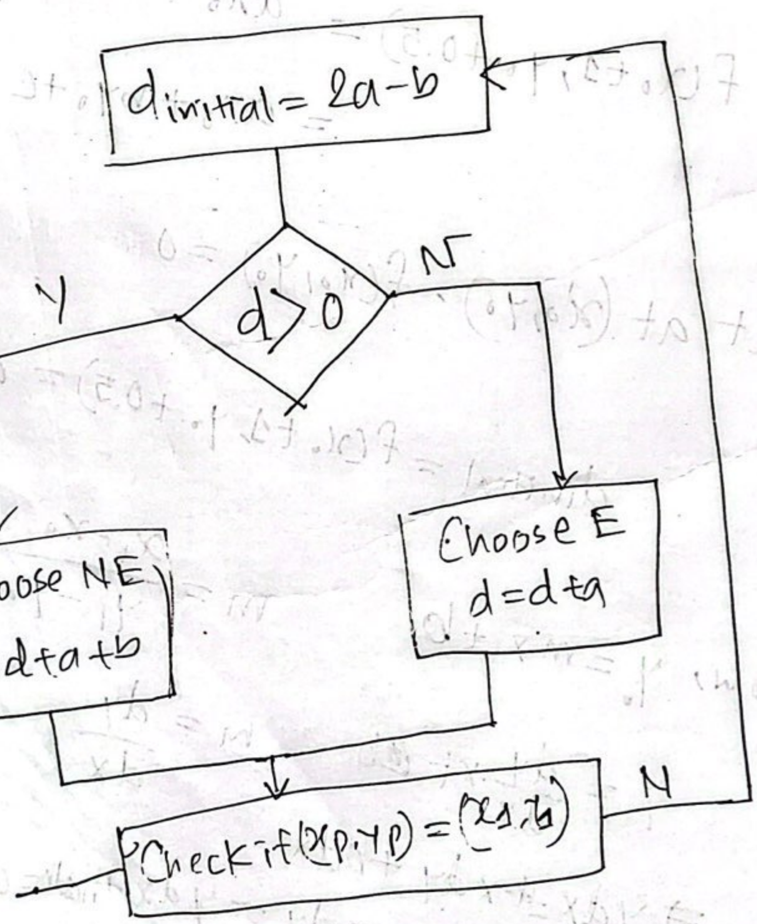
$$\cancel{\boxed{\begin{array}{l} a = dy \\ b = -dx \end{array}}}$$

$a = 2dy$

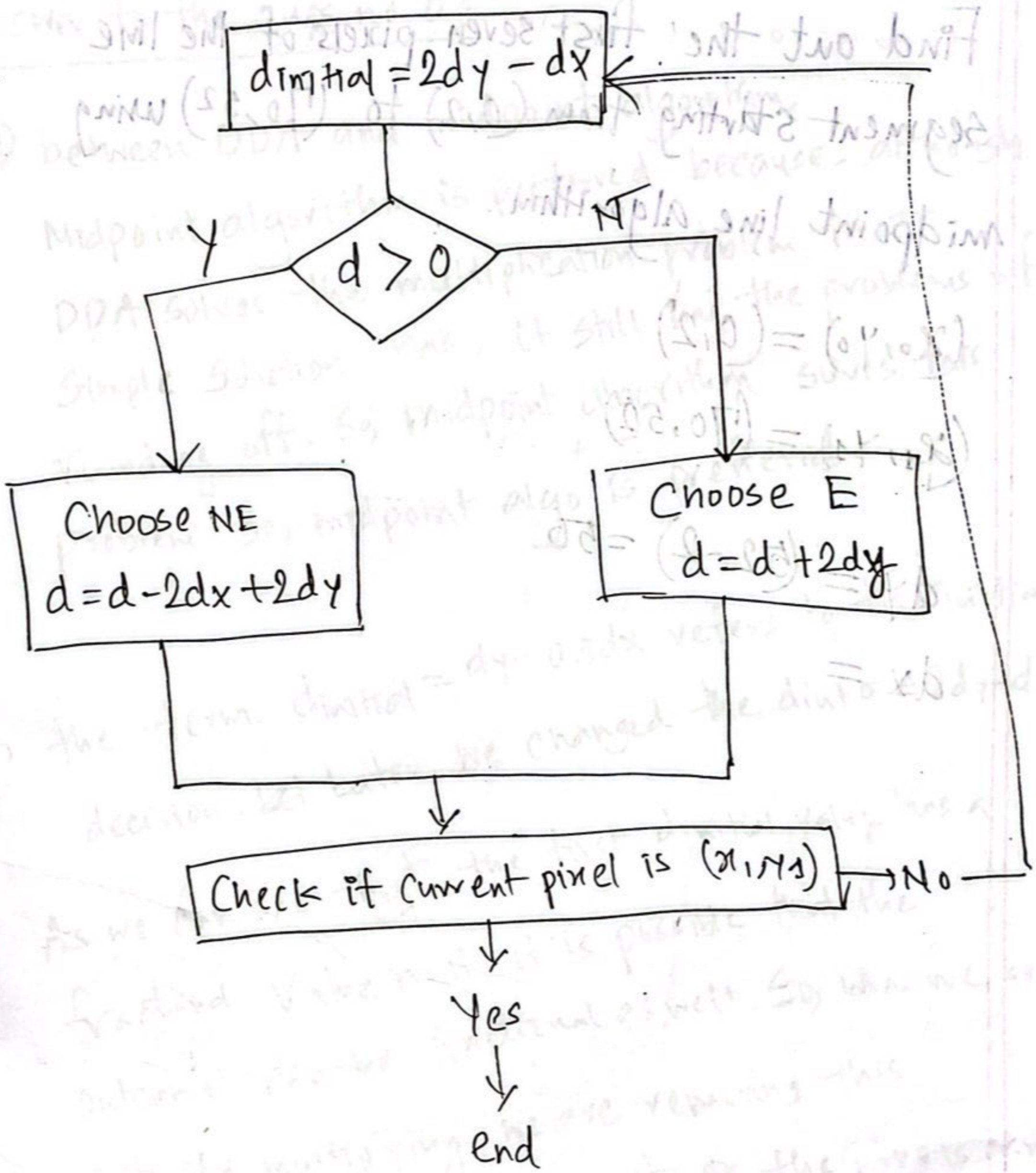
$b = -2dx$

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

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



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 has, it still has the problems of rounding off. So, midpoint algorithm solves this problem. So, midpoint algo is preffered.

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

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

Scanned with CamScanner

figure out the value for

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

then we multiply it with 2,

$$2dyx - 2dxy + 2dx.b = 0$$

The value on the right hand side isn't changed as well. So it ~~is~~ 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)

$$d_{initial} = 2dy - dx$$
$$= 2(y_1 - y_0) - (x_1 - x_0)$$
$$= 2(13-7) - (6-3) = 12 - 3 = 9$$

Scanned with CamScanner

$d = d + 2dy - 2dx$

$= 9 + 12 - 6$

$= 21 - 6 = 15$

$d = \cancel{d + 2dy - 2dx}$

$\cancel{= 15 +}$

$d = d + 2dy - 2dx$

$= 15 + 2 \times 6 - 2 \times 3$

$d = 21 + 2 \times 6 - 2 \times 3$

$= 27$

$d = 27 + 2 \times 6 - 2 \times 3$

$= 33$

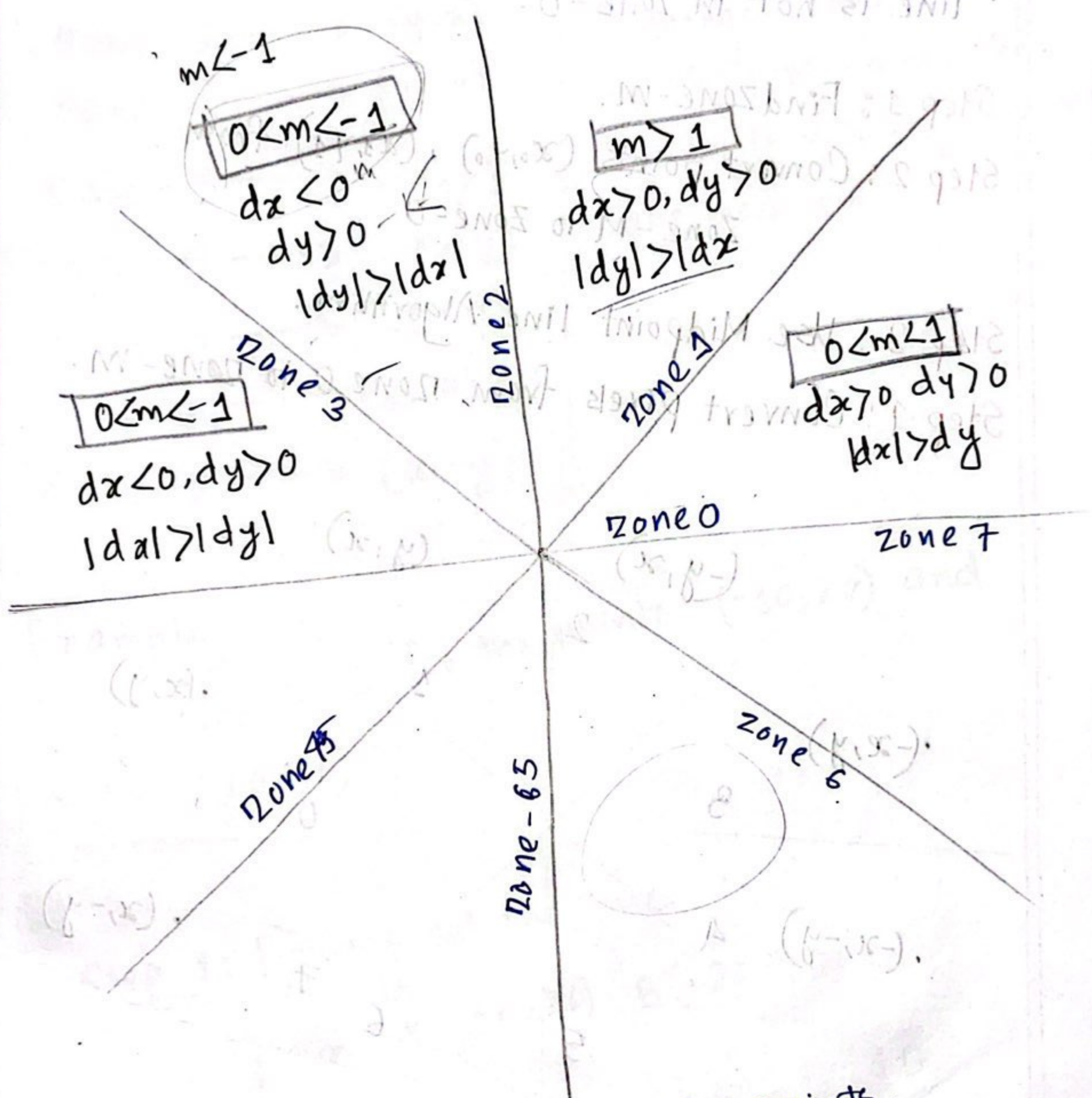
$d = d + 2dy - 2dx$

$= 9 + 2(13 - 7) - 2(6 - 3)$

$= 15$

Scanned with CamScanner

Eight Way Symmetry



* How to find out zone from given points.

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

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

Scanned with CamScanner

Scanned with CamScanner

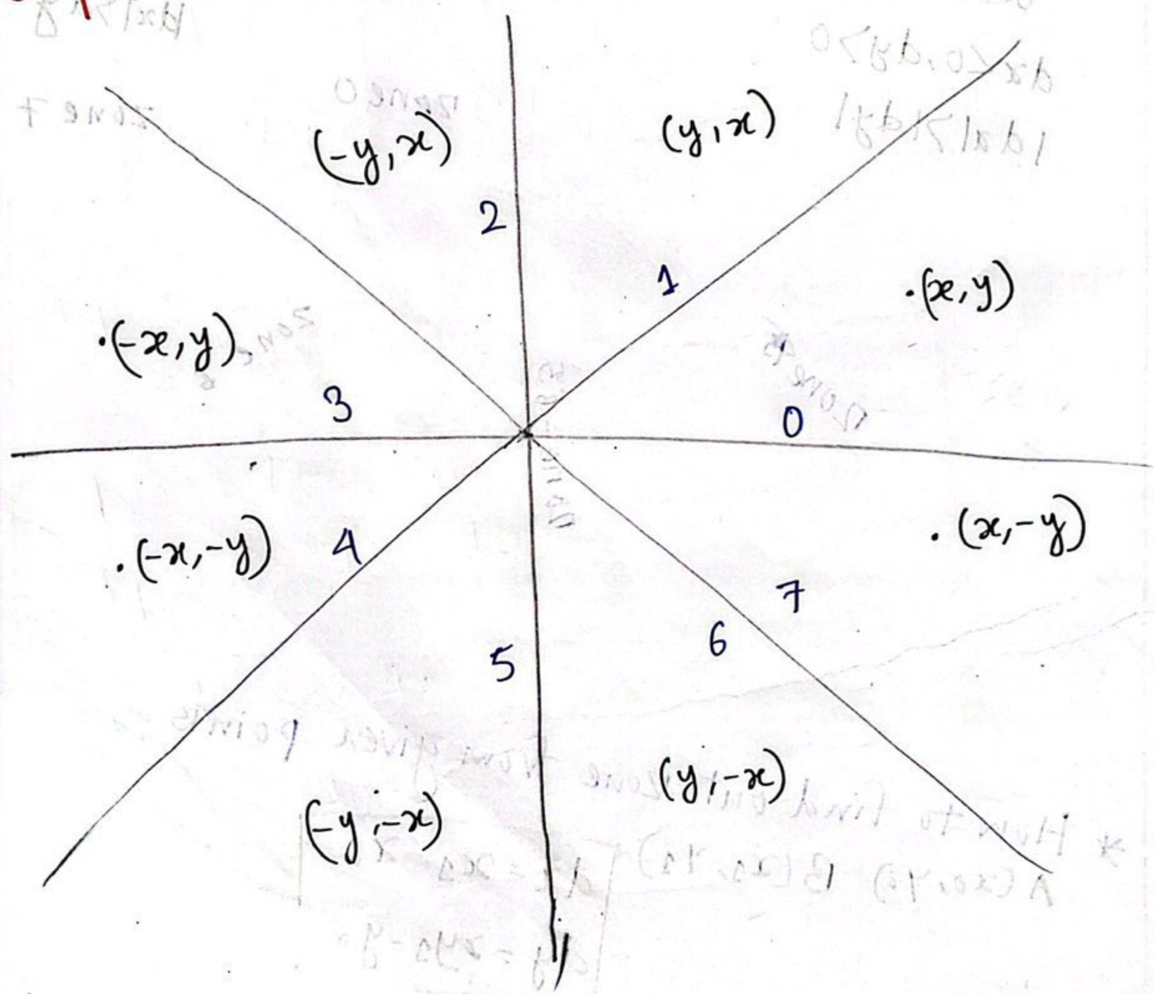
☐ How to use Midpoint Line Algorithm when the line is not in zone-0.

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

Zone - 0 $\longrightarrow$ (x, y)

Zone - 1 $\longrightarrow$ (y, x)

Zone - 2 $\longrightarrow$ $(-y, x)$

Zone - 3 $\longrightarrow$ $(-x, y)$

Zone - 4 $\longrightarrow$ $(-x, -y)$

Zone - 5 $\longrightarrow$ $(-y, -x)$

Zone - 6 $\longrightarrow$ $(y, -x)$

Zone - 7 $\longrightarrow$ $(x, -y)$

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

Example

Use MPL for the points $(-10, 20)$ and $(-20, 70)$

Sol^n:

Step 1: Figuring out the zone.

$A(-10, 20)$ $B(-20, 70)$

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

$dy = 70 - 20 = 50$

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

$\boxed{\text{Zone } 2}$

Scanned with CamScanner

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
$(-10, 20)$	$\longrightarrow$	$(-20, 10)$
$(-y, x)$		(x, y)
$(-20, 70)$	$\longleftrightarrow$	$(70, 20)$

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

$x = -y$

$y = -x$

Step 3

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

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

$dy = 20 - 10 = 10$

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

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

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

Scanned with CamScanner

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

x	y	d	E/NE	dupdate	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

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

$(-20, 10) \longrightarrow (-10, -20)$

$(-19, 10) \longrightarrow (-10, -19)$

$(-18, 10) \longrightarrow (-10, -18)$

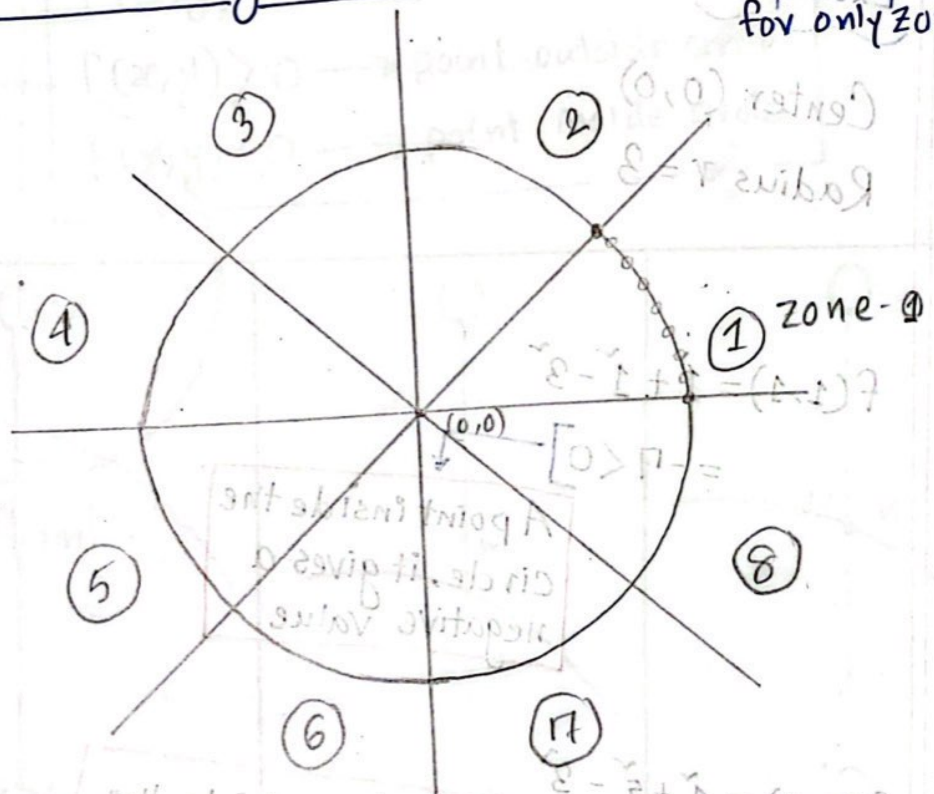
$(-17, 10) \longrightarrow (-10, -17)$

$(-16, 10) \longrightarrow (-10, -16)$

Scanned with CamScanner

Midpoint Circle Algorithm

• We draw for only Zone-1



center (0,0)

radius r

$x^2+y^2-r^2=0$ → Eq^n for Circle.

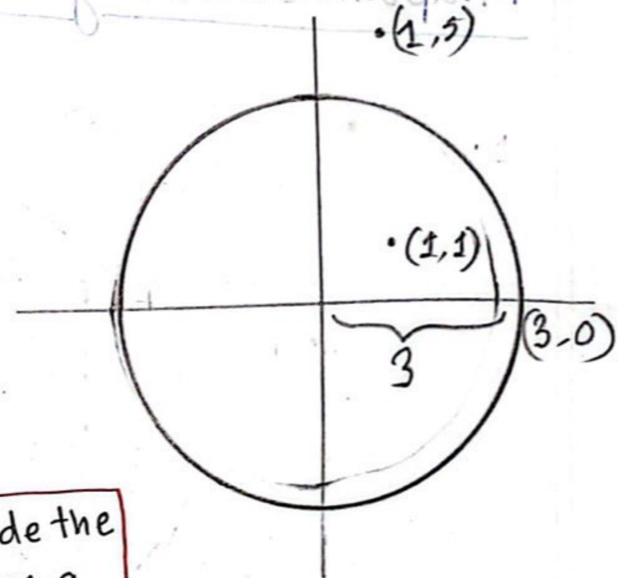
$f(x,y)=x^2+y^2-r^2$

Scanned with CamScanner

Example

Center (0,0)

Radius $r = 3$



$f(1,1) = 1^2 + 1^2 - 3^2$

$= -7 < 0$ → A point inside the circle, it gives a negative value

$f(1,5) = 1^2 + 5^2 - 3^2$

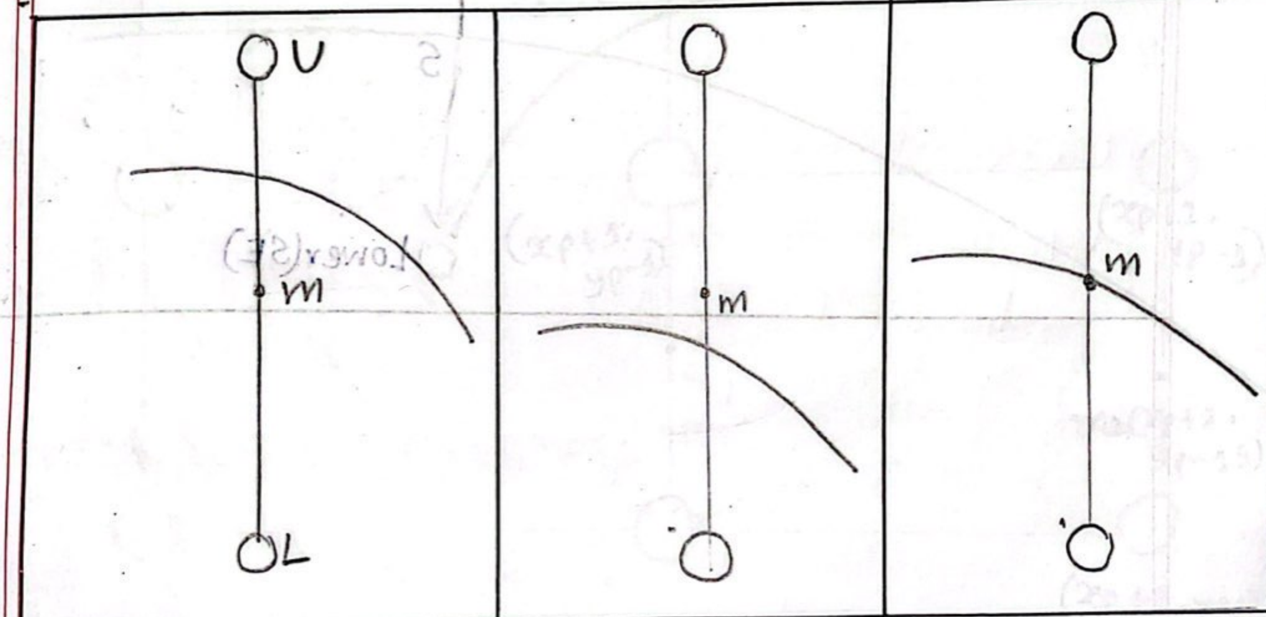
$= 17 > 0$ → A point outside the circle, gives positive value

$f(3,0) = 3^2 + 0^2 - 3^2$

$= 0$ → A point on the circle gives out 0 as output

Scanned with CamScanner

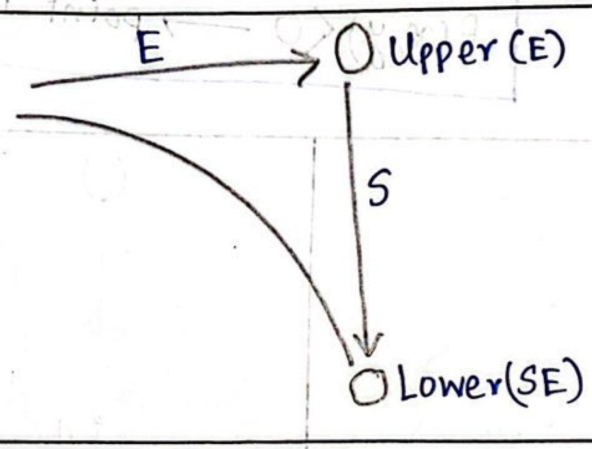
$f(x,y) = 0 \longrightarrow$ point on circle
$f(x,y) > 0 \longrightarrow$ point outside circle
$f(x,y) < 0 \longrightarrow$ point inside circle



• point on circle $f(m) < 0$ Upper Pixel	• point outside circle $f(m) > 0$ Lower Pixel	point on surface of cycle $f(m) = 0$ Lower Pixel We can chose UP as well .

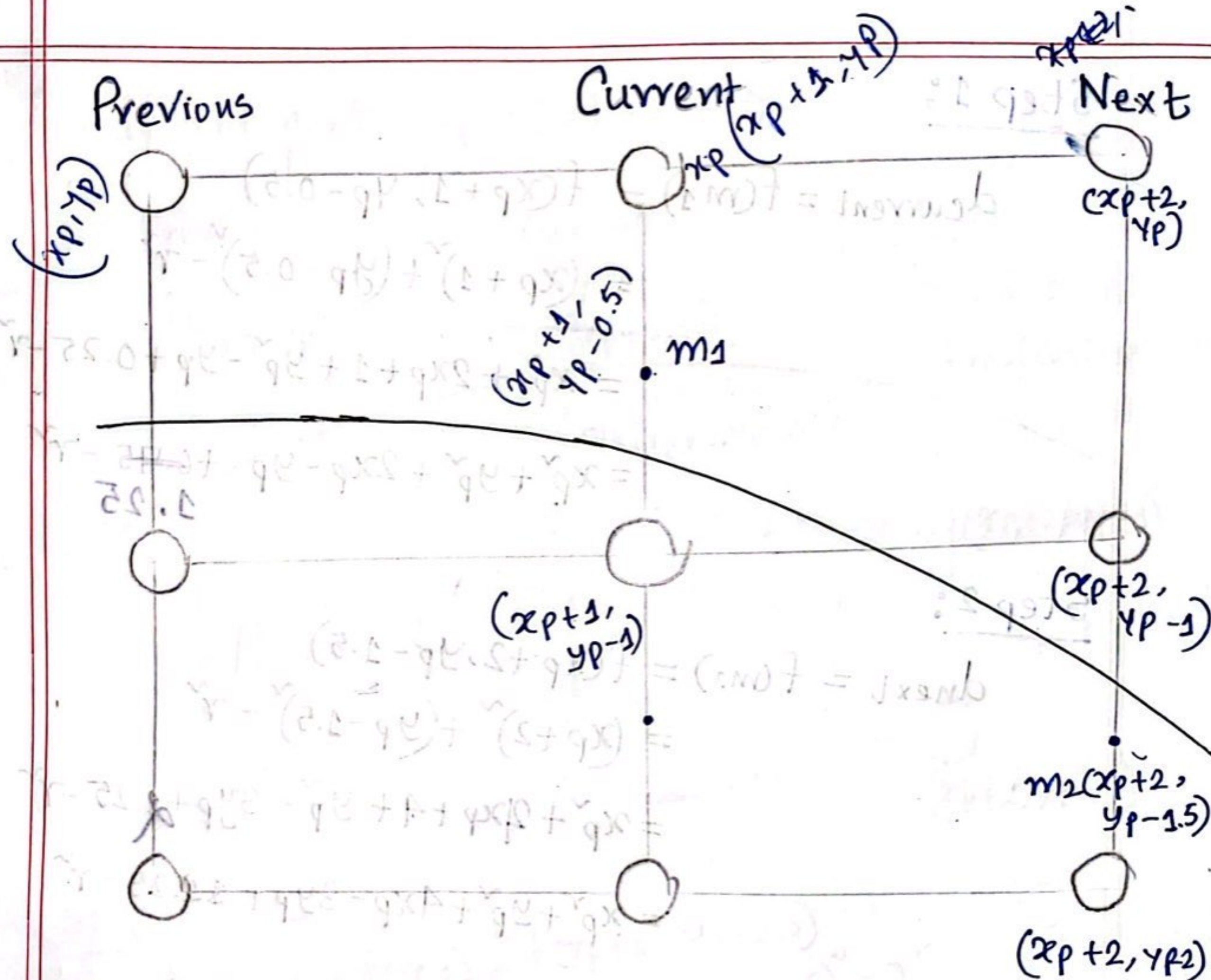
Scanned with CamScanner

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



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

Step 1:

$$d_{current} = f(m_1) = f(x_p+1, y_p-0.5)$$
$$= (x_p+1)^2 + (y_p-0.5)^2 - r^2$$
$$= x_p^2 + 2x_p + 1 + y_p^2 - y_p + 0.25 - r^2$$
$$= x_p^2 + y_p^2 + 2x_p - y_p + \cancel{0.75}\ 1.25 - r^2$$

Step 2:

$$d_{next} = f(m_2) = f(x_p+2, y_p-1.5)$$
$$= (x_p+2)^2 + (y_p-1.5)^2 - r^2$$
$$= x_p^2 + 4x_p + 4 + y_p^2 - 3y_p + 2.25 - r^2$$
$$= x_p^2 + y_p^2 + 4x_p - 3y_p + 12.25 - r^2$$

Step 3:

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

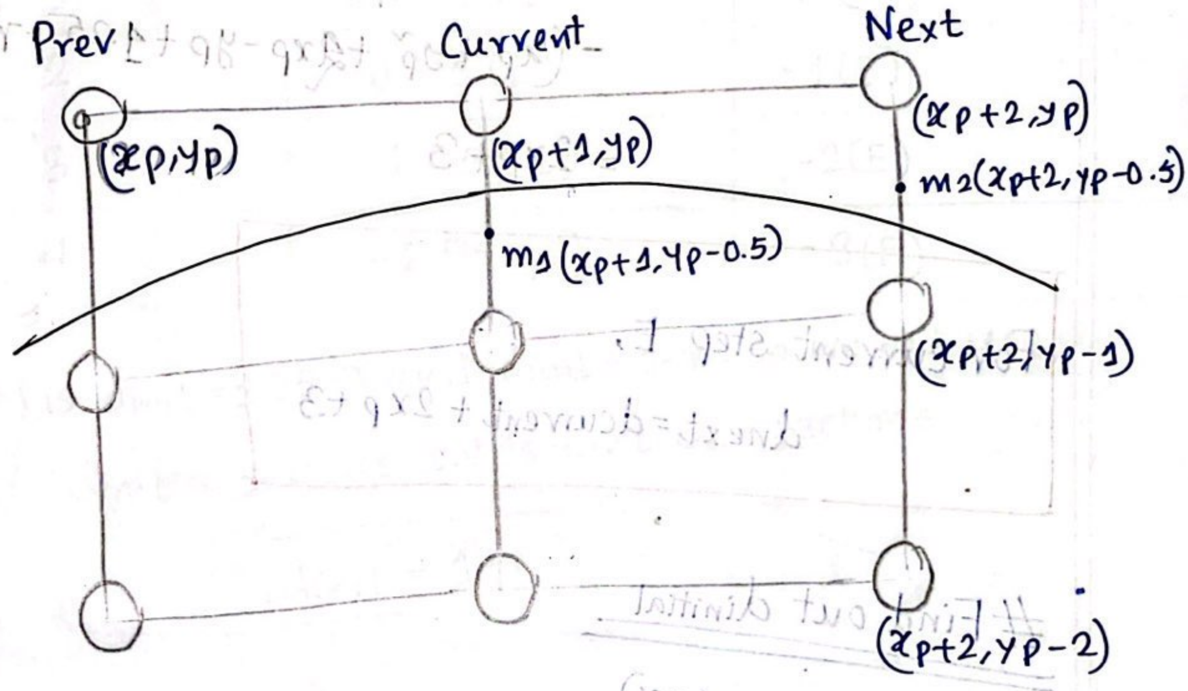
$$\therefore \boxed{d_{update} = 2x_p + 3 - 2y_p + 5}$$

if the next step is SE, ~~next~~ (dcurrent+

$$d_{next} = 2x_p - 2y_p + 5)$$

Scanned with CamScanner

• If the next step pixel is E ⇒



Step 01:

$$d_{current} = f(m_1) = f(x_p+1,\ y_p-0.5)$$
$$= (x_p+1)^2 + (y_p-0.5)^2 - r^2$$
$$= x_p^2 + 2x_p + 1 + y_p^2 - y_p + 0.25 - r^2$$
$$= x_p^2 + y_p^2 + 2x_p - y_p + 1.25 - r^2$$

Step 02:

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

Scanned with CamScanner

Step 03:

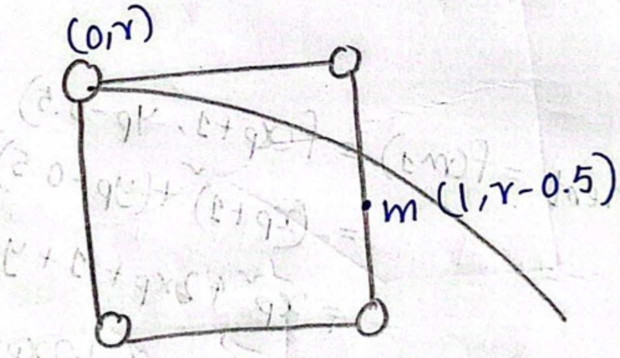
$$d_{next} - d_{current} = (x_p^2 + y_p^2 + 4x_p - y_p + 4.25 - r^2) - (x_p^2 + y_p^2 + 2x_p - y_p + 1.25 - r^2)$$

$$= 2x_p + 3$$

For current step E,

$$\boxed{d_{next} = d_{current} + 2x_p + 3}$$

Find out $d_{initial}$



$$f(M_0)$$

$$d_{initial} = f(1, r-0.5) = 1^2 + (r-0.5)^2 - r^2$$

$$= 1 + r^2 - r + 0.25 - r^2$$

$$= \boxed{1.25 - r}$$

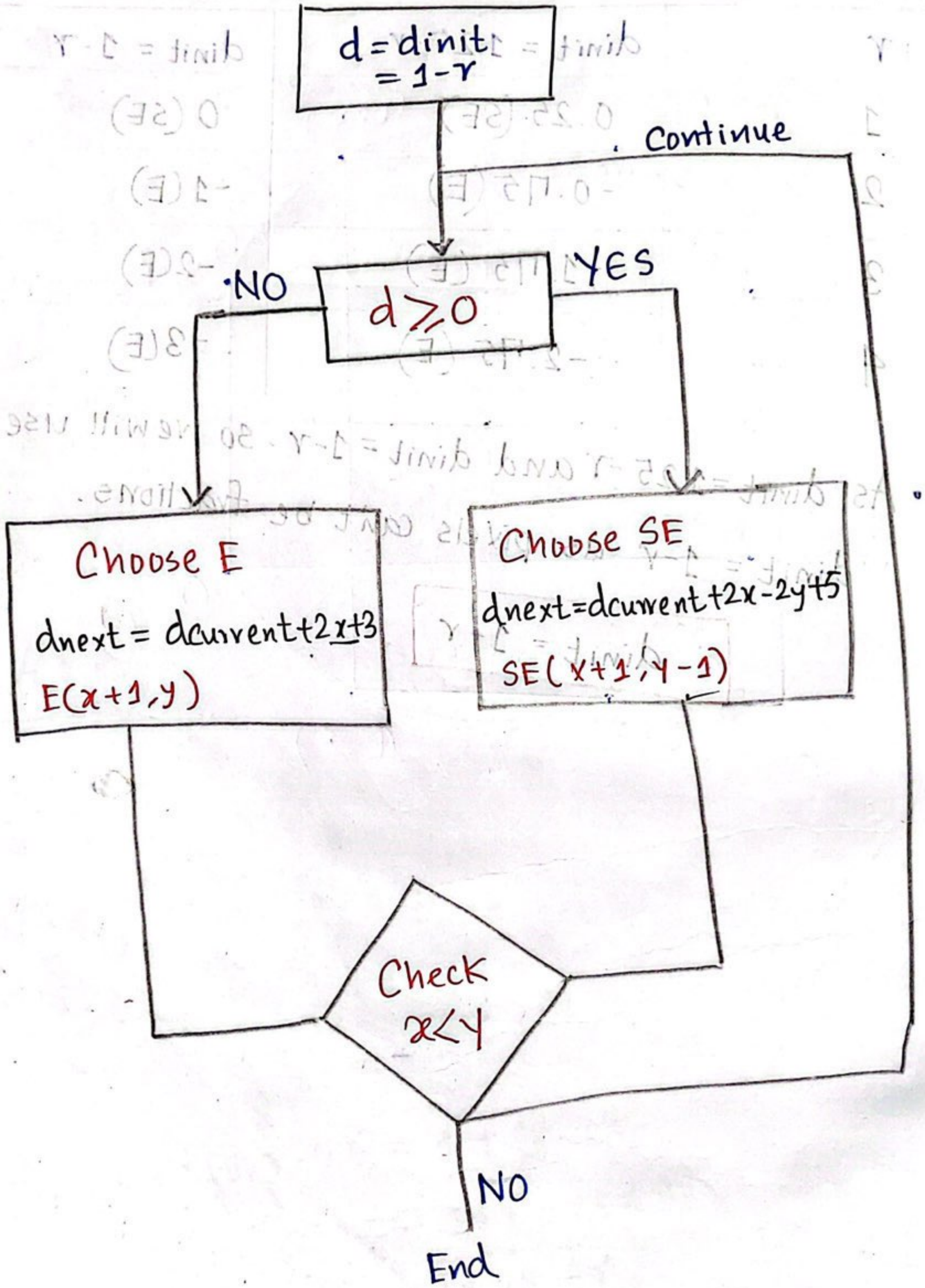
Scanned with CamScanner

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.

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

Scanned with CamScanner



Scanned with CamScanner

Example

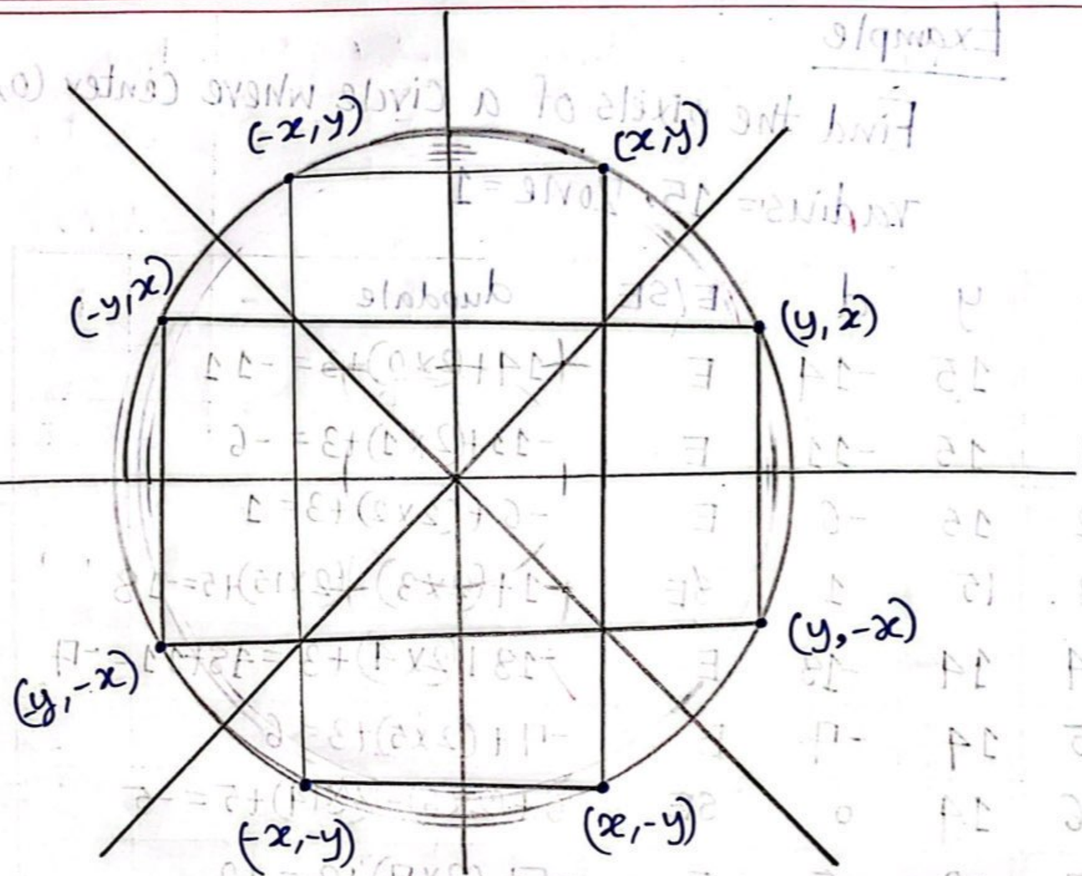
Find the pixels of a Circle where Center (0,0)

radius = 15, Zone = 1

x	y	d	E/SE	dupdate	Pixels
0	15	-14	E	$-14+(2\times0)+3=-11$	(0,15)
1	15	-11	E	$-11+(2\times1)+3=-6$	(1,15)
2	15	-6	E	$-6+(2\times2)+3=1$	(2,15)
3	15	1	SE	$1+(2\times3)-(2\times15)+5=-18$	(3,15)
4	14	-18	E	$-18+(2\times4)+3=18+11=-7$	(4,14)
5	14	-7	E	$-7+(2\times5)+3=6$	(5,14)
6	14	6	SE	$6+(2\times6)-(2\times14)+5=-5$	(6,14)
7	13	-5	E	$-5+(2\times7)+3=12$	(7,13)
8	13	12	SE	$12+(2\times8)-(2\times13)+5=7$	(8,13)
9	12	7	SE	$7+(2\times9)-(2\times12)+5=6$	(9,12)
10	11	6	SE	$6+(2\times10)-(2\times11)+5=8$	(10,11)
11	10	here $x>y$ →			

$$d_{init} = 1-r$$
$$= 1-15=-14$$

Scanned with CamScanner



Scanned with CamScanner

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

Change Zone

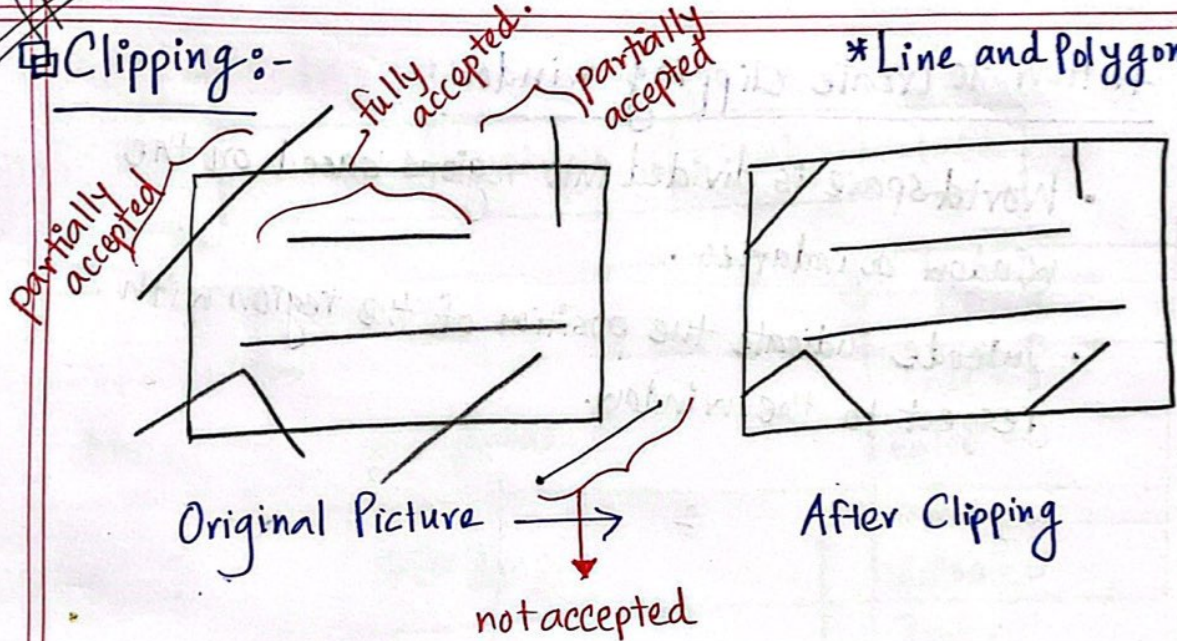
Scanned with CamScanner

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

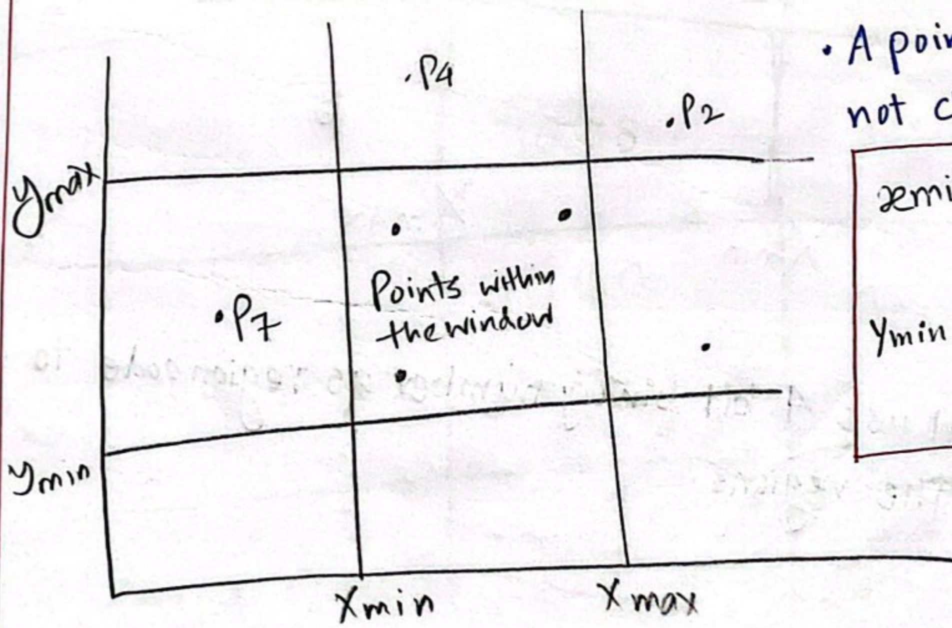
Scanned with CamScanner

Clipping :-

*Line and Polygon



- 2 Algorithm ⟶ i) Cohen Sutherland Algorithm.
  ii) Cyrus beck Algorithm



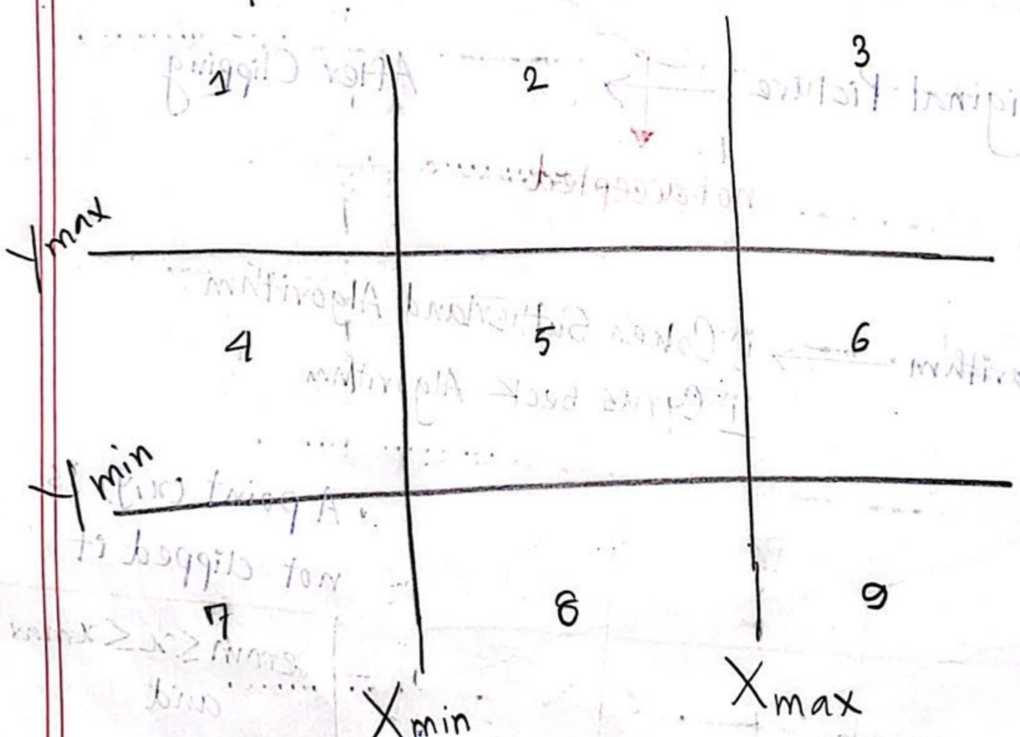
- A point (x,y) is not clipped if

$x_{min} \le x \le x_{max}$ and $y_{min} \le y \le y_{max}$

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.



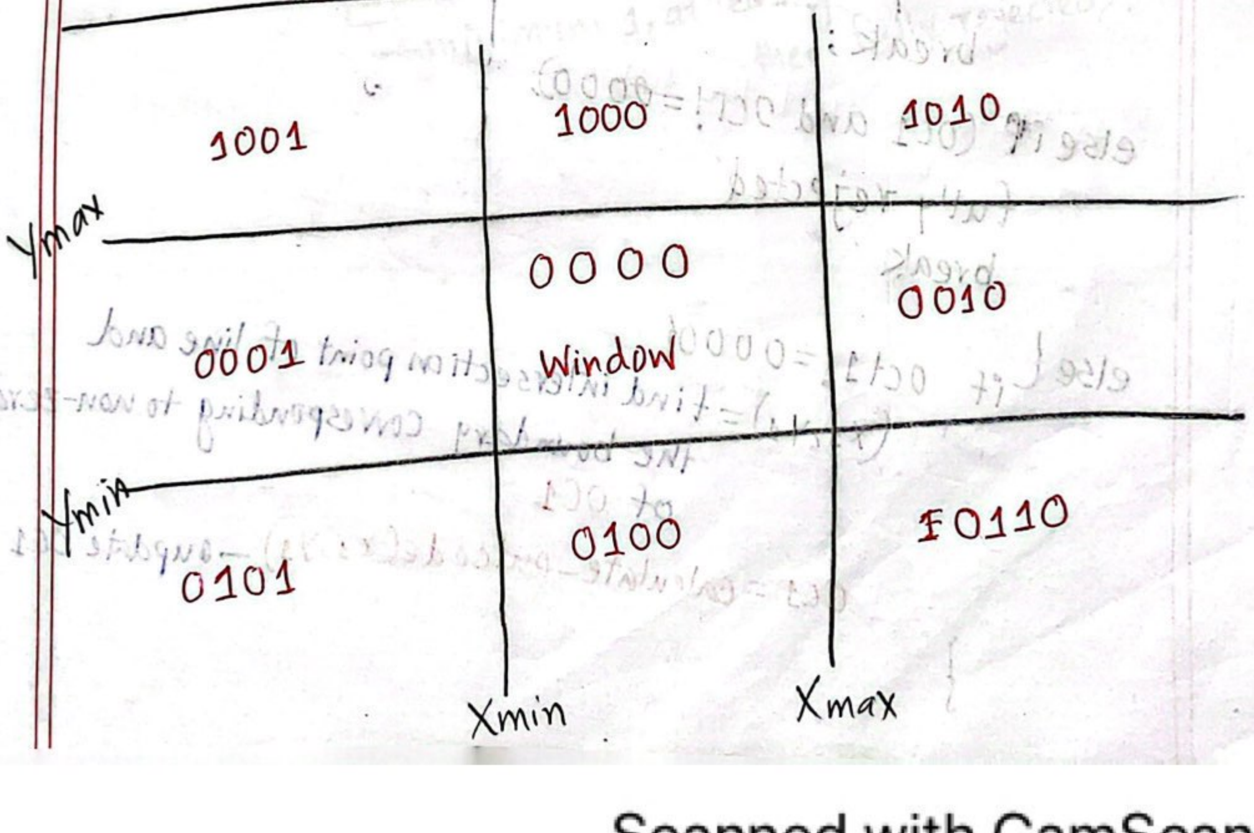
We will use 4 bit binary number as region code to name the regions.

Scanned with CamScanner

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



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.

Psendo Code:

```
Cohen-Sutherland(x1, y1, x2, y2):
OC1 = Calculate outcode(x1, y1)
OC2 = Calculate outcode(x2, y2)
while True:
    if (OC1 = OC2 = 0000)
        fully accepted
        break;
    elseif (OC1 and OCT != 0000):
        fully rejected
        break
    else{ if oct1 != 0000{
            (x1, y1) = find intersection point of line and
                       the boundary corresponding to non-zero bit
                       of OC1
            OC1 = calculate_outcode(x1, y1) → update OC1
    }
```

Scanned with CamScanner

Scanned with CamScanner

else {

$OC_1 = 0000$] (x_2, y_2) = find intersection point of line and the boundary corresponding to non-zero ~~set~~ bit of OC_2.

OC_2 = Calculate_outcode(x_2, y_2) → update OC_2

}

Continue

}

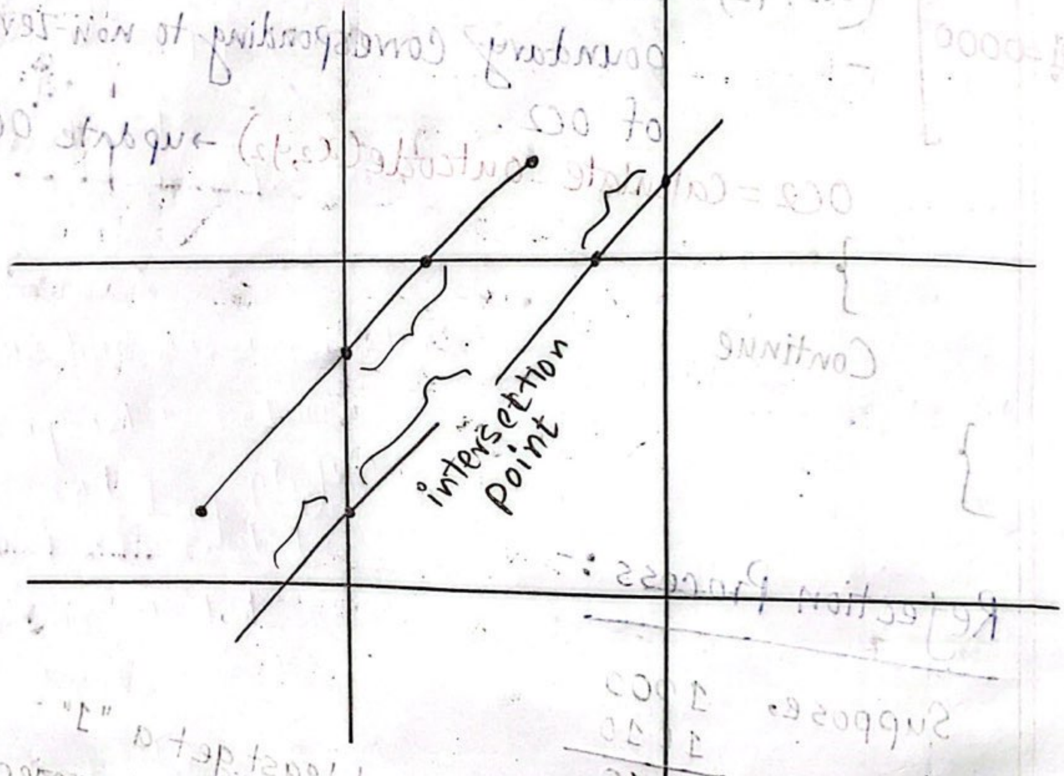
Rejection Process :-

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

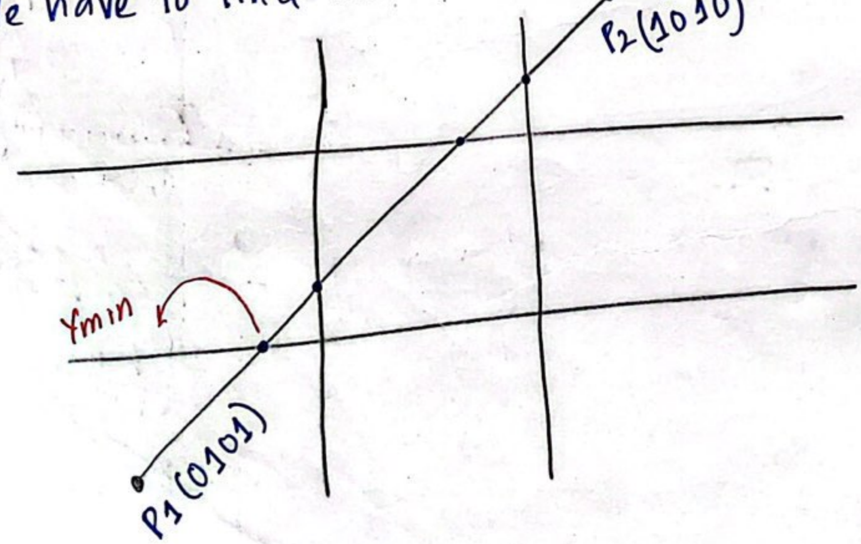
minimum 1, at least get a "1" then fully rejected.

Scanned with CamScanner

What if, partially accepted?



• We have to find the intersection point.

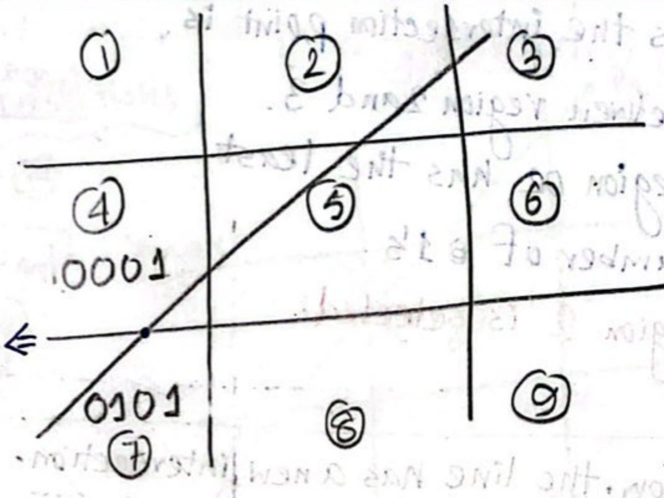


Scanned with CamScanner

This intersection point shares region 4 and 7.

To select the intersection point, we will select the region with least 1's.

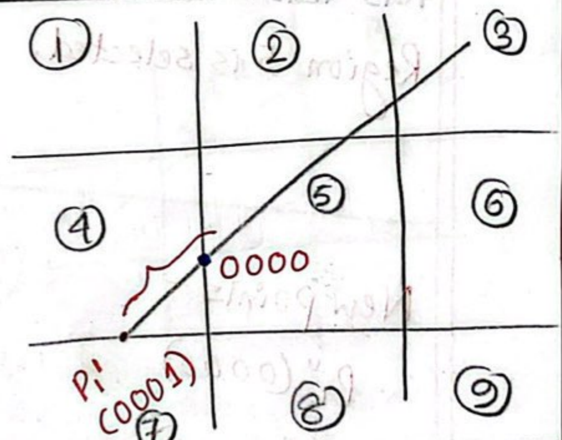
∴ Region 4 is selected.



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

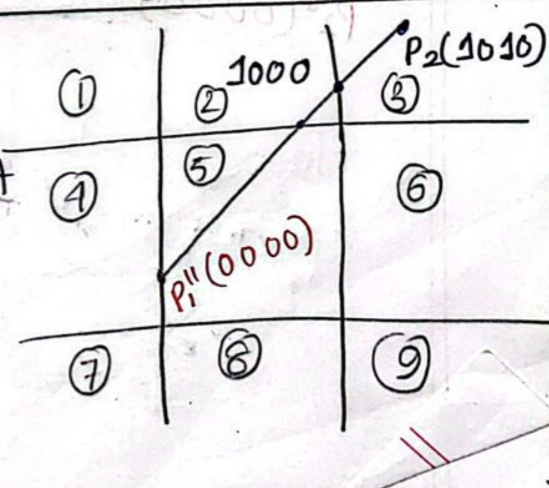
Number of 1's less in Region 5.

∴ Region 5 is selected.



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

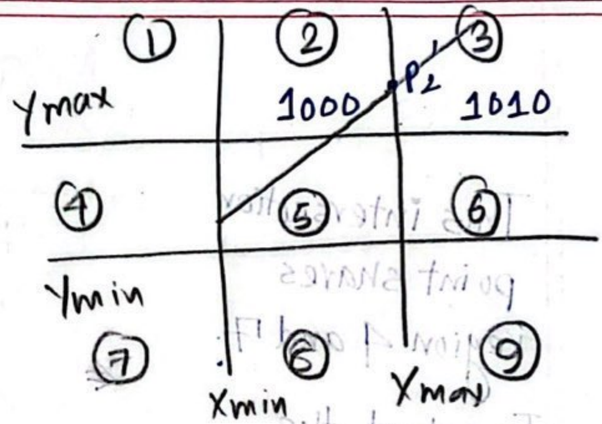
Now, P_2 is on 3rd region. And its not on the window. So, we would look for an intersection point.



Scanned with CamScanner

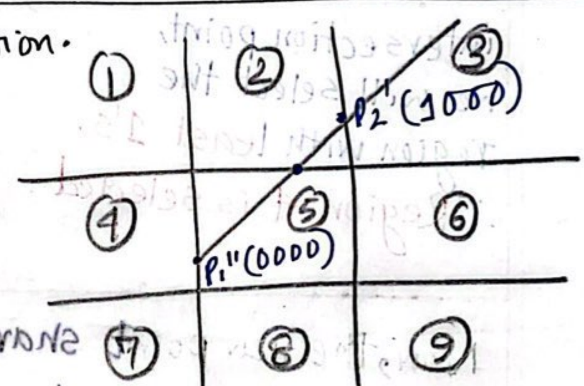
As the intersection point is betwen region 2and 3. Region 02 has the least number of 1's.

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.

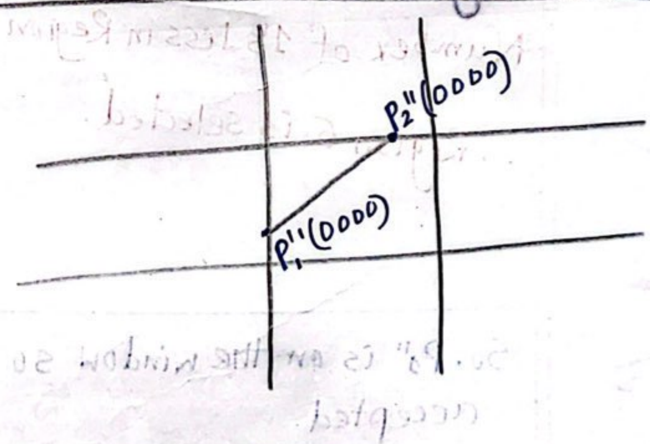
Region 5 is selected.



New points -

P_1''(0000)

P_2''(0000)



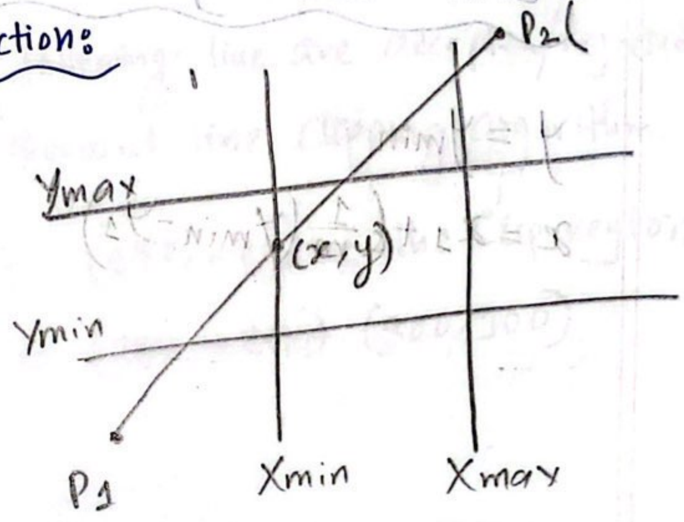
Scanned with CamScanner

Formula

· Left Boundary Intersection:

$X = X_{min}$

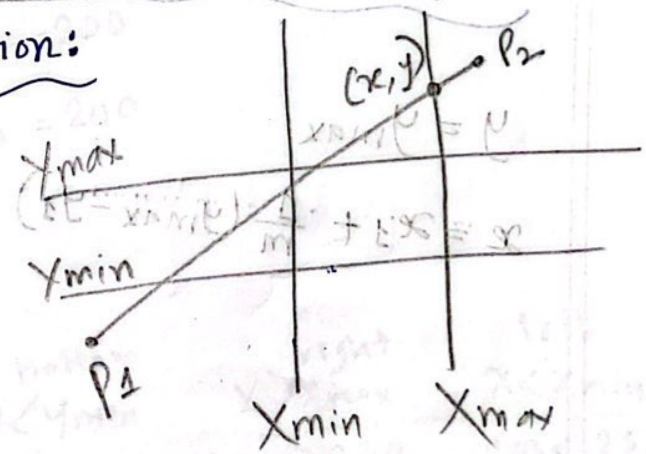
$y = y_1 + (X_{min} - X_1)$



· Right Boundary Intersection:

$X = X_{max}$

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



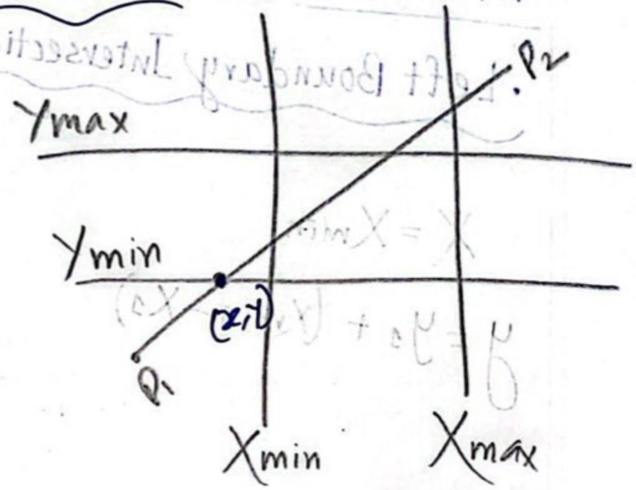
Scanned with CamScanner

Scanned with CamScanner

▫ Bottom Boundary Intersection:

$y = y_{min}$

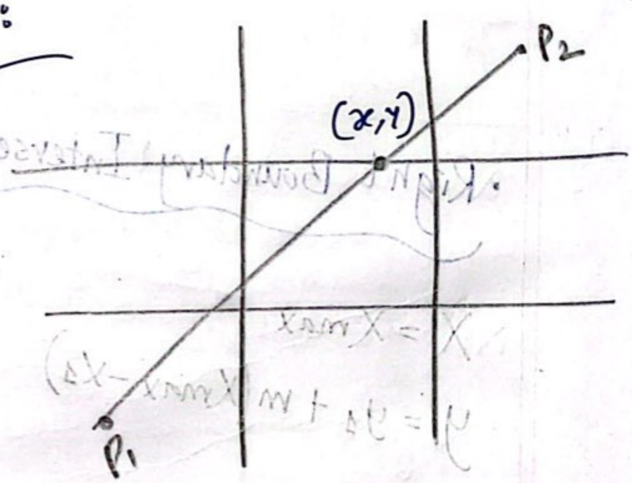
$x = x_1 + \frac{1}{m}(y_{min} - y_1)$



▫ 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.

Scanned with CamScanner

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) $(100, \cancel{220}\ 50)$ to $\cancel{(330, 240)}$ (300, 100)

Soln:

Step 01: Check Condition and Make Opcode.

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

$X_{max} = 250$ $Y_{max} = 200$.

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

	top $y > y_{max}$	bottom $y < y_{min}$	right $x > x_{max}$	left $x < x_{min}$
	50 > 200	50 < -200	100 > 250	100 < -250
OC1 = 0000	0	0	0	0
	100 > 200	100 < -200	300 > 250	300 < -250
OC2 = 0010	0	0	1	0

Scanned with CamScanner

Step 02: Check Loop Conditions

#First if Condition:

$\boxed{OC1 = OC2 = 0000} \rightarrow \times$

#Second if Condition:

$\boxed{OC1 \text{ and } OC2 \neq 0000} \rightarrow \times$

#3rd if Condition:

$\boxed{OC1 \neq 0000}$

↳ Find Non Zero Bit intersection

#4th if Condition:

$\boxed{OC2 \neq 0000} \rightarrow$ Find Intersection.

Step 03:

Intersect $P_2 \longrightarrow X_{max}$

$\rightarrow x$

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

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

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

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

Scanned with CamScanner

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

$= 100 - 12.5$

$= 87.5$

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

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

$\downarrow x$ $\quad \downarrow y$ (New Value)

Check Conditions again:

$x = 250 \quad y = 87.5$

$y > y_{max} \rightarrow 87.5 > 200 = 0$ [0 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)

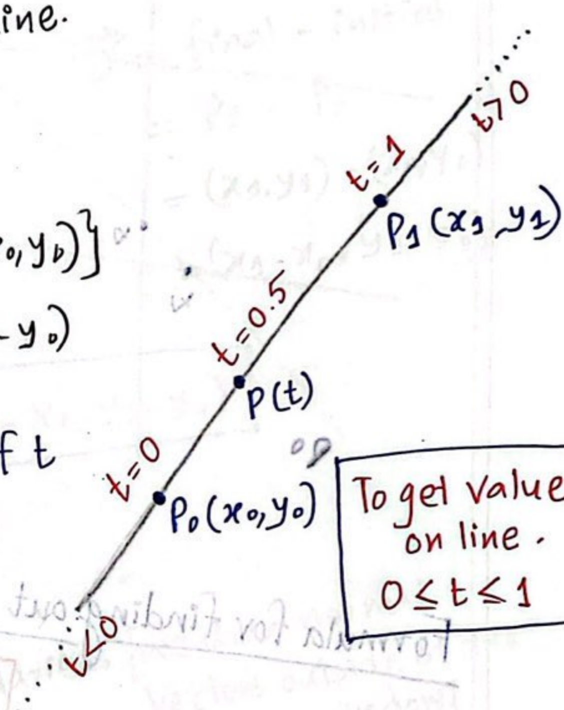
Again Start from Condition 1 from Loop.

Scanned with CamScanner

⊕ CyRus Beck Algorithm

Step 01: Parametric Eq^n of a Line.

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



To get value on line.
$0 \leq t \leq 1$

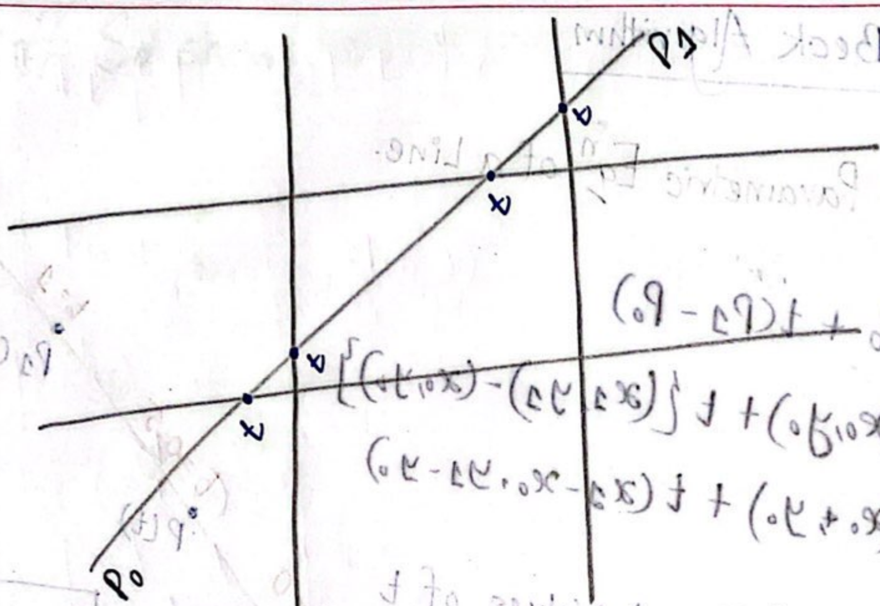
We can use different Values of t and get Values.

t=0,

$$P(0) = (x_0, y_0) \rightarrow \text{initial point}$$
$$P(1) = (x_0, y_0) + 1(x_1 - x_0, y_1 - y_0)$$
$$= (x_0 + x_1 - x_0, y_0 + y_1 - y_0)$$
$$= (x_1, y_1)$$

P.T.O

Scanned with CamScanner



Formula for finding out t :

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

Normal Vectors

Dot Product

Initial Point (x_0, y_0)

Random point on Boundary

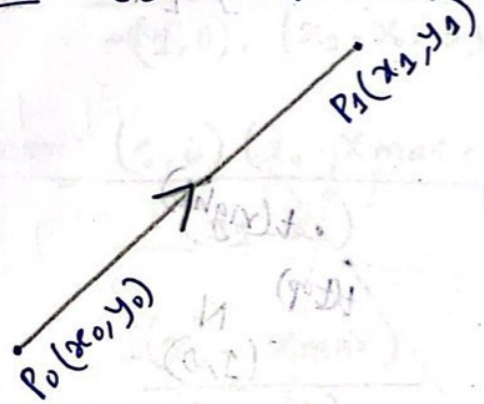
Dot Product

Normal Vector

$(x_1 - x_0, y_1 - y_0)$

Scanned with CamScanner

Step 02: Vector Representation of a Line.



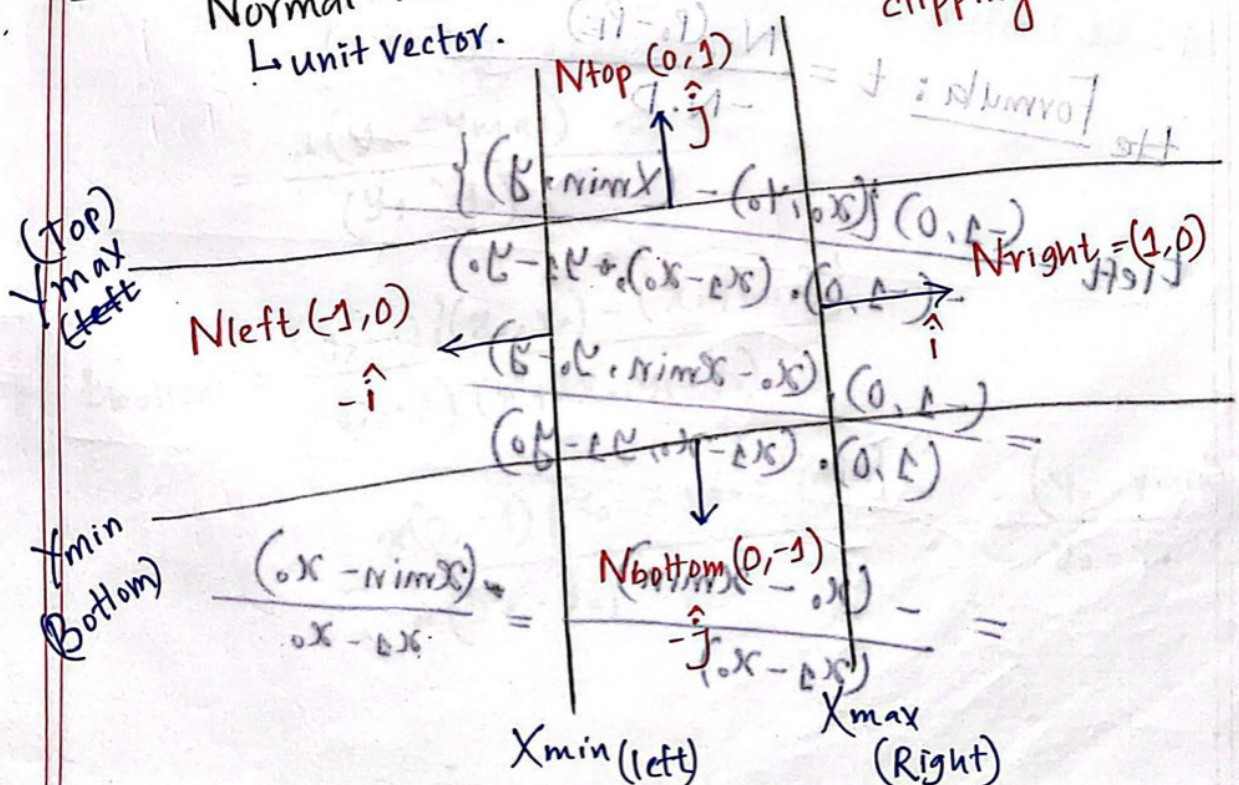
$$\vec{D} = \text{final} - \text{initial}$$
$$= P_1 - P_0$$
$$= (x_1, y_1) - (x_0, y_0)$$
$$= (x_1 - x_0, y_1 - y_0)$$

$$\boxed{\vec{D} = (x_1 - x_0, y_1 - y_0)}$$

Step 03:

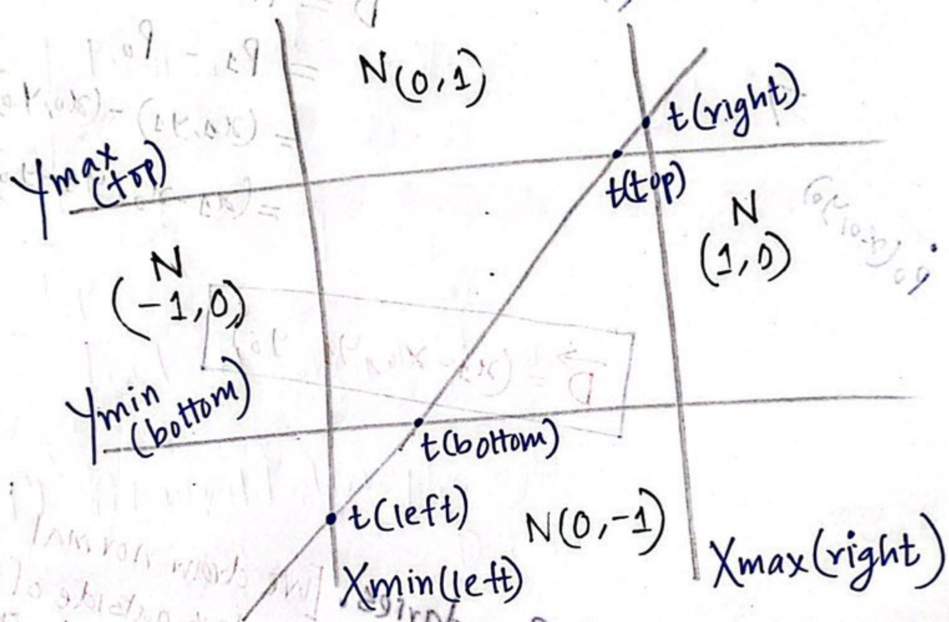
Normal Vectors of Boundaries. [We draw normal vectors outside of the clipping window]

↳ unit vector.



Scanned with CamScanner

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



~~the~~ Formula: $t = \frac{N.(P_0 - P_E)}{-N.D}$

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

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

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

Scanned with CamScanner

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

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

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

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

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

$$= \frac{-(y_0-y_{max})}{(y_1-y_0)}$$

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

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

Scanned with CamScanner

$$t_{top} = \frac{-(y_0 - y_{max})}{y_1 - y_0} \qquad t_{left} = \frac{\cancel{x_{min}} - (x_0 - x_{min})}{x_1 - x_0}$$

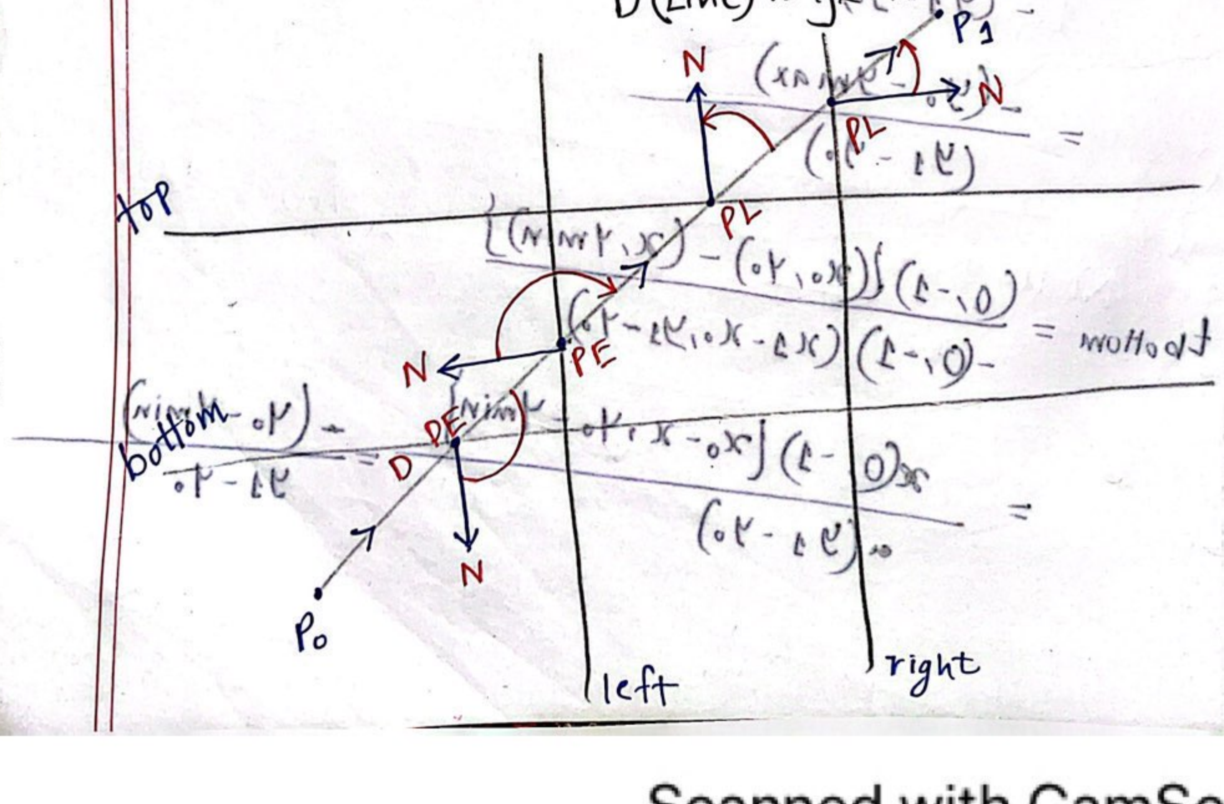
$$t_{bottom} = \frac{-(y_0 - y_{min})}{y_1 - y_0} \qquad t_{right} = \frac{-(x_0 - x_{max})}{x_1 - x_0}$$

Step 05:

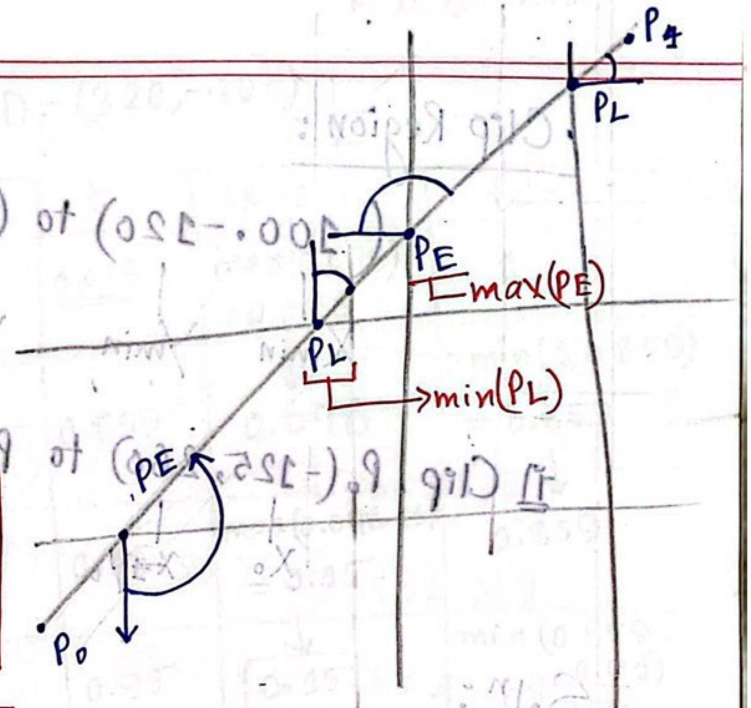
Find out when to draw and when to discard.

PE (Potential Entering): Angle between N (Normal) and D (Line) is greater than 90°

PL (Potential Leaving): Angle between N (Normal) and D (Line) is ~~greater~~ less than 90°



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

Discard Condition

PE ⟶ N.D < 0

PL ⟶ N.D > 0

Scanned with CamScanner

• **Clip Region:**

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

X_{min} Y_{min} X_{max} Y_{max}

i) Clip, $P_0(-125, 260)$ to $P_1(195, -140)$

X_0 $\cancel{X_1}Y_0$ X_1 Y_1

Soln:

Line Vector $= (x_1 - x_0, y_1 - y_0)$

$= (195 - (-125), -140 - 260)$

$= (320, -400)$

Scanned with CamScanner

$N.D < 0 \rightarrow PE$

$N.D > 0 \rightarrow PL$

initially $\boxed{t_E = 0,\ t_1 = 1}$; $D = (320, -400)$

if $t_E < t_L$

Boundary	N_i	$N_i.D$	P_E/P_L	t	$t_E = 0$	$t_L = 1$
left	(-1,0)	-320	PE	0.078	max(0,0.078) = 0.078	1
right	(1,0)	320	P_L	0.859	0.078	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	P_L	0.95	$\boxed{0.15}$ t_E	min(0.859, 0.95) = $\boxed{0.859}$ t_L

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

if $t_E > t_L$
L line segnet outside window

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

$$t_{top} = \frac{-(y_0 - y_{max})}{y_1 - y_0} = \frac{-(260-200)}{-140-260} = 0.15$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{y_1 - y_0} = \frac{-(260+200)}{-140-260} = 0.95$$

Scanned with CamScanner

if $t_E < t_L \longrightarrow$ draw

$0.15 < 0.859 \longrightarrow$ we can draw.

Step 01

$P(t) = (-125, 260) + t(320, -400)$

$P(0.15) = (-125, 260) + 0.15(320, -400) = (-77, 200)$

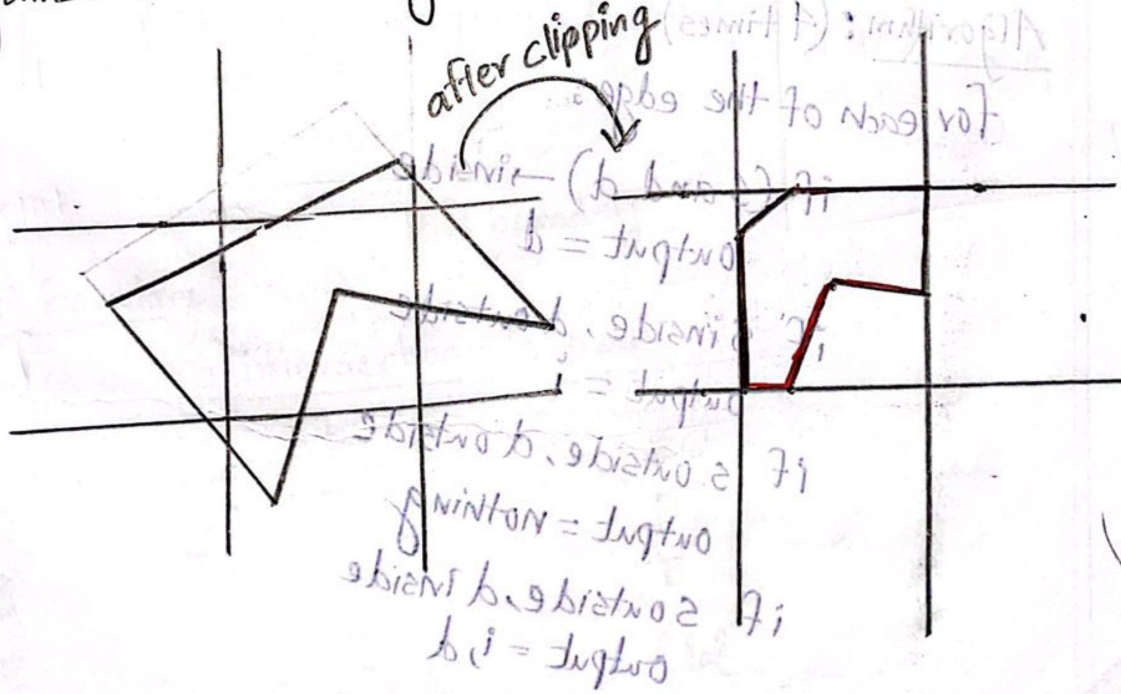
$P(0.859) = (-125, 260) + 0.15(320, -400) = (150, -84)$

Scanned with CamScanner

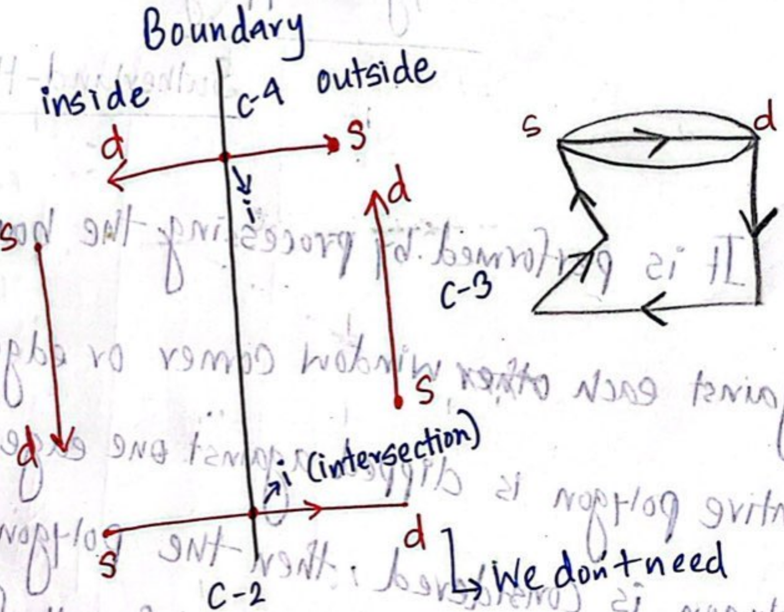
Polygon Clipping

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.



Scanned with CamScanner



Algorithm: (4 times)

for each of the edge:

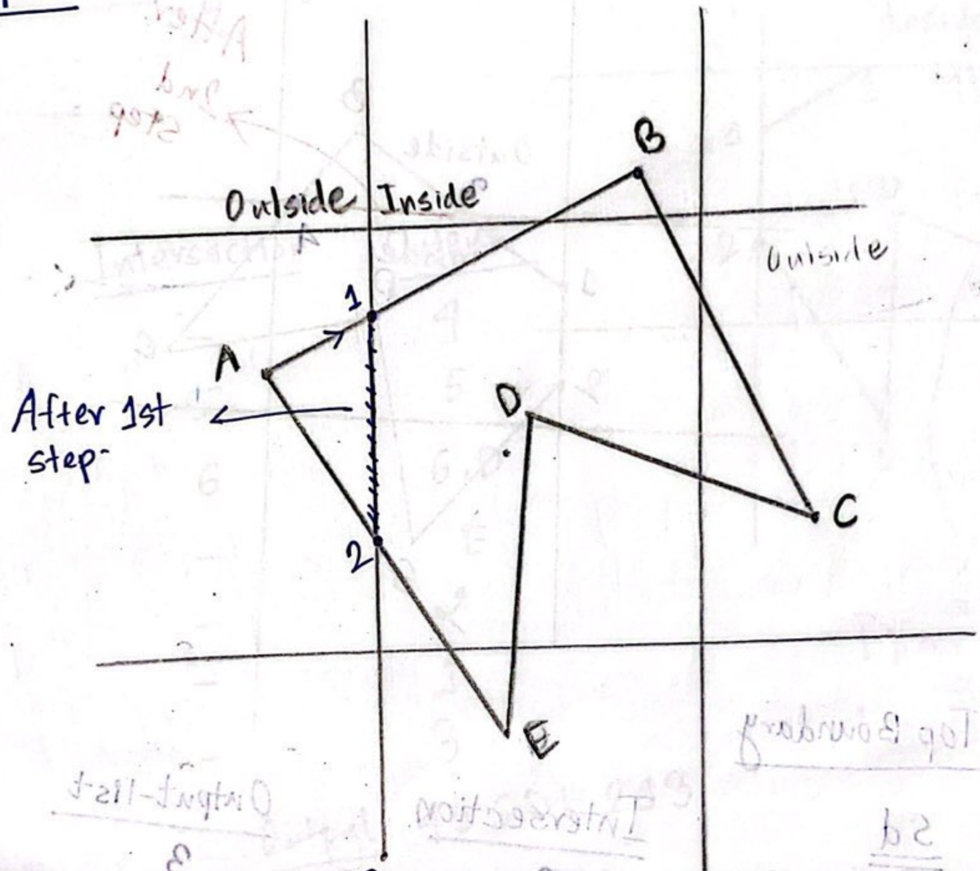
if (s and d) → inside
output = d

if s inside, d outside
output = i

if s outside, d outside
output = nothing

if s outside, d inside
output = i, d

Example:



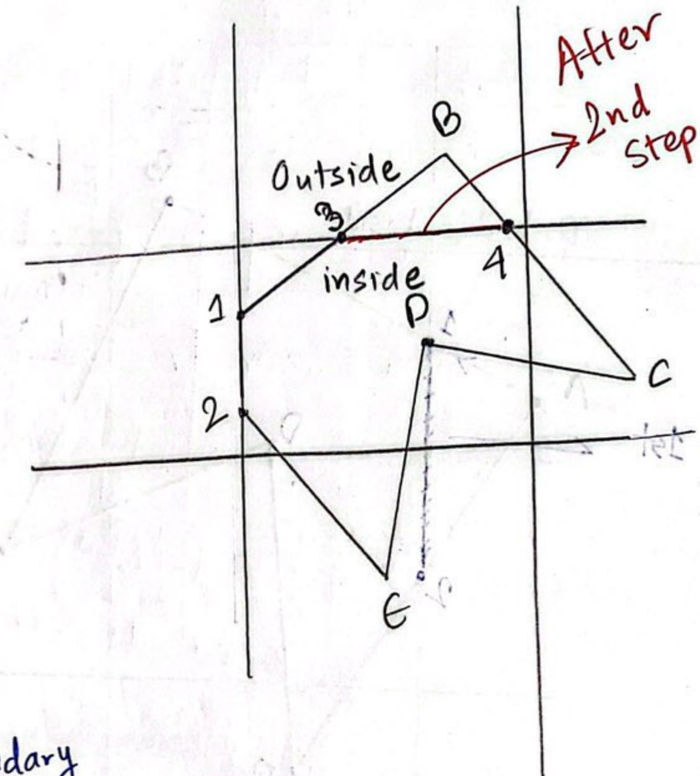
left boundary

Left Boundary

sd	intersection	Output list
AB	1	1, B
BC	–	BC
CD	–	D
DE	–	E
EA	2	212

Output list = 1, B, D, E, 2

Scanned with CamScanner



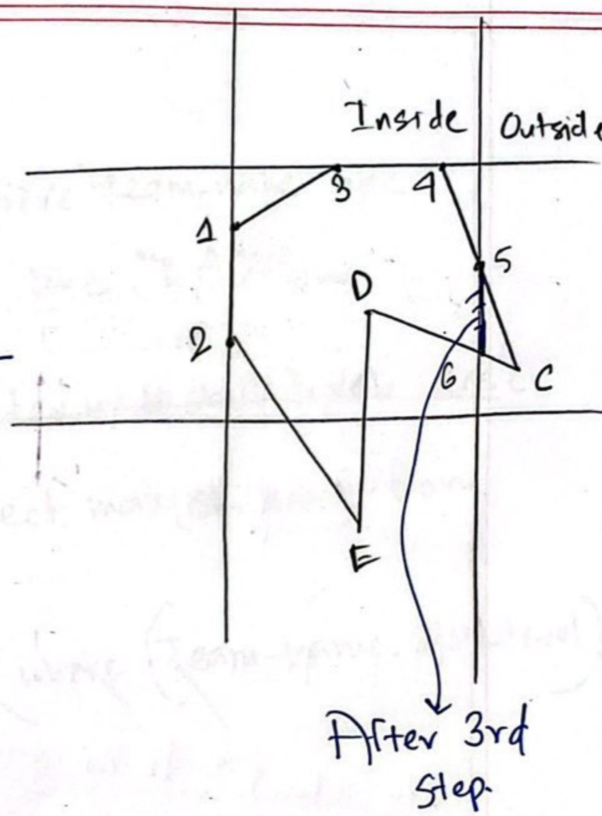
Top Boundary

Sd	Intersection	Output-list
1B	3	3
BC	4	4, C
CD		D
DE		E
E 2		2
2 1		1

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

Scanned with CamScanner

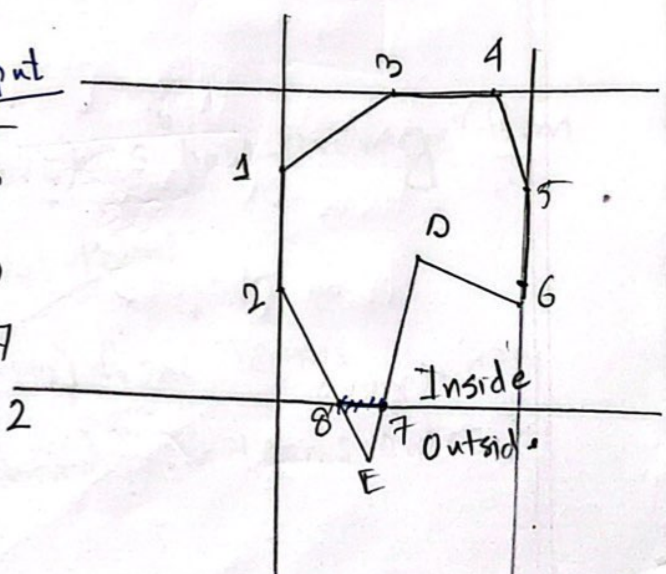
Right Boundary



Sd	Intersection	Output
34	–	4
4C	5	5
CD	6	6, D
DE	–	E
E2	–	2
21	–	1
13	–	3

Output: 456DE213

Bottom Boundary



Sd	Intersection	Output
45	–	5
56	–	6
6D	–	D
DE	7	7
E2	8	8, 2
21	–	1
13	–	3
34	–	4

Output: 56D782134

Scanned with CamScanner

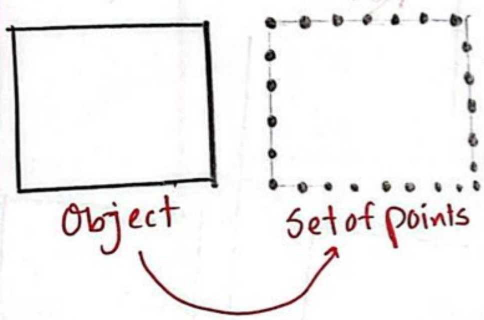
CSE-423

Transformation

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

Reposition and resizing of an object.

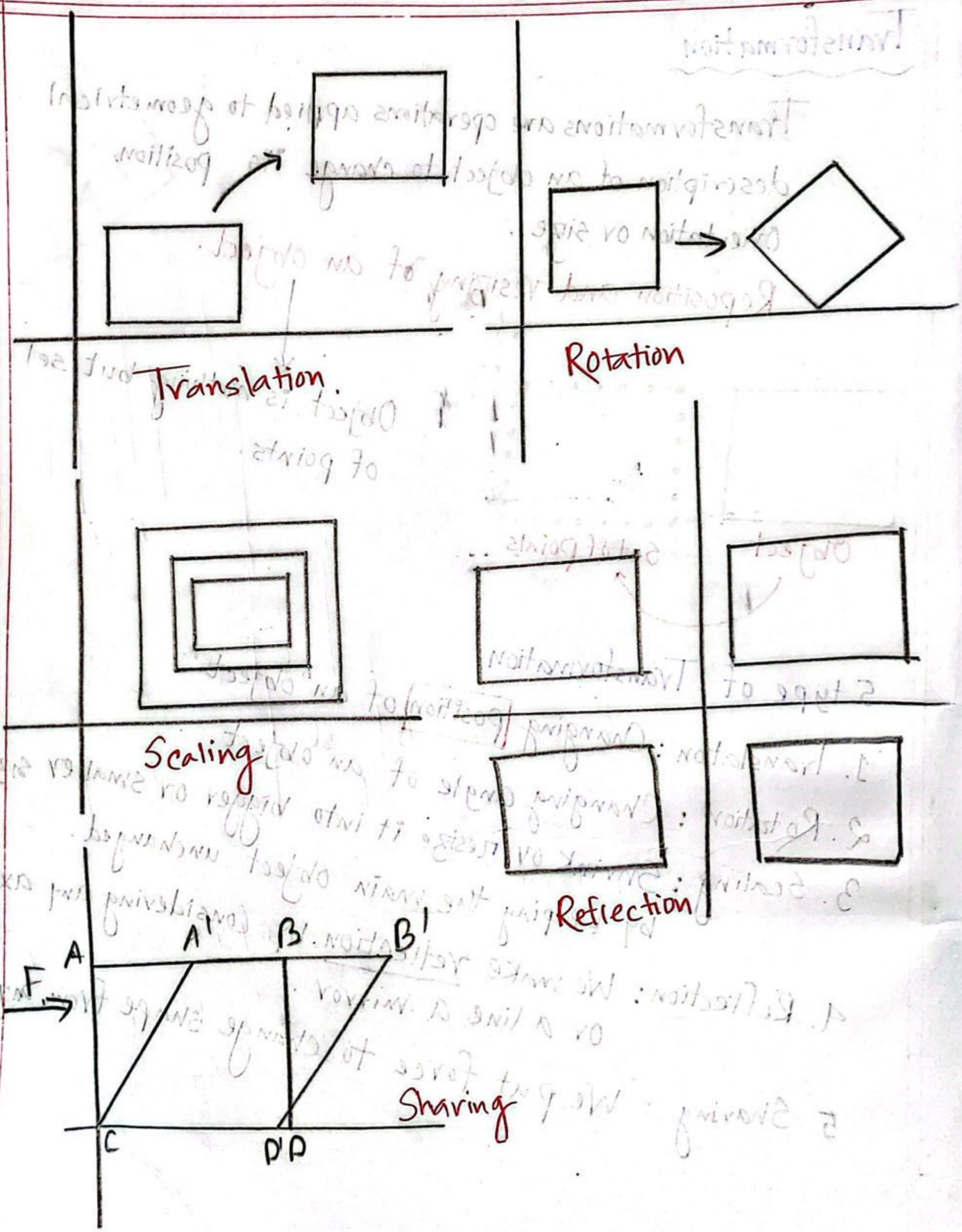
Object is nothing but set of points.



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. Sharing : We put force to change shape from any side.

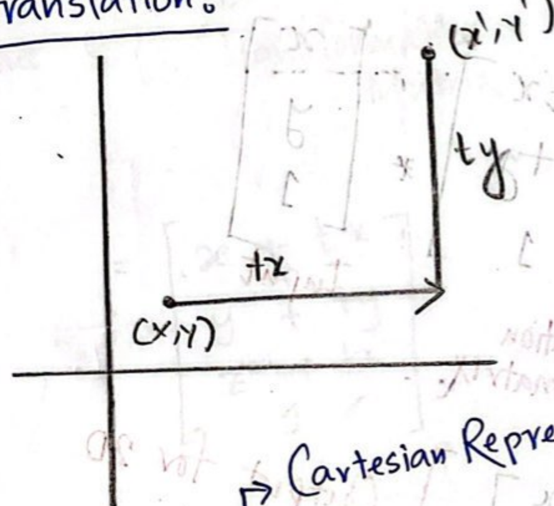
Scanned with CamScanner



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:



$$x' = x + tg$$
$$y' = y + ty$$

→ Cartesian Representation

$$\underset{\text{Output}}{\begin{bmatrix} x' \\ y' \end{bmatrix}} = \underset{\text{Transformation Matrix}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} * \underset{\text{Input}}{\begin{bmatrix} x \\ y \end{bmatrix}} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

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

Scanned with CamScanner

Homogeneous Representation:

For n-dimension

we use $(n+1)*(n+1)$ matrix.

2D

So, for 2D - we use $3*3$ Matrix

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

Output — Transformation matrix — Input

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

Output for 2D

3D

we need $(4*4)$ Matrix

$x' = x + tx$

$y' = y + ty$

$z' = z + tz$

Scanned with CamScanner

Scanned with CamScanner

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

Output Transformation Matrix Input

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

Note:

Transformation Matrix for Translation:

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

Scanned with CamScanner

Example

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

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

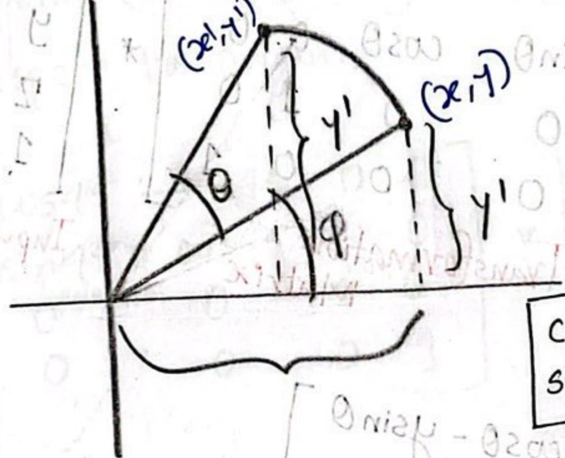
$$= \begin{bmatrix} 9 \\ -3 \\ 12 \\ 1 \end{bmatrix}$$

$\therefore$ point (9, -3, 12)

Scanned with CamScanner

Rotation:

z based rotation $\longrightarrow z' = z$



We rotate it θ amount anti-clockwise.

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

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

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

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

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

$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned}}$$

as it is being rotated z based so z is unchanged.

Scanned with CamScanner

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

Output Transformation Matrix Input

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

Scanned with CamScanner

Example

Rotate point (9,2,5) at angle 45° (anti-clockwise) with respect to origin.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos45 & -\sin45 & 0 & 0 \\ \sin45 & \cos45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 9 \\ 2 \\ 5 \\ 1 \end{bmatrix}$$

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

Answer: (4.95, 7.78, 5)

P.T.O

Scanned with CamScanner

Example

Rotate Point (6,2) at an angle 90° anti-clockwise with respect to point (2,2)?

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

Sequence: $\boxed{\text{Translate to origin} \rightarrow \text{Rotation} \rightarrow \text{Translate to origin}}$

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

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

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

∴ Rotation Point (2,6)

Scanned with CamScanner

Note:

- Output for Rotation with respect to another point:

$$\text{Output} = \underbrace{T(x,y)}_{\text{Translation to original}} * \underbrace{R(\theta)}_{\text{Rotation}} * \underbrace{T(x,y)}_{\text{Translation to origin}} * \text{Input}$$

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

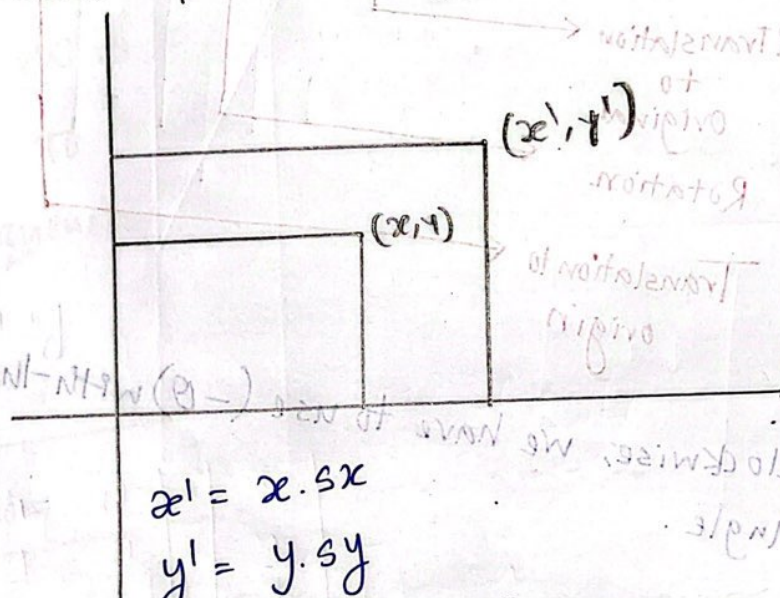
Scanned with CamScanner

Scaling:

Cases:

Scale sx units in x with respect to origin.

Scale sy units in y with respect to origin.



$x' = x \cdot sx$

$y' = y \cdot sy$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Scaling Matrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot sx \\ y \cdot sy \\ 1 \end{bmatrix}$$

Scanned with CamScanner

Scanned with CamScanner

Example

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

Soln:

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

Example

Scale point (3,5) $\left\{ \begin{array}{l} 5 \text{ units} \longrightarrow x \\ 8 \text{ units} \longrightarrow y \end{array} \right\}$ with respect to (8,6)

Soln: Output = T(8,6) * Scale(5,8) * T(-8,-6) * Input

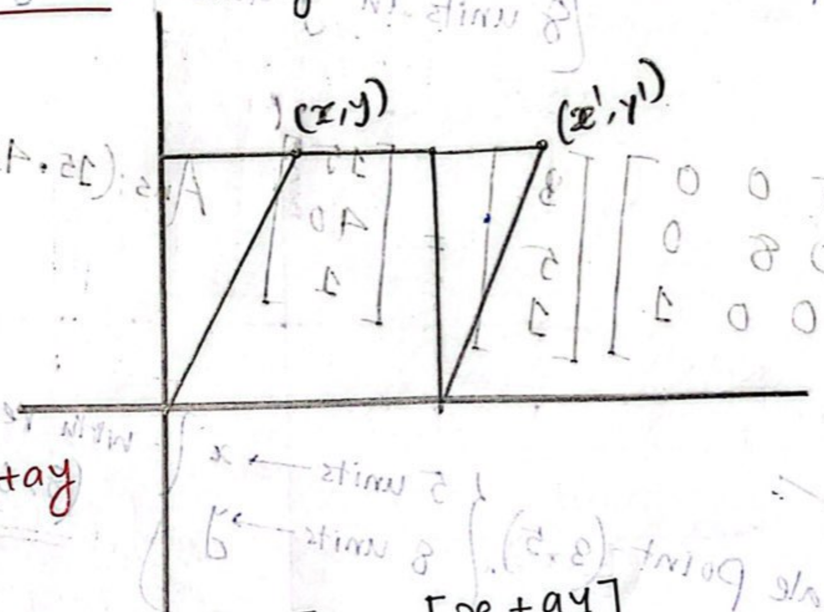
$$= \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 5 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -8 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

=

Scanned with CamScanner

Shearing:-

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



$x' = x + ay$

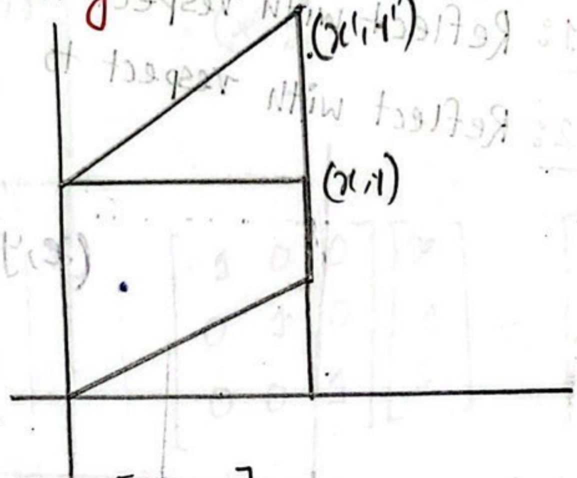
$y' = y$

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

Scanned with CamScanner

Case 02:

Shearing with y-axis. (b amount)



$x' = x$

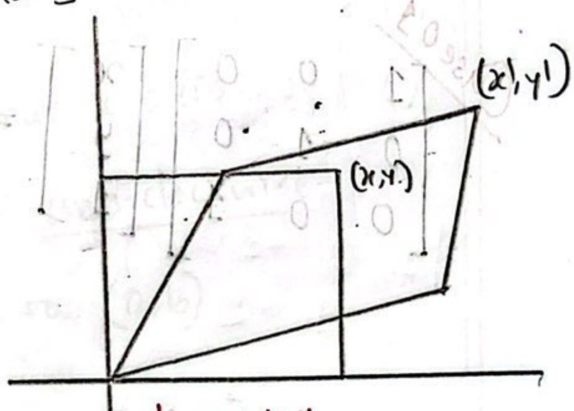
$y' = y + bx$

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

Case 03:

Shearing with both x and y axis.

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



$x' = x + ay$

$y' = y + bx$

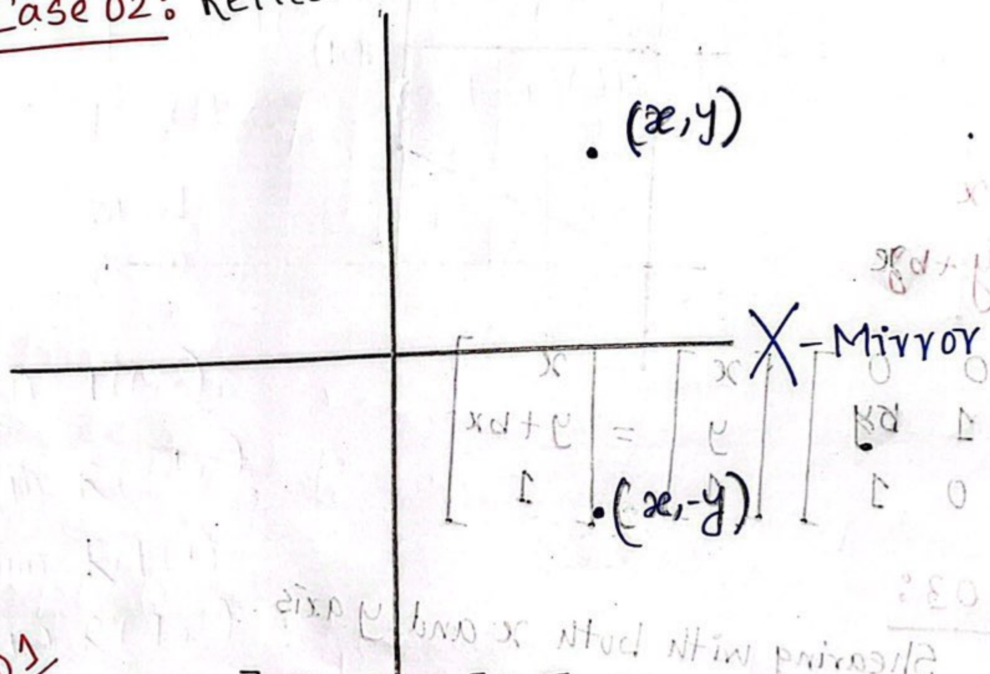
$z' = z$

Scanned with CamScanner

Reflection:

Case 01: Reflect with respect to x axis.

Case 02: Reflect with respect to y-axis.

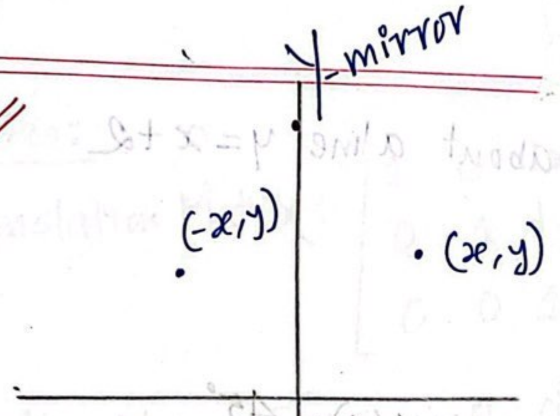


Case 01

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

Scanned with CamScanner

Case:02



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

Case 03: Reflection with respect to a line.

Step 01: Translate (a,b) 0,b) to origin.

Step 02: Rotate- θ- degrees clockwise

Step. 03: Mirror reflect about x-axis.

Step 04: Rotate θ- degree. anti-clockwise.

Step 05: Translate origin to (0,b)

Scanned with CamScanner

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

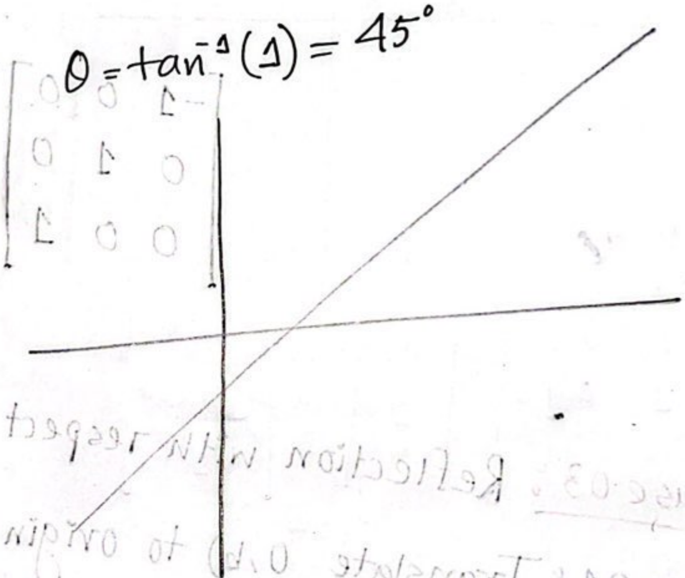
Soln:

$y = mx + c$

$y = x + 2$ $\qquad \theta = \tan^{-1}(1) = 45^\circ$

$m = 1$

$c = 2$



(i) $T(0,-2)$

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

(iii) $Ref(x)$

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

(v) $T(0,2)$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}}_{T(0,2)} \times \underbrace{\begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(\theta)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{Ref(x)} \underbrace{\begin{bmatrix} \cos(-45) & -\sin(-45) & 0 \\ \sin(-45) & \cos(-45) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(-45^\circ)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}}_{T(0,-2)} \begin{bmatrix} 10 \\ 5 \\ 0 \end{bmatrix}$$

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

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
$$\underset{\text{X-axis}}{\begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \quad \underset{\text{Y-axis}}{\begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \quad \underset{\text{both axis}}{\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

vi) Reflection:
$$\underset{\text{X-axis}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \quad \underset{\text{Y-axis}}{\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

$T(0,b)$
$R_o(\theta)$
Ref
$R_o(\theta)$
$T(0,-b)$

Scanned with CamScanner

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 Substractive color model.

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

Scanned with CamScanner

Scanned with CamScanner

Primary color of light / (RGB)

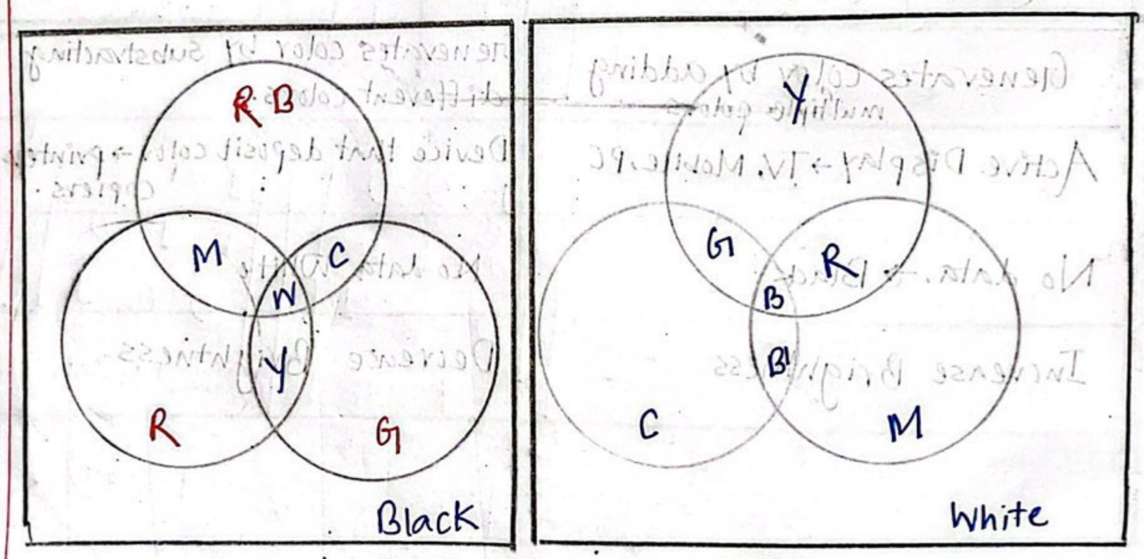
red(R); Green (G), Blue (B)

Secondary Colors of light (CMY)

Cyan(C) → Green + Blue

Magenta → Red + Blue

Yellow → red + green.



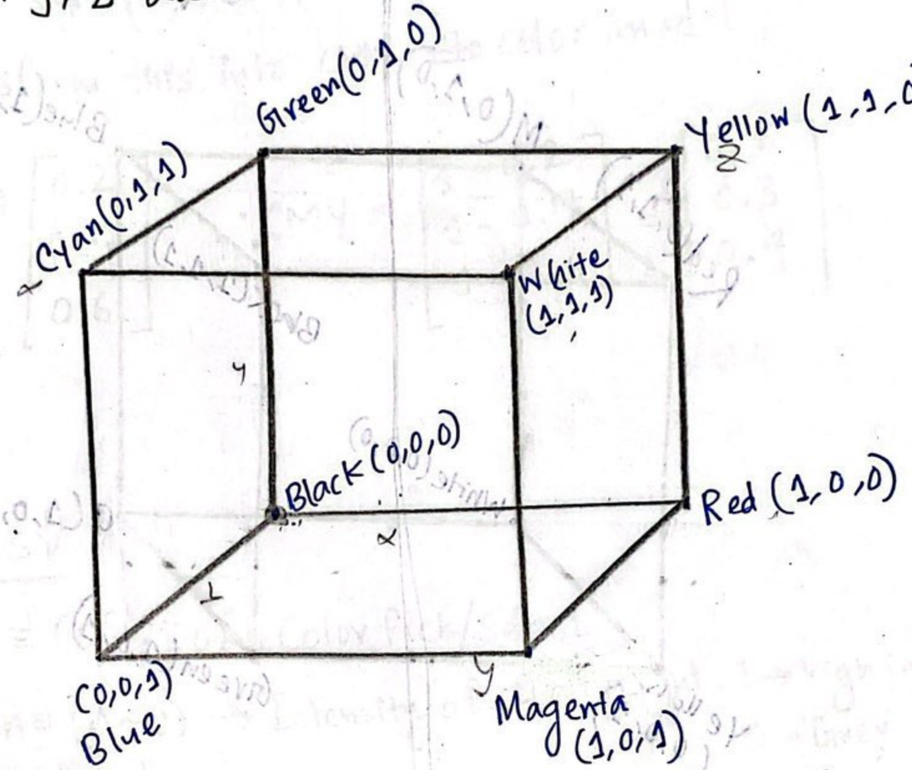
RGB (1, 0, 0) → Red Ranges (0 - 1)

RGB (1, 1, 0) → Yellow

Scanned with CamScanner

Color Cube

x, y, Z axis.

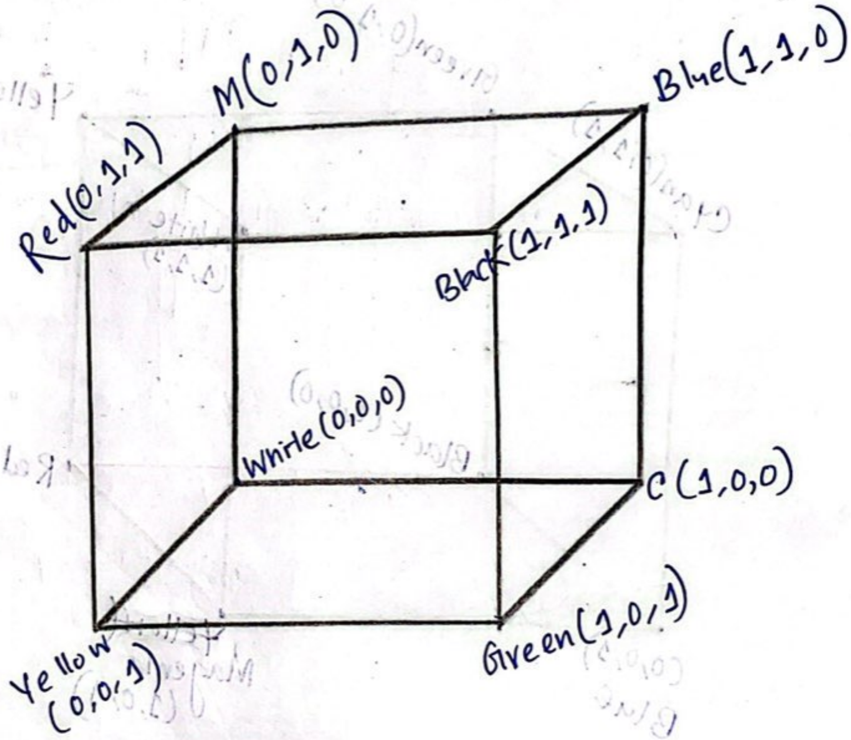


Total Combination that can be generated $= 2^8 \times 2^8 \times 2^8$

$= 16$ million.

Scanned with CamScanner

CMY Color Cube



RGB → CMY

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

* Switching between RGB and CMY

Scanned with CamScanner

Scanned with CamScanner

Example

Let RGB(0.2, 0.7, 0.6)

Transform this into CMY ~~clo~~ color model.

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

HSL/HSV

HSL

Hue ≡ (0 - 360°) → Color Pick/Select.

Saturation ≡ (0 ~ 1) → Intensity of Color/Light 1 → high intensity, 0 → Grey

Lightness (0 ~ 1) → Background Light 1 → White, 0 → Black

Hue ≡ (0 ~ 360°)

Saturation ≡ (0 ~ 1)

Value ≡ (0 ~ 1) → Brightness.

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

Scanned with CamScanner

Red → (0-60°)

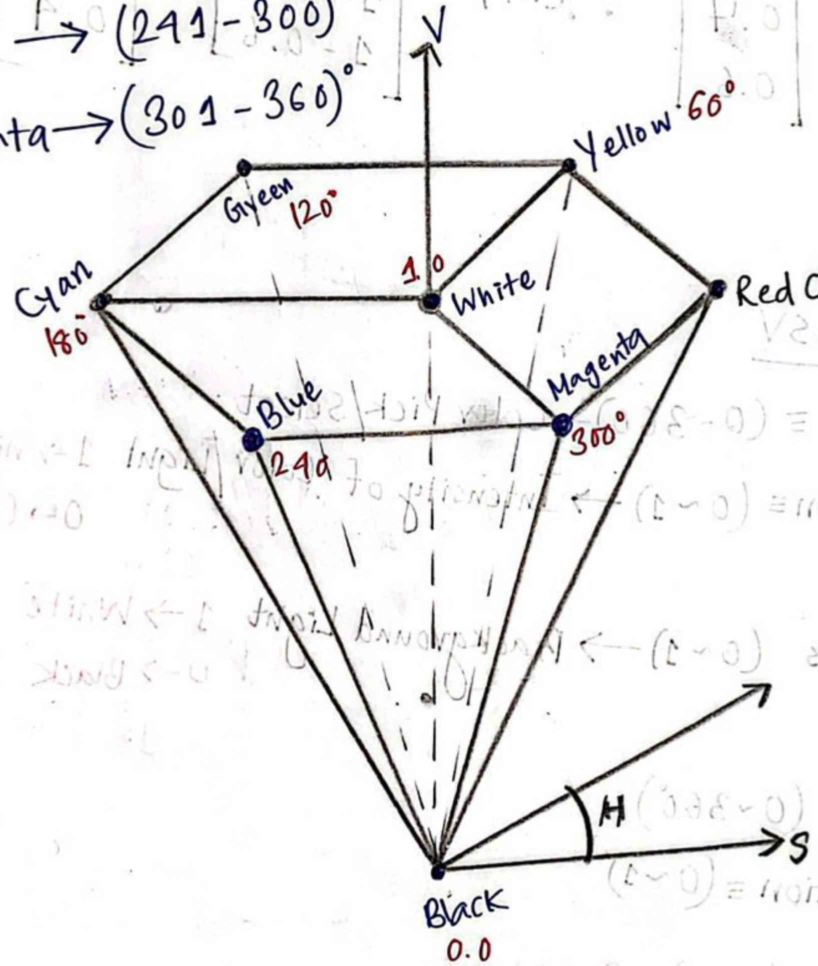
Yellow → (61 - 120)°

Green → (121 - 180)°

Cyan → (181 - 240)°

Blue → (241 - 300)°

Magenta → (301 - 360)°



Scanned with CamScanner

RGB to HSL/HSV

Step 01:

Find out Maximum and Minimum Value from RGB Values.

$Max = max(R, G, B)$

$Min = min(R, G, B)$

$d = Max - Min$

Step 02:

Find out ~~H~~ Lightness and Saturation for HSL

& Find out Value and Saturation for HSV.

HSL → Lightness: $\frac{Max + Min}{2}$

HSL → Saturation: $\frac{d}{1 - |2L - 1|}$ here L = Lightness

HSV → Saturation: $\frac{d}{Max}$

HSV → Value : Max(Max Lightness)

Scanned with CamScanner

Step 03:

Calculate Hue

if R==Max:

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

if G==Max:

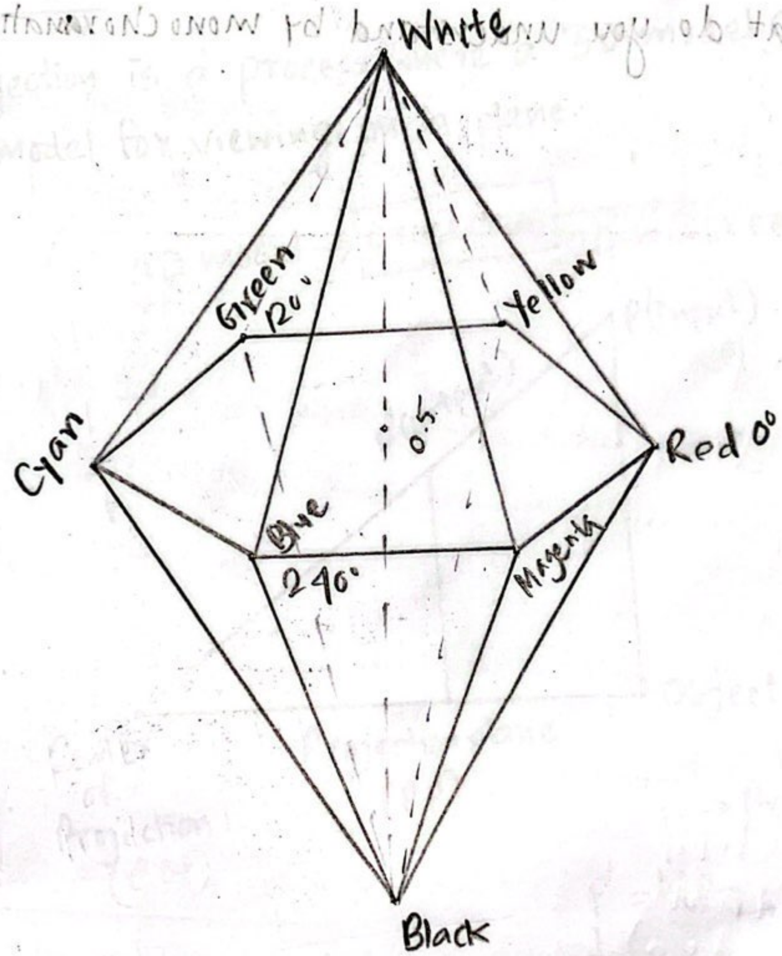
$$H=\left(\frac{B-R}{d}\right)*60^\circ+120^\circ \quad \left[\begin{array}{l}\text{By default} \\ \text{add } 120^\circ\end{array}\right]$$

if B==Max:

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

Scanned with CamScanner

HSL Color Model

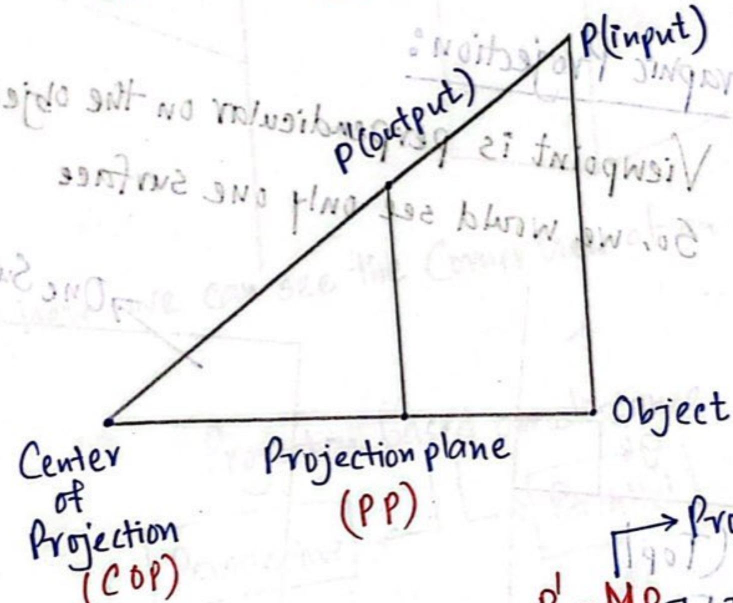


Scanned with CamScanner

Projection

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

3D model → $\boxed{\text{projection}}$ → 2D model output.



$P' = MP$

P' → Output

M → Projetion Matrix

P → Input

Scanned with CamScanner

Projection

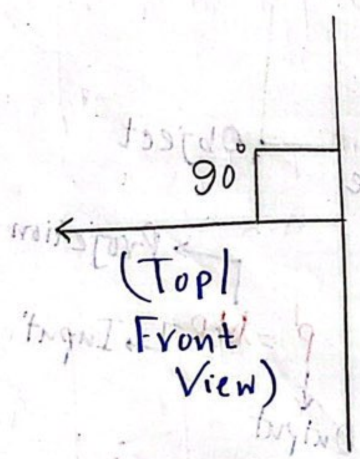
Orthographic Projection

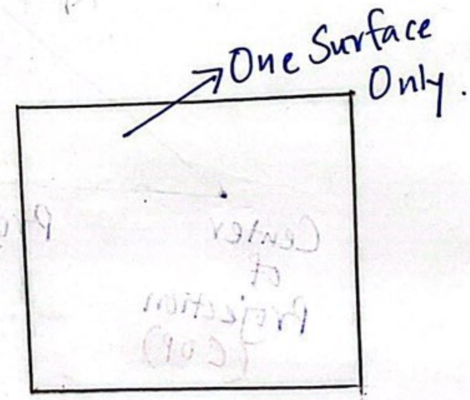
Oblique Projection

1) Orthographic Projection:

Viewpoint is perpendicular on the object.

So, we would see only one surface

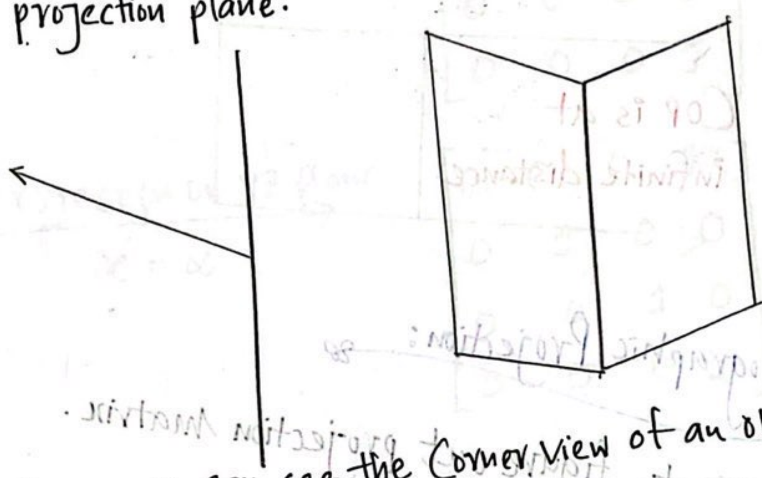




Scanned with CamScanner

Scanned with CamScanner

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

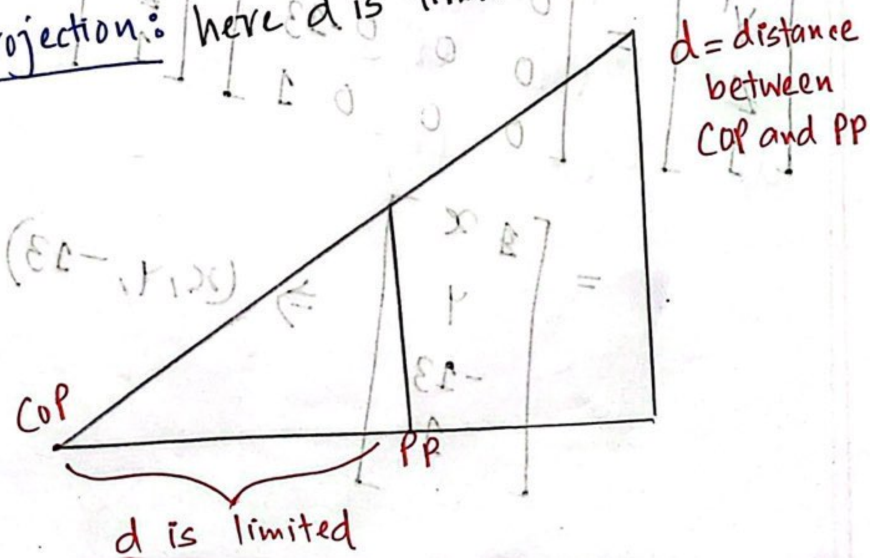


So ~~where~~ here we can see the Corner view of an object.

Projection based on distance

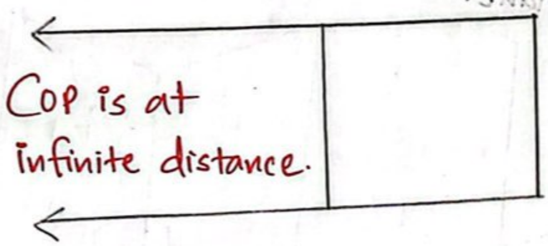
$\boxed{\text{Perspective}}$ ← → $\boxed{\text{Parallel}}$

i) **Perspective Projection:** here d is limited



Scanned with CamScanner

ii) Parallel Projection: d is infinite. Example: Telescope



Orthographic Projection:

we have to figure out projection matrix.

Example

$$(x,y,z) \xrightarrow[z=-13]{\text{projection on xy plane}} (x,y,-13)$$

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

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

Scanned with CamScanner

Formula:

$$(x, y, z) \xrightarrow[z = \alpha]{\text{projection on xy plane}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(x, y, z) \xrightarrow[x = \alpha]{\text{projection on yz plane}} \begin{bmatrix} 0 & 0 & 0 & \alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(x, y, z) \xrightarrow[y = \alpha]{\text{projection on zx plane}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

General Projection Matrix

for xy plane

$p' = mp$

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

$dx = cop.x - pp.x$

$dy = cop.y - pp.y$

$dz = cop.z - pp.z$

cop.x = center of projection at x axis

pp.x = projection plane at x coordinate

Scanned with CamScanner

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

$dx = COP_x - PP_x = 50 - 0 = 50$
$dy = COP_y - PP_y = 40 - 0 = 40$
$dz = COP_z - PP_z = 100 - (-200) = 300$

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

$$= \begin{bmatrix} 115/3 \\ 170/3 \\ 700/3 \\ \boxed{1.167} \end{bmatrix} \xrightarrow{\text{Normalize}} \begin{bmatrix} 32.85 \\ 48.5 \\ -200 \\ 1 \end{bmatrix}$$

As this result is not in 1. So we have to normalize it

Scanned with CamScanner

Case 01

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

COP (0, 0, 0)
PP (0, 0, d)

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

Projection Matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \cancel{d}\,0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$

$d\left(1 + \frac{d}{-d}\right)$
$= d \times (1 - 1)$
$= 0$

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

Scanned with CamScanner

Case 02:

Find out the matrix where $\boxed{\text{PP is at origin}}$ and $\boxed{\text{COP is at d-distance}}$ from PP

COP(0, 0, d)

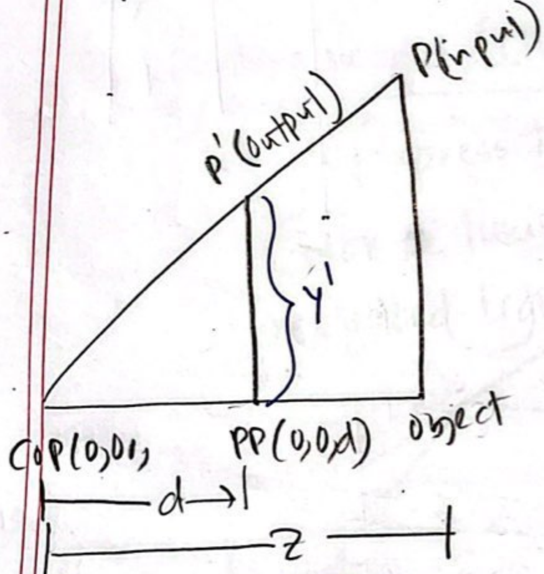
PP (0, 0, 0)

$dx = 0$

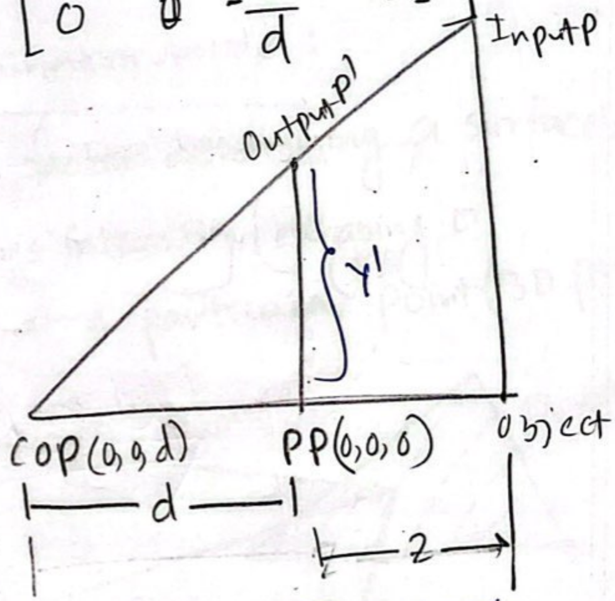
$dy = 0$

$dz = d - 0 = d$

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



Origin at COP



PP at origin

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

Lighting / Illumination / Reflection Model

Illumination:

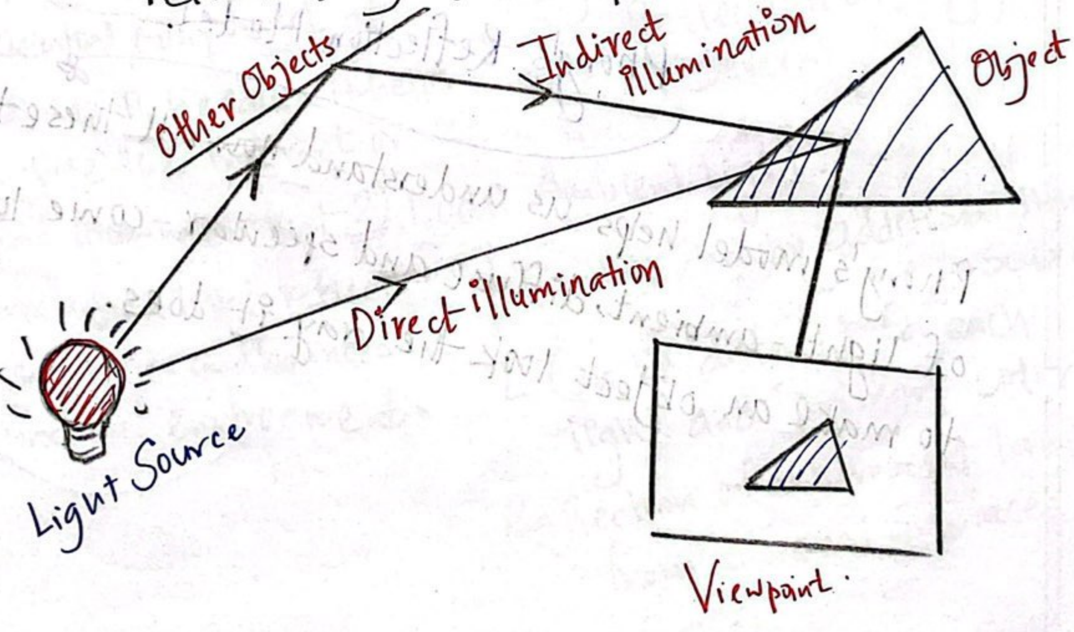
The transport of energy from light Sources to surfaces & points.

→ Local illumination

→ Global illumination.

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)



Scanned with CamScanner

Scanned with CamScanner

☐ Components of Illumination:

1. Light source
2. Surface Properties.

I = luminance/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.

Scanned with CamScanner

$$I = I_a + I_d + I_s$$

Final Intensity — Ambient — Diffuse — Specular.

Ambient Light

- No identifiable Source or direction
- Product of multiple reflections of light from the many surface 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 fluroscent lamps from several Surfaces.

* Ambient light is an essential Component of the model because it helps simulate the indirect or global illumination that occurs when light reflects off surfaces and bounces around the environment.

Scanned with CamScanner

Scanned with CamScanner

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 ~~under~~ equally on all objects.

$$\boxed{A = I_a k_a}$$

here, I_a = Intensity of the ambient light

k_a = Ambient co-efficient.

$k_a \in [0, 1]$

Nighttime scene — Bright scene.

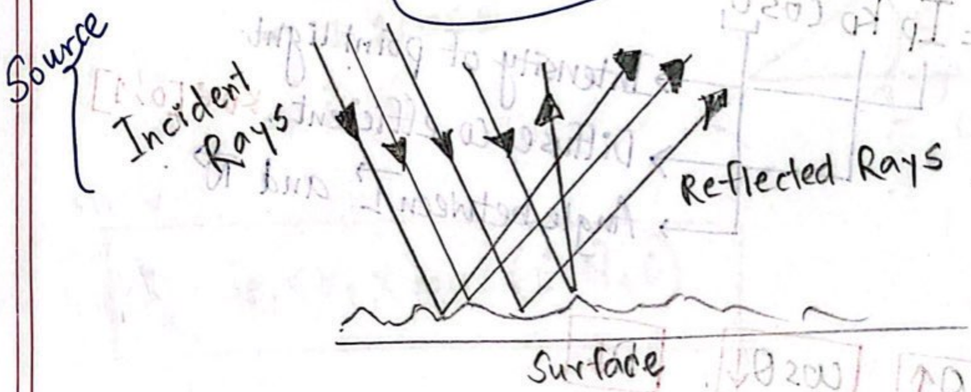
* **Ambient Reflection Co-efficient:-**

Moving from left to right the ambient.

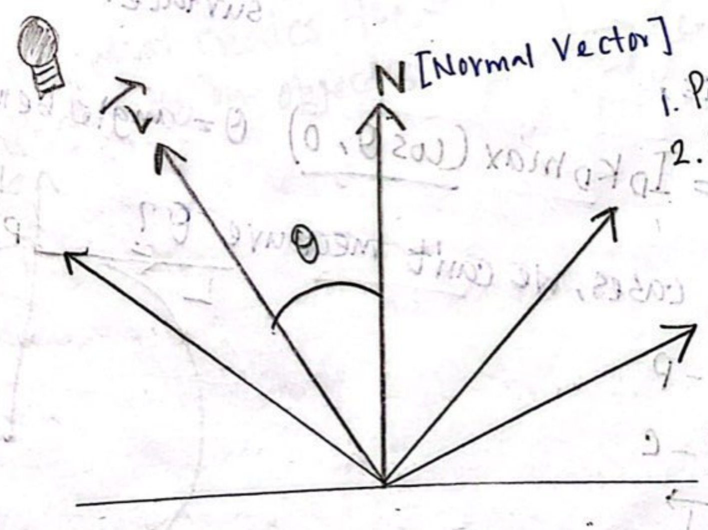
Ambient light ↓↓	Shadows too deep and harsh.
Ambient Light ↑↑	Picture looks washed out and blend.

Scanned with CamScanner

Diffuse Reflection



* 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.



1. Point light source.
2. Light reflected equally in all direction.
3. Magnitude of Reflection depends on θ.

Direction of L reversed.

Scanned with CamScanner

Formula:

$D = I_P K_D \cos\theta$

I_P → Intensity of point light.

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

θ → Angle between $\vec{L}$ and $\vec{N}$.

here, $\boxed{\theta\uparrow}$ $\boxed{\cos\theta\downarrow}$ $\boxed{D\downarrow}$

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

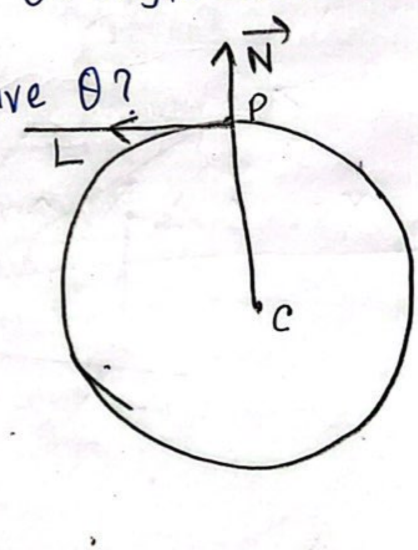
So, $\boxed{\theta > 90^\circ}$, $\boxed{\cos\theta < 0}$ $\boxed{\therefore \cos\theta = -ve}$ $\boxed{\therefore D = -ve}$

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

To solve this,

$D = I_D K_D \max(\cos\theta, 0)$ θ = angle between L and N

** For the cases, we can't measure θ?



$\vec{L} = L - P$

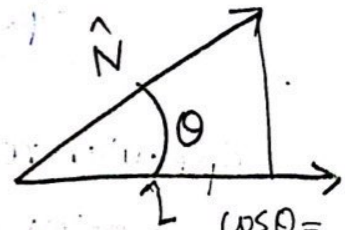
$\vec{N} = P - C$

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

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

Scanned with CamScanner

$$\hat{L}\cdot\hat{N} = |\hat{L}||\hat{N}|\cos\theta$$
$$= 1 \cdot 1 \cos\theta$$
$$= \hat{L}\cdot\hat{N}$$



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

$$\boxed{D = I_P \times k_P \times \max(\hat{L}\cdot\hat{N}, 0)}$$

$\vec{L} = L - P \longrightarrow$ illumination point
$\quad\hookrightarrow$ light source

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

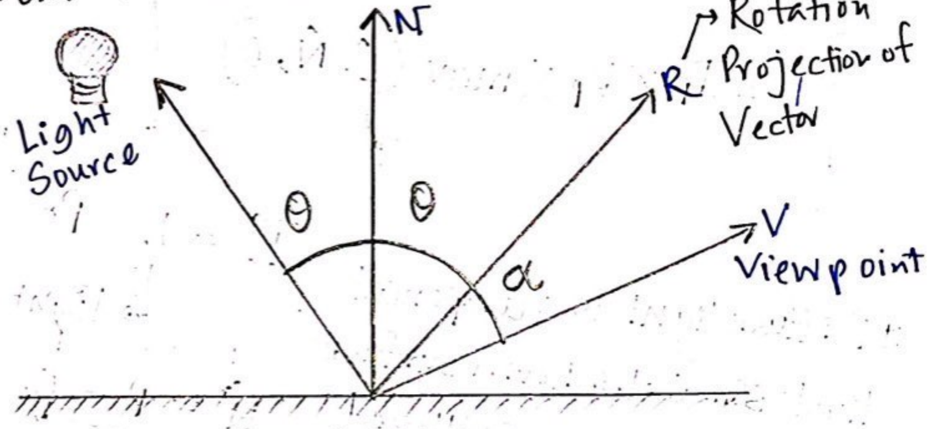
$\vec{N} = P - C$
$\hookrightarrow$ illumination of point
$\hookrightarrow$ Center of sphere.

Scanned with CamScanner

Specular Reflection

Shineness – Depends on Viewpoint.

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



Formula:

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

n → specular exp / shineness factor

I_p → Intensity of point Light.

K_s → Specular Co-efficient.

α → Angle between R and V

if $\boxed{n\uparrow \ (\cos\alpha)^n \downarrow \ \ S\downarrow}$

Modified Formula:

$$S = I_p k_s (\hat{R}\cdot\hat{V})^n$$
$$= I_p K_s \{\max(\hat{R}\cdot\hat{V}, 0)\}^n$$

Scanned with CamScanner

Final Formula for Phong's Reflection Model:

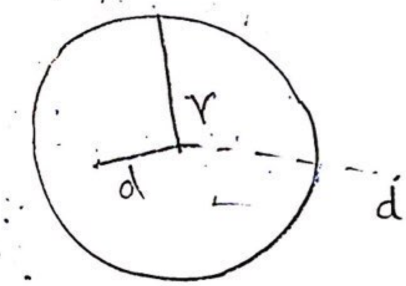
$$I = I_a k_a + I_p k_s \max(\hat{L}\cdot\hat{N}, 0) + I_p k_s \{\max(\hat{R}\cdot\hat{V}, 0)\}^n$$

Attenuation

Attenuation is the loss of light energy over space.

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

$$f_{att} = \frac{1}{d^2}$$ d = 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,

$$d > r,$$

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

Scanned with CamScanner

Final Formula:

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

Updated Formula for Phong's Reflection Model:

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

For Multiple Light Sources.

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

For each light source-

I_p ~~is~~, L, R is multiple

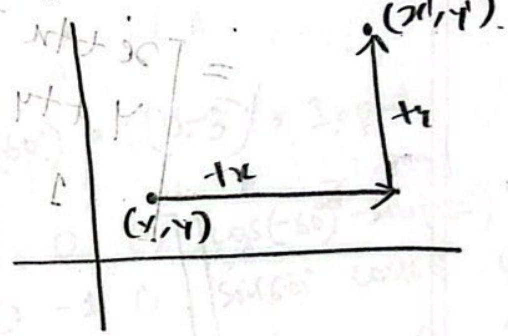
Scanned with CamScanner

Answer to the ques no. 01

1 (a)

Homogeneous co-ordinates is better fit for use in translation instead of Cartesian co-ordinates. Initially, translation is expressed as:

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



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

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

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 3X3 matrices and W is 1 for affine transformation in graphics.

P.T.O

Scanned with CamScanner

Suppose, for 2D, we use a 3 * 3 matrix.

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

$$= \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

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

Rotation Angle 60° Clockwise.

Rotation with respect to (6, 6)

Output = T(6, 6) * R(-60°) * T(-6, -6) * Input

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

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

∴ Answer: (5.5, 6.866)

Scanned with CamScanner

1(c)

Given line.

$$Y = \sqrt{3}x + 3$$

$m = \sqrt{3}$ $\quad \theta = \tan^{-1}(\sqrt{3}) = 60^\circ$

$c = 3$

Output:

$T(0,-3) * R(-60^\circ) * Ref(x) * R(60^\circ) * T(0,3) * \text{Input}$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}}_{T(0,3)} * \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ & 0 \\ \sin 60^\circ & \cos 60^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-60^\circ) & -\sin(-60^\circ) & 0 \\ \sin(-60^\circ) & \cos(60^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 5 \\ 6 \\ 1 \end{bmatrix}$$

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

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.

$$Max = max(R, G, B)$$
$$Min = min(R, G, B)$$
$$l = Max - Min$$

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

$$Value = Max$$

Scanned with CamScanner

Scanned with CamScanner

Hue:

if R==Max:

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

if G==Max:

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

if B==Max:

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

3(c) $C=0.4$

$M=0.2$

$Y=0.3$

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

Max = $G=0.8$

Min $=R=0.6$.

$l=(0.8-0.6)=0.2$

Scanned with CamScanner

$$\text{Saturation} = \frac{l}{\text{Max}} = \frac{0.2}{0.8} = 0.25$$

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

Hue:

here, $G = max$

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

$$= 150^\circ$$

Scanned with CamScanner

Projection:

3(a)

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

Scanned with CamScanner

Scanned with CamScanner

3(b) Mr. Joy Could not take Jaguar's full picture because of ~~parallel~~ perspective projection. 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) $P.P(0,0,0)$

$COP(0,0,175)$

Given points $(35,60,-300)$

$dx = COP.x - PP.x = 0$

$dy = COP.y - PP.y = 0$

$dz = COP.z - PP.z = 175$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{175} & 1 \end{bmatrix} \begin{bmatrix} 35 \\ 60 \\ -300 \\ 1 \end{bmatrix}$$

Scanned with CamScanner

Scanned with CamScanner

$$= \begin{bmatrix} 35 \\ 60 \\ 0 \\ -5/7 \end{bmatrix} \xrightarrow[\text{divide by } -5/7]{\text{Normalize}} \begin{bmatrix} -49 \\ -84 \\ 0 \\ 1 \end{bmatrix}$$

$\therefore$ Output $= (-49, -84, 0)$

Scanned with CamScanner

Final Formula Sheet for CSE423

Transformation

1) Translation:

$x' = x + t_x$
$y' = y + t_y$
$z' = z + t_z$

Matrix:

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

2) Rotation:

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

Matrix:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ → for 3D. z axis

θ will be (-ve) for clockwise

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

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

3) Scaling:

$x' = x \cdot s_x$
$y' = y \cdot s_y$

Matrix:

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

Scanned with CamScanner

4) Shearing:

*Case 01: Shear with x-axis

$$x' = x + ay$$
$$y' = y$$

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

*Case 02: Shear with y-axis

$$x' = x$$
$$y' = y + bx$$

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

Case 03: Shear with x-axis and y-axis

$$x' = x + ay$$
$$y' = y + bx$$

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

Scanned with CamScanner

5) Reflection:

Case 01: Reflect with respect to x-axis.

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

Case 02: Reflect with respect to y-axis.

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

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 (0,b)

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

Scanned with CamScanner

Color Models

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

Primary: R, G, B

Secondary: Cyan → ~~Red +~~ Green + Blue

Magenta → Red + Blue

Yellow → Red + Green.

RGB to HSL / HSV

$Max = max(R, G, B)$

$Min = min(R, G, B)$

$l = Max - Min$

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

Scanned with CamScanner

HSV:

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

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

Hue:

if R == Max:

$$\text{Hue} = \left(\frac{G - B}{l}\right) * 60 \quad [\text{if } H < 0,\ H = H + 360^\circ]$$

if G == Max:

$$\text{Hue} = \left(\frac{B - R}{l}\right) * 60 + 120$$

if B == Max:

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

Scanned with CamScanner

Projection:

Orthographic Projection:

$$(x,y,z) \xrightarrow[z=\alpha]{\text{Projection on xy-plane}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(x,y,z) \xrightarrow[x=\alpha]{\text{Projection on yz-plane}} \begin{bmatrix} 0 & 0 & 0 & \alpha \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(x,y,z) \xrightarrow[y=\alpha]{\text{Projection on xz-plane}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

General Projection Matrix:

(xy-plane)

$$dx = COP_x - PP_x$$
$$dy = COP_y - PP_y$$
$$dz = COP_z - PP_z$$

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

Cases 01: COP(0, 0, 0)
PP (0, 0, d)

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

Scanned with CamScanner

Case 02:

COP $(0, 0, d)$ 　　 $d_x = 0$

PP $(0, 0, 0)$ 　　 $d_y = 0$

$d_z = d$

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

Scanned with CamScanner

Lighting / Projection Illumination

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

$I_a \rightarrow$ Ambient, $I_d \rightarrow$ Diffuse, $I_s \rightarrow$ Specular

Ambient: $A = I_a K_a$

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

Diffuse:

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

Formula Without Attenuation:

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

Formula with Attenuation:

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

$$f_{att} = \max\left(1 - \left(\frac{d}{r}\right)^2, 0\right)$$

Scanned with CamScanner

Scanned with CamScanner

Formula for Reflection $\hat{E}$ / $\hat{R} = 2(\hat{L} \cdot \hat{N})\hat{N} - \hat{L}$

I = Intensity of the light.

I_a = Intensity of ambient light

K_a = ambient reflection Co-efficient

I_p = Intensity of 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.

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

n = shininess exponent

Scanned with CamScanner