Prepared and Written By

RAZEEN AHMED

CSE-423

[COMPUTER GRAPHICS]

HANDWRITTEN NOTE

Computer Science and Engineering

BRAC University

Github: github.com/razeen

LinkedIn: linkedin.com/in/razeenahmed

3D Data → CG → 2D Image

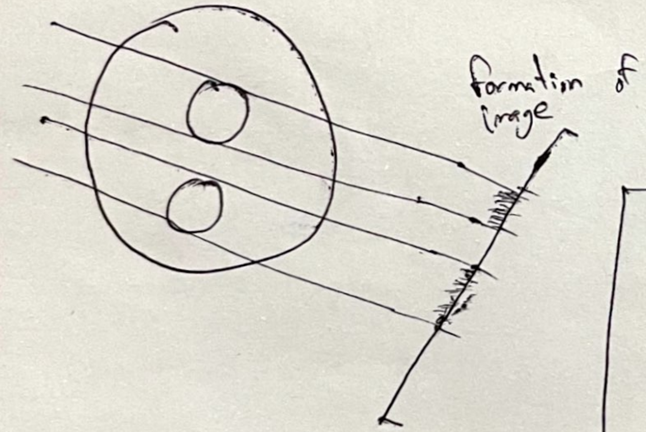
why do we need computer graphics?

→ we can add different elements (we can achieve our goal)

Uses:-

- Movie/animation
- medical Imaging.

Leg



computer graphics takes stack of slides and converts them to

Rendering Pipeline → The order is strictly maintained

Making image in computer

Data —3D Data→ motion → - - - - → Scan Conversion —2D→ image

Display system

(Nowadays display system is hybrid) → internally vector graphic
→ externally raster "

Display system :-

Types of display :- (i) Vector graphics
(ii) Raster " → contains many horizontal lines.

(speed of display is data independent)

advantage:
(i)

(this signal is "horizontal synchronization pulse".)

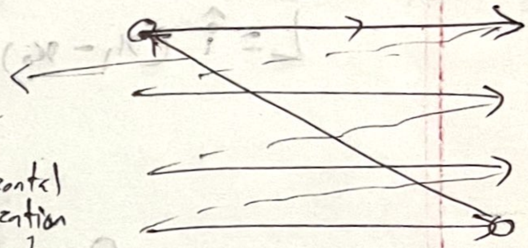


Image on raster graphics display ⇒ bit map.

1920 x 1800
↓
every line contains 1920 pixel

Any image in vector graphics is set of instructions.

TTF → set of instruction to draw a

language of vector graphics → PS (e.g.) → pdf

VIDEO CONTROLLER :-

→ Independent device in display system.

→ video controller can read from

FRAME BUFFER :-

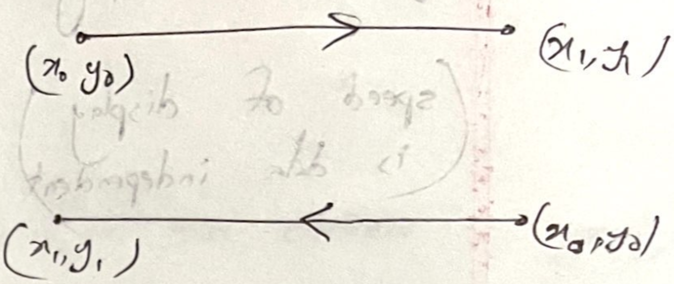
→ sequentially memory it is producing image.

→ can be adjusted on RR, color depth.

$\boxed{\text{line} \equiv \text{vector}}$

Line - Drawing - Algorithm

from (x_0, y_0) to (x_1, y_1)



$$L = \hat{i}\,(x_1 - x_0) + \hat{j}\,(y_1 - y_0)$$

DDA Algorithm :

$\boxed{\text{step 1}}$:- Input (x_0, y_0) and (x_1, y_1)

$dx = x_1 - x_0$ & $dy = y_1 - y_0$

$signdx = sign(dx)$ and $signdy =$ ~~sing~~ $sign(dy)$

C CODE :-

```
int  sign(int x){
    if (x>=0);
        return 1;
    else:
        return -1;
```

$\boxed{\text{Step 2}}$:

step size = max (abs (dx), abs (dy)) ;

dx = abs(dx) / step size ;

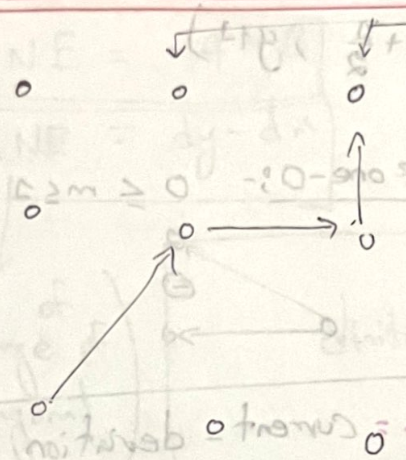
dy = abs(dy) / step size ;

[any one is 1
another is less than 1]

```
For ( i=0; i<= stepsize ; i++ ) {
    draw Pixel (x0, y0);
    x0 += (dx * sign dx);
    y0 = y0 + (dy * sign dy);
}
```

MID POINT LINE DRAWING ALGO   05/10/23

BRESSENHAM LINE DRAWING ALGORITHM :-



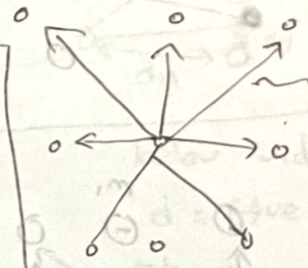
how to find the closest point.

line chooses the lowest

New deviation = current deviation + rate of change

$y = mx + c$ —— (i)

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



(x & y = coordinates of pixel)

↓ if on the line

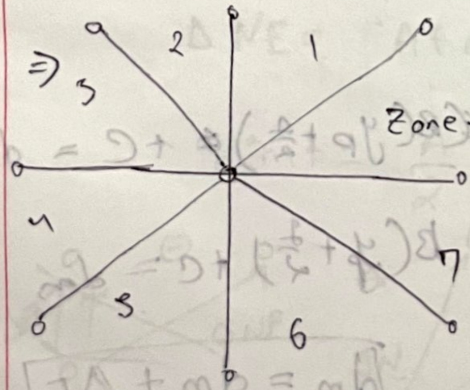
↓ we will get 0

Imp ★ $Ax + By + C = 0$ —— (ii)

Implicit form

$A = dy$, $B = -dx$

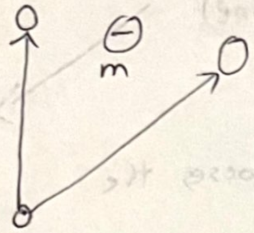
→ How to choose the lowest value



(each 45° is a zone)

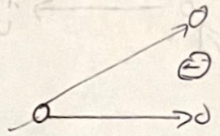
(→ for different zone the rule is different)

Zone - 1

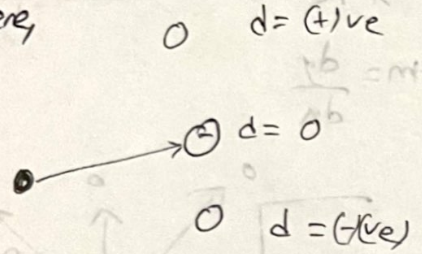


$m = (x + \frac{1}{2}, y+1)$

Zone-0 :- $0 \leq m \leq 1$

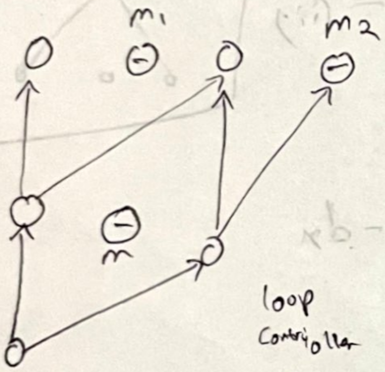


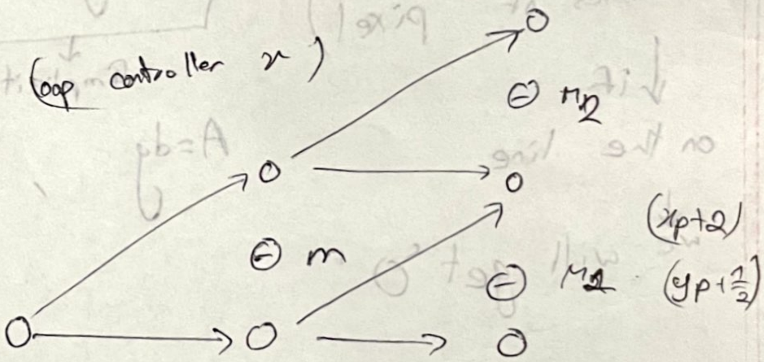
Here,



New deviation = current deviation + rate of change

rate of change = change of diagonal.
$= (dm_1 - dm_2)$





$f(x,y) = Ax + By + C = 0$

$f(M_1) = A(x_p + 2) + \cancel{B}(B(y_p + \frac{1}{2}) + C = dm_1$

$f(M) = A(x_p + 1) + B(y_p + \frac{1}{2}) + C = dm$

rate of change of deviation per pixel, (horizontal line) $\Delta E = A = dm_1 - dm$

$\boxed{dm_1 = dm + \Delta E}$

$$\boxed{\Delta NE = A + B}$$

$$\boxed{\therefore \Delta NE = dy - dx}$$

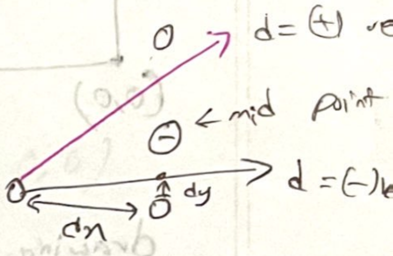
Rate of change of diagonal line.

$[\Delta E = A = dy]$

$A = dy$

$B = -dx$

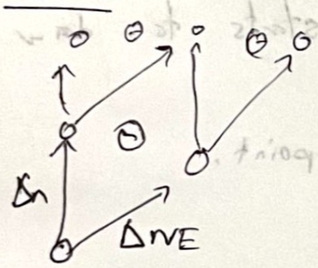
$$d_{initial} = A + \frac{B}{2} = dy - \frac{dx}{2}$$



below mid point
$d = (-)ve$

Others d will be
$d = (+)ve$

Zone -1



$m = (x + \frac{1}{2}, y + 1)$

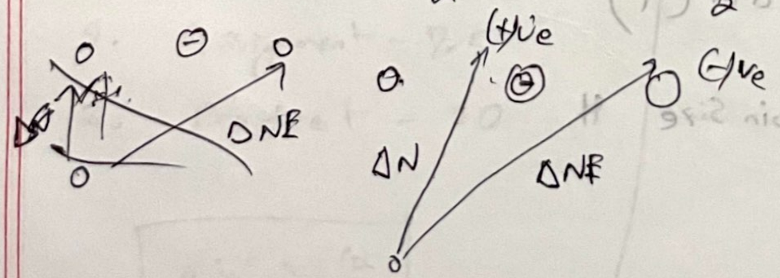
$m = (x + \frac{1}{2}, y + 2)$

$m = (x + \frac{3}{2}, y + 2)$

$\Delta N = B = -dx$

$\Delta NE = A + B = dy - dx$

$$d_{init} = \frac{A}{2} + B = \frac{dy}{2} - dx$$

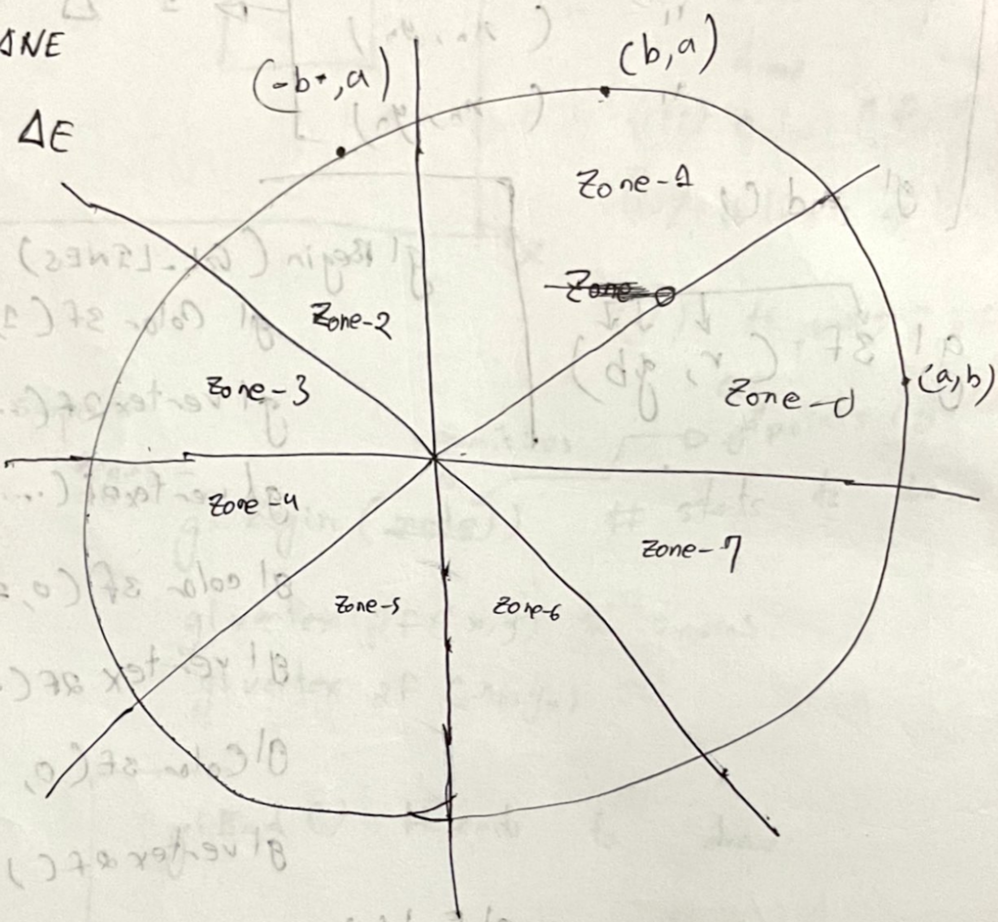


How to blink: change of colors.

\# import time

$d_i \geq 0 \Rightarrow \Delta NE$

$d_i < 0 \Rightarrow \Delta E$



(i) sign change.

(ii) value interchange.

DDA Algorithm

[If $-1 < m < 1$ line is not steep:]

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k + m$$

Otherwise

next point $y_{k+1} = y_k + 1$

next point $x_{k+1} = x_k + \frac{1}{m}$

$dx < 0$ $dy < 0$ zone 4

$dx > 0$ $dy < 0$ zone 7

Slope determination Algo:-

$|dx| \geq |dy| \Rightarrow 0, 3, 4, 7$

$dx > 0,\ dy > 0 \Rightarrow$ zone 0

$dx < 0$ $dy > 0 \Rightarrow$ zone 3

$|dy| > dx \Rightarrow 1, 2, 5, 6$

$dx > 0$ $dy > 0 \Rightarrow$ zone 1

$dx < 0$ $dy > 0 \Rightarrow$ zone 2

$dx < 0$ $dy < 0 \Rightarrow$ zone 5

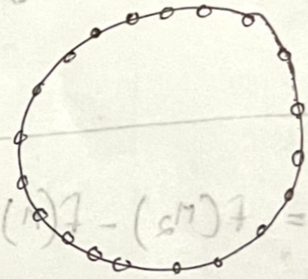
$dx > 0$ $dy < 0 \Rightarrow$ zone 6

(MID POINT CIRCLE DRAWING)

Code for drawing

cartesian)

```
For (x = -r to r):
    drawPixel(x, √(r²-x²))
    drawPixel(x, -√(r²-x²))
```



$x = R\cos\theta$ ← independent variable

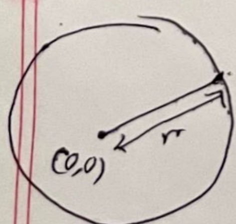
$y = R\sin\theta$

polar)

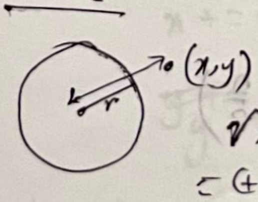
```
for (θ = 0 to 360):
    drawPixel(rcosθ, rsinθ)
```

Imp equation:

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

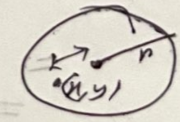


Outside



$\sqrt{x^2+y^2} - r^2$
$= (+)ve$

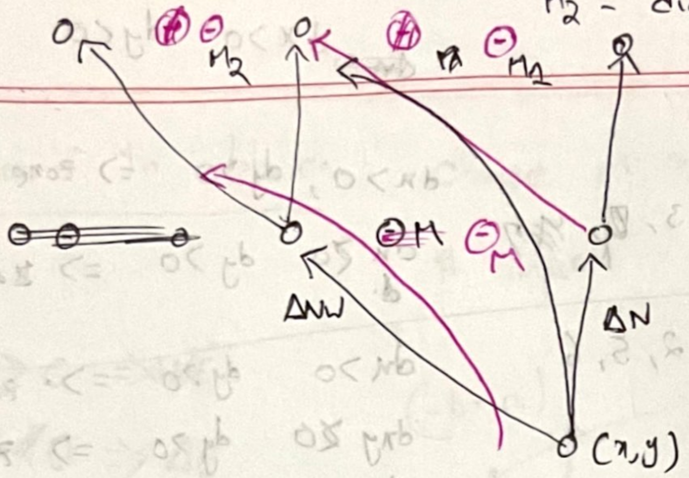
inside



$\sqrt{x^2+y^2} - r^2$
$= (-)ve$

$M_1 =$ either vertical or horizontal

$M_2 =$ diagonal (always)



$M = (x-\frac{1}{2}, y+1)$

$\to M_2 = (x-\frac{1}{2}, y+2)$

$(x-\frac{3}{2})$

$\to M_2 = (\cancel{x+\frac{1}{2}}, y+2)$

$\Delta N = f(M_1) - f(M)$

$f(x-\frac{1}{2}, y+2) = (\cancel{x-\frac{1}{2}})^2 + (y+2)^2 - \cancel{r^2}$

$f(x-\frac{1}{2}, y+1) = (\cancel{x-\frac{1}{2}})^2 + (y+1)^2 - \cancel{r^2}$

(-)

$\Delta NW = f(M_2) - f(M)$

$\Delta N = (y+2)^2 - (y+1)^2$

$= y^2 + 4y + 4 - (y^2 + 2y + 1)$

$= 2y + 3$

#

$f(x-\frac{3}{2}, y+2) = (x-\frac{3}{2})^2 + (y+2)^2 - r^2$

$f(x-\frac{1}{2}, y+1) = (x-\frac{1}{2})^2 + (y+1)^2 - r^2$

$\Delta NW = (x-\frac{3}{2})^2 + (y+2)^2 - (x-\frac{1}{2})^2 - (y-1)^2$

$= x^2 - 3x + \frac{9}{4} + y^2 + 4y + 4 - x^2 - x - \frac{1}{4} - y^2 - 2y - 2$

$= 2y - 2x + 5$

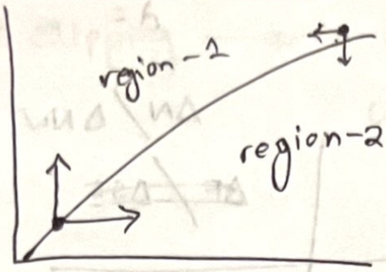
The first midpoint $= (r-\frac{1}{2}, 1)$

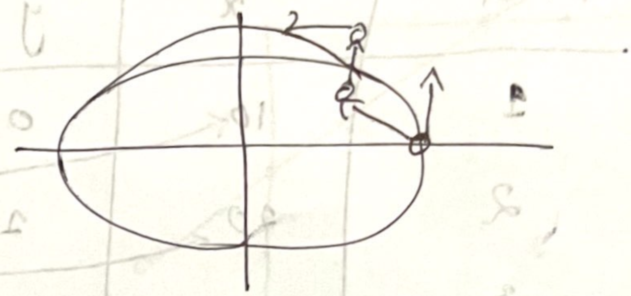
$d_{init} = 2dy - dx$

$= 2.1 - (r-\frac{1}{2})$

$= 2 - r + \frac{1}{2}$

$= 5/4 - r$





$\therefore$ d = (-)ve (choose/go outside

d = (+)ve " /go inside

* In region-2 • dot is inside we are bringing it outside ↑

* In region • dot is outside we are bringing it inside ↓

Code :-

```
def draw circle_2 (r)
    x = r
    y = 0
    d = 5-4*r
    draw 8way (x,y)
    while (x > y):
        if d<0    # del N
            d += 4*(2*y +3)
            y += 1
        else: # del NW
            d+= 4*(2*y - 2*x +5)
            x += 1
            y -= 1
    draw 8way(x,y)
```

#check
slide code will be opposite x, y variable

$r = \sqrt{x^2 + y^2}$

Solve (using region-2):

Initialize
$\to d = 5 - 4r$

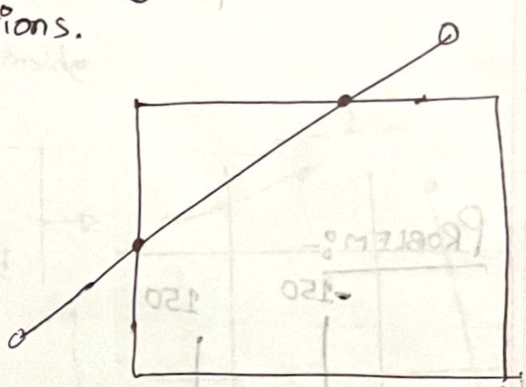
$d =$

Pixel	x	y	d	$\Delta N / \Delta NW$ ~~ΔE / ΔSE~~	
1	10	0	-35	ΔN	
2	10	1	-23	ΔN	-35 +12
3	10	2	-3	ΔN	-23 +20
4	10	3	~~33~~ 25	ΔNW	-3 +36
5	9	4	~~13~~ -11	~~ΔNW~~ ΔN	25 -36 ~~33 -20~~
6			33	ΔNW	

LINE CLIPPING Algorithm

Clipping Primitives

★ Clipping ⟶ improving the efficiency to avoid unnecesary calculations.



= 4 (2y + 3) + prev(d)

= 4 * (2y+3) + prev(d)

= 4 * (2y+3) + prev(d)

= 4(2y-2x+5) + prev(d)

= 4 (2y+3) + prev(d)

All the lines in three groups:

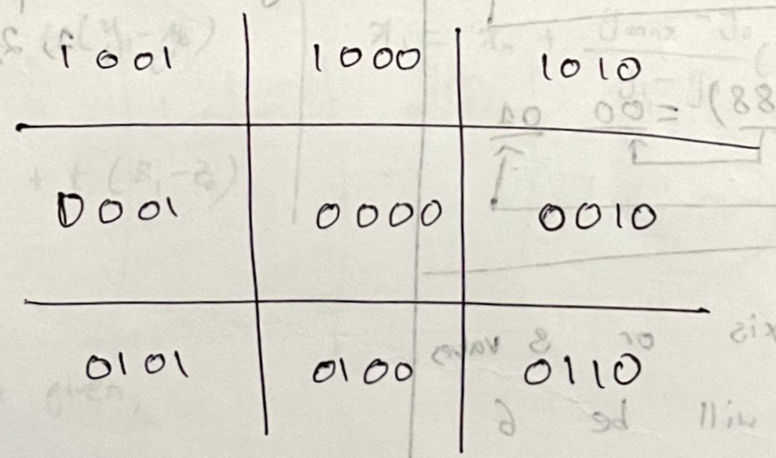
(i) Inside (ii) Outside (iii) partial

Cohen Sutherland Algorithm:-

line end point can be converted into 4 bit code.

b_3 b_2 b_1 b_0

y



1001	1000	1010
0001	0000	0010
0101	0100	0110

~~of~~ greater than ~~above~~ y~~max~~

10 Above y

10 - -

below y

01 - -

In between

00 - -

left :

- - 01

right:

- - 10

In between

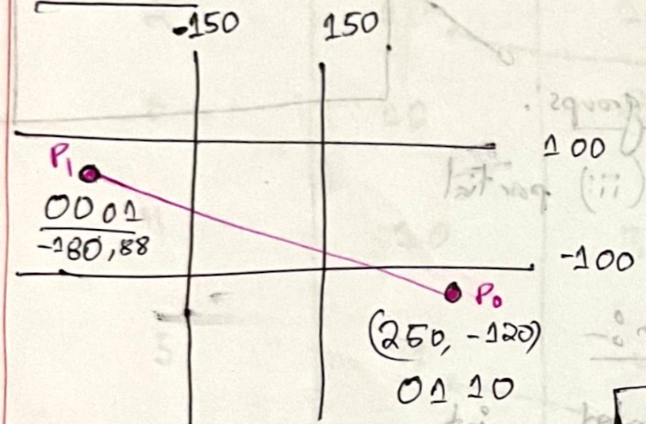
00

CODE:

"|=" can be used as "+="

00 ---- 1000
00 0010
~~00~~ 1010

PROBLEM:-



$P_0(250, -120) = 01\ 10$

$P_1(-180, 88) = 00\ 01$

QUIZ:

Point's will be given convert it into code. (based on 3 values)

left, Right, Bottom, Top = ~~1, 2~~, Far, Near =
1, 2, 4, 8, 16, 32

For 3 axis or 3 values the Code will be 6 digits:-

$d_6 | d_5 | d_4 | d_3 | d_2 | d_1$

The main point of Cohen Sutherland algorithm :-

(i) Completely accepted (ii) partially accepted

(iii) " rejected

Code:-

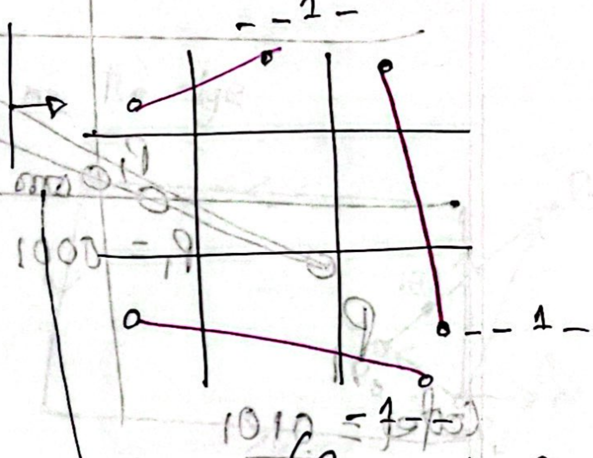
if (code0 | code1) == 0 :-

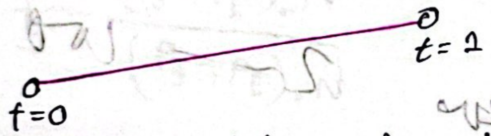
#then the line is inside

else if (code0 & code1) ≠ 0 :-

then the line is rejected

else:





$x = x_0 + t(x_1 - x_0)$

$y = y_0 + t(y_1 - y_0)$

If y is known,

$$x_1 = x_0 + \frac{y_{max} - y_0}{y_1 - y_0}(x_1 - x_0)$$

$y = y_{max}$

For 3D:-

$z = z_0 + t(z_1 - z_0)$

Quiz:

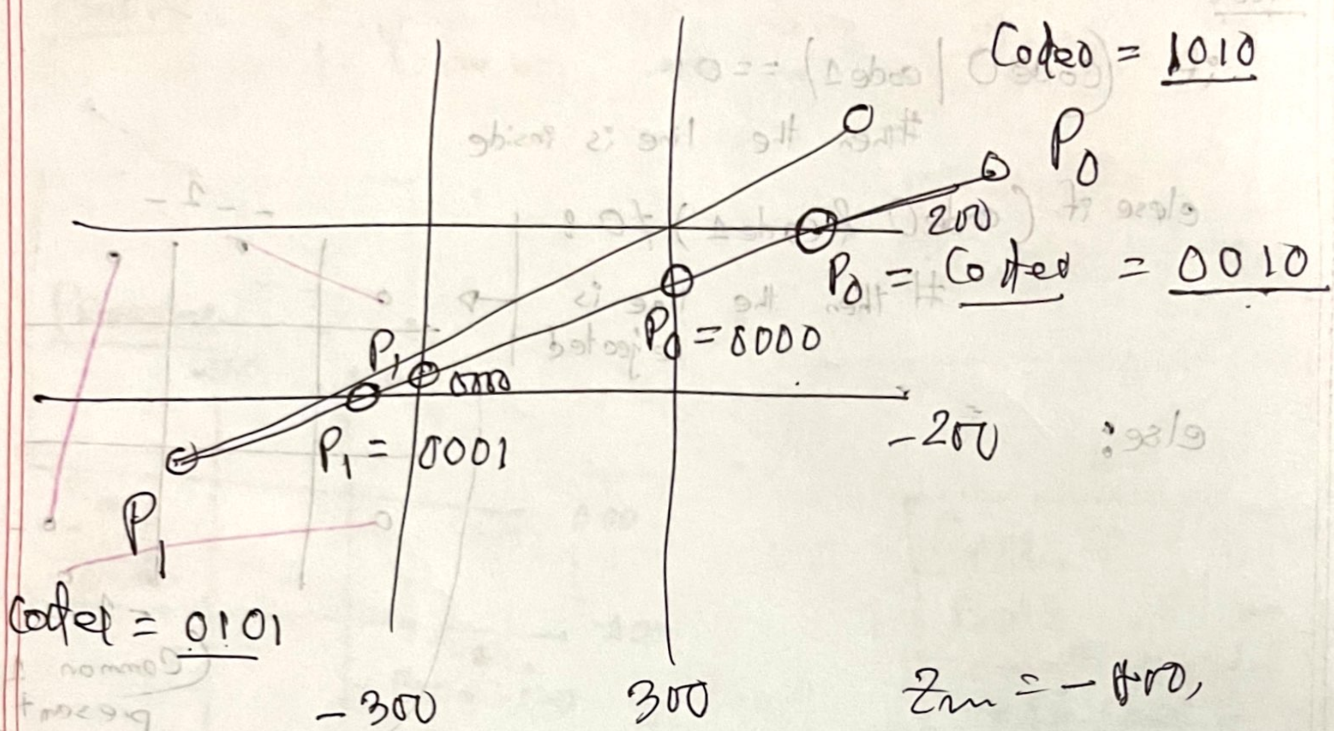
endpoints are given,

comments:-

completely accepted

" rejected

partiall accepted.



$Z_{min} = -400$,

$Z_{max} = 100$

$Y_{max} = 200$, $Y_{min} = -200$

P_0	P_1	Code0	Code1	Comment
(200, -300, 20)	(-400, 200, 120)			

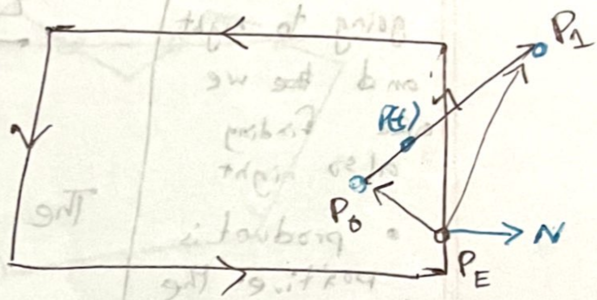
21/10/23

PARAMETRIC LINES AND INTERSECTIONS

$(P(t) - P_E) \cdot N > 0$ means point $P(t)$ is outside

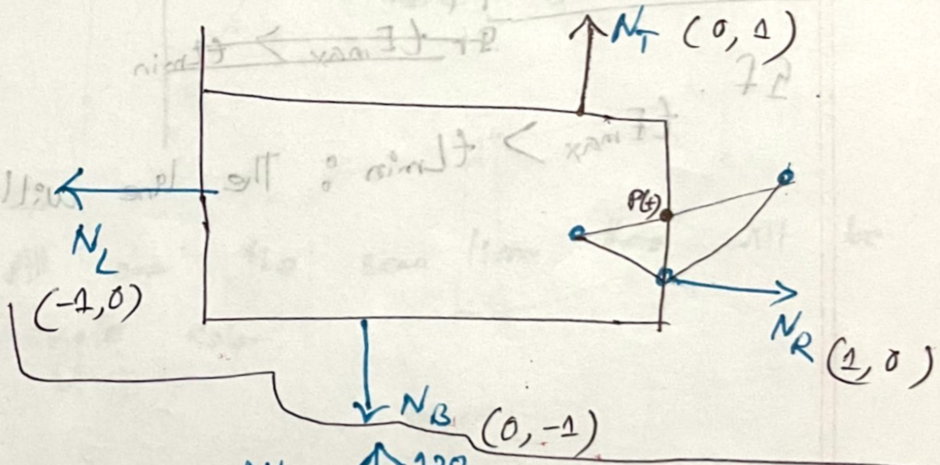
$(P(t) - P_E) \cdot N < 0$ means point $P(t)$ is inside

$(P(t) - P_E) \cdot N = 0$ " " $P(t)$ on the edge

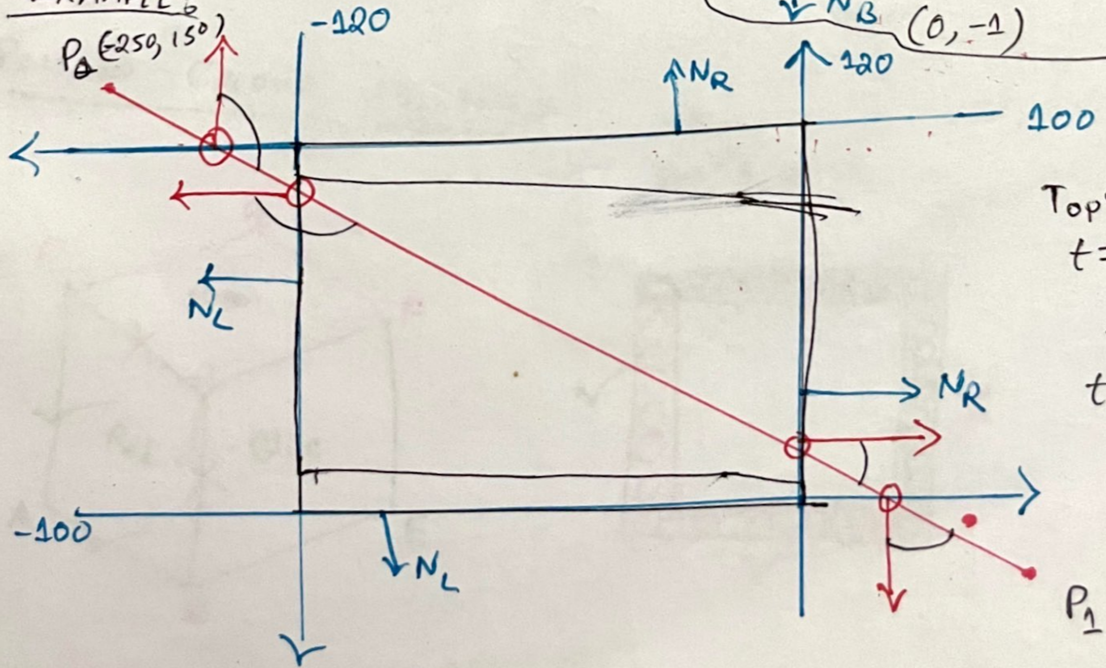


CYRUS BECK ALGORITHM:

$$t = -\frac{(P_0 - P_E) \cdot N}{(P_1 - P_E) \cdot N}$$



EXAMPLE:



Top: *

$$t = \frac{100-150}{-140-150}$$

$$= \frac{-50}{-290}$$

$$t = \frac{50}{290}$$

t_E → Top

P_1 (260, -140)

21/10/23

Bottom: $t = \dfrac{-100-150}{-140-150} = \dfrac{-250}{-290} = \dfrac{25}{29}$ t_L → bottom

Right: $t = \dfrac{120+150}{260+250} = \dfrac{37}{51}$ t_L

↓

> The line is going to right and ~~the~~ we are finding also right
> - product is positive the line is learing

Left: $t = \dfrac{-120+250}{260+250} = \dfrac{13}{51}$ t_E

The line is from $t_{E\,max}$ to $t_{L\,min}$.

$t_{E\,max} = \dfrac{13}{51}$ & $t_{L\,min} = \dfrac{37}{51}$

~~If $t_{E\,max} > t_{L\,min}$~~

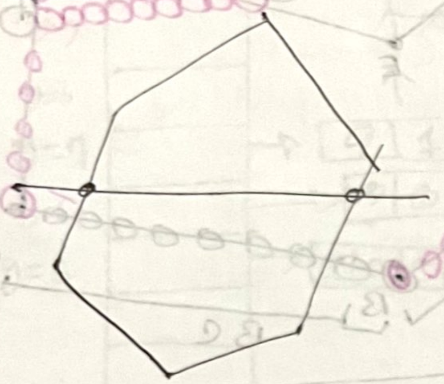
If

$t_{E\,max} > t_{L\,min}$: The line will be rejected.

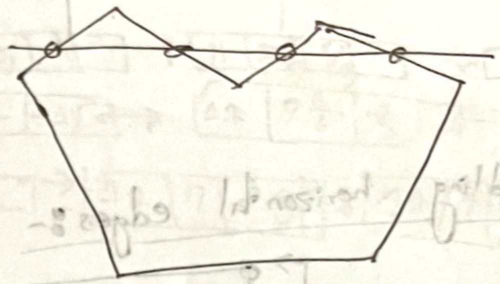
SCAN CONVERSION ALGORITHM

Polygon's are two types:- (i) Convex
(ii) Concave

Convex :-



Concave :-



• there is no broken scan line

• There is broken line

POLYGON PRIMITIVES:-

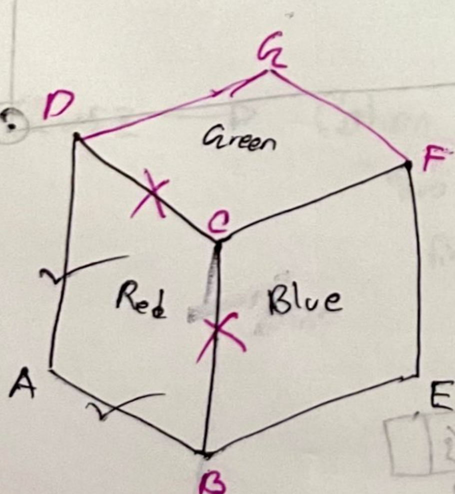
(i) Coordinates
(ii) Color filling

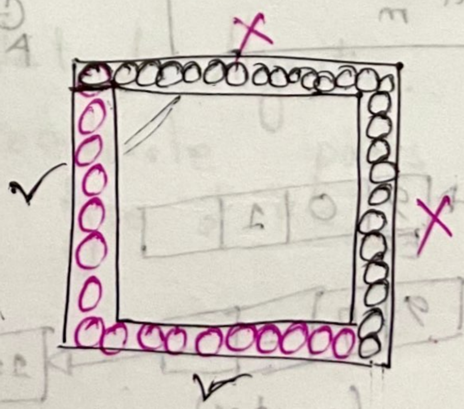
All over the scan line there will be single color.

POLYGON COLOUR FILLING:-

$x * y =$

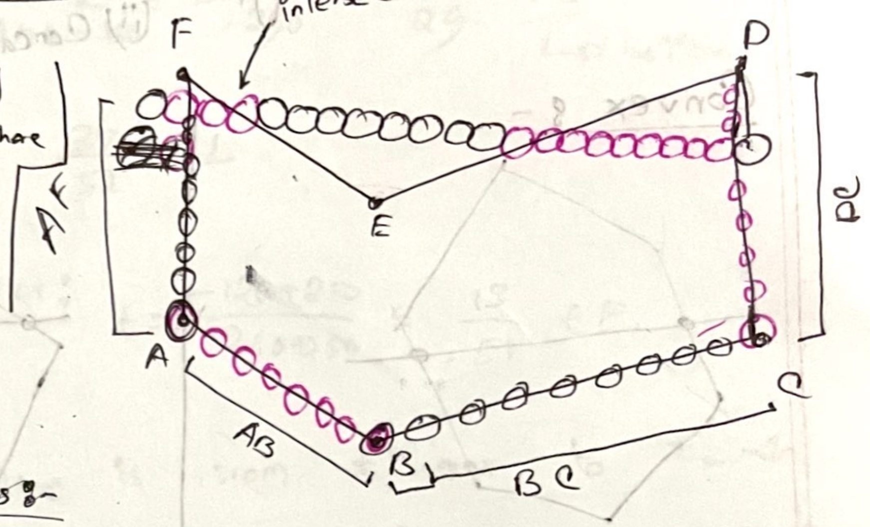
$(x-1) * (y-1) =$



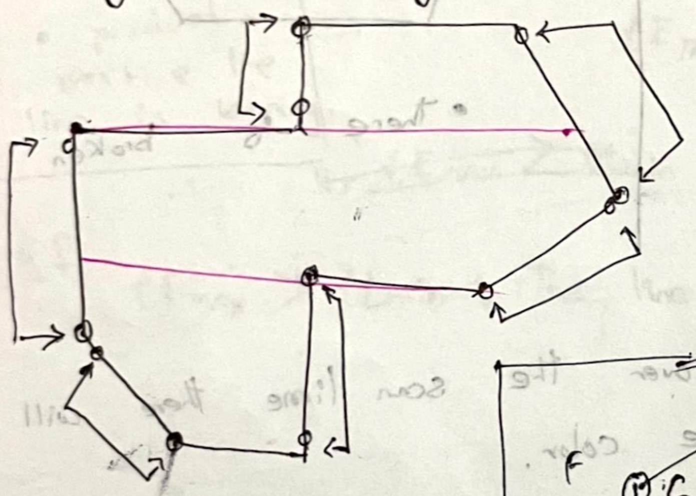


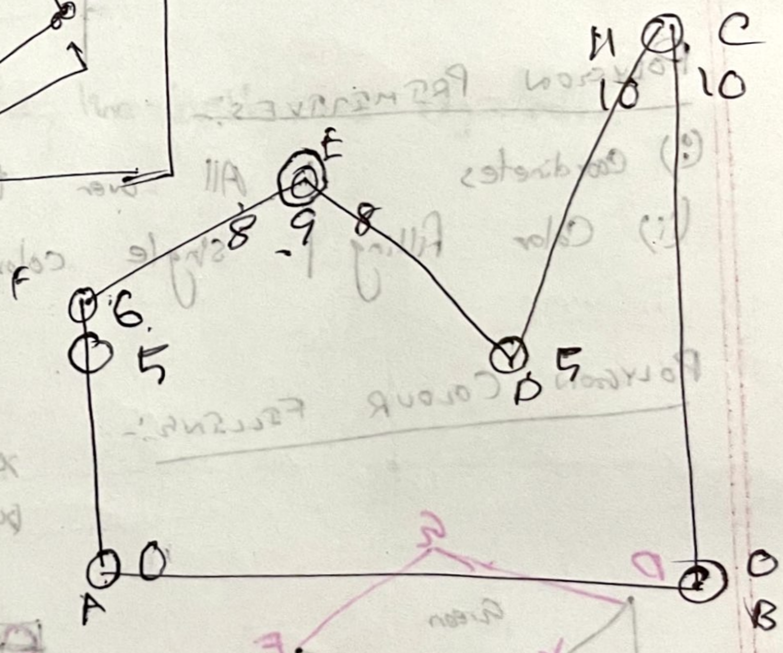
SCAN FILLING ALGORITHM:

* Our target for any pair of scan line there must be pair of intersected line



handling horizontal edges:-

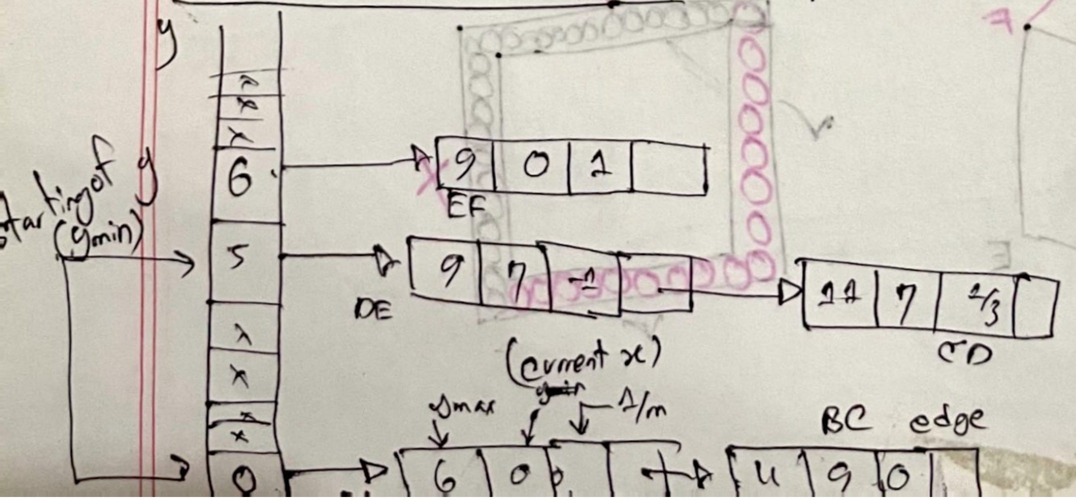




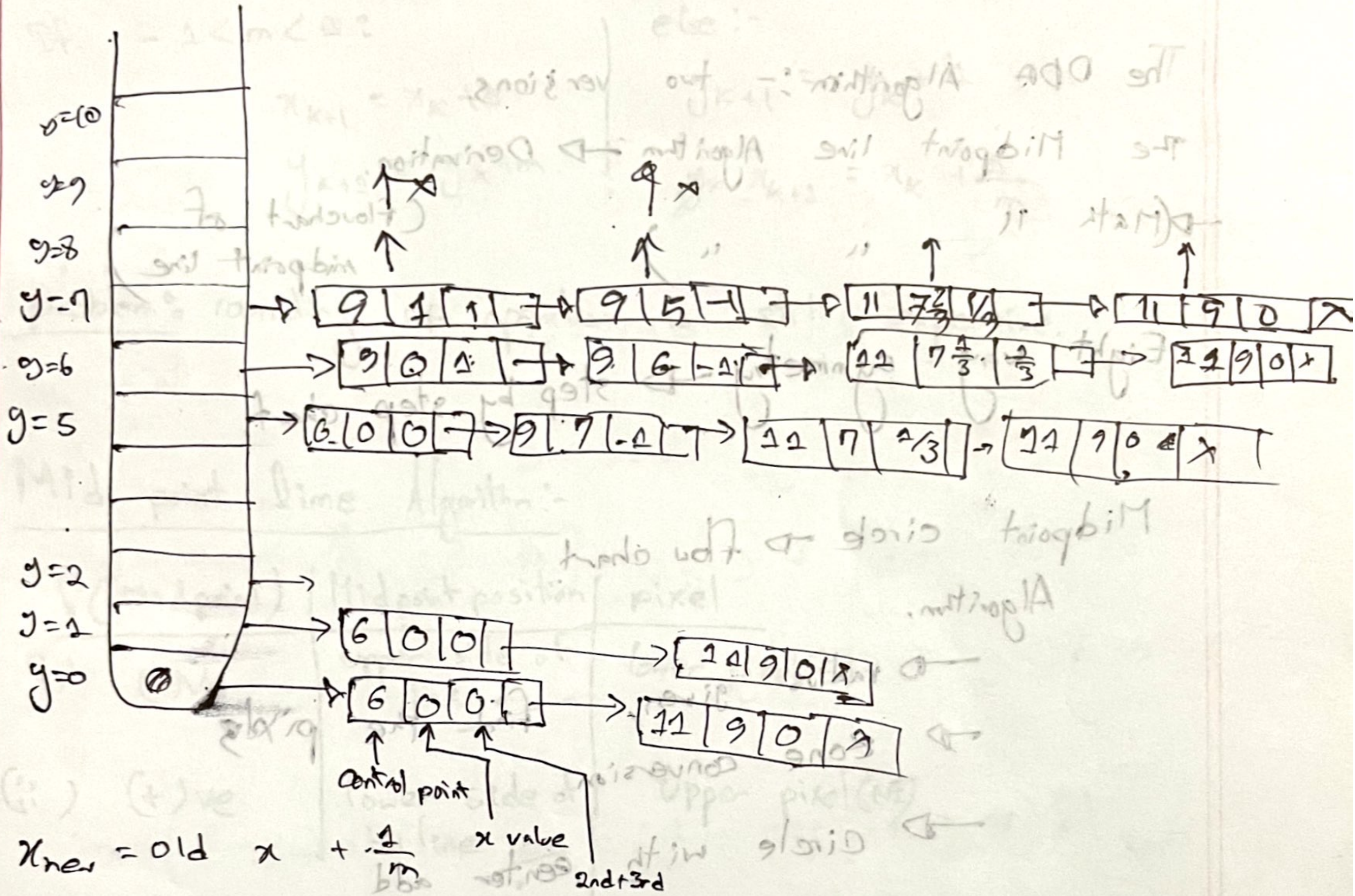
Boundary line rules:-

$y \longrightarrow y+1$

$x \longrightarrow x + \frac{1}{m}$



Active Edge Table



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

Questions:-

S table will given make active edge table
active edge will be make S table.

QUIZ → Cohen sutherland Algorithm.
two coordinate points will be given
find the type of line

COHEN SUTHERLAND POLYGON CLIPPING

MID syllabus pot up

The DDA Algorithm :- two versions.

The Midpoint line Algorithm → Derivation.

(Flowchart of midpoint line)

→ (Math " " ")

Eight way symmetry → step by step chart.

Midpoint circle → flow chart.

Algorithm.

→ radius given, find the pixels

→ Zone conversion.

→ circle with center add

$x \longrightarrow x + 30$

$y \longrightarrow y + 30$

Cohen sutherland line clipping.

2D & 3D

Cyrus-beck clipping (Math V. imp)

DDA Algorithm came instead of simple solution.

If $-1 < m < 1$:

$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k + m$$

else:-

$$y_{k+1} = y_k + 1$$
$$x_{k+1} = x_k + \frac{1}{m}$$

problem: rounding up operation is still expensive.

Mid point line Algorithm:-

f(Midpoint)	Midpoint position	pixel
(i) (-)ve	upper side of line	lower pixel(E)
(ii) (+)ve	lower side of line	upper pixel(NE)

$$\cancel{2} d_{old} = 2a + b$$

$$\boxed{d_{initial} = 2\Delta y - \Delta x}$$

when, ΔNE $\boxed{d>0}$

$$d_{new} = d_{old} + \underbrace{2\Delta y - 2\Delta x}_{\Delta NE}$$

when ΔE $\boxed{d<0}$

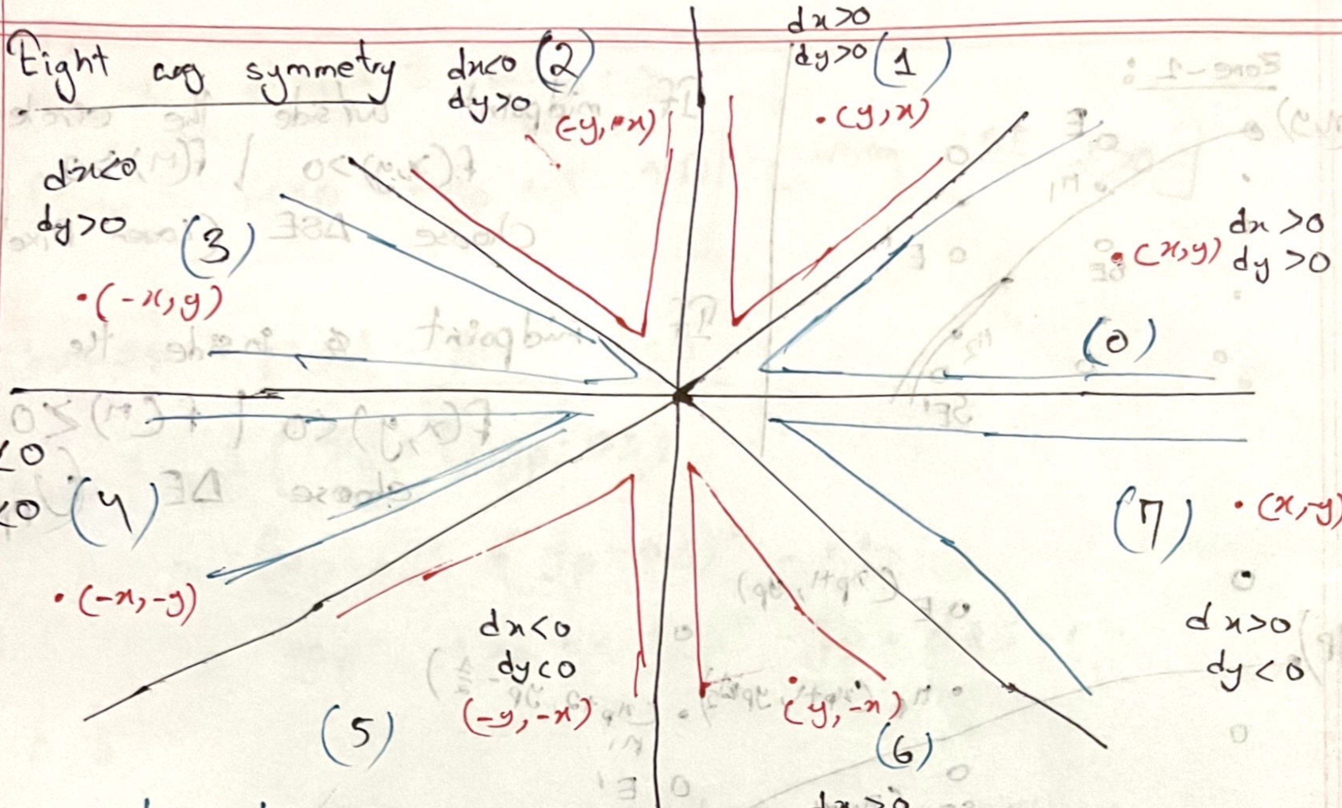
$$d_{new} = d_{old} + \underbrace{2\Delta y}_{\Delta E}$$

$$d_{\Delta NE} = 2(\Delta y - \Delta x)$$

$$d_{\Delta E} = 2\Delta y$$

$$d_{old} = \cancel{2(\Delta y -} \; 2\Delta y - \Delta x$$

Eight way symmetry



∠ → |dx| > |dy| zone: 0, 3, 4, 7

∠ → |dy| > |dx| zone: 1, 2, 5, 6

(To find in which zone the line lies in)

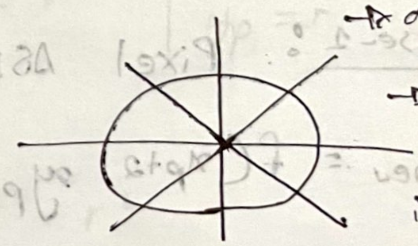
MID POINT CIRCLE DRAWING:-

$\cancel{f(x,y)}$ $r = \sqrt{(x-0)^2 + (y-0)^2}$ center (0,0)

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

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

$f(x,y) = x^2 + y^2 - r^2 = 0$



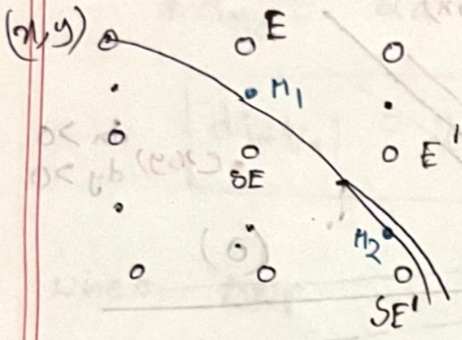
→ arc length same

→ equally divided into sector

If points on the boundary of circle,

$f(x,y) = 0$

Zone-1 :



If midpoint outside the circle

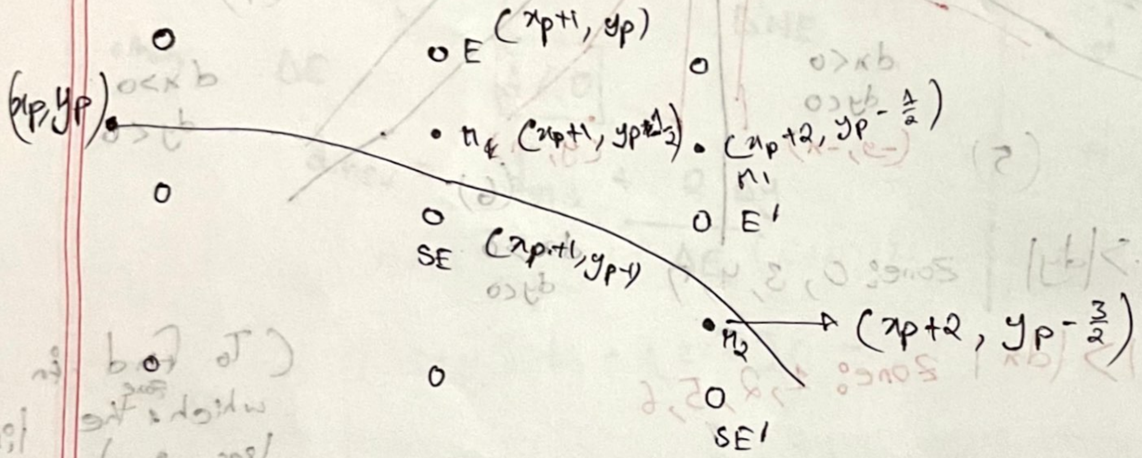
$f(x,y)>0 \mid f(M)>0$

Choose ΔSE (Lower Pixel)

If midpoint inside the circle

$f(x,y)<0 \mid f(M)<0$

choose ΔE (Upper Pixel)



$$d = f(x_p+1, y_p-0.5) = f(M) = (x_p+1)^2 + (y_p-0.5)^2 - r^2$$

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

Case-1 : Pixel ΔSE $d>0$

$$D_{new} = f(x_p+2, y_p-1.5) = (x_p+2)^2 + (y_p-1.5)^2 - r^2$$

$$= x_p^2 + 4x_p + 4 + y_p^2 - 3y_p + 2.25 - r^2$$

And,

$D_{new} - D = 2x_p + 3 - 2y_p + 2$

$$\boxed{D_{new} = 2x_p - 2y_p + 5 + D}$$ [when SE $d > 0$]

Case-2:- Pixel E $d < 0$

$D_{new} = f(x_p + 2, y_p - 0.5)$

$= (x_p+2)^2 + (y_p - 0.5)^2 - r^2$

$= x_p^2 + 4x_p + 4 + y_p^2 - y_p + 0.25 - r^2$

And,

$D_{new} - D = x_p^2 + 4x_p + 4 + y_p^2 - y_p + 0.25 - r^2$
$\quad - x_p^2 - 2x_p - 1 - y_p^2 + y_p - 0.25 + r^2$

$D_{new} - D = 2x_p + 3$

$$\boxed{D_{new} = 2x_p + 3 + D}$$ [when E $d < 0$]

Again,

~~Dinitial~~ $D_{initial} = f(x_p + 1, y_p - \frac{1}{2})$ | $x_p = 0$, $y_p = r$

$= f(1, r - \frac{1}{2})$

$= 1^2 + (r - 0.5)^2 - r^2$

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

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

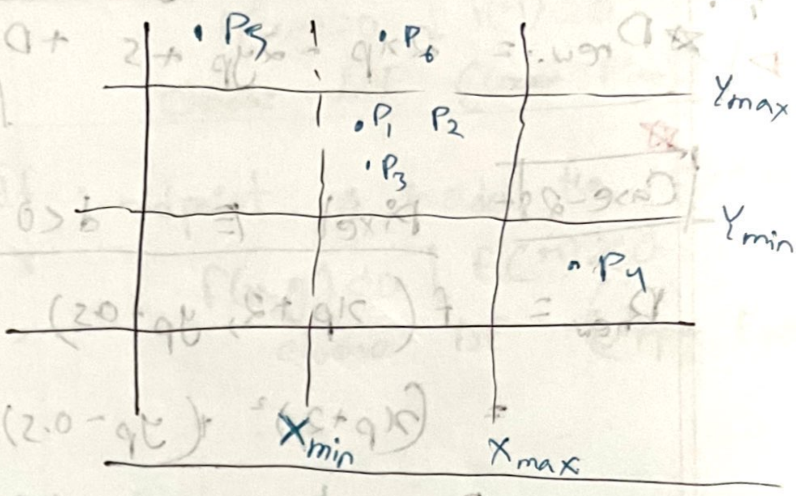
$$\boxed{D_{initial} = 1 - r}$$ → round up

LINE CLIPPING:

A point is not clipped if :-

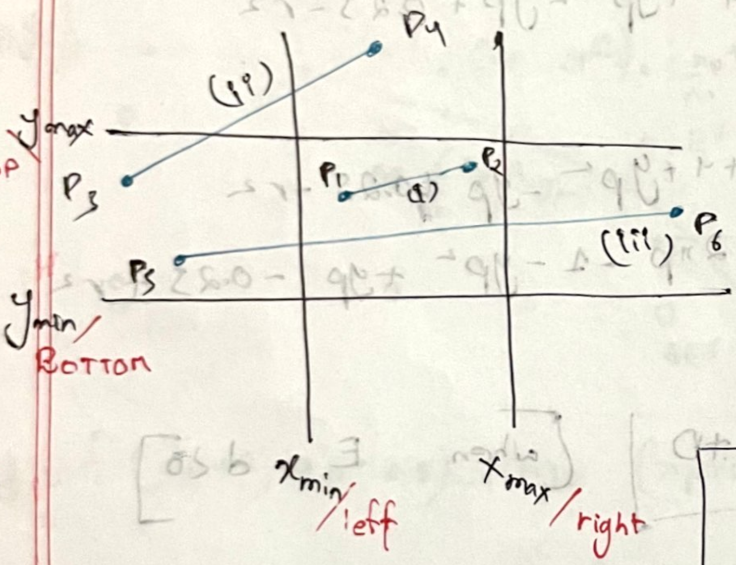
$$X_{min} \le x \le X_{max}$$
$$Y_{min} \le y \le Y_{max}$$

POINTS :-



CROPPED : P_1, P_2, P_3

LINE IN REGION:



3 cases :-

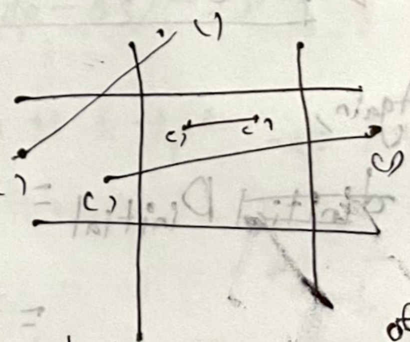
(i) line completely inside.

(ii) line " outside.

(iii) → line partially inside

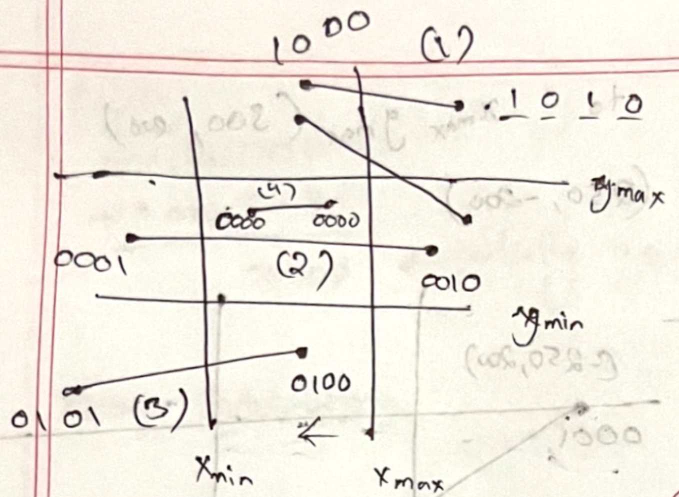
bit code :-

5	4	3	2	1	0
far	near	Top	bottom	right	left



To check the cases of the line we need to AND the bit-codes of starting (x_1, y_1) and ending point (x_2, y_2)

Violates :- 1
~~Not~~ Accepts :- 0



Case-1

FOR (1)

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

If any one of the bit in result is '1' the line is rejected

COMMENT :- LINE IS COMPLETELY OUTSIDE

Case-2

FOR (2)

$$\begin{array}{r} 0001 \\ \text{AND } 0010 \\ \hline 0000 \end{array}$$

In the question at least 1 bit is on

COMMENT :- LINE IS ~~COMPLETELY~~ Partially INSIDE

FOR (4)

$$\begin{array}{r} 0000 \\ \text{AND } 0000 \\ \hline 0000 \end{array}$$

Case-3

COMMENT :- LINE IS COMPLETELY INSIDE

Q. Types :-

(i) check the lines situation in the window by finding

(i) Find outcodes of starting & ending point of the line by performing some operation comment on the situation line.

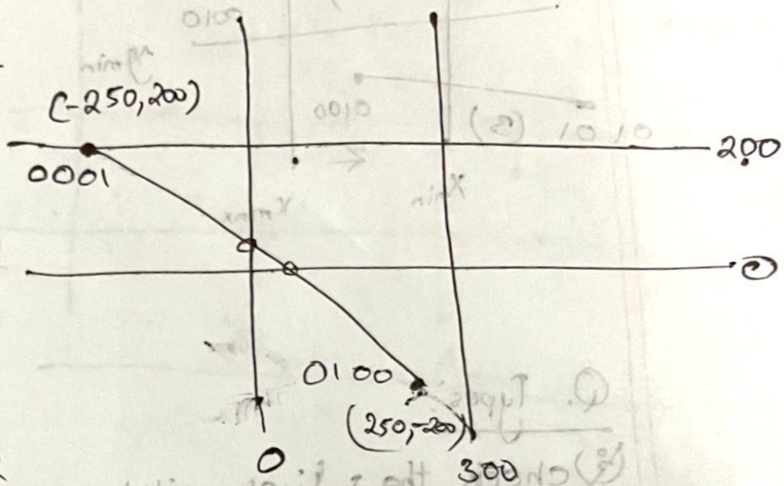
(ii) Comment on the situation of the line ~~on~~ of the window and if the line is partially inside find the clipped ~~part~~ coordinate of the new line. Show step by step calculation.

Example: $x_{min}\ y_{min}\ (0,0)$ to $x_{max}\ y_{max}\ (300, 200)$

line → $(-250, 200)$ to $(250, -200)$

$x_1 = -250$ | $y_1 = 200$

$x_2 = 250$ | $y_2 = -200$



$(x_1 y_1) = 0001$ = out code 1

$(x_2 y_2) = 0100$ = out code 2

Now,

$$\begin{array}{r} 0001 \\ 0100 \\ \hline 0000 \end{array}$$

at least '1' present

The line is partially inside.

left

$y_1 = y_1 + m(x_{min} - x_1)$

$= 200 + \frac{-200-200}{250-(-250)}(0-(-250))$

$= 200 + \frac{-400}{500} \times 250$

$= 0$

New coordinatie, $y_1 = 0$, $(x, y_1) = (0, 0)$

New outcode1 = 0000

Again,

new otde 1 0000

old otde 0100

00001 ← partially inside

~~New outcode 2 =~~

Now,

New outcode 1 = 0000

" outcode 2 = 0000

0000

New coordingates

$(x_1, y_1) = (0,0)$

$(x_2\ y_2) = (0,0)$ Ans,

Bottom,

$y_2 = y_{min}$

$x_2 = x_2 + \frac{1}{m}(y_{min} - y_2)$

$= 250 + \frac{1}{\frac{-400}{500}}(0 - (-200))$

$= 250 \cancel{+} - \frac{500 \times 200}{400}$

$= 0$

$= 250 + \frac{x_2 - x_1}{y_2 - y_1}(0 + 200)$

$= 250 + \frac{250 - 0}{-200 - 0} \times 200$

$x_2 = 0$

New coordinate,

$(x_2\ y_2) = (0,0)$

New outcode 2 = 0000

Clipping equations:

Top:-

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

$y_1 = y_{max}$

bottom:

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

$y_1 = y_{min}$

Right:-

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

$x = x_{max}$

left:

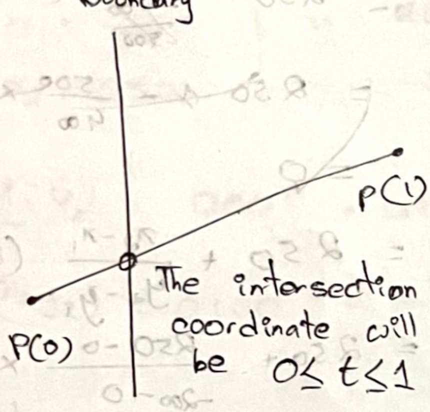
$y_1 = y_1 + m(x_{min} - x_1)$

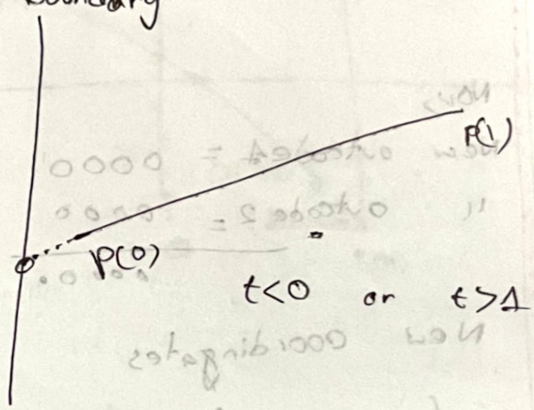
$x = x_{min}$.

Cyrus Beck Clipping Algorithm.

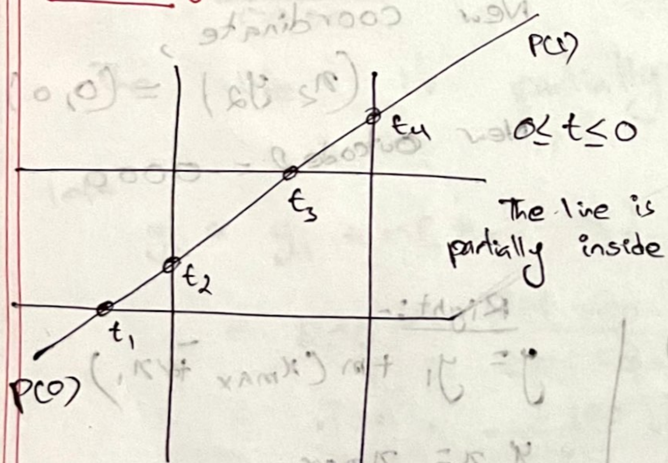
Conditions: $0 < t < 1$ points lie inside the line.

$t < 0$ or $t > 1$ coordinates will be outside the line segments.

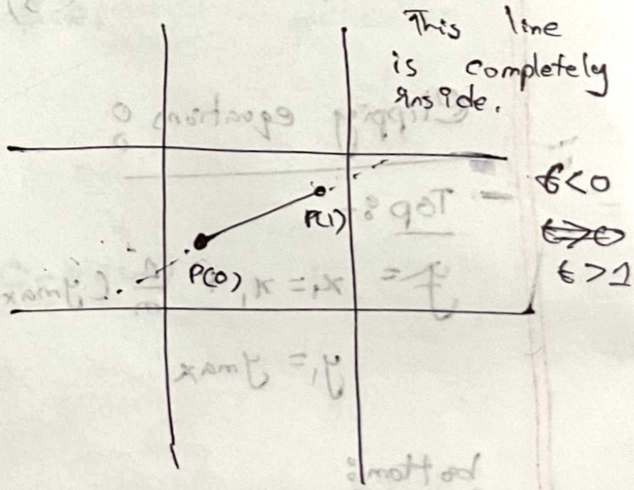




Case-1:

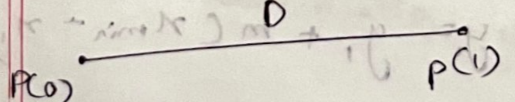


Case-2



Vector

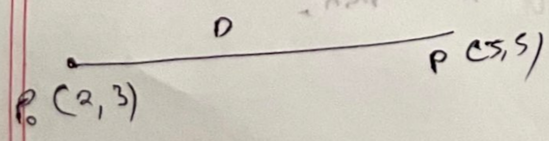
(D is a directional vector)



D is a vector from P to P_1

$D = P_1 - P_0$

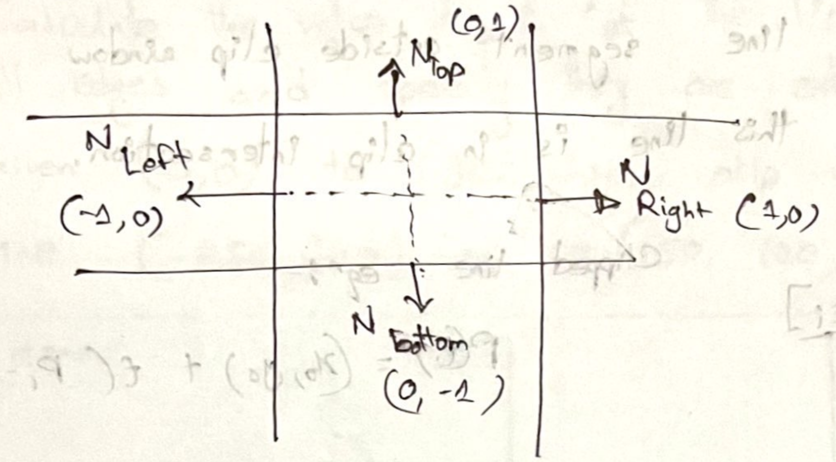
$= (x_1 - x_0, y_1 - y_0)$



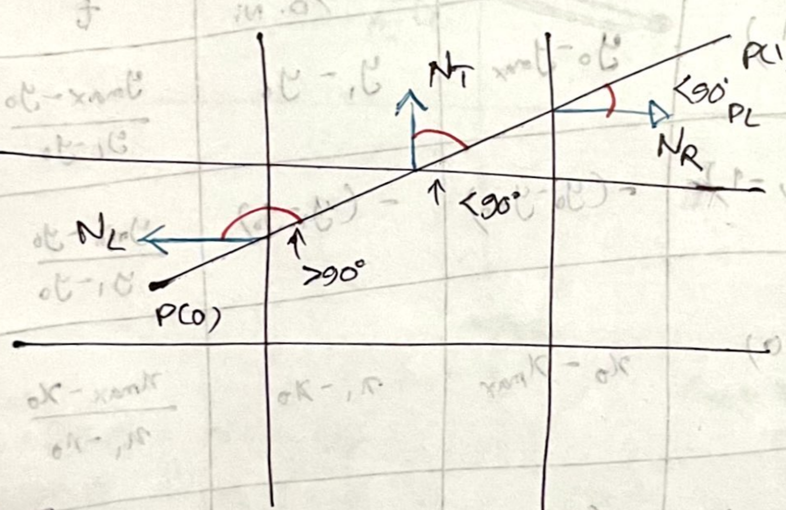
$D = (5-2, 5-3)$

$= (3, 2)$

Normal ~~to~~ vector to boundary:



D with normal ~~to~~ vector boundary



Multiply

$N_i \cdot D < 0$ >90°

→ PE
Potentially entering

$N_i \cdot D > 0$ <90°

PL
Potentially leaving

ALGORITHM:-

(i) Calculate t values of intersections points with each boundaries
(ii) Classify intersection points whether it is PE/PL

→ Select PE with highest t/t_E and P_L with ~~highest~~ lowest value of t/t_L

$N_i \cdot D < 0$ PE → t_E (max)

$N_i \cdot D > 0$ PL → t_L (min)

~~(iii) Use paramet~~

(iii) $t_E > t_L$ line segment outside clip window

$t_E \le t_L$ this line is in clip intersection.

Clipped line eqⁿ:-

$$t = \frac{\cancel{\emptyset} N_i \cdot [P_0 - P_{Ei}]}{-N_i \cdot D}$$

$$P(t) = (x_0, y_0) + t(P_1 - P_0)$$

Boundary	N	$(P_0 - P_E) N$	$(P_1 - P_0) N_i / D \cdot N_i$	t
Top	$(0, 1)$	$y_0 - y_{max}$	$y_1 - y_0$	$\frac{y_{max} - y_0}{y_1 - y_0}$
Bottom	$(0, -1)$	$-(y_0 - y_{min})$	$-(y_1 - y_0)$	$\frac{y_{min} - y_0}{y_1 - y_0}$
Right	$(1, 0)$	$x_0 - x_{max}$	$x_1 - x_0$	$\frac{x_{max} - x_0}{x_1 - x_0}$
Left	$(-1, 0)$	$-(x_0 - x_{min})$	$-(x_1 - x_0)$	$\frac{x_{min} - x_0}{x_1 - x_0}$

$D = P_1 - P_0$

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

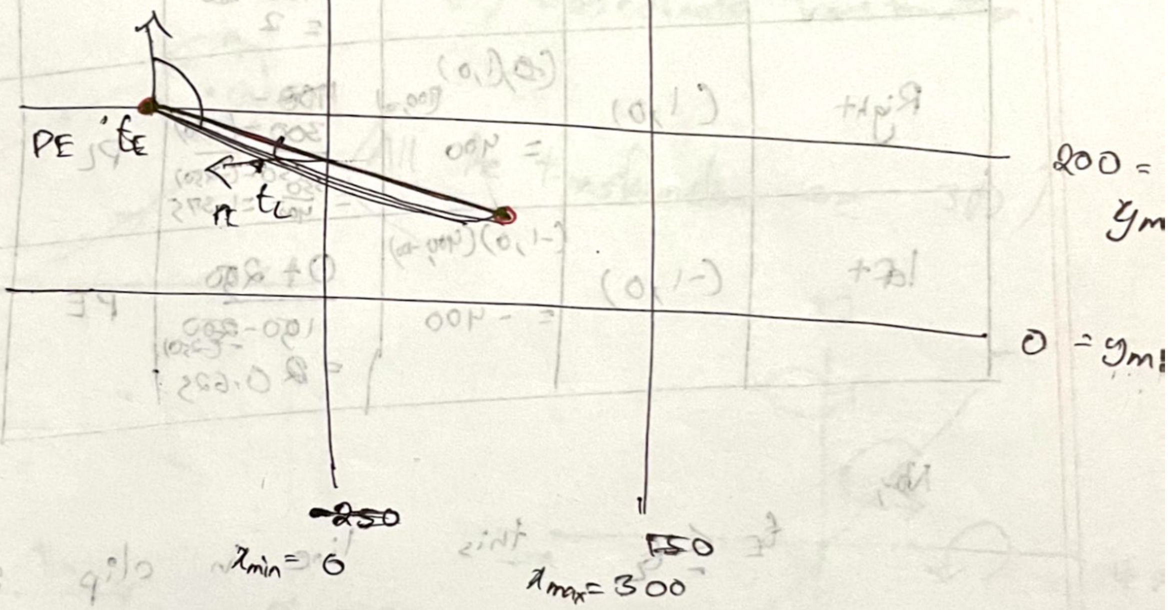
Example :-

Calculate the value of t of the line given below for all edges and specify they are entering or leaving t.

Given (0,0) to (300, 200) clip region.

line (-250, 200) to Q(150, 100)

Here,



$x_0 = -250 \mid x_1 = 150$

$y_0 = 200 \mid y_1 = 100$

$D = P_1 - P_0$

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

$= [(150 - (-250)), (100 - 200)]$

$D = (400, -100)$

$y_{max} = 200$ | $x_{max} = 300$ | $x_0 = -250$ | $x_1 = 1$
$y_{min} = 0$ | $x_{min} = 0$ | $y_0 = 200$ | $y_1 = 100$

Initiate

Table: $t_E = 0$ | $t_L = 1$

dot product

Boundary	N_i	$N_i \cdot D$	t	P_E/P_L	t_E	t_L
Top	$(0,1)$	$(0,1)(400,-100) = -100 < 0$	$\frac{200-200}{100-200} = 0$	PE	0	1
Bottom	$(0,-1)$	$(0,-1)(400,-100) = 100$	$\frac{0-200}{100-200} = 2$	PL	0	1
Right	$(1,0)$	$(1,0)(400,-100) = 400$	$\frac{\cancel{400}\ 300-(-250)}{150-(-250)} = \frac{550}{400} = 1.375$	PL	0	1
left	$(-1,0)$	$(-1,0)(400,-100) = -400$	$\frac{0+250}{150-(-250)} = 0.625$	PE	0.625	1

Now,

$t_E \le t_L$ this line in clip i.e. intersection.

Entering coordinate, (endpoint)

$$P(0.625) = (x_0, y_0) + t(P_1 - P_0)$$
$$= (-250, 200) + t(400, -100)$$
$$= (-250, 200) + 0.625(400, -100)$$
$$= (0, 137.5)$$

Leaving coordinate, endpoint =

$$P(1) = (150, 100)$$

endpoints of line

~~GEOMETRIC~~ TRANSFORMATION

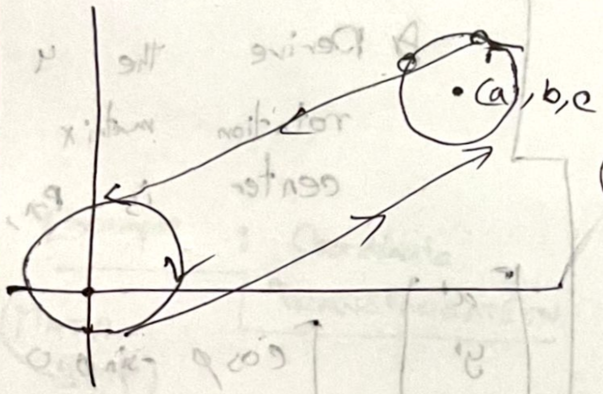
(i) Translation
(ii) Rotation
(iii) Reflection
(iv) Scaling

$\rightarrow$ Motion

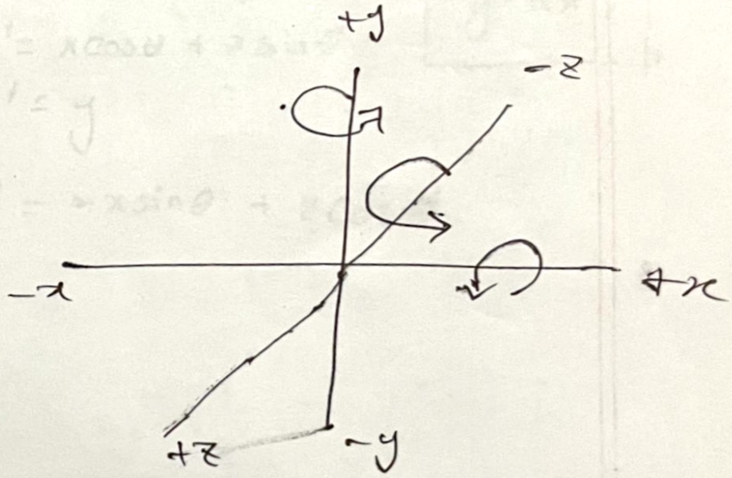
Classification :-

(1) Based on scaling :-
- (i) Rigid-body Motion (Uniform scaling)
- (ii) Affine motion (non-uniform scaling)

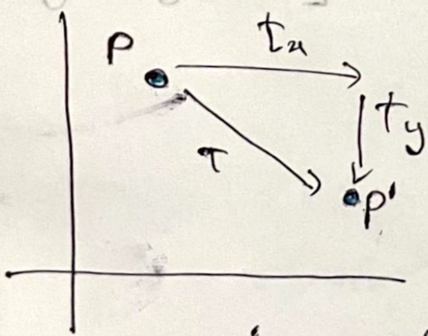
(2) Based on center of rotation :-
- (i) Geometric transformation
- (ii) Coordinate transformation.



(All the transformation are 3D)



Rotation across x-axis
" " y-axis
" " z-axis



$P' = P + T$

$x' = x + t_x$

$y' = y + t_y$

$z' = z + t_z$

$|P'| = |T| \cdot |P|$

$$\underbrace{\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}}_{P'} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T} \cdot \underbrace{\begin{vmatrix} x \\ y \\ z \\ 2 \end{vmatrix}}_{P}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

x y coordinate

translation

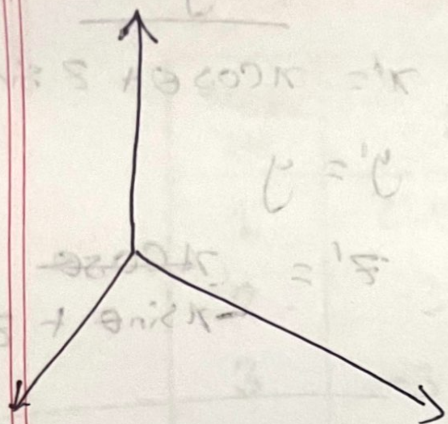
Rolling means → rolling up & dam

* Derive 4 by 4 rotation matrix while thee center is origin

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

* Derive the 4 by 4 rotation matrix while the center is p_x, p_y, p_z

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & p_x(1-\cos \\ \sin\varphi & \cos\varphi & 0 & -p_x(1-c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$R_z$$

$$x' = x\cos(t) - y\sin(t)$$
$$y' = x\sin(t) + y\cos(t)$$
$$z' = z$$

rotating across z axis

$$R_x$$

$$x' = x$$
$$y' = y\cos(t) - z\sin(t)$$
$$z' = y\sin(t) + z\cos(t)$$

rotating ~~y~~ across ~~y~~ x axis

Example : Coordinate TRANSFORMATION

(MATH IMP)

$$R_y$$

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

rotating across y-axis

Transformation Matrix :-

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & C_x(1-\cos\varphi \\ \sin\varphi & \cos\varphi & 0 & \\ 0 & 0 & & \\ 0 & 0 & & \end{bmatrix}$$

R_y

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

$y' = y$

$z' = \cancel{x\cos\theta}$
$-x\sin\theta + z\cos\theta$

$x' = x$

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

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

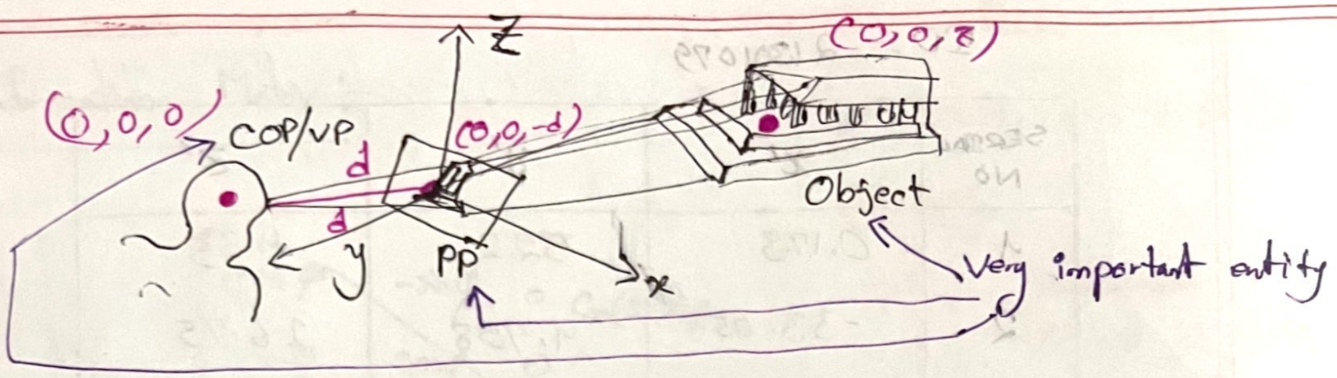
3D $\rightarrow$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & P_x(1-\cos\varphi) + P_y\sin\varphi \\ \sin\varphi & \cos\varphi & 0 & P_y(1-\cos\varphi) - P_x\sin\varphi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2D $\rightarrow$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & P_x(1-\cos\varphi) + P_y\sin\varphi \\ \sin\varphi & \cos\varphi & P_y(1-\cos\varphi) - P_x\sin\varphi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

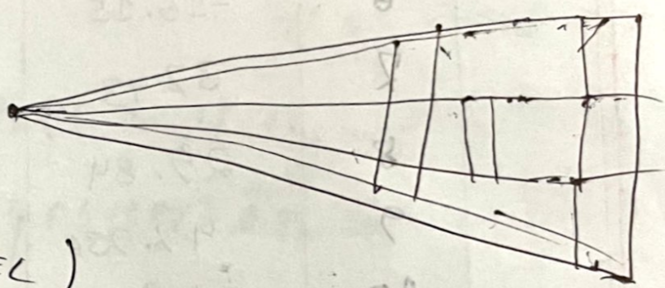
VIEWING OR PROJECTION



Three center of rotation.

1 vanishing point

T



ORTHOGRAPHIC PROJECTION (PARALLEL)

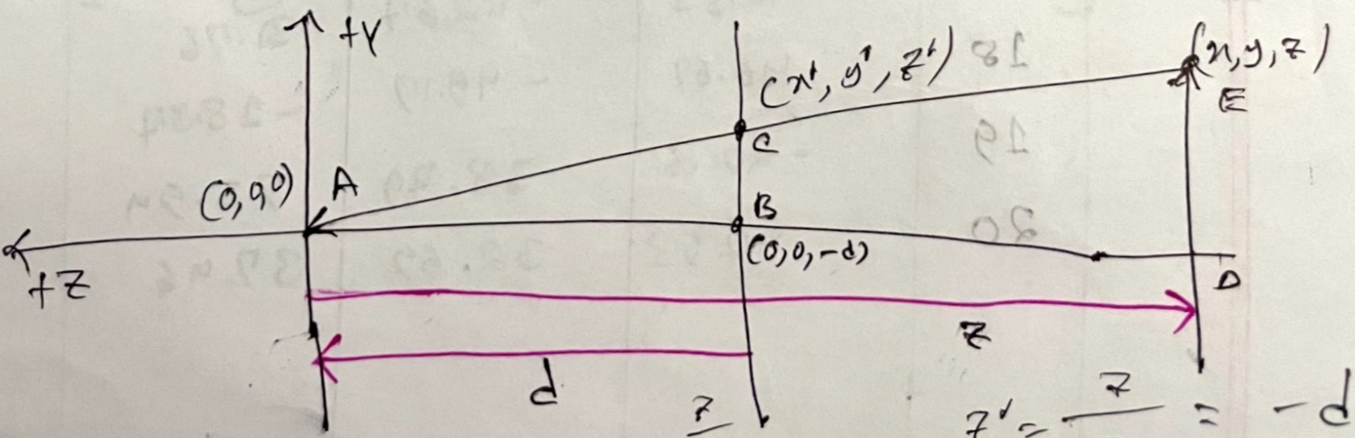
PERSPECTIVE PROJECTION ☆

☆ Description of PP

PERSPECTIVE MATRICES:

VIEWING TRANSFORMATION:

Derivation of simple perspective Projection mattm



$w = -\frac{z}{d}$

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

☆ COP is on the ~~origin~~ origin (Simple perspective projection matrices) (☆ imp) [derivation]

☆

~~Derive~~ General Purpose Projection Matrix

23/11/23.

☆ Derive 4x4 general Purpose Projection matrix

Example :-

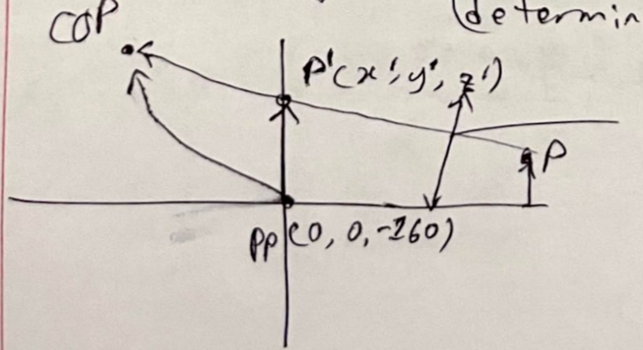
$Q_x = ?$
$Q_y = ?$
$Q_z = ?$
$z_p = ?$

if
$Q = (120, -40, 160)$
Q_x Q_y Q_z

COP is given then we have to determine Q_x, Q_y, Q_z

Question :- (i) diagram given

(ii) coordinate of projection plane, cod of cop are give (determine P')



this two will be the same after calculation, if $w=1$ the magnitude of Homogenous and cartesian are same

$$\left[\text{homogenous} \leftarrow \begin{cases} x'' = () \\ y'' = () \\ z'' = () \\ w = 1 \end{cases} \right. \quad \left| \quad \left. \begin{array}{l} x' = ()/w \\ y' = ()/w \\ z' = ()/w \\ w = ()/w \end{array} \right] \rightarrow \text{cartesian} \right.$$

Quiz

.P will be given | find .P'

(Z_p
Q_x
Q_y
Q_z) given

CMY is opposite to RGB.

COLOUR MODEL OF LIGHT :

Two models :-

(i) Monitor $\longrightarrow$ RGB

(ii) Printer $\longrightarrow$ CMY

The Color Model of Light :

(i) HSV

Hue, Saturation, Value

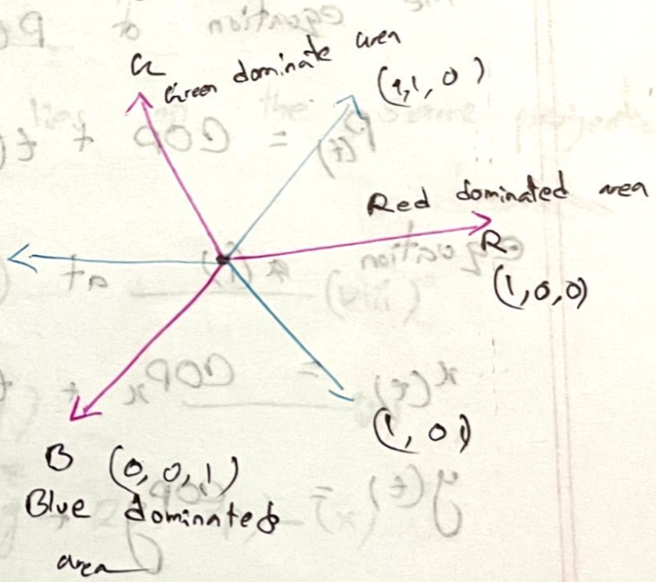
(ii) HSL

In some operating system

$H = 0$ to $360/240$

$S = 0$

RGB to HSV conversion :-



$RGB = (0.26,\ 0.19,\ 0.89)$

$V = 0.89$

$$S = \frac{(0.89 - 0.\cancel{26}\,19)}{0.89} = 0.787$$

$$H = \left(\frac{0.26 - 0.19}{0.89 - 0.19} \times 60\right) + 240$$

$= 246°$

RGB to HSL conversion :

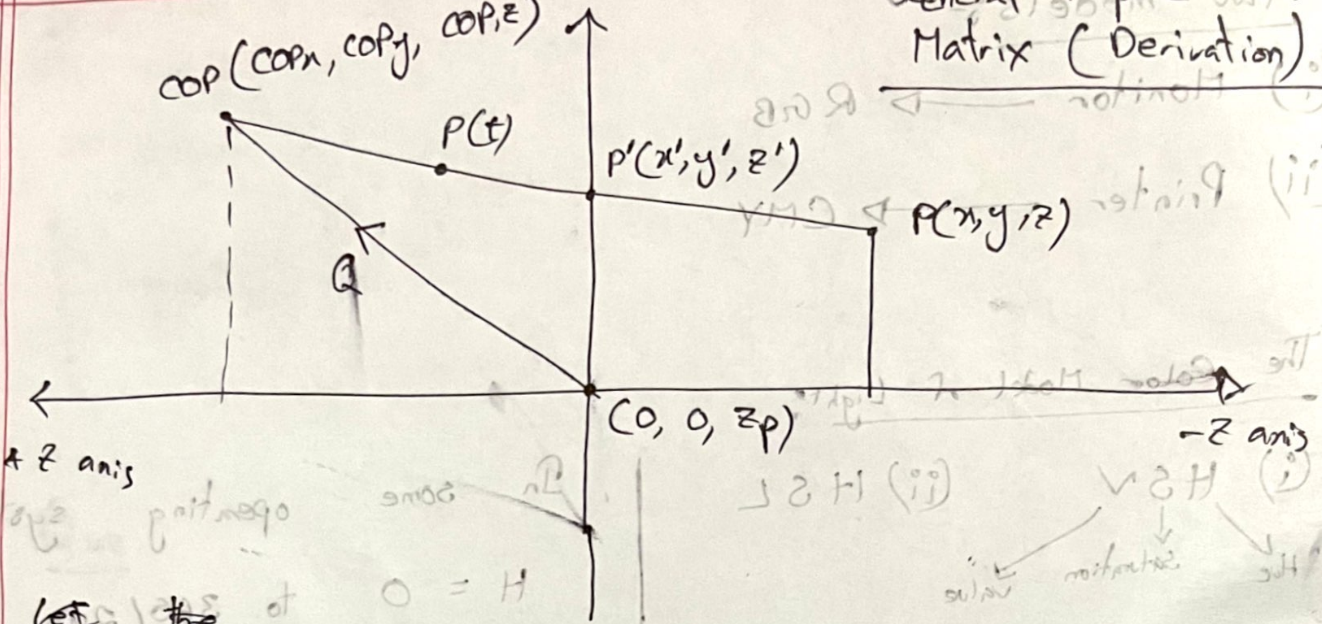
$$L = \frac{\max(R, G, B) + \min(R, G, B)}{2}$$

$RGB = (1.0,\ 1.0,\ 1.0)$

$L = 0.5$

PROJECTION :-

General Purpose Projection Matrix (Derivation) :-



~~Let, the~~

The equation of P(t) is,

$$P_{(t)} = COP + t(P - COP) \quad \text{---} \quad (i)$$

equation (i) at all planes,

$$x(t) = COP_x + t(P_x - COP_x) \quad \text{---} \quad (ii)$$

$$y(t) = COP_y + t(P_y - COP_y) \quad \text{---} \quad (iii)$$

$$z(t) = COP_z + t(P_z - COP_z) \quad \text{---} \quad (iv)$$

Now, the distance will be,

$$Q = COP - (0, 0, Z_p)$$

we can say,

$COP_x = Q_x$ | $COP_y = Q_y$ | $COP_z = Q_z + Z_p$

Substituting the values of COP_x, COP_y, and COP_z at (ii), (iii) and (iv), respectively,

[~~Let $t = t_p$, $P(t)$ and lies on the same projection plane P'~~]

$$x(t) = Q_x + t_p (x - Q_x) \quad \text{--- (v)}$$

$$y(t) = Q_y + t_p (y - Q_y) \quad \text{--- (vi)}$$

$$z(t) = Q_z + z_p + t_p (z - Q_z - z_p) \quad \text{(vii)}$$

~~From equation (vii) we get~~

Let, $t = t_p$ and, $P(t)$ lies on the same projection plane P'

$$x' = Q_x + t_p (x - Q_x) \quad \text{--- (viii)}$$

$$y' = Q_y + t_p (y - Q_y) \quad \text{--- (ix)}$$

$$z' = Q_z + z_p + t_p (z - Q_z - z_p) \quad \text{--- (x)}$$

From (x) we get, $[z' = z_p]$

$$z_p = Q_z + z_p + t_p (z - Q_z - z_p)$$

$$t_p = \frac{-Q_z}{z - Q_z - z_p} = \frac{1}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

Substituting the value t_p in ~~x', y', z'~~ (viii), (ix), (x).

$$x'_t = Q_x + \frac{x - Q_x}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$= \frac{-z\frac{Q_x}{Q_z} + Q_x + z_p\frac{Q_x}{Q_z} + x - Q_x}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$x' = \frac{-z\frac{Q_x}{Q_z} + z_p\frac{Q_x}{Q_z} + x}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

and,

$$y' = Q_y + \frac{y - Q_y}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$y' = \frac{-z\frac{Q_y}{Q_z} + z_p\frac{Q_y}{Q_z} + y}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

~~and~~, and,

$$z' = Q_z + z_p + \frac{z - Q_z - \frac{z_p}{Q_z}z}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$z' = \frac{\cancel{z_1} - \frac{Q_z}{Q_z}\times z + \cancel{Q_z} + \frac{z_p \times Q_z}{Q_z} - \frac{z_p^2}{Q_z} + \cancel{z_p} + \frac{z_p^2}{Q_z} + z - \cancel{Q_z} - \cancel{z_p}}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$z' = \frac{-z + \cancel{z_p} - \frac{z_p}{Q_z} z + \frac{z_p^2}{Q_z} + z}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}}$$

$$= \frac{\cancel{-z} \; \cancel{+} - \frac{z_p}{Q_z} z + \frac{z_p^2}{Q_z}}{-\frac{z}{Q_z} + 1 + \frac{z_p}{Q_z}} = \frac{-\frac{z_p}{Q_z} z + z_p + \frac{z_p^2}{Q_z}}{}$$

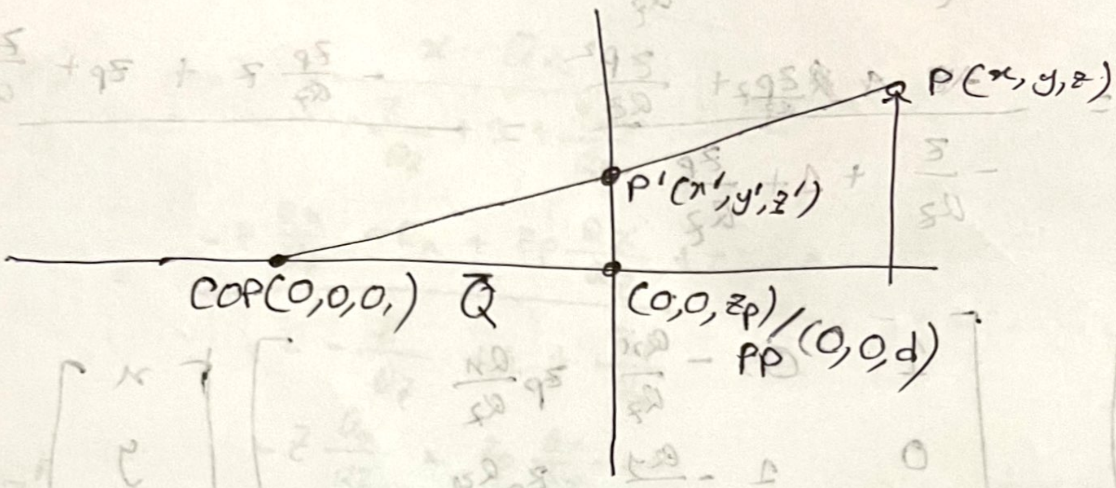
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{Q_x}{Q_z} & z_p \frac{Q_x}{Q_z} \\ 0 & 1 & -\frac{Q_y}{Q_z} & z_p \frac{Q_y}{Q_z} \\ 0 & 0 & -\frac{z_p}{Q_z} & z_p\left(1 + \frac{z_p}{Q_z}\right) \\ 0 & 0 & -\frac{1}{Q_z} & 1 + \frac{z_p}{Q_z} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Origin

(Center of Projection)

~~Comparison~~.

① ORIGIN on COP :-



$Q = COP - (0,0,d)$

~~$COP_x = Q_x = 0$~~

~~$COP_y = Q_y = 0$~~

~~$COP_z = Q_z =$~~

$Q_x = COP_x = 0$

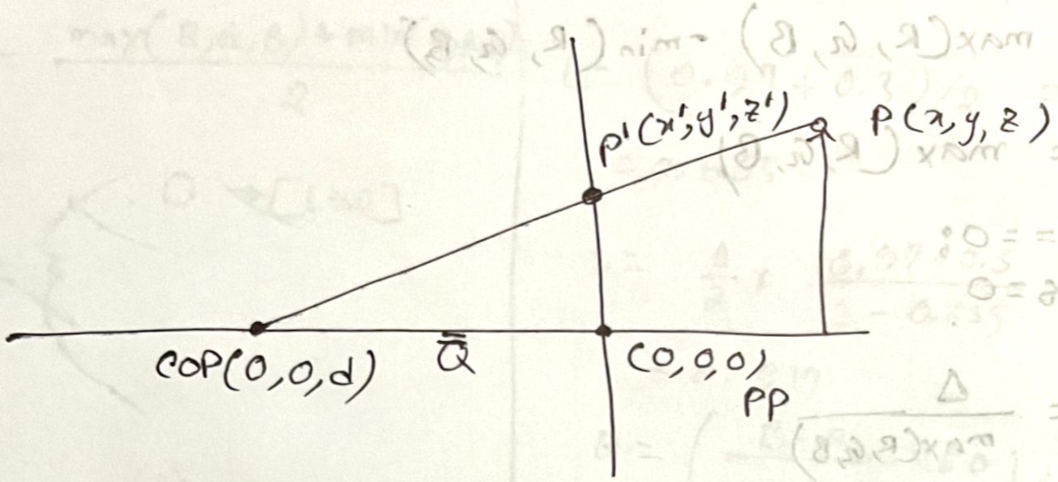
$Q_y = COP_y = 0$

$Q_z = COP_z - z_p$

$= 0 - d = -d$

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

ORIGIN ON PP (Projection Plane):



$Q_x = COP_x = 0$

$Q_y = COP_y = 0$

$Q_z = COP_z - 0 = d$

$Q = COP - (0, 0, 0)$

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

COLOR MODEL OF ~~LIGH~~ LIGHT.

RGB to HSV

$\Delta = \max(R, G, B) - \min(R, G, B)$

$V = \max(R, G, B)$

if $V == 0$:

$S = 0$

else:

$$S = \frac{\Delta}{\max(R, G, B)}$$

if $\max(R, G, B) == R$:

$$H = \frac{G-B}{\Delta} \times 60$$

elif: $\max(R, G, B) == G$:

$$H = \left(\frac{B-R}{\Delta} \times 60\right) + 120$$

else:

$$\left(\frac{R-G}{\Delta} \times 60\right) + 240$$

RGB to HSL :

$$L = \frac{max(R,G,B) + min(R,G,B)}{2}$$

$$S = \begin{cases} \text{(i)} \quad 0 \longrightarrow [L=0] \\ \text{(ii)} \quad \frac{1}{2} \times \frac{max(R,G,B) - min(R,G,B)}{L} \Rightarrow \frac{max(R,G,B) - min(R,G,B)}{max(R,G,B) + min(R,G,B)} \longrightarrow [0 < L < 0.5] \\ \text{(iii)} \quad \frac{1}{2} \times \frac{max(R,G,B) - min(R,G,B)}{1 - L} \\ \Rightarrow \frac{max(R,G,B) - min(R,G,B)}{2 - [max(R,G,B) + min(R,G,B)]} \longrightarrow [0.5 \le L < 1] \\ \text{(iv)} \quad 0 \longrightarrow [L = 1] \end{cases}$$

"H" is same as HSV

[if H is (-)ve add 360° with it.]

Example :-

$R = 0.3$ | $G = 0.97$ | $B = 0.67$

$L = (0.97 + 0.3)/2 = 0.5$

$= 0.635$

$S = \frac{1}{2} \times \frac{0.97 - 0.3}{1 - 0.635}$

$= 0.917$

$H = \left(\frac{B - R}{\Delta} \times 60\right) + 120$

$= \frac{0.67 - 0.3}{0.97 - 0.3} \times 60 + 120$

Ambient reflection:- It is the reflection of light [which does] not come directly from a light source.

Intensity of light → I_a, ambient coefficient → K_a

★ $A = I_a K_a$

PHONG'S REFLECTION MODEL:-

(i) Colour the pixel. (ambient reflection).

(ii) Try to show 3D shape. (Diffuse reflection)

(iii) (i) + (ii) => 3D image

(iv) Give a shinning effect (specular reflection)

(v) (iii) + (iv) => Real model

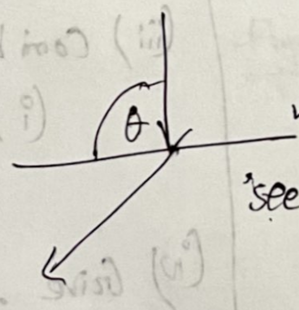
Diffuse reflection:- always occurs in rough surface

It is the reflection of light such that a ray incident on the surface is scattered at many angle.

→ (equal change of being reflected in any direction).

→ every reflection will have different intensity

~~Diffus-~~



we will see no light it

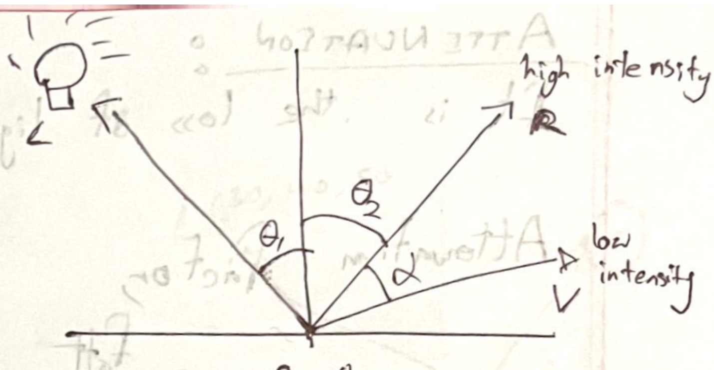
$D = I_p k_d \max(\cos\theta, 0)$

$L \cdot n = \cos\theta$

★ $D = I_p K_d \max(L \cdot n, 0)$

$D = I_p K_d (N \cdot L)$

Specular Reflection:-



$\theta_1 = \theta_2$

angle increase = ligh intensity decrease

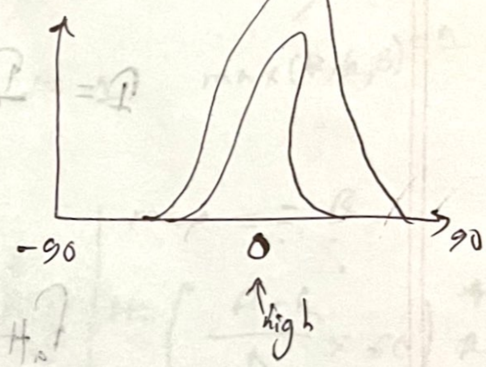
intensity of light

☆ $S = I_p K_s \cos^n(\alpha)$

specular coefficient

$K_s = [0, 1]$

the closer to 0 K_s value the Light intensity decreases



α = angle between R & V

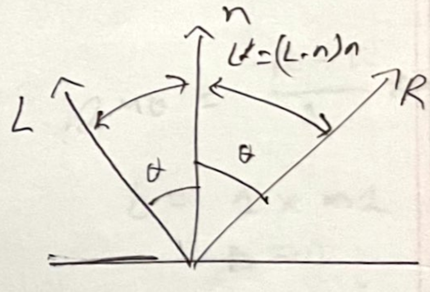
n = amount of shininess

less n value → shininess↑

☆ $\cos\alpha = V \cdot R$ [V & R are unit vectors]

finding R → (i) using rotation

(ii) projection of vector.



$L + R = 2(L \cdot n)n$

$R = 2(L \cdot n)n - L$

Together we can say:-

(phong's reflection model):-

$$I = \underbrace{I_a K_a}_{\text{Ambient Reflection}} + \underbrace{I_p K_d \max(L \cdot n, 0)}_{\text{Diffuse Reflection}} + \underbrace{I_p K_s (\max(\overset{\cos\theta}{V \cdot R}, 0))^n}_{\text{Specular Reflection}}$$

[$L \cdot n > 0$ must be (+)ve]

ATTENUATION :

It is the loss of light energy over space.

high attenuation factor = lower intensity.

Attenuation factor,

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

For one light source Intensity (with attenuation)

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

$$f_{att} = \frac{1}{a_0 + a_1 d + a_2 d^2}$$

FORMULA's

Cohen Sutherland

Top :-

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

$y_1 = y_{max}$

Right :-

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

$x = x_{max}$

1 - violates
0 - accepts

far near Top bottom right left

The eight way

Cyrus beck

$t_E < t$

$N_i \cdot D < 0$ PE $\rightarrow t_E$ (max)

$N_i \cdot D > 0$ PL $\rightarrow t_L$ (min)

$t_L > t$

$D = P_1 - P_0$

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

Clipped line

$P(t) = (x_0, y_0) + t(P_1 - P_0)$

Boun	N_i	$N_i \cdot D$	t	PE/PL	t_E	t_L

MIDPOINT LINE (Zone-0)

$dNE = 2\Delta y - 2\Delta x \quad d \geq 0$

$dE = 2\Delta y \quad d < 0$

$d_{init} = 2\Delta y - \Delta x$

$d_{new} = d_{old} + \Delta dNE$

" = " $+ \Delta dE$

MID POINT Circle (Zone-1)

$\Delta SE = 2x_p - 2y_p + 5 \quad d > 0$

$\Delta E = 2x_p + 3$

$d_{new} = d_{old} + dSE$

" = " $+ dE$

$d_{init} = \frac{5}{4} - r$

$= 1 - r$

DDA

$-1 < m < 1$

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k + m$

else:-

$y_{k+1} = y_k + 1$

$x_{k+1} = x_k + \frac{1}{m}$