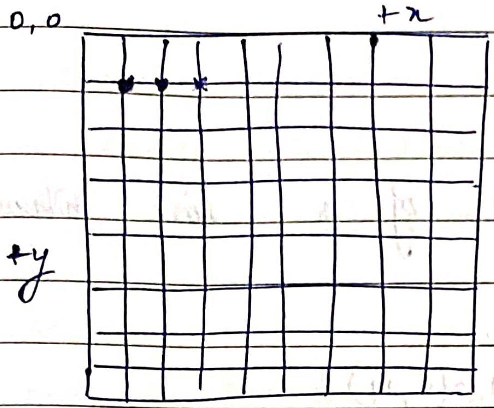
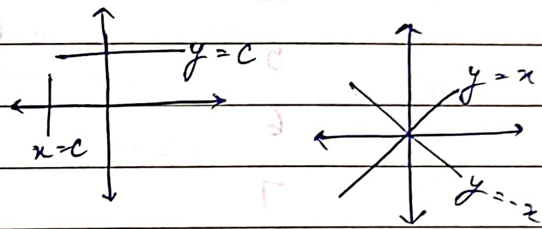
chapter 3



pixels → on
└→ off

$y = c$ $m = 0$



$x = c$ $m = \infty$

$y = x$ $m = 1$

$y = -x$ $m = -1$

CHAPTER 3 (1 Ref.)

$\boxed{\text{DDA (Digital Differential Algo)}}$

It is also called incremental algo.

In this algo 2 points will be given A & B (endpoints) and algo will give all points that needs to be illuminated.

$A = (x_1, y_1)$ $B = (x_2, y_2)$

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

$x_i, y_i \rightarrow (x_{i+1}, y_{i+1})$

$\downarrow x_i + 1$ $\rightarrow y_i + m$

$$y_i = m x_i + c$$

$$y_{i+1} = m x_{i+1} + c$$
$$= m(x_i + 1) + c$$
$$= m x_i + m + c$$
$$= y_i + m$$

Ques Generate all the points of a line whos coordin are (5,8) and (9, 11)

$(x_1, y_1) = (5,8)$ $(x_2, y_2) = (9,11)$

$m = 3/4$

x	y
5	8
6	$8 + \frac{3}{4} = 8.75 \simeq 9$
7	$8.75 + \frac{3}{4} = 9.5 \simeq 10$
8	$9.5 + \frac{3}{4} = 10.25 \simeq 10$
9	$10.25 + \frac{3}{4} = 11$

DDA line algo works only for $-1 \leq m \leq 1$

Ques Using DDA line algo illuminate the points of the line (1,4) (2,10)

$(x_1, y_1) = (1,4)$ $(x_2, y_2) = (2,10)$

$m = \frac{10-4}{2-1} = 6$

As the slope of the line is greater than 1 we canno apply DDA

* Special case will be used.

Limitations:

(i) Works only for slope b/w -1 and 1
(ii) Round off error is there at every step.

Solution -

Mid point line algo

4 Special cases of DDA line algo -

(1) Horizontal line $[(y=c) \Rightarrow m=0]$

$(x_1, c) \quad (x_2, c)$

↳ Here we can directly write the points as x will increment by 1 and y will be constant.

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

$y = c$

(2) Vertical line : $[(x=c) \Rightarrow m = \infty]$

$(c, y_1) \quad (c, y_2)$

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

$x = c$

(3) m = 1 $\quad \Delta y = \Delta x$

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

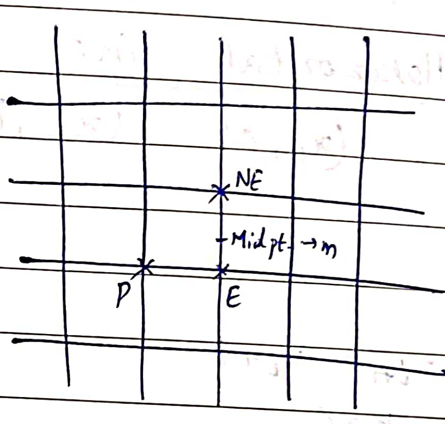
$x_{i+1} = x_i + 1$

(4) $m > 1$ $\qquad \Delta y > \Delta x$

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

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

Mid point line Algo (Bresenhem's line drawing Algo)



$y = mx + B$

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

$dx\, y = dy\, x + dx\, B$

If m the mid pt. of E & NE is above the line ~~...~~ E is illuminated

If m is below NE is iluminated

$F(x,y) = dy\, x - dx\, y + dx\, B$

$$\boxed{F(x,y) = ax + by + c}$$

$a = dy$ $\qquad dy = y_2 - y_1$

$b = -dx$ $\qquad dx = x_2 - x_1$

$c = dx\, B$

→ point P is a pt. on the line and integral

$P = (x_p, y_p)$

$E = (x_p + 1, y_p)$

$NE = (x_p + 1, y_p + 1)$

$m = (x_p+1, y_p+\frac{1}{2})$ # No actual pt. on the screen (imaginary point), can never be iluminated.

$F(m) = F(x_p+1, y_p+\frac{1}{2})$

$d = F(m) = a(x_p+1) + b(y_p+\frac{1}{2}) + c \longrightarrow d_{old}$

↓

decision variable.

If $d > 0$

↳ mid point is below the line

NE

If $d < 0$

↳ mid point is above the line

E

If $d = 0$

↳ mid-point is on the line

E/NE any can be taken

but we take E

→ Case 1

Chose E

New P $= x_p+1, y_p$

New E $= x_p+2, y_p$

New NE $= x_p+2, y_p+1$

New m $= x_p+2, y_p+\frac{1}{2}$

$d_{new} = F(\text{New } m)$
$= a(x_p+2) + b(y_p+\frac{1}{2}) + c$

$\Delta d_E = d_{new} - d_{old}$

$= a = dy = y_2 - y_1$

The increment in d when E/NE is chosen is a constant

→ <u>Case 2</u>

chose NE

New P $= x_p+1, y_p+1$

New E $= x_p+2, y_p+1$

New NE $= x_p+2, y_p+2$

New m $= x_p+2, y_p+3/2$

$d_{new} = f(\text{New } m)$

$= a(x_p+2) + b(y_p+3/2) + c$

$\Delta d_{NE} = d_{new} - d_{old}$

$= a + b$

$= dy - dx$

→ Initial value of d

Initial point $= x_1, y_1$

Initial m $= x_1+1, y_1+\frac{1}{2}$

$d_m = f(\text{Initial } m)$

$d_m = f(x_1+1, y_1+1/2)$

$= a(x_1+1) + b(y_1+1/2) + c$

$= ax_1 + by_1 + c + a + b/2$

$d_{init} = a + b/2 = dy - \frac{dx}{2}$

Ques. Generate the points of the line whos end points are (5,8), (9,11). Using Bresenham's line Algo

ini d = $\boxed{dy - \frac{dx}{2}}$ $\qquad$ $dy = 11-8 = 3$

$= 3 - 4/2 = 1$ $\qquad$ $dx = 9-5 = 4$

$\Delta d_E = dy$

$\Delta d_{NE} = dy - dx$

As $d = 1 > 0 \longrightarrow NE$

$1 + \Delta d_{NE}$

$1 + (3-4)$

$1 - 1 = 0 \Rightarrow E$

$\Rightarrow 0 + \Delta d_E$

$0 + 3 = 3 \Rightarrow NE$

$\Rightarrow$ ~~Ole~~ $3 + \Delta d_{NE}$

$3 + (3-4) = 2 \Rightarrow NE$

$\Rightarrow 2 + \Delta d_{NE}$

$2 + (3-4) = 1 \Rightarrow NE$

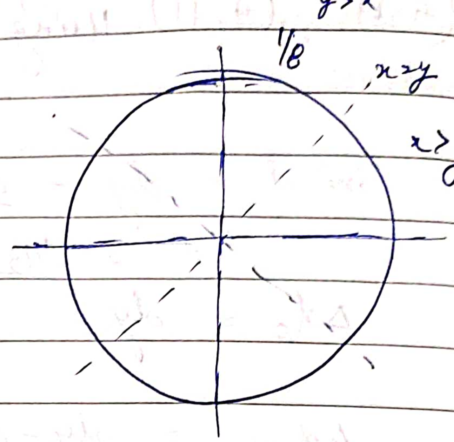
x	y
5	8
6	9
7	9
8	10
9	11

Bresenham's Circle Algo / Mid point circle Algo

$x^2 + y^2 = R^2$

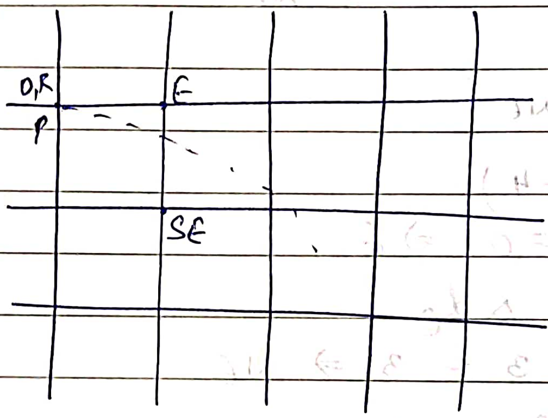
$(x-a)^2 + (y-b)^2 = R^2$

$\hookrightarrow (a,b)$ Radius $= R$



x, y
$-x, -y$
$-x, y$
$x, -y$

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



$P = x_p, y_p$

$E = x_p+1, y_p$

$SE = x_p+1, y_p-1$

$m = x_p+1, y_p-1/2$

$d = f(m)$

$= f(x_p+1, y_p - 1/2)$

$d = (x_p+1)^2 + (y_p - 1/2)^2 - R^2$

\# $d > 0 \Rightarrow SE$

\# $d < 0 \Rightarrow E$

\# $d = 0 \Rightarrow SE$ $\rightarrow$ any of the 2 can be chosen but we take SE

Case 1 →

chose E

New P $= x_p+1, y_p$

$E_{new} = x_p+2, y_p$

$SE_{new} = x_p+2, y_p-1$

$m_{new} = x_p+2, y_p-1/2$

$d_{new} = F(x_p+2, y_p-1/2)$

$= (x_p+2)^2 + (y_p-1/2)^2 - R^2$

$\Delta d_E = d_{new} - d_{old}$

$= (x_p+2)^2 + (y_p-1/2)^2 - R^2 - [(x_p+1)^2 + (y_p-1/2)^2 - R^2]$

$= 2x_p + 3$

Case 2 →

chose SE

New P $= x_p+1, y_p-1$

$E_{new} = x_p+2, y_p-1$

$SE_{new} = x_p+2, y_p-2$

$m_{new} = x_p+2, y_p-3/2$

$d_{new} = F(x_p+2, y_p-3/2)$

$= (x_p+2)^2 + (y_p-3/2)^2 - R^2$

$$\Delta d_{NE} = d_{new} - d_{old}$$

$$= \left[(x_p+2)^2 + (y_p - 3/2)^2 - R^2\right] - \left[(x_p+1)^2 + (y_p - 1/2)^2 - R^2\right]$$

$$= \cancel{x_p^2} + \cancel{4}^{3} + 4x_p + \cancel{y_p^2} + \frac{9}{4} - 3y_p - \cancel{x_p^2} - \cancel{1} - 2x_p - \cancel{y_p^2} - \frac{1}{4} + y_p$$

$$= 2x_p - 2y_p + 5$$

→ Initial value of d

NOTE: The change in d when E/SE is chosen is a linear function in x and y

If nothing is said, origin is taken as centre

Ques $(0,0)$ $R = 6$

x	y	d
0	6	d_{ini} $\frac{5}{4} - R = \frac{5}{4} - 6 = \frac{-19}{4} < 0 \Rightarrow E$
1	6	$\frac{-19}{4} + \Delta d_E \Rightarrow \frac{-19}{4} + (2x_p + 3)$ $\frac{-7}{4} < 0 \Rightarrow E$
2	6	$\frac{-7}{4} + \Delta d_E \Rightarrow \frac{-7}{4} + (2x_p + 3)$ $\frac{13}{4} > 0 \Rightarrow SE$
3	5	$\frac{13}{4} + 2x_p - 2y_p + 5 \Rightarrow$ $= \frac{13}{4} + 2 \times 2 - 2 \times 6 + 5$ $= \frac{1}{4} > 0 \Rightarrow SE$
4	4	

$(0,6) \rightarrow (6,0), (-6,0), (0,-6)$

$(1,6) \rightarrow (1,-6), (-1,-6), (-1,6)$
$(6,-1), (6,1), (-6,-1), (-6,1)$

$(2,6) \rightarrow (-2,6), (2,-6), (-2,-6), (6,-2), (6,2), (-6,-2)$
$(-6,-2)$

$(3,5) \rightarrow$

$(4,4) \rightarrow$

Ques Generate all the points of a circle whose equation is $x^2 + y^2 - 25 = 0$

$R = 5$ $R \neq 25$

Ques Generate all the points of the circle whose equation is

$$(x-1)^2 + (y-2)^2 = 16$$

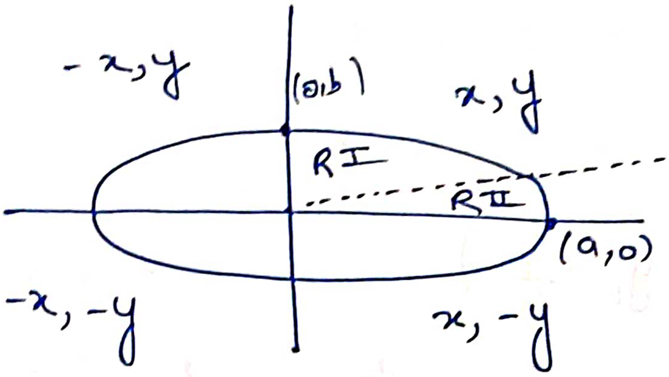
We will translate (shifting) the circle from the centre $(1, 2)$ to $(0, 0)$

$$x^2 + y^2 = 16$$

Ans
0, 4 → 1, 6 (0, −4), (−4, 0), (4, 0)
1, 4 → 2, 6 (1, −4), (−1, −4), (4, 1) (−1, 4), (−4, −1), (−4,
2, 3 → 3, 5
3, 3 → 4, 5

Mid Point Ellipse Algo

(4 way symmetry)



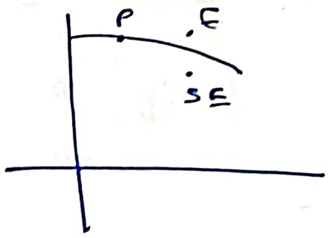
$\frac{dy}{dx} = -1$
(slope of tangent = -1)

RI → almost a horizontal line
RII → almost a verticle line

$$F(x,y) = b^2x^2 + a^2y^2 - a^2b^2$$

RI → slope of tangent > -1
RII → slope of tangent < -1

RI



choices → E & SE

RII

choices → S & SE

RI

$P = x_p, y_p$

$E = x_p + 1, y_p$

$SE = x_p + 1, y_p - 1$

$m = x_p + 1, y_p - \frac{1}{2}$

$d = F(m) = F(x_p + 1, y_p - \frac{1}{2})$

$d_{old} = b^2(x_p+1)^2 + a^2(y_p - \frac{1}{2})^2 - a^2b^2$

If $d < 0 \rightarrow E$

$d \geq 0 \rightarrow SE$

East $\rightarrow$ New $P = x_p + 1, y_p$

New $E = x_p + 2, y_p$

New $SE = x_p + 2, y_p - 1$

New $m = x_p + 2, y_p - \frac{1}{2}$

$d_{new} = F(\text{New } m)$

$= F(x_p + 2, y_p - \frac{1}{2})$

$= b^2(x_p+2)^2 + a^2(y_p - \frac{1}{2})^2 - a^2b^2$

$\Delta d_E = d_{new} - d_{old}$

$\Delta d_E = b^2(2x_p + 3)$

<u>South-East</u> $\rightarrow$ New $P = x_p + 1,\ y_p - 1$

New $E = x_p + 2,\ y_p - 1$

New $SE = x_p + 2,\ y_p - 2$

New $m = x_p + 2,\ y_p - \frac{3}{2}$

$d = F(\text{new } m)$

$= F(x_p + 2,\ y_p - \frac{3}{2})$

$d_{new} = b^2(x_p+2)^2 + a^2(y_p - \frac{3}{2})^2 - a^2b^2$

$\Delta d_{SE} = d_{new} - d_{old}$

$\Delta d_{SE} = b^2(2x_p + 3) + a^2(2 - 2y_p)$

<u>R II</u> I stay in region I till

$a^2(y_p - \frac{1}{2}) > b^2(x_p + 1)$ is true.

$P = x_p,\ y_p$

$S = \cancel{x_p}\ x_p,\ y_p - 1$

$SE = x_p + 1,\ y_p - 1$

$m = x_p + \frac{1}{2},\ y_p - 1$

$d = F(m) = F(x_p + \frac{1}{2},\ y_p - 1)$

$= b^2(x_p + \frac{1}{2})^2 + a^2(y_p - 1)^2 - a^2b^2$

$d < 0 \Rightarrow SE$

$d > 0 \Rightarrow S$

($d=0 \rightarrow$ see from the book)

Stop RII when we reach $(a, 0)$

SE

New $P = x_p + 1, y_p - 1$

New $S = x_p + 1, y_p - 2$

New $SE = x_p + 2, y_p - 2$

New $m = x_p + \frac{3}{2}, y_p - 2$

$d_{new} = F(\text{new } m)$

$= F(x_p + \frac{3}{2}, y_p - 2)$

$= b^2(x_p + \frac{3}{2})^2 + a^2(y_p - 2)^2 - a^2b^2$

$\Delta d_{SE} = d_{new} - d_{old}$

$= b^2(2x_p + 2) + a^2(-2y_p + 3)$

South

New $P = x_p, y_p - 1$

New $S = x_p, y_p - 2$

New $SE = x_p + 1, y_p - 2$

New $m = x_p + \frac{1}{2}, y_p - 2$

$d_{new} = F(\text{new } m)$

$= F(x_p + \frac{1}{2}, y_p - 2)$

$= b^2(x_p + \frac{1}{2})^2 + a^2(y_p - 2)^2 - a^2b^2$

$\Delta d_S = d_{new} - d_{old}$

$= a^2(3 - 2y_p)$

Initial d
(will be in RI)

$P_{ini} = (0, b)$

$m_{ini} = (1, b - \frac{1}{2})$

$d_{ini} = F(m_{ini})$

$= b^2(1)^2 + a^2(b - \frac{1}{2})^2 - a^2b^2$

$= b^2 + \cancel{a^2b^2} + \frac{a^2}{4} - a^2b - \cancel{a^2b^2}$

$d_{ini} = b^2 + \frac{a^2}{4} - a^2b$

Q Generate points of the ellipse whose eq^n is

$$\frac{x^2}{400} + \frac{y^2}{100} = 1$$

$a = 20$

$b = 10$

$d_{ini} = b^2 + \frac{a^2}{4} - a^2 b$

$= 100 + 100 - 4000$

$= \cancel{4000} - 3800 \quad < 0 \Rightarrow E$

Region I

x	y	d
0	10	-3800
1	10	$-3800 + b^2(2x_p + 3)$
		$= -3500 < 0 \Rightarrow E$
2	10	$-3500 + \Delta d_E$
		$= -3000 < 0 \Rightarrow E$
3	10	$-2300 \Rightarrow E$

continue till $a^2\left(y_p - \frac{1}{2}\right) > b^2(x_p + 1)$

$$400\left(y_p - \frac{1}{2}\right) > 100(x_p + 1)$$

$$4y_p - x_p > 3$$

x	y	d
4	10	-1400 < 0 => E
5	10	-300 < 0 => E
6	10	1000 > 0 => SE
7	9	-4700 < 0 => E
8	9	-3000 < 0 => E
9	9	-1100 < 0 => E
10	9	1000 > 0 => SE
11	8	-3100 < 0 => E
12	8	-600 < 0 => E
13	8	2100 > 0 => SE
14	7	-600 < 0 => E
15	7	2500 > 0 => SE
16	6	1000 > 0 => SE
17	5	

last point of region I
=> first point of region II

Region II

$m = 17 + \frac{1}{2}, 5 - 1$

$= \frac{35}{2}, 4$

d_{ini} for R II $= F\left(\frac{35}{2}, 4\right)$

$= b^2\left(\frac{35}{2}\right)^2 + 4^2(a^2) - a^2b^2$

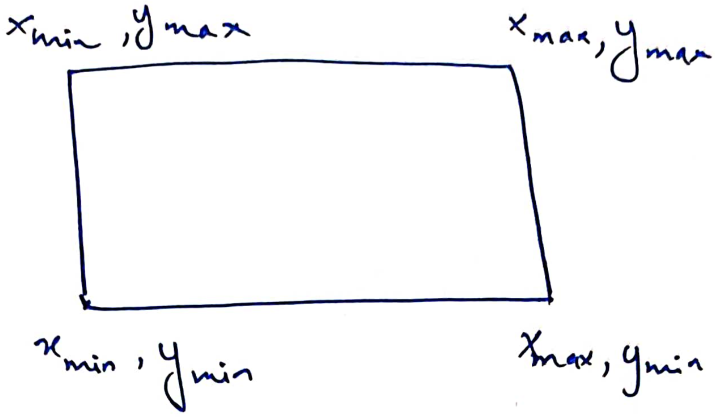
$= 100 \times \frac{1225}{4} + 6400 - 40000$

$= -2927 < 0 \Rightarrow \underline{SE}$

x	y	d
35/2	4	$-2927 < 0 \Rightarrow SE$
18	4	$-2175 < 0 \Rightarrow SE$
19	3	$-375 < 0 \Rightarrow SE$
20	2	$2425 > 0 \Rightarrow S$
20	1	$2024 > 0 \Rightarrow S$
20	0	

Replicate all the points using 4 way symmetry

Filling Rectangles



Rectangle is made up of 4 lines with eq^n's

$x = x_{min}$

$y = y_{min}$

$x = x_{max}$

$y = y_{max}$

Function to illuminate a pixel (present in header file graphics) → putpixel (x, y, color)

arguments

uppercase or ASCII

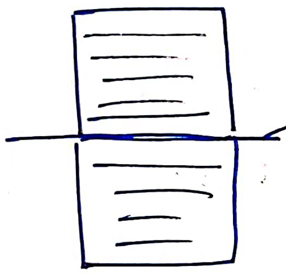
→ putpixel (x, y) → monochromatic system

→ In monochromatic system if you switch on a pixel twice using putpixel function, it will go to switch of state.

Algo →

```
for(int x = xmin ; x ≤ xmax ; x++)
    for(int y = ymin ; y ≤ ymax ; y++)
        putpixel(x, y, color);
```

Shared Edges



→ not illuminated in monochromatic system

Rules for Boundary Pixel → pixels on the bottom & left will be drawn. Pixels on top & right will not be drawn

(Follow this rule even if there is one rectangle)

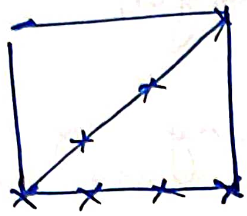
Updated algo →

$x < x_{max}$

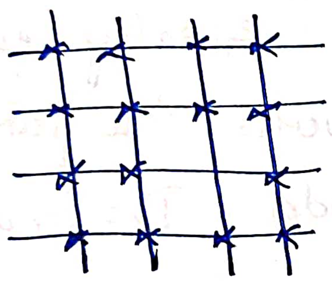
$y < y_{max}$

Varying Intensity

Total intensity of a line is a function of slope of the line



Let × be light bulbs.
Intensity of diagonal < Intensity of side
If equal no. of bulbs are placed on both



Diagonal might not appear as bright as the side.

⇒ Intensity of the pixel should be a function of the slope of the line.

Filling Polygon

It is a 3 step procedure

1. Find the intersection of current scan line ($y=c$) w/ all edges of the polygon
2. Sort the intersections by increasing x coordinate
3. Fill the pixels using odd-parity rule (Imp)

Odd parity rule states parity is initially even; every intersection encountered inverts the parity. We draw when the parity is odd. Do not draw when the parity is even

Case I

Intersection w/ fractional coordinates

Problem - which side to round? (ceil or floor)

Sol^n → stay inside the figure

(left) If we are inside the figure, we take floor

(right) If we are outside the figure, we take ceil case II

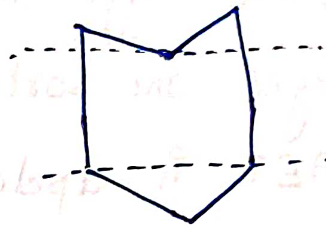
Intersection with exact integral coordinates

Problem - shared edges

Sol^n → left pixel is drawn, right is not drawn

Case III

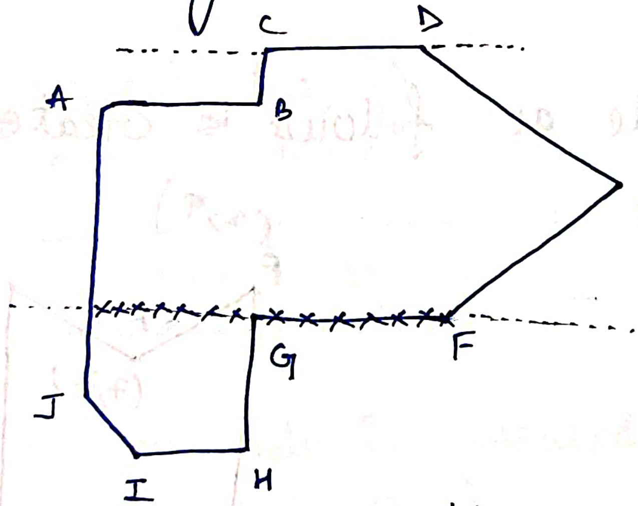
Shared vertices



Sol^n → y_{min} of every edge is counted in parity inversion, y_{max} is not counted

Case IV

Horizontal edges



C → BC (max) no inversion

C → CD (horizontal) won't take part in parity inversion

⇒ CD is not illuminated (say $y = 10$)

However for $y = 9$, the line within the polygon is illuminated

For GF, we are getting the desired result

Thus we are by default following the rule of drawing the bottom & not the top.

Data structures that will be used for scan fill algo (filling of a polygon)

- active edge ~~atge~~ table (AET)
- global table (edge table (ET))

Active edge table →

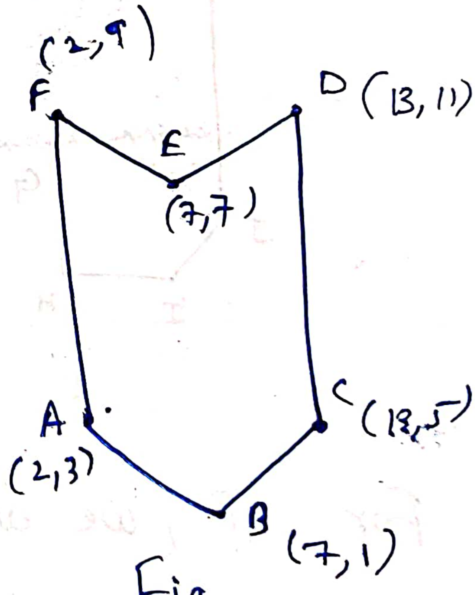
1. Edges are sorted on x values
2. AET is updated for every scan line (y=c)

Global edge table →

1. It contains all edges sorted by y_{min}
2. Global table is ~~a buct~~ maintained as a bucket

For every edge, a node as follows is created

y_{max}	x_{min}	$1/m$



Fig

max min

AB → (2,3) (7,1)

* Imp → based on y value, decide the min & max value.

AB

3	7	-5/2

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{1-3}{7-2}$

(When y is incremented by 1 x is incremented by 1/m

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

$x_{i+1} = x_i + \frac{1}{m}$)

(DDA line algo)

BC

5	7	3/2

(7,1) (13,5)
min max

CD (13,5) (13,11)
min max

11	13	0

(Note → no node is created for horizontal line as $1/m = \infty$)

DE (13,11) (7,7)
max min

11	7	3/2

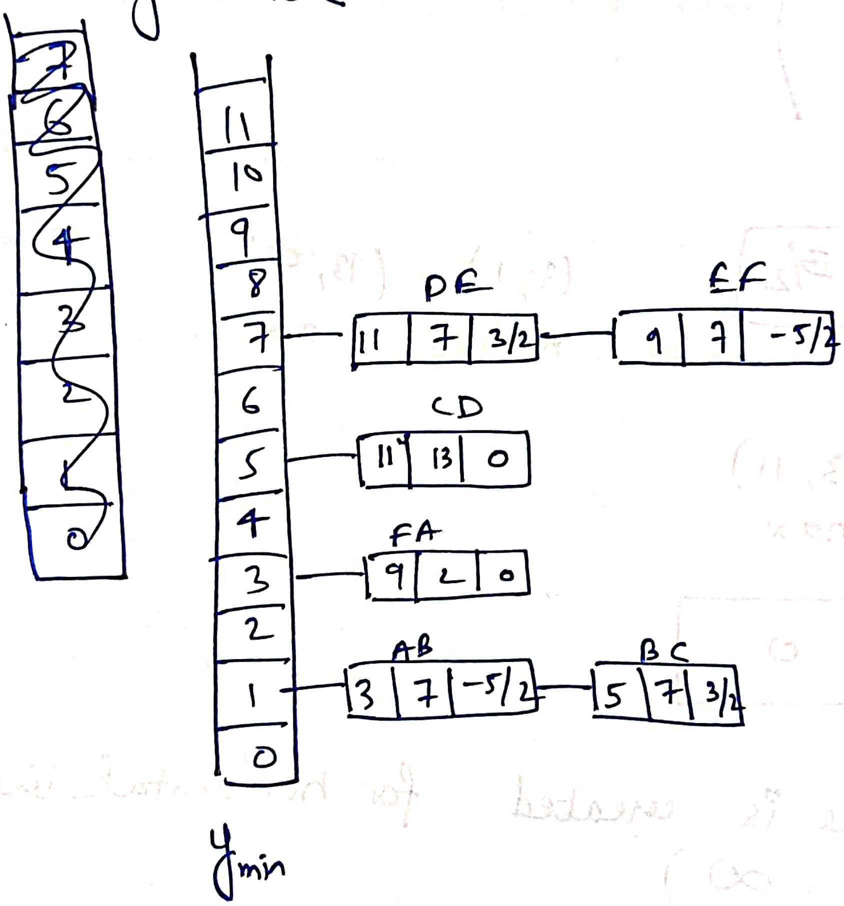
EF (7,7) min (2,9) max

9	7	-5/2

FA (2,9) max (2,3) min

9	2	0

Global edge table



GET is static, it won't change through the algo while AET will change for every y.

Edge Table

Start w/ $y=0$

Copy the nodes at bucket 0 from the GET.

If no nodes ⇒ NULL ⇒ no pixel illuminated.

$y=1$

| 3 | 7 | -5/2 | → | 5 | 7 | 3/2 |

sort them on the basis of x value.

span to be illuminated

(7, 1) ⟷ (7, 1)

x of left node

(∵ y = 1)

x of right node

Copy the nodes from the previous scan line to the next, while incrementing x by 1/m (∵ y = y+1)

$y=2$

| 3 | 9/2 | -5/2 | ⟷ | 5 | 17/2 | 3/2 | (Sort)

no node from GET.

span →

(5, 2) ⟷ (8, 2)

(take ceil for left & floor for right fractional value)

(this span illuminates (5,2), (6,2), (7,2), (8,2))

$y = 3$

$\boxed{9 \mid 2 \mid 0}$

we will not copy the node from one scan line to another if we have reached y_{max}

thus, we won't copy $\boxed{3 \mid 9/2 \mid -5/2}$

$\boxed{9 \mid 2 \mid 0} \longrightarrow \boxed{5 \mid 10 \mid 3/2}$

sort on x

(already sorted)

span $\rightarrow$ $(2, 3) \longleftrightarrow (10, 3)$

$y = 4$

$\boxed{9/2 \mid 0} \longrightarrow \boxed{5 \mid 23/2 \mid 3/2}$

$(2, 4) \longleftrightarrow (11, 4)$

$y = 5$

$\boxed{9 / 2 / 0} \longrightarrow \boxed{11 \mid 13 \mid 0}$

$(2, 5) \longleftrightarrow (13, 5)$

$y = 6$

| 9 | 2 | 0 | → | 11 | 13 | 0 |

$(2,6) \longleftrightarrow (13,6)$

$y = 7$

| 11 | 7 | 3/2 | → | 9 | 7 | -5/2 | → | 9 | 2 | 0 | → | 11 | 13 | 0 |

Note → AET will always have even no. of nodes

Sort →

| 9 | 2 | 0 | → | 11 | 7 | 3/2 | → | 9 | 7 | -5/2 | → | 11 | 13 | 0 |

$(2,7) \longleftrightarrow (7,7)$ $\quad$ $(7,7) \longleftrightarrow (13,7)$

[We illuminate pixels b/w pairs].
i.e. b/w (pair 1, pair 2) & (pair 3, pair 4)

$y = 8$

| 9 | 2 | 0 | → | 11 | 17/2 | 3/2 | → | 9 | 9/2 | -5/2 | → | 11 | 13 | 0 |

Sort

| 9 | 2 | 0 | → | 9 | 9/2 | -5/2 | → | 11 | 17/2 | 3/2 | → | 11 | 13 | 0 |

$(2,8) \longleftrightarrow (5,8)$ $\quad$ $(8,8) \longleftrightarrow (13,8)$

$\underline{y = 9}$

$\boxed{9 \mid 2 \mid 0} \longrightarrow \boxed{9 \mid 2 \mid -5/2}$ ($\because$ $y = y_{max}$ discard these)

$\boxed{11 \mid 10 \mid 3/2} \longrightarrow \boxed{11 \mid 13 \mid 0}$

scan $\longrightarrow$

$(10, 9) \longleftrightarrow (13, 9)$

$\underline{y = 10}$

$\boxed{11 \mid 23/2 \mid 3/2} \longrightarrow \boxed{11 \mid 13 \mid 0}$

scan

$(12, 10) \longleftrightarrow (13, 10)$

$y = 11$

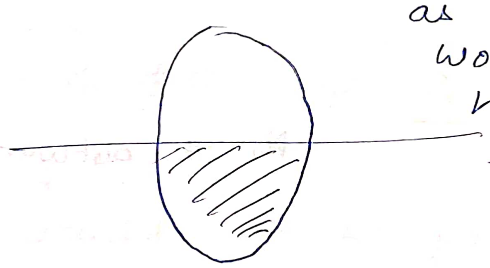
<u>Null</u>. ($\because$ $y = y_{max}$, node is not copied This is because we do not illuminate the top)

Edge Coherence Property

Many points intersected by scan line $y = i$ are also intersected by scan line $y = i+1$.

This edge coherence property is used for ellipse filling.

→ Ellipse has a 4-way symmetry however, we'll be using 2-way symmetry.



as the edge coherence property works only for the lower half of the ellipse & are then replicated.

Cyrus Beck Line Clipping Algo

Uses parametric form of eqn

$P(t) = P_0 + (P_1 - P_0)t$

$P(0) = P_0$

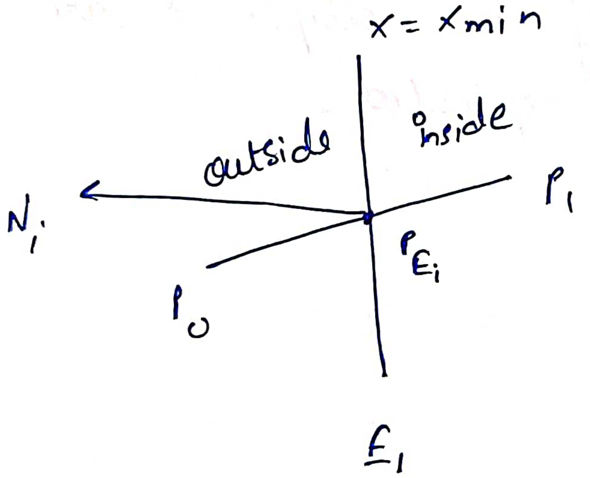
$P(1) = P_1$

$0 \leq t \leq 1$

$P(0.5)$

↓

mid-point of line



$N_i \rightarrow$ outward normal

$N_i \cdot [P(t) - P_{E_i}] = 0 \Rightarrow P_{E_i}$ is on the edge

$< 0 \rightarrow$ inside the plane

$> 0 \rightarrow$ outside the plane

To find P_{E_i}

$N_i \cdot [P(t) - P_{E_i}] = 0$

$$N_i \cdot [P_0 + (P_1 - P_0)t - P_{E_i}] = 0$$

$$N_i \cdot (P_0 - P_{E_i}) + N_i (P_1 - P_0)t = 0$$

$$t = \frac{N_i \cdot (P_0 - P_{E_i})}{-N_i \cdot D}$$ where $D = P_1 - P_0$

For a value of t, denominator $\neq 0$

$\Rightarrow D \neq 0$

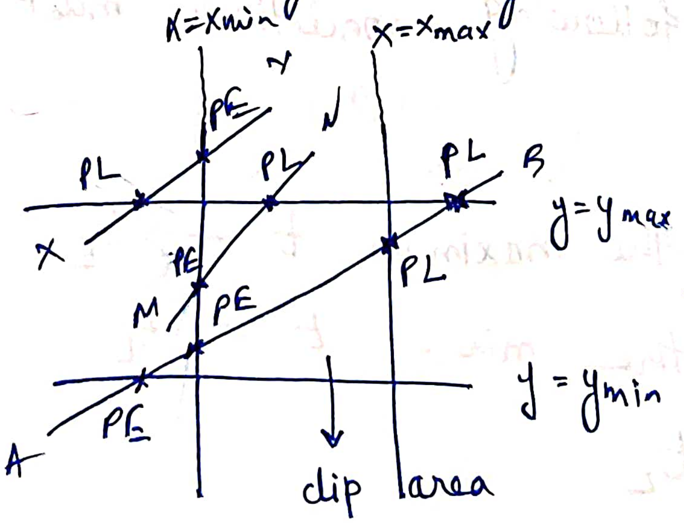
$\Rightarrow P_1 - P_0 \neq 0 \Rightarrow \underline{P_0 \neq P_1}$

$N_i \cdot D \neq 0$

$\Rightarrow P_0 \cdot P_1$ should not be parallel to E_i

---- x ----

PE → Potentially entering
PL → Potentially leaving



A line can have max 4 intersections with the clip edges.

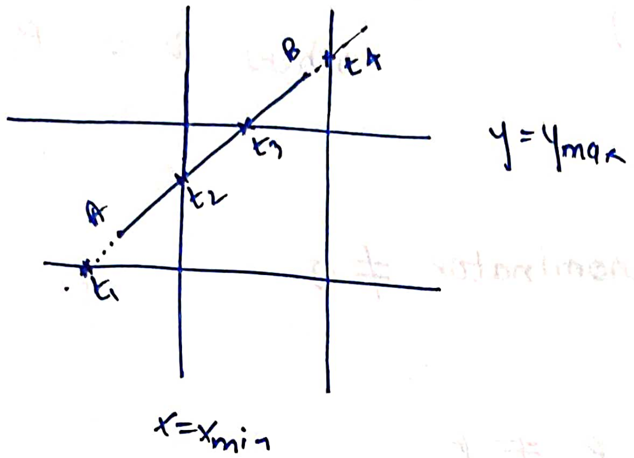
Move from left to right

To identify PE & PL look at the clip edge that has been intersected. Identify the areas on either side of the clip edge as outside & inside (wrt clip area)

→ A line can have multiple PE & PLs (eg. AB)

→ Order of PE & PL is not fixed

→ A line can have max 4 intersections



$0 \le t_2, t_3 \le 1$

$0 \nleq t_1, t_4 \nleq 1$

→ If $N_i \cdot D < 0 \Rightarrow P_E$

$N_i \cdot D > 0 \Rightarrow P_L$

To clip the line, the following conditions must be satisfied:

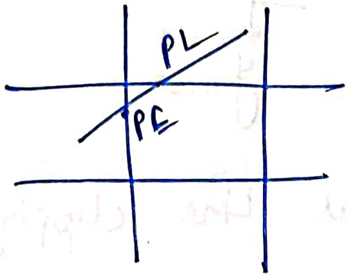
→ Of all PEs, find the maximum $t \Rightarrow t_E$

→ PLs, find the min. $t \Rightarrow t_L$

→ At all times $t_E < t_L$

→ If $t_L < t_E$, we reject the line

For a line that enters the clip area, the first edge will always be PE & the other PL



$$t = \frac{N_i \cdot (P_0 - P_{E_i})}{-N_i \cdot D}$$

$D = P_1 - P_0$

Clip edge	Normal N_i	PE_i	$P_0 - P_{E_i}$
$x = x_{min}$	$(-1, 0)$	(x_{min}, y)	$(x_0 - x_{min}, y_0 - y)$
$x = x_{max}$	$(1, 0)$	(x_{max}, y)	$(x_0 - x_{max}, y_0 - y)$
$y = y_{min}$	$(0, -1)$	(x, y_{min})	$(x_0 - x, y_0 - y_{min})$
$y = y_{max}$	$(0, 1)$	(x, y_{max})	$(x_0 - x, y_0 - y_{max})$

Q Give a line segment $P(2,10)$, $Q(20,40)$

& the clip are $(10,10)$ $(20,20)$

x_{min} y_{min} x_{max} y_{max}

Clip the line PQ using Cyrus Line clipping algo

$P_0 = P = (2,10)$

$P_1 = Q = (20,40)$

$x = x_{min} \Rightarrow x = 10 \Rightarrow t = \frac{x_0 - x_{min}}{x_1 - x_0} = \frac{-2(2-10)}{20-2}$

$\frac{8}{18} = \frac{4}{9}$ ✓ $t(PE)$

Negation of denominator $= -18 < 0 \Rightarrow PE$

$x = x_{max} \Rightarrow x = 20 \Rightarrow t = \frac{x_0 - x_{max}}{-(x_1 - x_0)} = \frac{2-20}{-(20-2)}$

$= \frac{-18}{-18} = 1$ ✓

$18 > 0 \Rightarrow PL$

$y = y_{min} \Rightarrow y = 10 \Rightarrow t = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$

$= \frac{-(10-10)}{40-10} = 0$ ✓

$-30 < 0 \Rightarrow PE$

$$y = y_{max} \Rightarrow y = 20 \Rightarrow t = \frac{y_0 - y_{max}}{-(y_1 - y_0)}$$

$$= \frac{10 - 20}{-(40 - 10)} = \frac{-10}{-30} = \frac{1}{3} \checkmark$$

$30 > 0 \Rightarrow PL$

Discard t if it's not b/w $[0, 1]$

$$t_E = \max\left(\frac{4}{9}, 0\right) = \frac{4}{9}$$

$$t_L = \min\left(1, \frac{1}{3}\right) = \frac{1}{3}$$

Calculate pt. of intersection for the two values of t.

$$P\left(\frac{4}{9}\right) = 2 + (20 - 2)\frac{4}{9} = 10 \quad (x \text{ value})$$

&

$$10 + (40 - 10)\frac{4}{9} = \frac{70}{3} \quad (y \text{ value})$$

$$pt \longrightarrow \left(10, \frac{70}{3}\right)$$

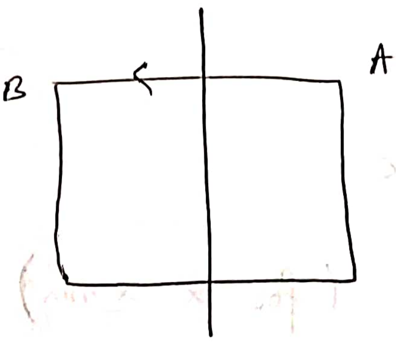
$$P\left(\frac{1}{3}\right) = 2 + (20 - 2)\frac{1}{3} = 8$$

$$= 10 + (40 - 10)\frac{1}{3} = 20$$

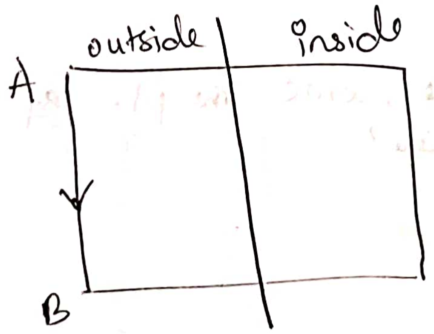
$$pt \longrightarrow (8, 20)$$

Q Given a line AB A(10,10), B(100,100)

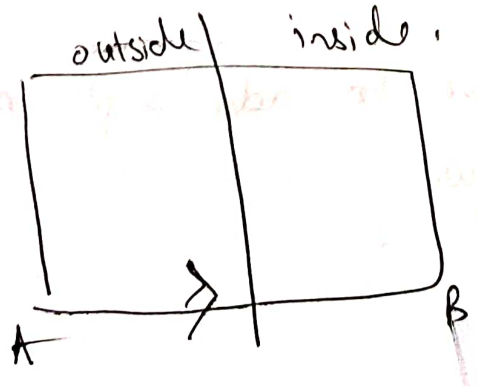
Clip the line against the clip area (50,50) (120,120) case 1



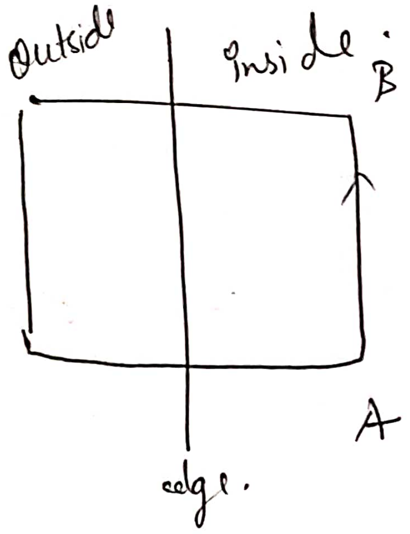
case 2



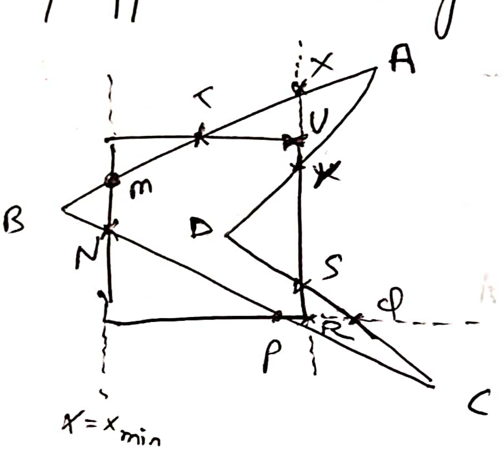
case 3



case 4



Clip the polygon ABCD against the given clip area



In vertex.	Out Vertex	
A	A (∵ inside)	
B	m N	(if outside, write the pt. of clipping for that line)
C	C	
D	D	
A	A	

(for $x = x_{min}$)

(close the polygon)

If we remove B, we'll have to add 2 pts m & N to make it a closed figure

Move anticlockwise & now consider the line ($y = y_{min}$)

→ Previous outvertex array becomes the new Invertex array for the next clip line. ($y = y_{min}$)

$y = y_{min}$.

Invertex	Outvertex
A	A
M	M
N	N
C	P
~~D~~ A	Q
	D
	A

$x = x_{max}$

Invertex.	Outvertex
	X
A	~~B~~
M	M
N	N
P	P
Q	R
D	S
A	D
	V
	X

(only X is required as we have to move anticlockwise & only X is required to close the figure)

should be a closed figure

$y = y_{max}$

In Vertex	Out Vertex.
X	T
M	M
N	N
P	P
R	R
S	S
D	D
Y	Y
X	U
	T

Draw the final clip polygon in exam! :/

Generating Character

2 methods :-

① <u>Method of splines</u>

Splines - higher order polynomials

$F(x, y) = \ldots$

Advantage - less storage space is needed

Disadvantage - difficult to implement

takes time in calculation

difficult to increase / decrease the size

② <u>Font Cache Method</u> / Bitmap Technique

1	1	1	1	0
1	0	0	0	1
1	1	1	1	0
1	0	0	0	1
1	1	1	1	1

→ Bitmap for B

Advantages → easy to implement

Less time wrt spline's method

easy to increase the size

easy to add faces (bold | italic | bold italic | normal

Disadvantage → larger storage

as you have to save the bitmap for every character

Jaggies / Zig-Zag / Staircase effect (Imp)

The jaggies is called aliasing ~~technique~~

Techiques to ~~add~~ remove / reduce aliasing are called anti-aliasing techniques.

3 techiques

① Increase the resolution

(no. of pixels ↑ ⇒ more the no. of pixels, better the resolution.. It reduces aliasing but doesn't remove it)

Improvement is at the cost of increasing memory & increasing scan converting time.

② ~~Unweighted Alias Sampling~~

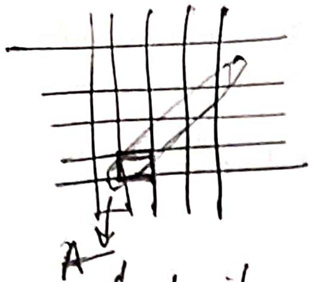
Unweighted Area Sampling.

3 properties :

① intensity of pixel intersected by a line edge ↓ as the distance b/w the line edge & pixel ↑

② primitive (polygon / line) cannot influence the intensity if the primitive doesn't intersect the pixel

③ Equal areas contribute equally, regardless of dist. from the centre.



Every box is a pixel in this & not every cross-section

And if 0 → white 10 → black, various shades of grey are possible.

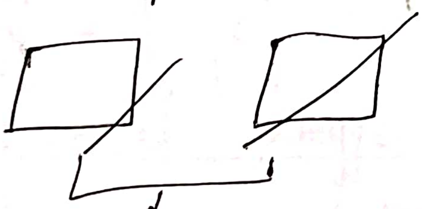
If say pixel A is to be illuminated & we only need to illuminate 10% of it, we'll assign it 1

③ Weighted Area Sampling

3 properties

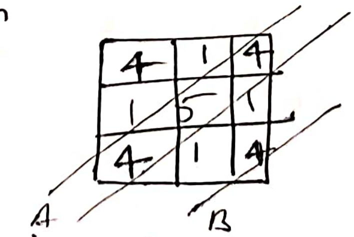
Properties ① & ② are same as before.

③ area nearer to the centre of the pixel has more effect on the color of the pixel than at a distance from the centre.



these 2 will have different ~~[illegible]~~ intensity.

weighing function



(just an example)

A → say 20% coverage with wt 13.

B → " " " " " 4

Z-Buffer Algo

```
void zbuffer (void)
{
    for (x=0 ; x <xmax ; x++)
     for( y=0 ; y < ymax ; y++ )
      {
          z[x][y] = 0 ;
          FB [x][y] = Background color ;
      }

      for (each polygon)
       for( each pixel (x,y) )
       {  pz = z value for (x,y)
          if (pz> z[x][y] )
          {   z[x][y] = pz
              FB[x][y] = current color;}
          }

       }
```

Equation of a polygon

$$Ax + By + Cz + D = 0$$

$$z = \frac{-Ax - By - D}{C}$$

if $x_{i+1} = x_i + 1$ [Assuming $y_{i+1} = y_i$]

$$z_i = \frac{-Ax_i - By_i - D}{C}$$

$$z_{i+1} = \frac{-Ax_i - A - By_i - D}{C}$$

$$= z_i - \frac{A}{C}$$

If $x_{i+1} = x_i$ $y_{i+1} = y_i + 1$

$$z_{i+1} = z_i - \frac{B}{C}$$

Advantages of zbuffer algo.

1. used to render (draw any object of any shape)
2. zbuffer can perform radix sort on x & y for the value of z, so only 1 comparison will be needed.
3. easy to implement

Disadv.

1. requires lot of space (zbuffer array, frame buffer array, size of the screen)
2. might face aliasing

* z buffer algo for transparent objects is called a buffer algo

Q

1	1	1	0	0
1	2	5	5	0
1	0	3	0	1
0	0	0	1	1
0	0	0	0	0

+

0	2	0	1	3
4				
0				
0				
0				

= ?

take max value for each box

1	2	1	1	3
4	2	5	5	0
1	0	3	0	1
0	0	0	1	1
0	0	0	0	0

Depth Sort Algo

3 steps:

1. sort on z value
2. resolve any ambiguity or split
3. scan convert (draw each polygon starting from the smallest z value i.e. farthest view pt)

* If we ignore the second step of splitting, the algo is called painter's algo.

To check if a split is required

We have a polygon P. We want to check the polygon P against every other polygon Q. If we are able to prove that P doesn't obscure Q neither does Q obscure P, we can draw P.

5 tests: (increasing order of complexity)
If any test is true, do not check others & move to the next value of Q.
If all the tests fail ⇒ P cannot be drawn before Q.

Test 1 : $\underbrace{x \text{ extents}}_{(\Rightarrow x_{min},\, x_{max})}$ do not overlap.

Test 2 : Y extents do not overlap.

Test 3 : P is completely behind Q surface.

Test 4 : P is completely infront of Q.

Test 5 : Do the projections of both polygons on the x-y plane not overlap.

If all the tests fail, perform split of the polygon P

C.G.

To prove - Parallel lines remain || even after transformation i.e. a ||gm transforms into another ||gm after transformation.

AB || CD
$\downarrow$ m $\quad$ | m

$A = [x_1 \quad y_1]$
$B = [x_2 \quad y_2]$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{bmatrix} A^* \\ B^* \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} [T]$$

$$= \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix}$$

$$\text{slope} = m^* = \frac{bx_2 + dy_2 - bx_1 - dy_1}{ax_2 + cy_2 - ax_1 - cy_1}$$

$$= \frac{d(y_2 - y_1) + b(x_2 - x_1)}{c(y_2 - y_1) + a(x_2 - x_1)}$$

$\div$ the n^r & d^r by $x_2 - x_1$

$$m^* = \frac{d\,\frac{y_2 - y_1}{x_2 - x_1} + b}{c\,\frac{y_2 - y_1}{x_2 - x_1} + a}$$

$$m^* = \frac{dm + b}{cm + a}$$

slope of $\perp$ $A^* B^*$

To prove: Intersecting lines remain intersecting even after transformation.

(Do it yourself refer Book)

Rotation

matrix →

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

This rotates the point counter clockwise about the origin

$\det[T] = 1$

determinent of $T = 1$

Note → Transformations w/ det. identically equal to +1 are called pure rotations

To rotate by $-\theta$

$$T_{-\theta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = T^{-1}$$

for this matrix → $T^{-1} = T^{T}$

The inverse of any pure rot^n matrix (det. = 1) is equal to its transpose

Q Given a ΔABC $A(3,-1)$, $B(4,1)$, $C(2,1)$
Rotate the Δ 90° about the origin in counter clockwise sense.

Matrix for Δ [write points rowwise]

$$\begin{array}{c} A \\ B \\ C \end{array} \begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix}$$

However if eqⁿs of lines are given, write the coefficients of lines columnwise & then multiply w/ T

$$\text{eg.} \quad \begin{array}{cc} \text{line 1} & \text{line 2} \\ \end{array}$$
$$\begin{bmatrix} - & - \\ - & - \\ - & - \end{bmatrix}$$

Ans.

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \overset{0}{\cancel{\cos 90}} & \overset{1}{\cancel{\sin 90}} \\ \underset{-1}{-\cancel{\sin 90}} & \underset{0}{\cancel{\cos 90}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix} \begin{array}{c} A^* \\ B^* \\ C^* \end{array}$$

Reflection through the lines

x-axis $(y=0)$

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

y-axis $(x=0)$

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

$x = y$

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

$x = -y$

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

Note → If the determinant of the matrix is $= -1$
It is said to be <u>pure reflection</u>.

Q Given a ΔABC $A(4,1)$ $B(5,2)$ $C(4,3)$.
Reflect it about x-axis, then reflect it about the line $x=-y$. Compare it w/ a rotation of 270°

(by default: rotation is counter clockwise, about the origin)

$$\Delta_{ABC} = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$T_1 \times T_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

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

$$\Delta_{ABC}^{*} = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix}$$

$$T_{R_{270}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\Delta^{**}_{ABC} = \left[\Delta^{*}_{ABC}\right]\left[T_{R_{270}}\right]$$

$$= \cancel{\begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}}$$

$$= \cancel{\begin{bmatrix} -4 & -1 \\ -5 & \end{bmatrix}} \quad \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix}$$

Note → $[T_1][T_2] \neq [T_2][T_1]$

i.e. order of transformation is important

⇒ matrix multiplication is not commutative :)

Q Given a ΔABC $A(2,2)$ $B(4,2)$ $C(4,4)$

Rotate 90° about the origin & then reflect through the line $y = -x$

$$\begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\Delta^*_{ABC} = \begin{bmatrix} -2 & 2 \\ -2 & 4 \\ -4 & 4 \end{bmatrix}$$

Reflection through $y = -x$

$$\begin{bmatrix} -2 & 2 \\ -2 & 4 \\ -4 & 4 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 2 \\ -4 & 2 \\ -4 & 4 \end{bmatrix}$$

Transformation of a unit square (each side = 1)

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$X^* = [X][T]$$

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

$$= \begin{bmatrix} 0 & 0 \\ c & d \\ a & b \\ a+c & b+d \end{bmatrix}$$

Note → origin is invariant ~~by~~
b, c have caused shearing
a & d are scaling factors

Area of transformed figure = area of original fig × det [T]

⇓

area scaling ~~area~~ property

Q ΔABC A(1,0) B(0,1) C(−1,0)
and the transformation matrix

$T = \begin{bmatrix} 3 & 2 \\ -1 & 2 \end{bmatrix}$. Find the area of tranformed Δ

$$= \frac{1}{2}\left[x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)\right] \times \text{Det}[T]$$

$$= \frac{1}{2}(1(1) + 0(0) + (-1)(-1)] \times \text{Det}[T]$$

$$= \frac{1}{2}(2) \times \text{Det}[T]$$

$$= 1 \times 8$$

$$= \underline{\underline{8}}$$