Ch-3

MARCH
WEDNESDAY
085 281 DAYS

25

DDA Line Algo (Digital Differential Analyzer)

* Solves D.E by numerical methods
* Traces out successive (x,y) by simul. inc x by 1 & y by m

(x_i, y_i)

x_{i+1} y_{i+m}

Qn Give all pts of a line whose co-ordinate are (5,8), (9,11)

Ans $m = 3/4 \simeq 0.75$

x_i	y_i	
5	8	We take ceil value
6	8.75	≃9
7	8.75+.75 = 9.25	≃10
8	10.25	≃10

[2 Mark]

Disadv: ① Works for slope b/w -1 & 1

② Round of error at every step.

4 special cases:

① Horizontal L : $y=c$ (x_1,c) (x_2,c) $m=0$

$x_i = x_{i+1}$ $y=c$ Till reach x_2

② Vertical Line: $x=c$ (c,y_1) (c,y_2) $m=\infty$

$y_i = y_{i+1}$ Till $x=c$

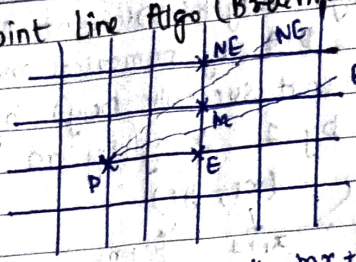
③ $m=1$ $\Delta y \simeq \Delta x$ $y_i = y_{i+1}$ | $x_i = x_{i+1}$

(1,1) (10,10) so [(2,2), (3,3) ...]

④ $m>1$ $\Delta y > \Delta x$

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

Mid-Point Line Algo (Bresenham's Line Draw Al.)



$P(x_p, y_p)$

$E(x_{p+1}, y_p)$

$NE(x_{p+1}, y_{p+1})$

$M(x_{p+1}, y_{p+1/2})$

$y = mx + B$

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

$F(x,y) = ax + by + C$

$a = dy \quad b = -dx$

$c = dx \cdot B$

$dy = y_2 - y_1 \quad dx = x_2 - x_1$

$d = F(M) = a(x_{p+1}) + b(y_{p+1/2}) + c$

$d > 0$ M·P below NE

$d < 0$ M·P above E

$d = 0$ M·P anywhere (we take E)

Choose E

$F = d_{new} = a(x_{p+2}) + b(y_{p+1/2}) + c$

$\Delta d_E = dy$

Choose NE :

$d_{new} = a(x_{p+2}) + b(y_{p+3/2}) + c$

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

$$\boxed{d_{ini} = dy - \frac{dx}{2}}$$

Qu Gen the points of line where end pt are. (5,8), (9,11)

Ans $dy = 3 \quad dx = 4$

$d_{ini} = 3 - \frac{4}{2} = 1$

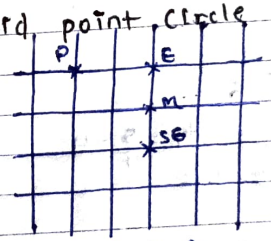
$\Delta d_E = dy$

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

x	y	d	
5	8	1	(NE)
6	9	1 - 1 = 0	(E)
7	9	0 + 3 = 3	(NE)
8	10	3 - 1 = 2	NE
9	11	2 - 1 = 1	NE

Ans (6,9), (7,9), (8,10)

Mid point Circle Algo



$P(x_p, y_p)$

$E(x_{p+1}, y_p)$

$SE(x_{p+1}, y_{p-1})$

$M(x_{p+1}, y_{p-1/2})$

$d = F(m)$

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

$d = (x_{p+1})^2 + (y_p - \frac{1}{2})^2 - R^2$

$d > 0$ SE

$d < 0$ E

$d = 0$ SE

Choose E

$$\boxed{d_E = 2x_p + 3}$$

Choose SE

$$\boxed{d_{SE} = 2x_p - 2y_p + 5}$$

$$\boxed{d_{ini} = \frac{5}{4} - R}$$

28 MARCH SATURDAY DAYS 088-278

Qn Gen all pts of circle where $r = 6$

$d_{init} = \frac{5}{4} - 6$

x	y	d	
0	6	$-19/4$	(E) → $2x_p + 3$
1	6	$-\frac{19}{4} + 3 = -7/4$	(E)
2	6	$-\frac{7}{4} + 5 = 13/4$	(SE)
3	5	$\frac{13}{4} + 4 - 12 + 5 = 1/4$	(SE)
4	4		

→ $(x,y), (-x,-y), (-x,y), (x,-y)$

$(y,x), (y,-x), (-y,x), (-y,-x)$

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

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

And so one

Qn Gen all pts for $(x-1)^2 + (y-2)^2 = 16$

$(1,2) \rightarrow (0,0)$ $\quad d_{in} = \frac{5}{4} - 4$

x	y	d	
0	4	$-11/4$	(E)
1	4	$-11/4 + 3$ $1/3$	(SE)
2	3	$\frac{1}{3} - 1$ $-2/3$	(E)
3	3		

29 SUNDAY

MARCH MONDAY 30 090-275 DAYS

Mid pt Ellipse Algo :-

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

(R1)

$d = F(M) = b^2(x_p+1)^2 + a^2(y_p - 1/2)^2 - a^2b^2$

$d \geq 0$ SE

$d < 0$ E

Choose E

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

Choose SE

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

Stay in R1 Till

$a^2(y_p - \frac{1}{2}) > b^2(x_p + 1)$ ★ [2 Marks]

(R2)

$d = F(m) = b^2(x_p + 1/2)^2 + a^2(y_p - 1)^2 - a^2b^2$

$d < 0$ SE

$d > 0$ S

Choose S

$\Delta d_S = a^2(3 - 2y_p)$

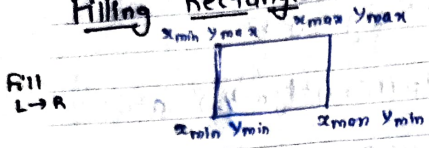
Choose SE

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

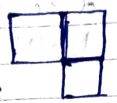
$\left[d_{init} = b^2 + \frac{a^2}{4} - a^2b \right]$

For Qun Refer to Rough ☺

Filling Rectangle



Problem: Shared Edges



Consider 2 rects. if we scan convert each rectangle in turn we'll write the pixel on shared edges twice which is undesirable.

☆ Soln: Rule for boundary Pixel
The pixels on the left & bottom drawn but pixel lie on top & right will not be drawn

CLIPPING LINES

Odd Parity: This rule state that parity is initially even every intersection encountered inverts the parity. We draw when parity is odd & do not draw when even.

① Intersection with fractional co-ordinate.
Prob: Which side to round
Soln If we r inside fig. take floor else ceiling

② Intersection with exact integral co-ordinates
Prob: Shared Edges
Soln: Left Pixel is drawn right is not

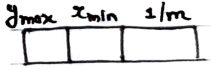
③ Shared Vertices
Soln: Ymin of every edge is counted in parity ie Ymax is not

④ Horizontal Edge

* Data stru. used for Scan fill Algo:-
1) Active Edge Table (AET)
2) Global Table (Edge Table) (ET)

AET :- ① Edges are sorted on x value
② AET is update for every scan line (y=c)

ET ① Contain all edges sorted by ymin
② Maintained as a bucket



Compare y value for max/min node

AB (2,3) max (7,1) min

3	7	-5/2

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

5	7	3/2

CD (13,5) (13,11)

11	13	0

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

11	7	3/2

EF (2,9) (7,7)

9	7	-5/2

AF (2,9) (2,3) min

9	2	0

No node created for horizontal line 1/m → ∞

02 APRIL THURSDAY DAYS 093-273

GET

ymin	
9	
8	11 | 7 | 3/2 → 9 | 7 | -5/2
7	
6	
5	11 | 13 | 0
4	
3	9 | 2 | 0
2	
1	3 | 7 | -5/2 → 5 | 7 | 3/2
0	

↗ Updated as scan line move down to poly

AET

y=0 No node for this

NULL: No pixel to be illuminated.

Sort by x-value

y=1 3 | 7 | -5/2 → 5 | 7 | 3/2

(7,1) → (7,1) illuminate all pixels.

y=2 3 | 9/2 | -5/2 → 5 | 17/2 | 3/2

(9/2, 2) → (17/2, 2)

(5,2) → (8,2) illuminate

y=3: 9 | 2 | 0 → 5 | 10 | 3/2

(2,3) → (10,3)

y=4: 9 | 2 | 0 → 5 | 23/2 | 3/2

(2,4), (11,4)

y=5: 9 | 2 | 0 → 11 | 13 | 0

(2,5) (13,5)

03 APRIL FRIDAY DAYS 094-272

y=6: 9 | 2 | 0 → 11 | 13 | 0

(2,6), (13,6)

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

(2,7) (7,7) (7,7) (13,7)

y=8: 9 | 2 | 0 → 9 | 9/2 | -3/2 → 11 | 17/2 | 3/2 → 11 | 13 | 0

(2,8) → (5,8) (8,8) (13,8)

y=9 11 | 10 | 3/2 → 11 | 13 | 0

(10,9) → (13,9)

y=10 11 | 23/2 | 3/2 → 11 | 13 | 0

(12,10) → (13,10)

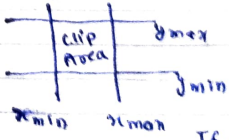
y=11 NULL

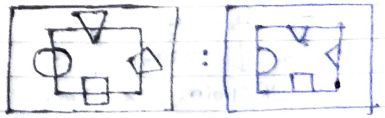
Edge Coherence Property:

Many points intersected by scan line (y=t) are also intersected by scan line (y=t+1). This ECP is used for Ellipse filling

Line Clipping

① Clipping End points





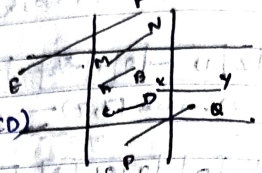
$x_{min} \le x \le x_{max}$

$y_{min} \le y \le y_{max}$

If any of ineqs does not hold point lie outside

② Clipping By solving Simultaneous Equations :-

(c1) Trivial Acceptance
If both ends pt lie inside clip Area (AB & CD)



(c2) 1 inside 1 outside (MN & XY)
We'll clip part which is outside

(c3) Both outside (EF & PQ)
We'll have to manually check & clip the line

Problem : This is a brute force Approach to check each line intersection

③ Cohen - Sutherland Line Clipping Algo

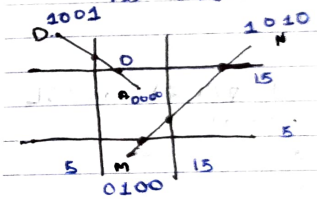
(S1) Check for TA
(S2) Check for Trivial Rejection
both end pts <xmin or <ymin
>xmax or >ymax
(S3) Perform clipping w/ outcode

05 SUNDAY

T B R L
y>ymax ↓ x>xmax →
y<ymin x<xmin

2	1001	1000	1010 ymax
0	0001	0000	0010
	0101	0100	0110 ymin
	xmin		xmax

Condition for TA : Both outcodes 0000 of end pts
" " TR : Take bit wise AND of outcode of both end pts if it is non-zero the line is TR



AD
0000 1001

① Not TA
② 0000
③ 1001
0000
Not TR

③ T B R L
1 0 0 1

AD have intersection with Top & left

A (12,13) D (3,17)
$(y-17) = \frac{-4}{9}(x-3)$
$4x+9y-165=0$
$y=15 \quad x=7.5$
O (7.5, 15)

M (7,3) N (20,20)
$17x-13y-80=0$
$y=5 \quad x=8.5$
(8.5, 05)

M 0100 N 1010

① Not TA
② Not TR
③ intsn. with bottom

N 1010 T & R

$17x - 13y - 80 = 0$

$y = 15 \quad x = 16.17$

$x = 15$

$y = 13.46$

(16.17, 15) (15, 13.46)

Clip line x

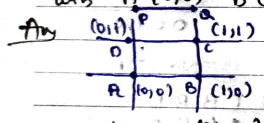
Disadv: Sometimes we have to form needless clipping.

2019

Q1 Cond. for TR of line P(0,5) Q(1,5) in CSL

with A(0,0) B(1,0) C(1,1) D(0,1)

Ans



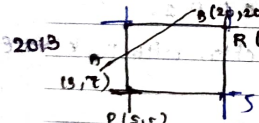
P (1000)

Q (1000)

AND 1000 (Non zero)

T R

2013



A B

0001 1000

① TA x

② TR x

③ A

0001 Intersection at L

AB ≡ $(y-7) = \frac{13}{17}(x-3)$

$17y - 119 = 13x - 39$

$13x - 17y + 80 = 0$

At $x = 5$ $y = 8.5$ (5, 8.5)

B: $y = 15 \quad x = 13.46$

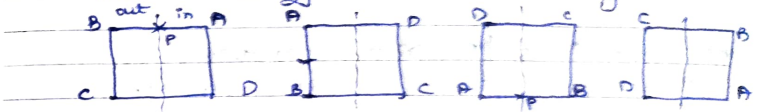
(13.46, 15)

Clipping Polygons :-

Sutherland-Hodgman Algo (Divide & Conq.)

There are 2 data struc. which are used

Inverter array & outverter array



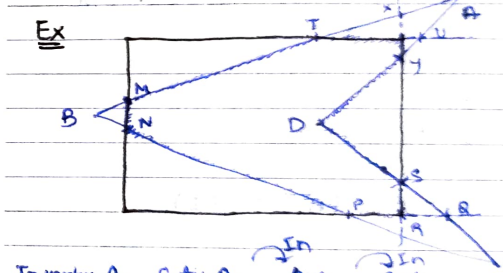
AB clipped at P

AB outside No point consid.

PB is inside clip Area

AB is comin in Tom lity.

Ex



Inverter A	Outv. A	In Out	In Out	Out
A	A	A	X	T
B	M	B	M	M
C	N	M	N	N
D	C	N	P	P
A	D	P	Q	R
	A	Q	S	S
		D	V	D
		A	Y	V
			X	U
				T

Generating Characters :-

(MI) Method of Splines (High Order Polynomials)

Adv : Less storage space is needed

Problem * Diff to Implement
* Takes time in calc.
* Diff to inc./dec. size

(MII) Font Cache / Bit Map Technique.

* Distinct font cache is needed for each font loaded by the application.

* For every caches, bitmap is saved

Adv: 1) Easy to Implement
2) Less time w.r.t splines
3) Easy to inc. size
4) Easy to Add faces (B, I, N, BI)

Bob: Storage Space

Jaggies / Zig-Zag / Stair-case Effect
Called Aliasing technique
Techn that reduce or eliminate aliasing ref. to Antialiasing.

① Inc. the resolution : ① Inc. the no. of pixels
② Make the no. of pixels better resolution.

③ Reduce Aliasing but does not remove it

Unweighted Area Sampling :-

1) Intensity of pixel intersected by a line edge decreases as the dist. bw the line edge & pixel increases

2) Primitive (Polygon/Line) can not influence the intensity at all if the primitive does not intersect the pixel.

3) Equal Areas contribute equal intensity regardless the distance bw pixel centers & area

Weighted Area Sampling. ↗ 1 & 2 same

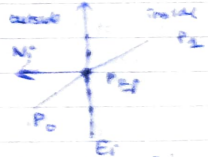
③ Area nearer to center of pixel has more effect on color of pixels

Cyrus-Beck line Clipping Also $0 \le t \le 1$

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

$P(0) = P_0 \quad P(1) = P_1$

for mid pt $P(0.5) = (P_0 + P_1)/2$



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

> 0

< 0

P_{E_i} is on edge E_i

outside plane

Inside Plane

Z-buffer algo

Two data struc which are used z buffer & frame buffer

$Ax + By + Cz + D = 0$

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

$\Rightarrow x_{i+1} = x_i + 1 \qquad y_{i+1} = y_i$

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

$$z_{i+1} = \frac{-A x_{i+1} - By_{i+1} - D}{C}$$

$$z_{i+1} = \frac{-A(x_i + 1) - B(y_i) - D}{C}$$

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

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

$\Rightarrow x_{i+1} = x_i \qquad y_{i+1} = y_i + 1$

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

Adv of Z-buffer

1) Used to render (draw) any obj of any shape
2) Z-buffer can perform radix short on x and y for value of z. So only 1 composition will be needed
3) Easy to implement

Disadv

1) Req. lot of space (z buffer array, frame buffer array, size of screen)
2) It might face aliasing

NOTE: Z-buffer algo for transparent obj like glass, window is called A buffer algo

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

Z-buffer

+

0	2	0	1	3
4				
0				
0				
0				

Z values of obj

=

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