2/1/18

COMPUTER GRAPHICS (CS335)

Books

- Hearn & Baker (computer Graphics)
- Edward Angel (Co. Graphics in Top down model)
- Geometric modelling by Mortinson
- John Vince
- F. S. Hill

- IEEE transaction on Graphics
- Annual conference called SIGGRAPH

Pre-requisites

- C ✓
- Coordinate Geometry ✓
- Linear Algebra (little bit)

Computer Graphics & related subjects

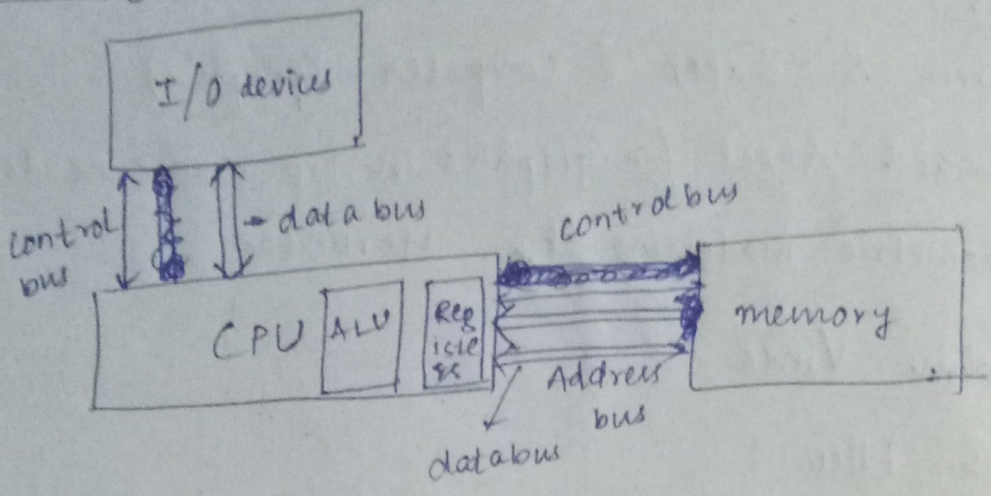
		Input	Output
1.	Computer Graphics	attributes that describes the image	image
2.	Image Processing	image	image
3.	Computer Vision	image	attributes that describes the image

- All the three are complementary to each other.

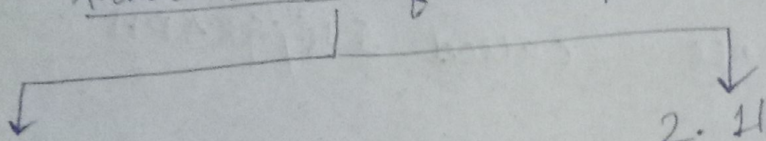
Scanned by CamScanner

4/1/18

Block diagram of a computer



Architecture of computer



1. Von neuman

2. Harvard

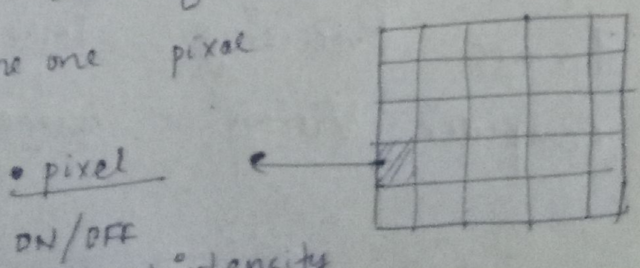
- Raster scan Graphics
- Vector scan Graphics

2D:

Pixel : Picture Element

the basic unit of measuring a display device is called a picture Element or pixel.

- Pixel organized in terms of matrix and each cell in the matrix is one one pixel



- ON/OFF
- Different intensity

- RGB : primary colours → can generate any colour

- CMY : Secondary colours → (2 primary colour)

cyan, magenta, yellow

- Primary colours don't produce all the colours in the light spectrum.

- <u>why RGB ?</u>
  - Linear combination of RGB produce all colour.
  - Better hardware implementation.

<u>D : Resolution</u> : Pixels per square inch.

- two aspects of resolution :
  - the way it is captured
  - the way it is displayed.

<u>D : ~~Raster~~ Vector Graphics</u> :

The geometric def^n of drawing / figures capture for e.g in a line $y = mx + c$

info : m, c (fixed)

- So, it is not a problem if we Zoom in or Zoom out, e.g HVG

<u>D: Rastor Graphic</u> :

- The image stored in form of pixel position information.
- So, it is a problem if we zoom in/out

e.g JPEG

11/1/18

Cathode Ray Tube

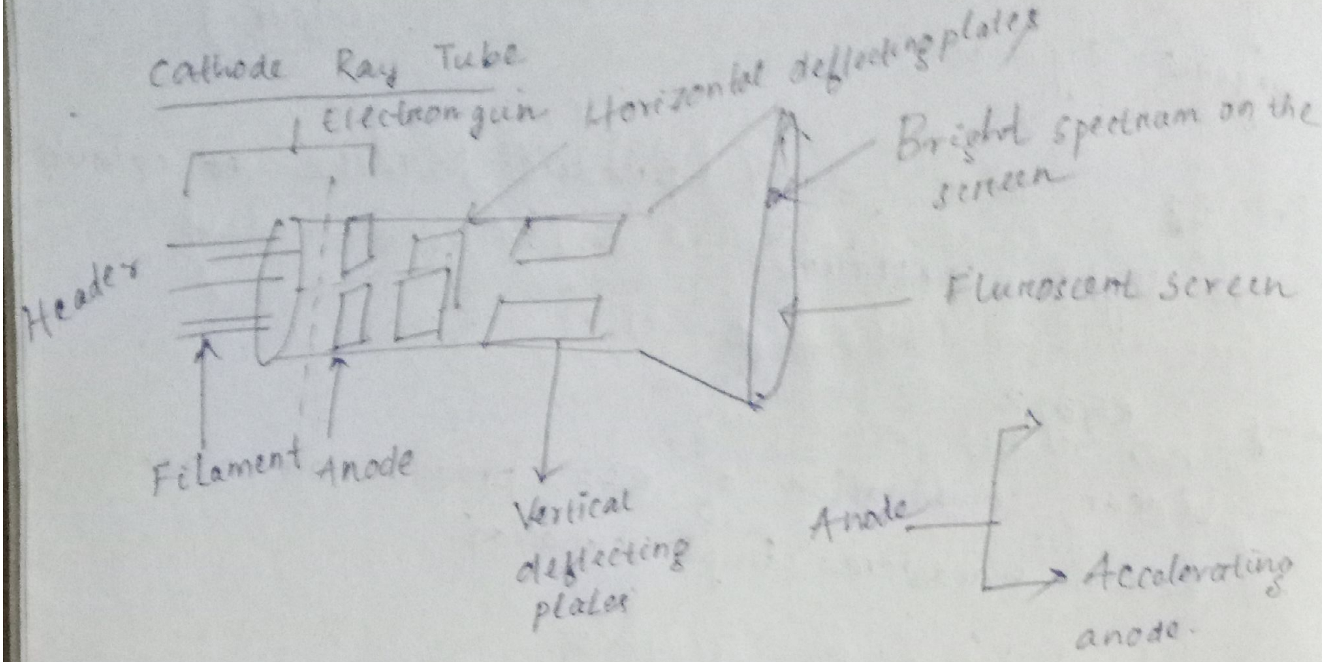
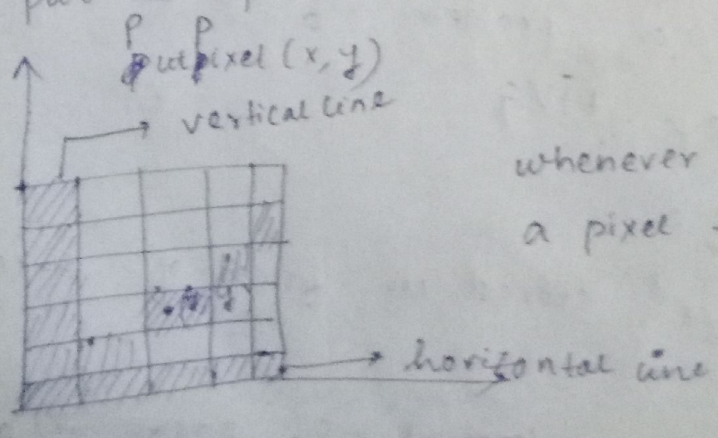


fig Cathode Ray Tube

- framebuffer: accomodates the loc of screen where light has/haven't be illuminated and the corresponding intensity of the light to be illuminated.
- for a colour TV there are three color gun

Aspect ratio : The ratio of length to width

- To plot the point :



whenever we select a point inside a pixel that pixel is illuminated.

Abstraction of a screen

DDA (Digital Differential Analyser) Algo for drawing line

$y = mx + c$

$y = mx$

(x_1, y_1) (x_2, y_2) (x_3, y_3) ... (x_m, y_m)

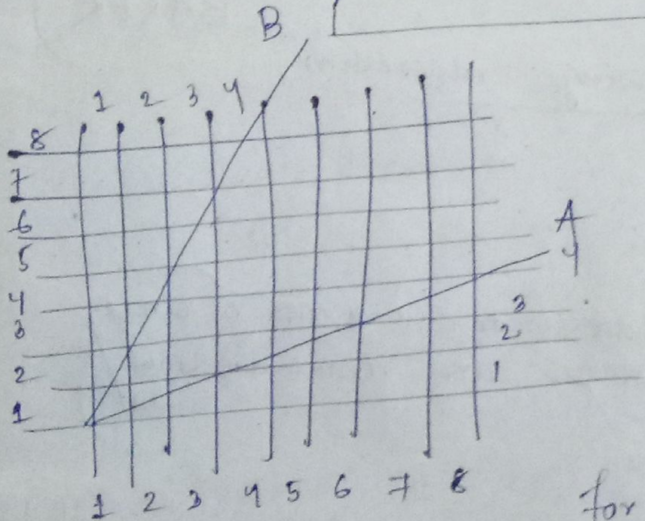
$\Delta x = x_m - x_1$

$\Delta y = y_m - y_1$

$$m = \frac{\Delta y}{\Delta x} = \frac{y_m - y_1}{x_m - x_1}$$

$$= \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$

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



$x_A > y_A$

$x_B < y_B$

for Line A

for each x find y

end for

for $x = 0$; $x \le 8$; $x{+}{+}$

$y = mx$

put pixel (x, y)

endfor

for line B

for each y find x

$x = \frac{y}{m}$

Note: In this example $m = \frac{y_{k+1} - y_k}{1}$ (as inc. value of x is 1)

for 0-45° or $y_{k+1} = m + y_k$

for $45^\circ - 90^\circ$ $m = \frac{1}{x_{k+1} - x_k}$

$\Rightarrow m x_{k+1} - m x_k = 1$

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

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

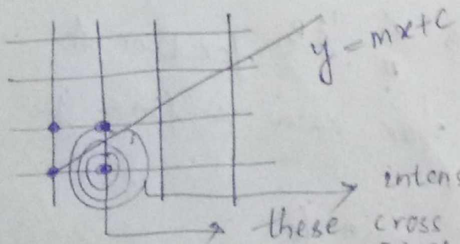
for $(y=0;\ y \leq 8;\ y{+}{+})$

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

PutPixel (x_k, y_k)

16/1/18 (READ FIRST CH. OF HANN & BAKER)

Bresenham's Line Drawing algorithm



→ intensity decreases with area of circle

→ these cross sections are illuminating light pixel

Drawback of DDA

- m : floating pt no.

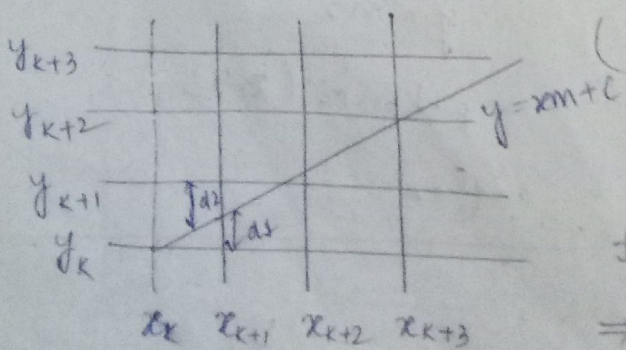


fig. 2

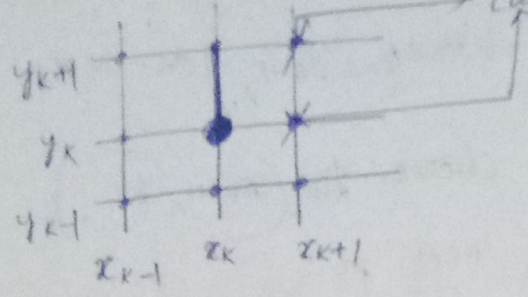
$0 \leq m \leq 1$

(x_k, y_k) : Current Pixel Location

$0 \leq m \leq 1$

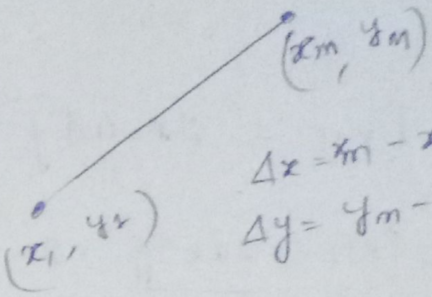
from LHS to RHS

$\Rightarrow$ it has been identified / illuminated candidate pixels



- Line drawing done incrementally
- once candidate point choosen the next point is choosen according to prev constraints.

(x_{k+1}, y_k) , (x_{k+1}, y_{k+1})



$\Delta x = x_m - x_1$

$\Delta y = y_m - y_1$

from the fig-2.

$$d_1 = y - y_k = \cancel{y\ m x_{k+1} + c} \quad m x_{k+1} + c - y_k$$

$$d_2 = y_{k+1} - y = y_{k+1} - m x_{k+1} - c$$

$$d_1 - d_2 = \cancel{\ldots} \quad 2 m x_{k+1} + 2c - (y_k + y_{k+1})$$

$$d_1 - d_2 = 2m(x_k + 1) + 2c - (y_k + y_{k+1}) \quad (\because x_{k+1} \equiv x_k + 1) \quad (\because y_{k+1} \neq y_k + 1)$$

$$= 2\frac{\Delta y}{\Delta x}(x_k + 1) - (y_{k+1} + y_k) + 2c$$

$$\Rightarrow (d_1 - d_2) \times \Delta x = 2\Delta y(x_k + 1) - (y_k + 1 + y_k)\Delta x + 2c\Delta x$$

$$\Rightarrow \underbrace{(d_1 - d_2) \times \Delta x}_{\text{decision}} = 2\Delta y(x_k) + 2\Delta y - (2y_k + 1)\Delta x + 2c\Delta x$$

$\Rightarrow$ decision parameter $(P_k) = 2x_k\Delta y - 2y_k\Delta x + 2(\Delta y) - \Delta x + 2c\Delta x$

$$\boxed{P_k = 2x_k\Delta y - 2y_k\Delta x + C \quad [\because C = 2\Delta y - \Delta x + 2c\Delta x]}$$

(1)

$$P_{k+1} = 2x_{k+1}\Delta y - 2y_{k+1}\Delta x + C \ \cancel{(2\Delta y - \Delta x + 2c\Delta x)}$$

Note: if $(d1-d2) \geq 0$ or P_K then choose (x_{K+1}, y_{K+1}) as next pt

$(d1-d2) < 0$ or P_K then choose (x_{K+1}, y_K) as next pt

$$P_{K+1} = 2x_{K+1}\Delta y - 2y_{K+1}\Delta x + C$$

$$P_{K+1} = \underline{2x_K\Delta y} + 2\Delta y - 2y_{K+1}\Delta x + \underline{C} \underline{- 2y_K\Delta x} + 2y_K\Delta x$$

$$P_{K+1} = P_K + 2\Delta y - 2y_{K+1}\Delta x + 2y_K\Delta x$$

$$\Rightarrow \boxed{P_{K+1} = P_K - 2(y_{K+1} - y_K)\Delta x + 2\Delta y} \quad (ii)$$

If $P_K > 0$, Next pt (x_{K+1}, y_{K+1})

$$P_{K+1} = P_K - 2(y_{K+1} - y_K)\Delta x + 2\Delta y$$

$$= P_K - 2(y_K + 1 - y_K)\Delta x + 2\Delta y$$

$$= P_K - 2\Delta x + 2\Delta y$$

$$= P_K - 2(\Delta x - \Delta y) = P_K + 2(\Delta y - \Delta x)$$

ELSE next point (x_{K+1}, y_K)

$$P_{K+1} = P_K + 2\Delta y$$

for line: $y = mx + 0$ [line passing through origin]

$$P_0 = 2 \times 0\Delta y - 2 \times 0\Delta x + 2\Delta y - \Delta x + 2c\Delta x$$

$$P_0 = 2\Delta y - \Delta x$$

Scanned by CamScanner

Bresenham's Algorithm

$P_0 = 2\Delta y - \Delta x$.

(1) Read two end points (x_1, y_1), $(x_m\ y_m)$

(2) $\Delta x = x_n - x_1$

(3) $\Delta y = y_m - y_1$

(3) $P_0 = 2\Delta y - \Delta x$

~~putpixel~~ putpixel (x_1, y_1)

(4) for $K=0$ to $(\Delta x - 1)$

if $P_K > 0$ putpixel (x_{K+1}, y_{K+1})

$$P_{K+1} = P_K - 2\Delta x + 2\Delta y.$$

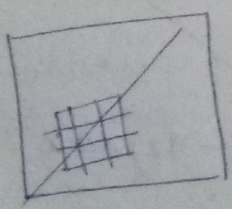
else putpixel (x_{K+1}, y_K)

$$P_{K+1} = P_K + 2\Delta y.$$

Remark: No floating point operation in this algorithm.

18/1/18 glcolor3f (1, 1, 1)

for each of these three values there would be three matrices.



$\Delta x = 4, \Delta y = 3$

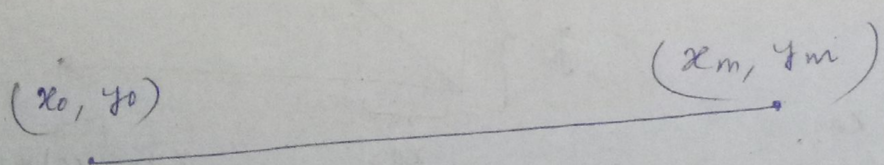
K	P_K	NextPixel	
			$P_0 = 2\times3-4=2$
0	2	(3,3)	$P_1 = 2 - 2\times4 + 2\times3 = 0$
1	0.	(4,3)	$\cancel{P_1 \geq 0}$
2	6	(5,4)	$P_2 = 0 + 2\times3 = 6$
3	4	(6,5)	$P_3 = 6 - 2\times4 + 2\times3 = 4$

23/1/18

- Both DDA and Bressenham are slow algorithm.
- flynns ~~Flings~~ classification:
  - classified m/c into 4 types

(SISD)	single	instruction	single	Data.
(MISD)	Multiple	"	Single	" (hypothetical)
(SIMD)	Single	"	Multiple	"
(MIMD)	Multiple	"	Multiple	"

most of the $\|^{le}$ m/cs use SIMD and MIMD



- Graphics processors are parallel line drawing processors.
- Graphic processor have many no. of small less/ low powered processor having small amount of memory and some common memory.

(method of parallelization of line plotting)

Intersection Line Drawing

> read (x_0, y_0) and (x_n, y_n)

① (single)

Note: In parallel line drawing only a part of the code is parallel rest is sequential.

$$m = \frac{y_n - y_0}{x_n - x_0}$$

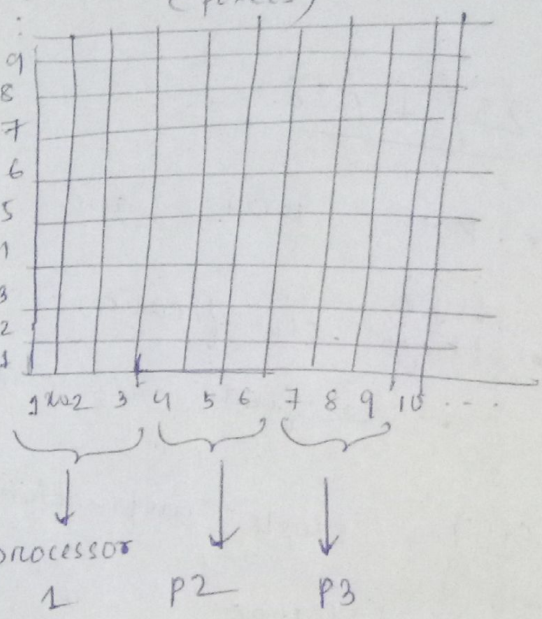
$w = 1/m$ if $m > 0$ GOTO ③

② (parallel) else GOTO ②

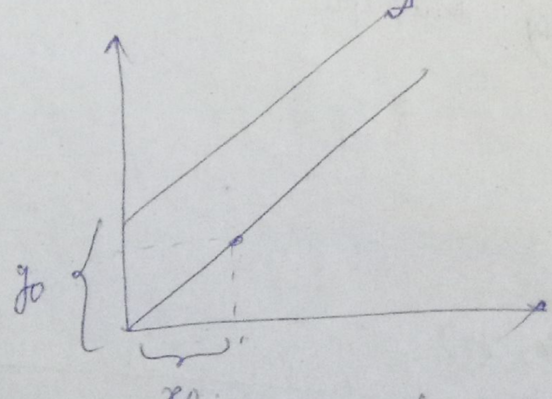
if $y_0 == y_n$ (horizontal line)

$y$co-ordinate $= y_0$

else $y$co-or $= m(\text{col} - x_0) + y_0$



[col is P_i's territory e.g (P1 → 1 2 3)]



$y$co-ord $= m(\text{col} - x_0) + y_0$

parallel

③ if $x_0 = x_m$

$x$co-ord $= x_n$

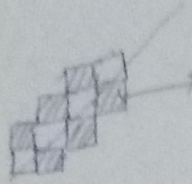
$x$co-ord $= x_0$

else

$x$co-ord $= w(\text{row} - y_0) + x_0$.

~~col = col + x_0~~

25/1/18

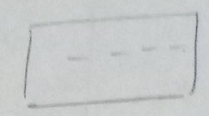


low intensity pixel (doing this jaggery effect will be less and the line would appear smoother)

<u>Staircase effect/ jaggery effect/ Aliasing</u>: comes into play if we don't follow Nyquist theorem ($f_s \geq 2 \times f_m$).

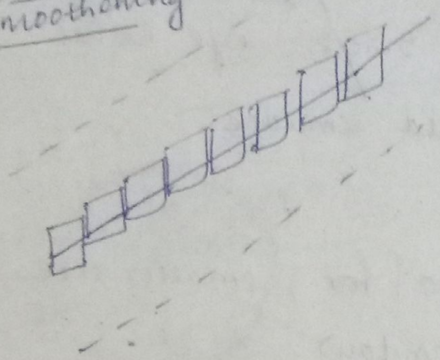
To avoid this use anti-aliasing method.

<u>Another method of smoothening</u>

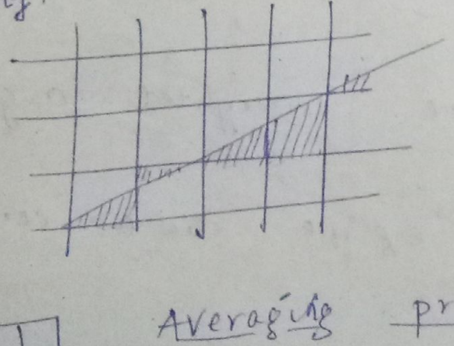


reference line

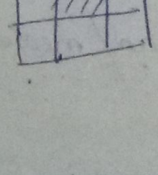
3 reference lines to draw pixels.



Note: We can't colour the part of the pixel so we can colour the pixel with proportionate amount of intensity.



∴ <u>Normalization</u> handles the problem of disproportional intensity.



- <u>Averaging problem</u>

- more size of pixel e.g (5x5 / 7x7)

nol (4x4), (2x2)

then

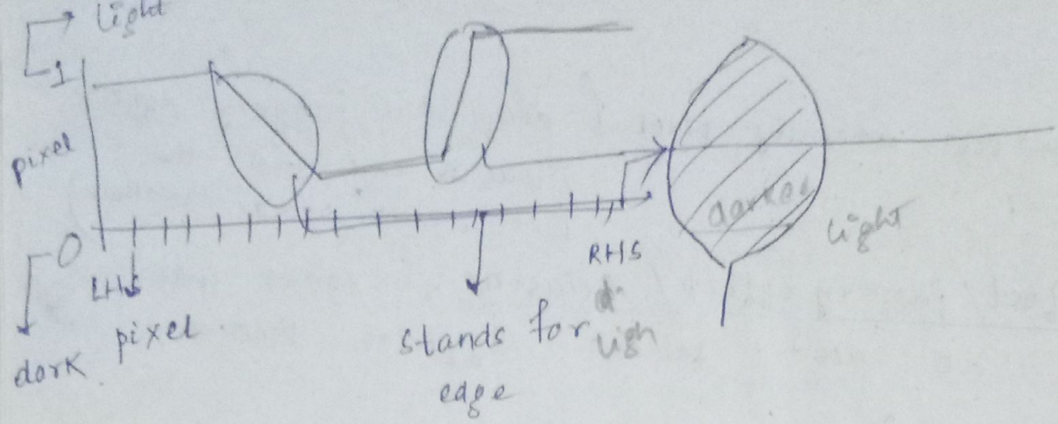
<u>Adv</u>

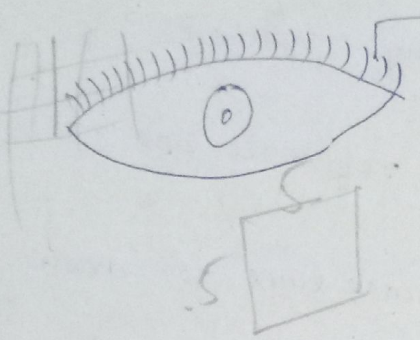
- faster coz lesser no of pixel

<u>disadv</u>

The accuracy decrease.

Edge





- In the prev. e.g

for 3x3: the eye laches visible.

- for 5x5/ 9x9: eye lashes would smudge.

Conclusion: - more edges: go for smaller size window (3x3)

- less no of edges: go for larger size window.

• There are cases where weighted avging is to be done:

- In our prev. e.g we did a centre weighted avging.

- We can do gausian avging / mean / median.

• In the bottomline: is ~~in~~ by doing the above methods we want a smooth outlines of the figures

→ Search parallel line drawing algorithm → Assignment

29/1/18

Midpoint Circle drawing Algorithm

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

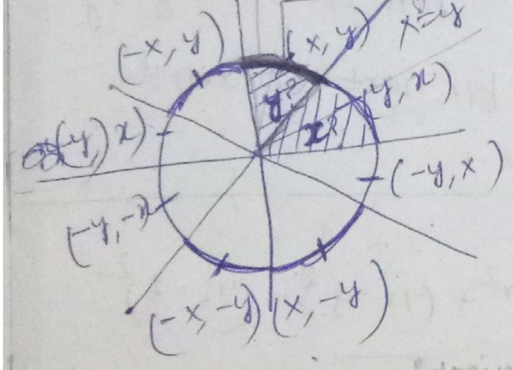
$x = r\cos\theta + a$

$y = r\sin\theta + b$

→ involves squaring and trigonometric operation that are costlier

→ first of all we're drawing only this quandrant

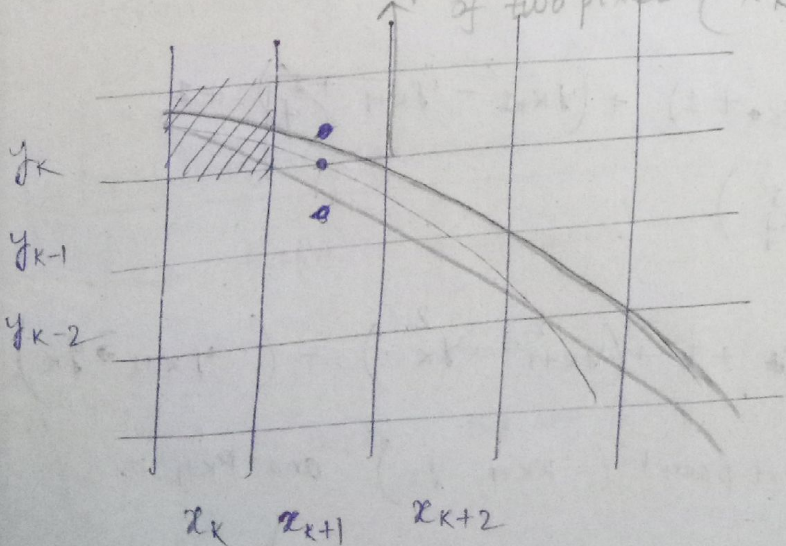
→ symmetric about octant (also about quadrant)



	point	slope
(a)	(0, r)	0
(b)	(x, x)	-1
(c)	(r, 0)	∞

as the circle is symmetric about the quadrant we can similarly find the corresponding pts in the other quadrants

midpoint of two pixel $(x_{k+1},\ y_k - \frac{1}{2})$



Note: We draw the line assuming the centre at origin, then it can be easily shifted.

circle eqⁿ = $x^2 + y^2 = r^2$

$$f(x, y) = x^2 + y^2 - r^2 \begin{cases} < 0 & \text{pt lies inside the circle} \\ = 0 & \text{point on the circle} \\ > 0 & \text{" " outside " "} \end{cases}$$

$f(x_{k+1}, y_k - \frac{1}{2})$: decision paramete

↳ if < 0 then choose pixel (x_{k+1}, y_k)

↳ if > 0 " " " (x_{k+1}, y_{k-1})

$P_k = f(x_{k+1}, y_k - \frac{1}{2}) = (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2$

$P_{k+1} = f(x_{k+1}+1, y_{k+1} - \frac{1}{2}) = (x_{k+1}+1)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2$

> $\boxed{\text{Note:}}$ There is nothing called y_{k+1} as we're going down. Rather y_{k+1} stands for next ordinate ie it can be y_k or y_{k-1}.

$= (x_k+1)^2 + 2(x_k+1) + 1 + (y_{k+1} - \frac{1}{2})^2 - r^2 + (y_k - \frac{1}{2})^2 - (y_k - \frac{1}{2})^2$

$P_{k+1} = P_k + 2(x_k + 1) + (y_{k+1}^2 - y_{k+1} + \frac{1}{4}) + 1$
$- (y_k^2 - y_k + \frac{1}{4})$

$P_{k+1} = P_k + 2(x_{k+1}) + 1 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$

if $P_k < 0$ Next point (x_{k+1}, y_k) and $P_{k+1} =$

$P_{k+1} = P_k + 2(x_{k+1}) + 1$

else Next point (x_{k+1}, y_{k-1}) and ~~P_{k+1}~~

~~$P_{k+1} = P_k + 2(x+1) + 1 + (y_k^2 - (y$~~

~~$P_{k+1} = P_k + 2$~~

Scanned by CamScanner

$P_{k+1} < y$

$$P_{k+1} = P_k + 2(x_{k+1}) + 1 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$$

$$= P_k + 2(x_{k+1}) + 1 + y_{(k+1)}^2 - y_k^2 - y_{(k+1)} + y_k$$

$$= P_k + 2(x_{k+1}) + 1 + (y_k - 1)^2 - y_k^2 - (\cancel{y_k} + 1 + \cancel{y_k})$$

$$= P_k + 2(x_{k+1}) + 1 + \cancel{y_k^2} + 1 - 2y_k - \cancel{y_k^2} \; 2y_k + 1$$

$$= P_k + 2(x_{k+1}) + 1 - 2y_k + 2$$

$$P_0 = (0+1)^2 + \left(r - \frac{1}{2}\right)^2 - r^2 \quad (P_0 \text{ with } (0, r))$$

$$= 1 + r^2 - r + 1/4 - r^2$$

$$= \frac{5}{4} - r \approx 1 - r.$$

terminating $cond^n = (x = y)$ or $(x > y)$

Problem statement for next lab.

~~for~~ take 4 random variables (radius, x-centre, y-centre, colour)

draw circles. such that for every fourth circle the ~~cir~~ first circle is erased.

30/1/18

Circle

$P_0 = \frac{5}{4} - r \approx 1 - r$.

If $P_K < 0$ Plot (A)

$P_{K+1} = P_K + 2x_{K+1} + 1$

Else PLOT (B)

$P_{K+1} = P_K + 2x_{K+1} + 1 - 2y_{K+1}$

(A) $x_{K+1} = x_K + 1$, $y_{K+1} = y_K$

(B) $x_{K+1} = x_K + 1$, $y_{K+1} = y_K - 1$.

this part is inside while $(x < y)$.

(A) $P_{K+1} = P_K + 2(x_K + 1) + 1$

$= P_K + 2x_K + 3$

(B) $P_{K+1} = P_K + 2(x_K + 1) + 1 - 2(y_K - 1)$

$= P_K + 2x_K + 2 + 1 - 2y_K + 2$

$= P_K + 2(x_K - y_K) + 5$.

Plot all points

for $r = 10$.

(0, 10)

$P_0 = 1 - 10 = -9 < 0$.

Origin is centre.

$P_1 = P_0 + 2 \times (0 + 1) + 1$

$= -9 + 2 + 1$

$= -9 + 3$

$= -6. < 0$.

next pt

(1, 0)

(2, 0)

$P_2 = -6 + 2 \times 2$

K		next pt
0	$P_0 = 1 - 10 = -9 < 0$	$(0, 10)$
1	$P_1 = -9 + 2 \times (x_k + 1) + 1$ $-9 + 2 \times (0 + 1) + 1$ $= -9 + \cancel{2 \times 2 +1}\ 2 + 1$ $= \cancel{-9 + 5 = -4}\ -6 < 0$	$\cancel{(2,10)}\ (1, 10)$
2	$P_2 = \cancel{-4 + 2 \times (2+1)} + 1$ $-6 + 2 \times (x_1 + 1) + 1$ $= -6 + 2 \times (1 + 1) + 1$ $= \cancel{-6 + 6 + 1}\ -6 + 5$ $= \cancel{1 > 0}\ -1 < 0$	$(2, 10)$
3	$P_3 = -1 + 2 \times (2 + 1) + 1$ $= 6 > 0$	$(3, 10)$
4	$P_4 = 6 + 2 \times (4) + 1 - 2 \times (9)$ $= 6 + 8 + 1 - 2 \times 9$ $= 6 + 8 + 1 - 18$ $= \cancel{15 - 22}\ 15 - 18$ $= \cancel{-7} = -3 < 0$	$(4, 9)$

Scanned by CamScanner

5 $P_5 = -3 + 2\times(5) + 1$ $= -3 + 10 + 1$ $= -3 + 11 = 8 > 0$	$(5, 109)$
6. $P_6 = 8 + 2(6) + 1 - 2(8)$ $= 8 + 12 + 1 - 16$ $= 21 - 16$ $= 5 > 0$	$(6, 8)$
7 $P_7 = 5 + 2\times(7) + 1 - 2\times 7$ $= 5 + \cancel{14} + 1 - \cancel{14}$ $= 6 > 0$	$(7, 7)$
8	$(8, 6)$

$\therefore$ As $x > y$ hence here the algo stops.

$$P_{K+1} = \left(\boxed{P_K + 2x_{K+1} + 1}\right)$$

$$P_{K+1} = P_{K+2} = P_K + 2x_{K+2} + 1 + 1 = \left(P_K + 2x_{K+1} + 1\right) + 2x_{K+2} + 1$$

$$= P_K + 2x_K + 4 + 1 + 1 = P_K + 2x_{K+1} + 1 + 2(x_K + 2) + 1$$

$$= (P_K + 2x_K + 1) + 5 = P_K + 2x_{K+1} + 1 + 2(x_{K+1} + 1) + 1$$

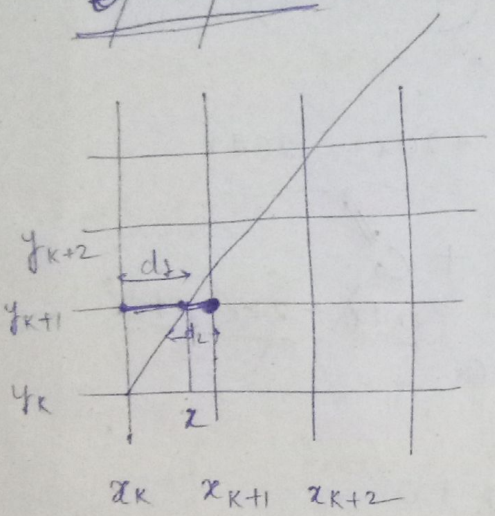
$= P_K + 2x_{K+1} + 1 + 2x_{K+1} + 2 + 1$

$= P_K + 4x_{K+1} + 1 + 3$

$= P_K +$

- Do the derivation with $1/4^{th}$ of the circle.

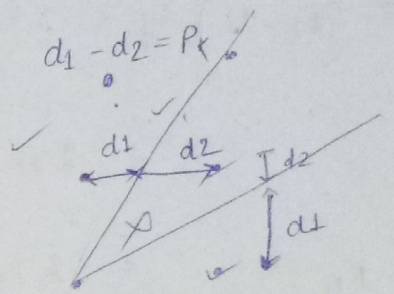
6/2/18



$y = mx + C$

$\Rightarrow x = \frac{y - C}{m}$

$d_1 - d_2 = P_K$



$d_1 = x - x_K$

$= \frac{y_{K+1} - C}{m} - x_K$

$d_2 = x_{K+1} - x$

$= x_{K+1} - \frac{y_{K+1} - C}{m}$

$$d_1 - d_2 = \left(\frac{y_{K+1} - C}{m} - x_K\right) - \left(x_{K+1} - \frac{y_{K+1} - C}{m}\right)$$

$$= \frac{2y_{K+1}}{m} - \frac{2C}{m} - x_K - x_{K+1}$$

$$= \frac{2y_{K+1}}{m} - \frac{2C}{m} - (x_K + x_{K+1})$$

$$d_1 - d_2 = \frac{2y_{k+1}}{m} - \frac{2c}{m} - (x_k + x_{k+1})$$

$$m = \frac{\Delta y}{\Delta x}$$

$$d_1 - d_2 = \frac{2(y_k+1)}{\frac{dy}{dx}} - \frac{2c}{\frac{dy}{dx}} - (x_k + x_{k+1})$$

$$\Rightarrow (d_1 - d_2)\,dy = 2(y_k+1)\,dx - 2c\,dx - (x_k + x_{k+1}) \times dy$$

$$= \underbrace{2y_k\,dx - (x_k + x_{k+1})\,dy + 2dx - 2c\,dx}$$

$$P_k = 2y_k dx - (x_k + x_{k+1})dy + C_0$$

$$= 2y_k dx - (2x_k + 1)dy + C_0 = 2y_k dx - 2x_k dy + C_0'$$

$$P_{k+1} = \cancel{2y_{k+1}dx - (2x_k + 1)\,dy + C_0}$$

$$= \cancel{2y_k} \quad 2y_{k+1}dx - 2x_{k+1}dy + C_0'$$

$$= 2(y_k + 1)dx - 2(x_{k+1})dy + C_0' + 2x_k dy - 2x_k dy$$

$$= 2y_k dx + 2dx - 2x_{k+1}dy + C_0' + 2x_k dy - 2x_k dy.$$

$$P_{k+1} = P_k + 2dx - 2x_{k+1}dy + C_0' + 2x_k dy.$$

$$= P_k - 2(x_{k+1} - x_k)dy + 2dx.$$

If $P_k > 0$ Next pt (x_{k+1}, y_{k+1}).

$$P_{k+1} = P_k - 2(x_{k+1} - x_k) + 2dx$$

$$= P_k - 2(\cancel{x_k} + 1)dy + 2dx$$

$$= P_k - 2dy + 2dx.$$

Scanned by CamScanner

If $p_k < 0$ Next pt (x_k, y_{k+1})

$$p_{k+1} = p_k - 2(x_k - x_k) + 2dx = p_k + 2dx.$$

p_0 (line passing through origin

$$p_0 = 2\overset{0}{y_k}dx - (2\overset{0}{x_k}+1)dy + 2dx - \overset{0}{2c}dx$$

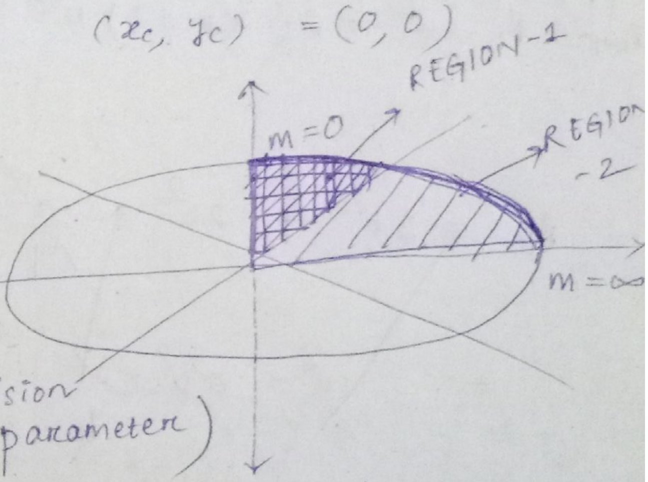
$$= -\cancel{2}dy + 2dx.$$

$$= 2dx - dy$$

Ellipse

$$\left(\frac{x - x_c}{a}\right)^2 + \left(\frac{y - y_c}{b}\right)^2 = 1$$

if centre of ellipse $(x_c, y_c) = (0, 0)$

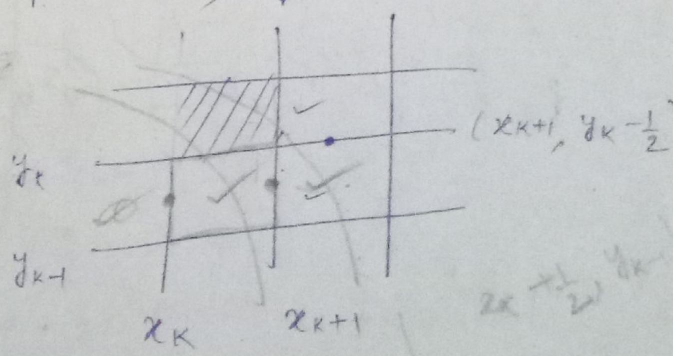


$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$$

$\Rightarrow b^2x^2 + ay^2 - a^2b^2 = 0$ (decision parameter)

$\Rightarrow$

$f(x_{k+1}, y_k - \frac{1}{2})$



$= p_k$

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

$\longrightarrow$ $\boxed{\text{REGION ONE}}$

$$p_{k+1}^{(1)} = b^2(x_{k+1} + 1)^2 + a(y_{k+1} - \tfrac{1}{2})^2 a^2 - a^2b^2$$

$$P_{K+1}^{(1)} = \underline{b^2(x_k+1)^2} + b^2 + 2(x_K+1)b^2 + a^2(y_{K+1}-\tfrac{1}{2})^2$$

$$\underline{-a^2b^2} + \underline{a^2(y_K-\tfrac{1}{2})^2} - \underline{a^2(y_K-\tfrac{1}{2})^2}$$

∴ Add & subtract

$$P_{K+1}^{(1)} = P_K^{(1)} + b^2 + 2(x_K+1)b^2 + a^2(y_{K+1}-\tfrac{1}{2})^2 - a^2(y_K-\tfrac{1}{2})^2$$

$$P_{K+1}^{(1)} = P_K^{(1)} + b^2 + 2(x_K+1)b^2 + a^2\left(y_{K+1}^2 + \cancel{\tfrac{1}{4}} - y_{K+1} - y_K^2 - \cancel{\tfrac{1}{4}} + y_K\right)$$

$$P_{K+1}^{(1)} = P_K^{(1)} + b^2 + 2(x_K+1)b^2 + a^2\left((y_{K+1}^2 - y_K^2) - (y_{K+1}-y_K)\right)$$

$$P_0 = 2b^2 + 2b^2 + \cancel{\frac{a^2}{4}} - a^2b^2 - \cancel{\frac{a^2}{4}} - \frac{a^2}{4}$$

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

$$P_0 = b^2 + \frac{a^2}{4} - a^2b$$

$P_K^{(2)} = ?$

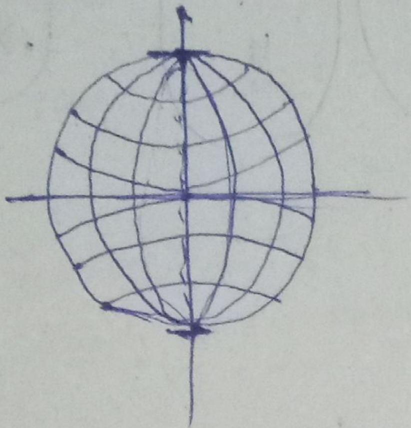
$P_{K+1}^{(2)} = ?$

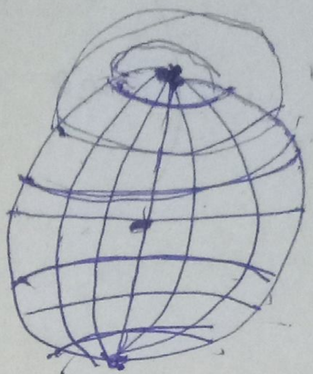
$P_0^{(2)} = ?$

<u>midpoint</u>

$f(x_k + \frac{1}{2}, y_k + \frac{1}{2})$

Note: How to transit from region 1 to region-2.





<u>12/2/18</u>

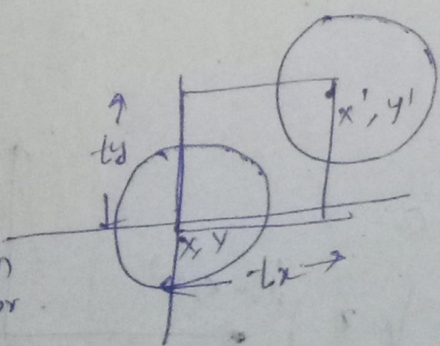
<u>Transformation</u>

- translation
- rotation
- scaling
- reflection:

<u>translation</u>

has two parameters t_x, t_y.

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

(*) (t_x, t_y) : translation vector or shift vector

<u>Rotation</u>

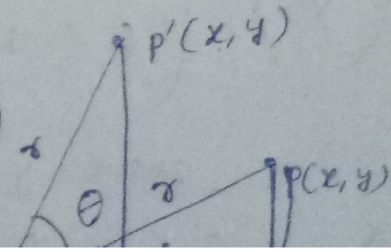
$x = r\cos\alpha$

$\boxed{x' = r\cos(\theta + \alpha)}$

$= r\cos\theta\cos\alpha - \sin\theta\sin\alpha)$

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

$\boxed{y' = r \sin(\theta + \alpha)}$



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

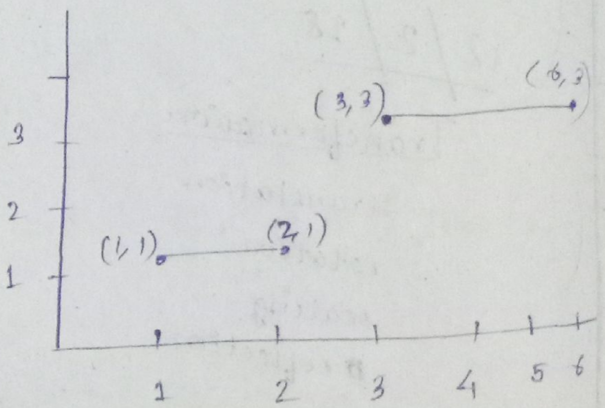
$$y' = y\cos\theta + x\sin\theta$$

in form of matrix.

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Scaling

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

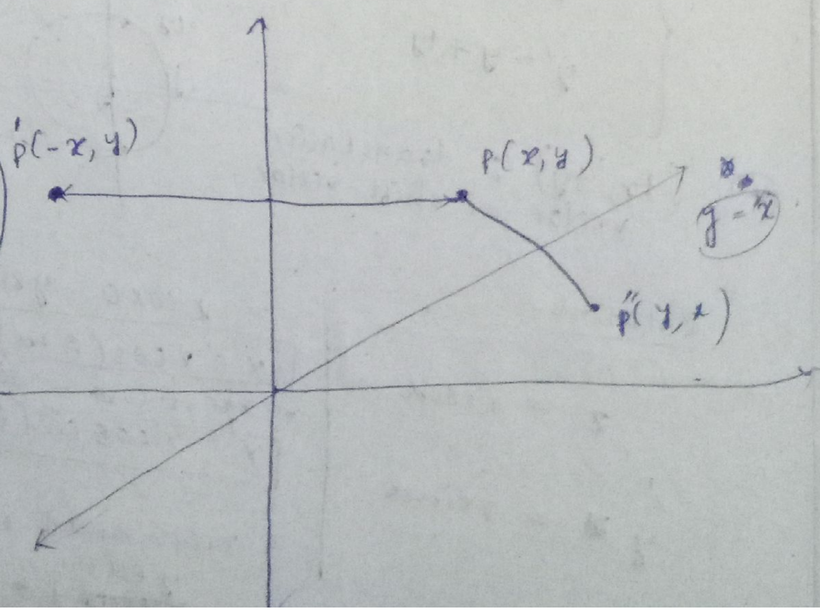


Scaling by a factor of 3 (x3)

Reflection

about $x = 0$.

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



about $y = 0$.

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

Lab

take arbitrary shape and rotate it by an arbitrary angle about the origin. DO for all the given combinations Do both raster graphically and vector graphically.

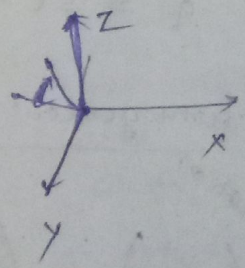
problem: cos θ, sin θ values may approximate to same (single values).

- rotate a pt (X, Y, Z) by an angle θ about origin (i.e around X-Y plane)

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

3×3   3×1

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



YZ plane

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

$x' = x$

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

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

Scanned by CamScanner

XZ plane

$$\begin{pmatrix} \sin\theta & 0 & \cos\theta \\ 0 & 1 & 0 \\ c\theta & 0 & -s\theta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

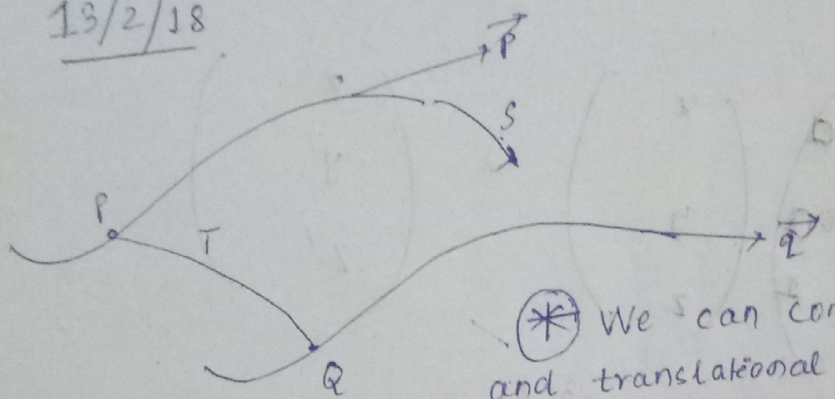
$x' = x\sin\theta + z\cos\theta$

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

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

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

13/2/18



$Q = T(P)$

$\vec{q} = s(\vec{P})$

(*) We can combine the multiplicative and translational terms of 2d geometric transformations to 3 by 3 matrices which allows us to express all transformation eq^n s as matrix multiplication

homogeneous co-ordinates / Representation

for translation

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

transformation matrix

$$\begin{pmatrix} c\theta & -s\theta & t_x \\ s\theta & c\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

both translation & rotation.

for rotation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

COMPOSITE TRASFORMATION MATRIX

- first rotation then

Scanned by CamScanner

$$(R_x(R_y(R_z(\overset{\text{point}}{P})))) \equiv (R_x R_y R_z)(P)$$

Rotation w.r.t x

P → variable

these transformation matrices don't change
↓
Less overhead on multiplying $R_x R_y R_z$ once.

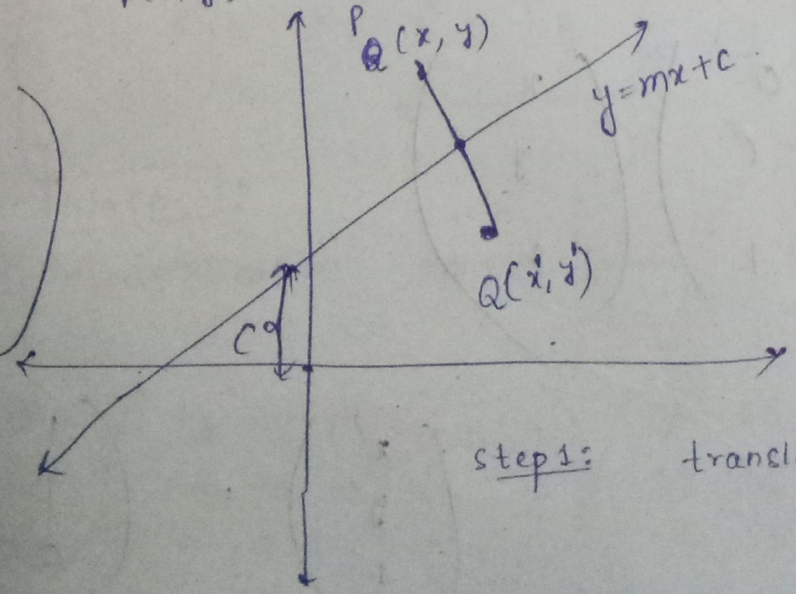
Affine Transformation: $x' = a_{xx}x + a_{xy}y + b_x$
$y' = a_{yx}x + a_{yy}y + b_y$

Each transformed co-ordinate x' and y' is a linear funⁿ of x and y.

3D for both translation and rotation:

$$\begin{pmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Find the transformation matrix to reflect a point $P(x, y)$ about an arbitrary line $y = mx + c$



Sol^n: ~~rat~~ rotate the axes to co-incide with the line

or rotate the line to pass through axes.

Step 1: translation of line to make $y = mx$

Step 2: Rotate the line by $-\theta$ to co-incide with axes

Step 1.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ mx + c \end{pmatrix} + \begin{pmatrix} 0 \\ -c \end{pmatrix}$$

Step 2 (rotation by $-\theta$)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ mx \end{pmatrix}$$

Step3 (reflection about $y=0$)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \underbrace{\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}}_{2\times 2} \underbrace{\begin{pmatrix} x \\ mx \end{pmatrix}}_{2\times 1}$$

transformation

$$\begin{pmatrix} C\theta & S\theta & 0 \\ -S\theta & C\theta & -C \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & +c \\ 0 & 0 & 1 \end{pmatrix}$$

Step 4 (rotation by $+\theta$)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} C\theta & S\theta \\ -S\theta & C\theta \end{pmatrix} \begin{pmatrix} x \\ mx \end{pmatrix}$$

Step 5 (tranlation by c)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} C\theta & -S\theta \\ S\theta & C\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} C\theta & S\theta \\ -S\theta & C\theta \end{pmatrix} \begin{pmatrix} x \\ mx \end{pmatrix} + \begin{pmatrix} 0 \\ c \end{pmatrix}$$

$$\left(T_{+c} \left(R_{\theta} \left(X_{x} \left(R_{-\theta} \left(T_{-c} \left(P \right) \right) \right) \right) \right) \right)$$

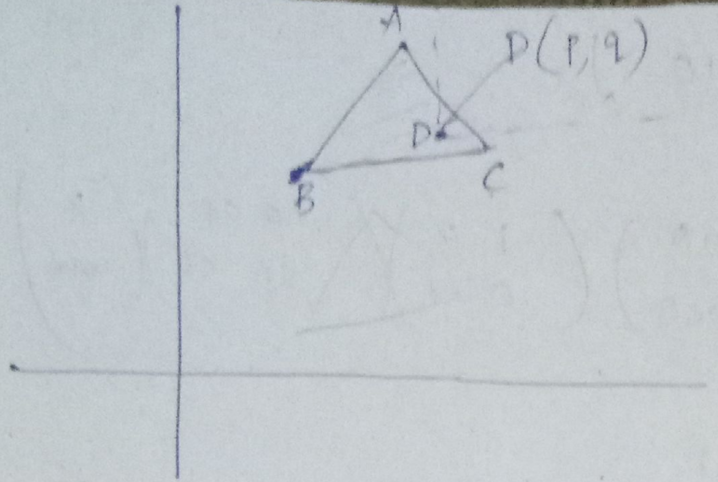
$\theta = \tan^{-1}(m)$

$$\left\{ T_{+c}\, R_{+\theta}\, X_{x}\, R_{-\theta}\, T_{-c} \right\} (P)$$

$\boxed{\text{Note}}$:
Multiplication done from R to L

$$\begin{pmatrix} & 0 \\ 1 & +c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C\theta & S\theta & 0 \\ -S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$\boxed{\text{Remark}}$: multeply it finally.



Rotate the triangle by an angle θ about the point D.

Step 1 translate to bring point D to origin

$$\begin{pmatrix}1 & 0 & p\\ 0 & 1 & q\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}c\theta & -s\theta & 0\\ s\theta & c\theta & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & -p\\ 0 & 1 & -q\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

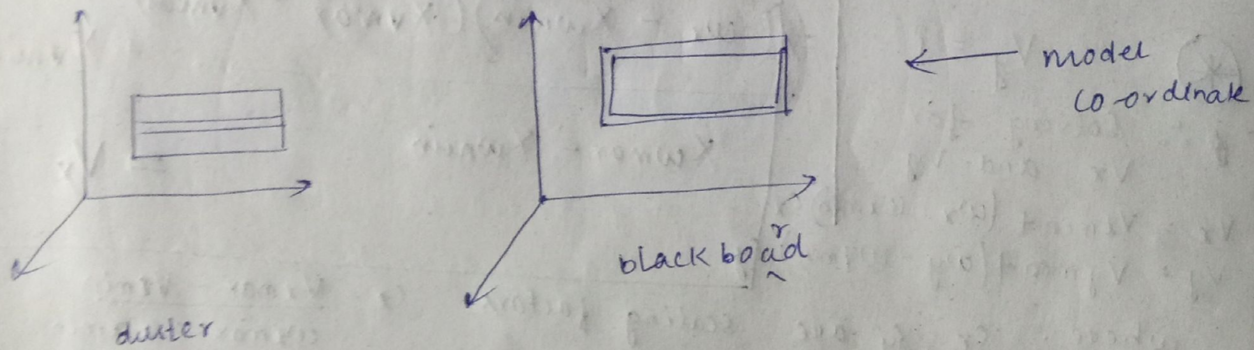
As with translations, rotations are rigid body transformations that move objects without deformation. Every point on an object is rotated through the same angle.

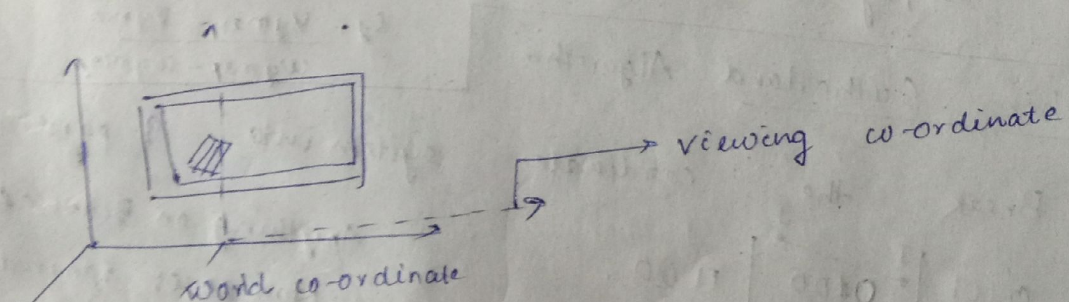
15/2/18

- Co-ordinate system with which each unit is prepared is called "Model co-ordinate system". (individual co-ordinate)

- The co-ordinate system in which all the units are merged is called "world co-ordinate system".

- The co-ordinate system that depicts which part of the world co-ordinate is to be visualized is called "viewing co-ordinate".



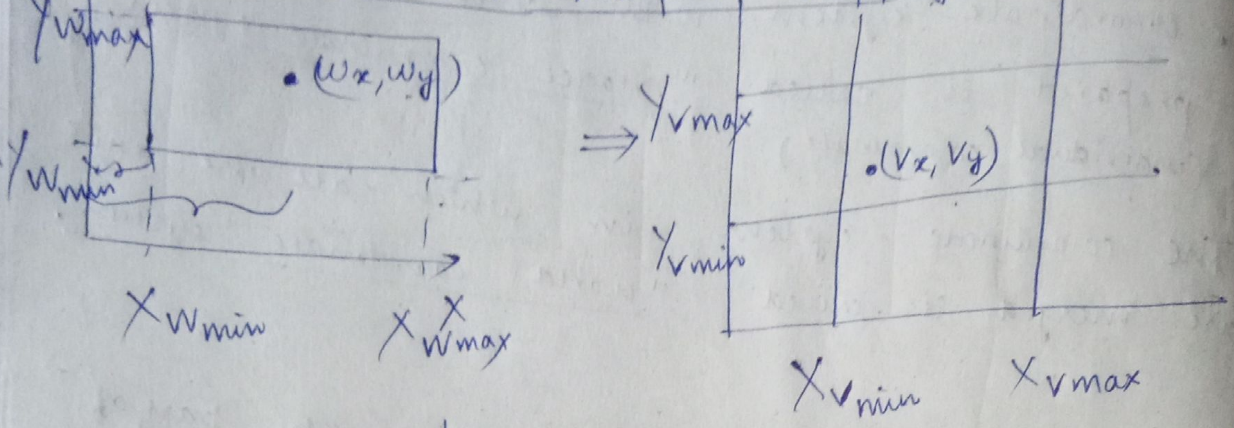


- Device co-ordinate (projector etc) and viewing coordinate can have some gaps hence to bridge the gap we use "Normalized coordinate".

- Normalized co-ordinates: represented as [0, 1]

Transforming one system to another system

Window to viewport mapping



$V_x = ?$

$$\frac{w_x - X_{wmin}}{X_{wmax} - X_{wmin}} = \frac{V_x - X_{vmin}}{X_{vmax} - X_{vmin}}$$

$V_y = ?$

$$\Rightarrow \boxed{\frac{(w_x - X_{wmin})(X_{vmax} - X_{vmin})}{X_{wmax} - X_{wmin}} + X_{vmin} = V_x}$$

∴ Solving for V_x and V_y

$V_x = V_{xmin} + (w_x - w_{xmin})\, s_x$

$V_y = V_{ymin} + (w_y - w_{ymin})\, s_y$

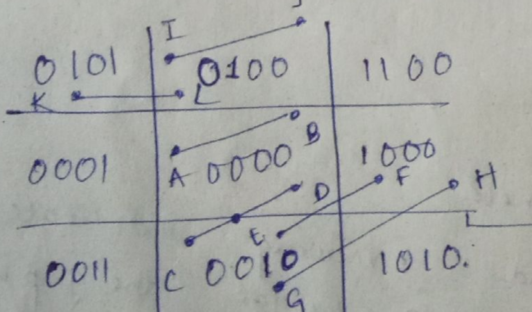
where. s_x, s_y are scaling factors.

$$s_x = \frac{V_{xmax} - V_{xmin}}{w_{xmax} - w_{xmin}}$$

$$s_y = \frac{V_{ymax} - V_{ymin}}{w_{ymax} - w_{ymin}}$$

Cohen Sutherland Algorithm.

Break the coordinate system into 9 parts



this axes are Y_{wmin}, Y_{wmax}, X_{wmin}, X_{wmax}

(RTBL code system)

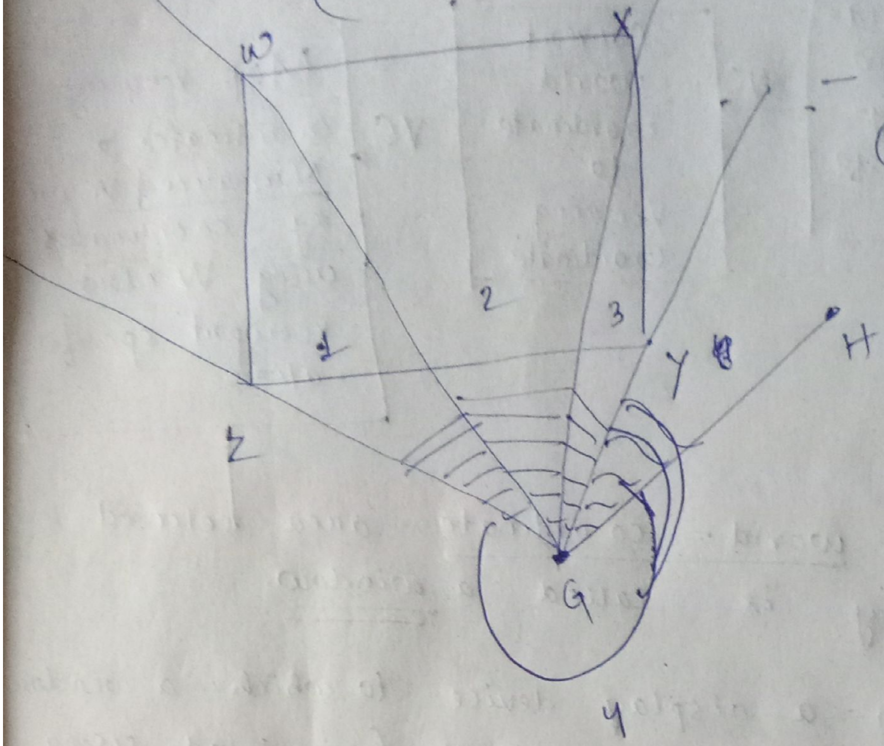
Anything on Bitwise AND

Non-zero: ignored (completely inside) (completely outside)

Zero: tested (partially inside).

- for testing whether V_x and V_y lie inside the given boundary or not.

Nikol-L-Nikol (makes 4 regions w.r.t the lines that needs to be tested e.g GH)
(N-L-N)



- draw ang lines through (say G) that pass through 4 corners of rectangle rendering 4 regions

$\overline{GH}$: A line outside the boundary.

cohensutherland algo.

N-L-N : advancement of cohensutherland algo. It ~~prevents~~ saves the work of testing for line $\overline{GH}$.

If GH lies in 4th region no-need to test. else need to be tested.

Note: If the line is completely inside N-L-N would be costlier. Hence, it is recomended to take combination of cohensutherland and N-L-N algo.