NAME : TARUN
ROLL ... 4 # 2

1) $Q(u) = \sum_{i=0}^{n} P_i \, B_{in}(u)$

$= ({}^{n}C_i)(u^i)(1-u)^{n-i}$

$n = 2 \qquad 0 \le u \le 1$

$Q(u) = P_0 B_{02}(u) + P_1 B_{12}(u) + P_2 B_{22}(u)$

$Q(u) = P_0\left[{}^{2}C_0 (u)^0 (1-u)^{2-0}\right] + P_1\left[{}^{2}C_1 (u)^1 (1-u)^{2-1}\right]$

$+ P_2\left[{}^{2}C_2 (u)^2 (1-u)^{2-2}\right]$

$= P_0\left[(1-u)^2\right] + P_1\left[2u(1-u)\right] + P_2\left(u^2\right)$

for x

$Q(u) = 4(1+u^2-2u) + 8(2u-2u^2) + 1(u^2)$

$= 4 + 4u^2 - 8u + 16u - 16u^2 + u^2$

for y

$Q(u) = 2(1+u^2-2u) + 8(2u-2u^2) + 4(u^2)$

$= 2 + 2u^2 - 4u + 16u - 16u^2 + 4u^2$

$= 2 + 12u - 10u^2$

$u = 0.2 \rightarrow Q(u) = 4 + 1.6 - 0.36 = 5.24$

$Q(u) = 2 + 2.4 - 0.4 = 4$

Ambient light reflection

- Ambient light is assumed to impinge equally on all surfaces from all direction

$I = I_a K_a$

$I_a \rightarrow$ Intensity of ambient light

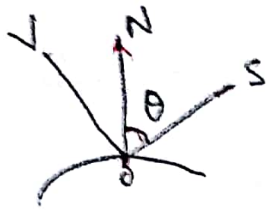
$K_a \rightarrow$ ambient reflection coefficient $(0 \leq K_a \leq 1)$

more realistic

ambient + diff.

$I = I_a K_a + I_s K_d \vec{N}.\vec{S}$

Diffuse reflection

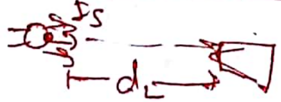


$I = I_s K_d \cos\theta$

$I = I_s K_d \vec{N}.\vec{S}$

For a Lambertian surface (Dull) the amount of light seen by viewer is independent of viewer direction

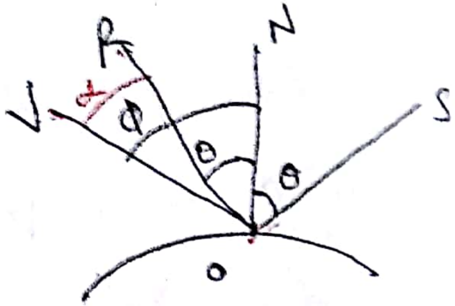
$0 \leq K_d \leq 1$ and $0 \leq \theta \leq 90$



$f_{at} = \frac{1}{d_L^2}$

$I = f_{at} I_s K_d \vec{N}.\vec{S}$

Specular reflection



$R \rightarrow$ reflection vector (all...

$I = I_s K_s \cos^n \alpha$

$n \rightarrow$ specular intensity

$R = 2N(N.S) - S$

$\cos\alpha = R.V.$

Phong Illumination

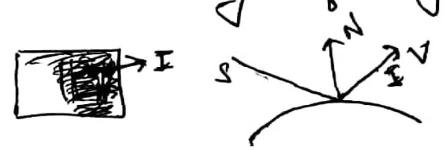
I = ambient + diffuse + spe...

Illumination Model

diffuse ⟷ specular

paper — metal — mirror

1) Illumination models is used to calculate the intensity of light that is reflected a given point on surface.



2) Rendering methods use the intensity calculations from the illumination mode to determine the light intensity at all pixels in the image, by possibly, Considering light propagation between surfaces in the scene.

Light Source:

1) Point Source

2) Parallel Source

infinite

3) Distributed Source finite ray

Ambient

$I = I_a K_a$ — diffuse reflection

Diffuse

→ Roughe

Specular

Shading

1) Illumination → How to color single point

Shading → How to color whole object $I = I_s K_d \vec{N} \cdot \vec{S}$

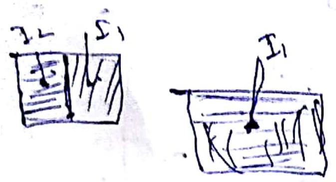
Flat shading (Constant Shading)

(One light calculation per polygon)

- → Each entire polygon drawn with same color
- → Color is computed once for each polygon

Drawback -

1) Mach Band effect



Gouraud shading

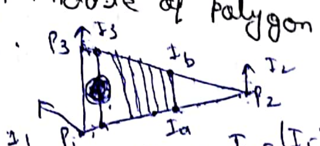
(Color Interpolation shading)

- → Color is computed once per vertex using normal of vertex
- → Colors are interpolated across polygon

Drawback!

Slower than Flat shading

- → It can miss highlights that occur in middle of polygon



$I_a = I_1 - (I_1 - I_3)\frac{(y_1 - y_a)}{(y_1 - y_2)}$

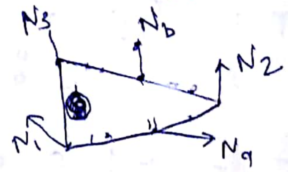
or

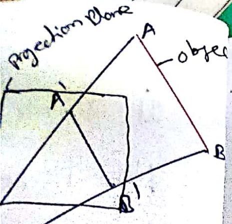
$I_a = \frac{I_1 + I_2}{2}$

Phong shading

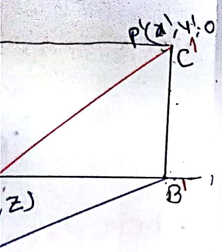
(Normal Interpolation shad

- → Normals are interpolate across polygon





re of Projection



$$\frac{Z_c}{Z_c - Z} = \frac{A}{A}$$

1%. abble apple (m) → Broken into

Perspective Projection Example

Q:- Perform perspective Projection on to $Z=0$ plane of unit cube where center of projection is at $x_c = 10$ and $y_c = 10$

Solution:

$$T_{pq} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & q & 0 & 1 \end{bmatrix} \rightarrow T_{pq} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(x, y, z)

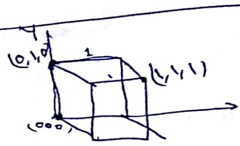
$$\underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\text{column}} \rightarrow \underbrace{[x\ y\ z]}_{\text{row}}$$

$$T = T_{pq} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -0.1 \\ 0 & 1 & 0 & -0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$p = -1/x_c = -1/10 = -0.1$

$q = -1/y_c = -1/10 = -0.1$

$$T = \begin{bmatrix} 1 & 0 & 0 & -0.1 \\ 0 & 1 & 0 & -0.1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



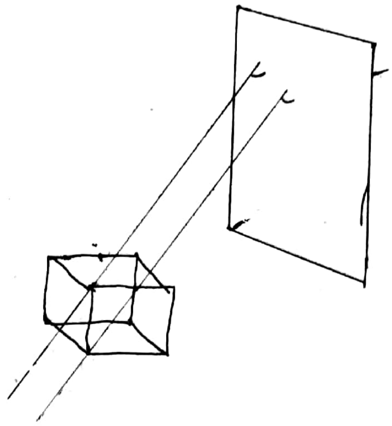
$$T_w = 0 \cdot T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -0.1 \\ 0 & 1 & 0 & -0.1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0.9 \\ 1 & 1 & 0 & 0.8 \\ 0 & 1 & 0 & 0.9 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0.9 \\ 1 & 1 & 0 & 0.8 \\ 0 & 1 & 0 & 0.9 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1.11 & 0 \\ 1.25 & 1.25 \\ 0 & 1.11 \\ 0 & \\ 1.11 & \\ 0.25 & \\ 0 & \end{bmatrix}$$

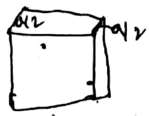
Oblique Projection

- It is method of drawing object in 3dimension
- Non perpendicular projectors to the plane of projection
- True shape and size for the faces parallel to projection plane is preserved



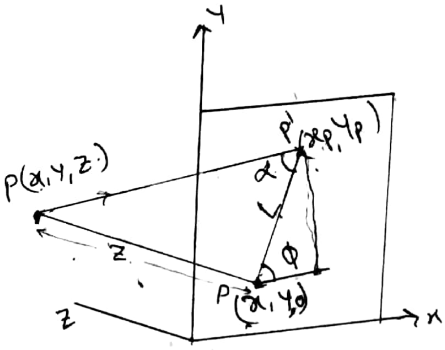
When $\alpha = 45°$ → Cavalier (Lines perpendicular to projection plane are not foreshortened)

When $\cot\alpha = \frac{1}{2}$ → Cabinet



Lines perpendicular to projection plane are foreshortened by half

ϕ is typically 30° or 45°



$$x_p = x + L\cos\phi$$

$$y_p = y + L\sin\phi$$

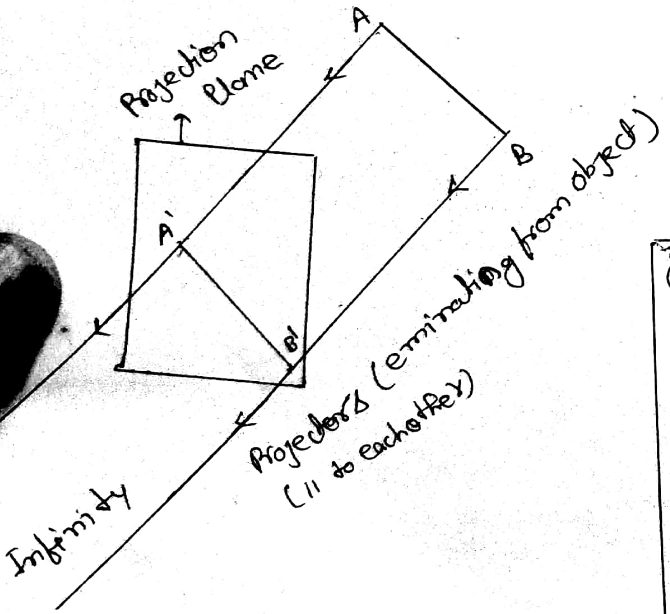
$$\tan\alpha = z/L$$

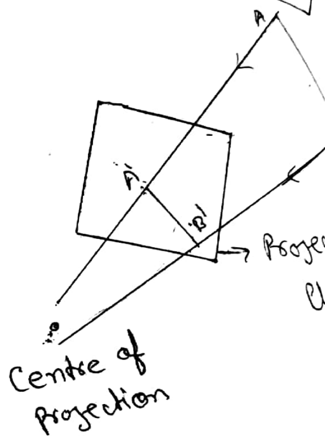
$$\cot\alpha = L/z$$

$$P = \begin{bmatrix} 1 & 0 & L\cos\phi & 0 \\ 0 & 1 & L\sin\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projection

- Parallel Projection
  - Oblique
    - Cavalier
    - Cabinet
  - Orthographics
    - Axonometric
      - Isometric
      - Dimetric
      - Trimetric
    - Multivew Orthographic
- Perspective Projection
  - Single Pt
  - 2 Point
  - 3 Point





Orthographics Multiview ($x=0$, or $y=0$ or $Z=0$ planes)
One view is not adequate

On $z=0$ plane

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

B-Splines

→ Each control point is associated with unique basis function

→ Each point affects the shape of the curve over range of parameter values where the basis function is non-zero

$$x(u) = \sum_{i=0}^{i=n} \underbrace{N_{i,k}(u)}\, x_i \quad , \quad 0 \le u \le n-k+2$$

→ Control points → $n+1$, and order of curve → k → Each Segment Influenced by k points

→ Curve is made up of $n-k+2$ segments

Basis function:

$$N_{i,k}(u) = \frac{(u-t_i)\, N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k}-u)\, N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}}$$

t_i $(0 \le i \le n+k)$ – Knot values

$t_i = 0$ if $i < k$

$t_i = i-k+1$ if $k \le i \le n$

$t_i = n-k+2$ if $i > n$

$N_{i,k}(u) = 1$ if $t_i \le u \le t_{i+1}$

$= 0$ otherwise

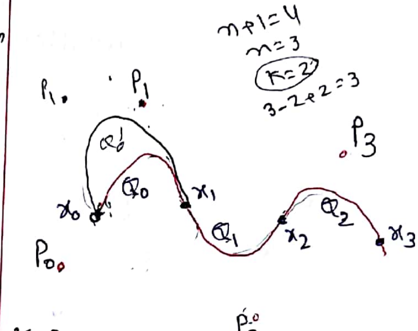
Ex → $n = \overset{6}{\cancel{5}}$ and $k=3$.

t_i $(0 \le i \le 8)$ are knot value

$t_i = \{0,0,0,1,2,3,4,4,4\}$

$N_{0,3}(u) = (1-u)^2 N_{2,1}(u)$

B-Spline
(Local Control)

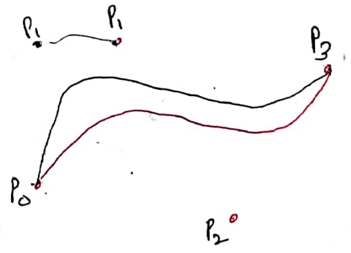


$K=2$ $\quad P_2$

P_0, P_1, P_2, P_3 – Control Points

x_0, x_1, x_2, x_3 – Knot Value

Bezier Curve
(Global Control)



$$x(u) = \sum_{i=0}^{n} B_{i,n}(u)\, x_i \, , \quad 0 \le u \le 1$$

Example

Bezier Curve

Q: Construct Bezier curve for control Points (4,2) (8,8) and (16,4)
P_0 P_1 P_2

$$Q(u) = \sum_{i=0}^{2} P_i \, B_{i,2}(u) \quad , 0 \le u \le 1$$

$$Q(u) = P_0 \, B_{0,2}(u) + P_1 \, B_{1,2}(u) + P_2 \, B_{2,2}(u)$$

$$x(u) = x_0 \, B_{0,2}(u) + x_1 \, B_{1,2}(u) + x_2 \, B_{2,2}(u)$$

$$y(u) = y_0 \, B_{0,2}(u) + y_1 \, B_{1,2}(u) + y_2 \, B_{2,2}(u)$$

$$x(u) = x_0 (1-u)^2 + x_1 \cdot 2 \cdot u(1-u) + x_2 u^2 = 4u^2 + 8u + 4$$

$$y(u) = y_0 (1-u)^2 + y_1 \, 2 \cdot u(1-u) + \cancel{x_2} \, y_2 u^2 = -10u^2 + 12u + 2$$

$$B_{0,2}(u) = (1-u)^2$$

$$B_{1,2}(u) = {}^{2}C_{1} \, u^1 (1-u)$$

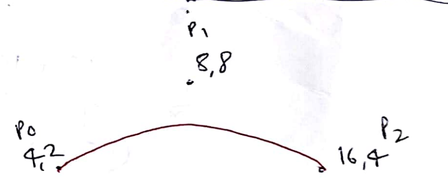
$$= \frac{2!}{1! \, 1!} u(1-u)$$

$$= 2u(1-u)$$

$$B_{2,2} = u^2$$

if

u = 0,	x(u)	y(u)	
u = 0	4	2	
u = 0.2	5.76	4.0	$-10(0.2)^2 + 12 \times 0.2 + 2$ $= -0.4 + 2.4 + 2$
u = 0.4	7.84	5.20	
u = 0.6	10.24	5.6	
u = 0.8	12.96	5.2	
u = 1	16	4	



→ Each Contr
→ Each point
parameter

x(u) =

→ Control p
→ Curve is

Basis function

Ex → n = 5 and

t_i $(0 \le i \le 8)$ ar

i = { 0, 0, 0, 1, 2,

$_{0,3}(u) = (1-u)^2 \, N_2$

Cohen Sutherland Cliping - Example

Q ⇒ Window is defined as A (10, 20), B (20,20), C = (20,10), D(10,10) find visible portion of line P (15, 15) and Q (15,5)

$Q(15,5) \rightarrow$

T	B	R	L
0	1	0	0

T → $y > y_{max}$

B → $y < y_{min}$

R → $x > x_{max}$

L → $x < x_{min}$

$P(15,15) \rightarrow 0000$

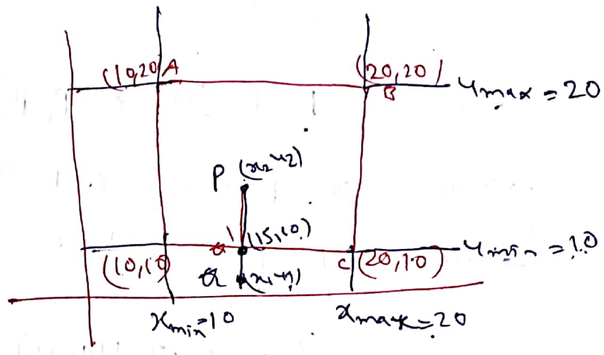
$P \,\&\, Q = 0 \rightarrow$

$x = x_1 + (y - y_1)/m$

$= 15 + (10-5)/\infty$

$\boxed{x = 15}$

$(15,10)$



$m = \frac{20}{0} = \infty$

Bezier Curve

Bezier curve use a constsuction curve, In which the interpolating polynomials depend on certain Control points

→ A Bezier curve is a parametric curve that uses Bernstein Polynomial as a basis function/ blending function.

Degree 'n' (order n+1) · Bezier Curve

$$Q(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u), \quad 0 \le u \le 1.$$

P_i → Control Point, $\quad B_{i,n}(u) = {}^{n}C_i \, u^i (1-u)^{n-i}$

$${}^{n}C_i = \frac{n!}{i!\,(n-i)!}$$

If $n = \overset{\text{degree}}{3}$ {4 Control Point}.

$$Q(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

$B_{0,3}(u) = {}^{3}C_0 \, u^0 (1-u)^3$

$= (1-u)^3$

$B_{1,3}(u) = 3u(1-u)^2$

$B_{2,3}(u) = 3u^2(1-u)$

$B_{3,3}(u) = u^3$

$$P(u) = Q(u) = P_0(1-u)^3 + P_1 3u(1-u)^2 + P_2 3u^2(1-u) + P_3 (u)^3$$

$$x(u) = x_0(1-u)^3 + x_1 3u(1-u)^2 + x_2 3u^2(1-u) + x_3 u^3$$

{$0 \le u \le 1$}

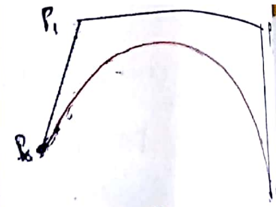
$0 \le u \le 1$

$u = 0.1$

$u = 0.2$

$u = 0.3$

$y(u) =$

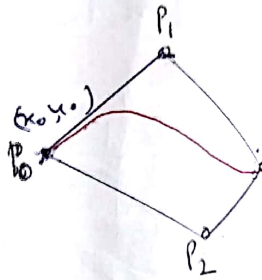


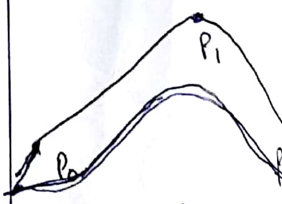
Cubic Bezier Curve

→ 4 Control Points

→ degree 3 {n...

Cont...





Quadratic Bezier Curve

Clipping

- Identify portion of a picture that is inside the viewport and removing the portion that is outside the view port

Cohen-Sutherland Line Cliping Algorithm

Algorithm:

1) Assign a 4 bit Code to each end point C_0, C_1

2) If $C_0 | C_1 = 0000$ — Completly accepted (Inside window)

else if $C_0 \& C_1 \neq 0000 \rightarrow$ rejected

else Clip

↓

If line crosses $x_{w min}$ or $x_{w max}$

$y = y_1 + m(x - x_1)$

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

here $x = x_{w min}$ or $x_{w max}$

else ($y_{w min}$ or $y_{w max}$)

$x = x_1 + (y - y_1)/m$

here $y = y_{w min}$ or $y_{w max}$

Verify -

$x_{w min} \le x \le x_{w max}$ }
$y_{w min} \le y \le y_{w max}$ } If not satisfy repeat clip

T B R L — Left

Top ↓ Right

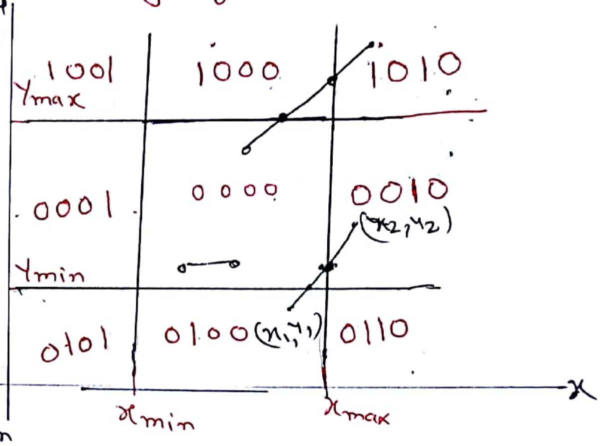
Bottom

Top - 1 if $y > y_{max}$

B - 1 if $y < y_{min}$

R - 1 if $x > x_{max}$

L is 1 if $x < x_{min}$

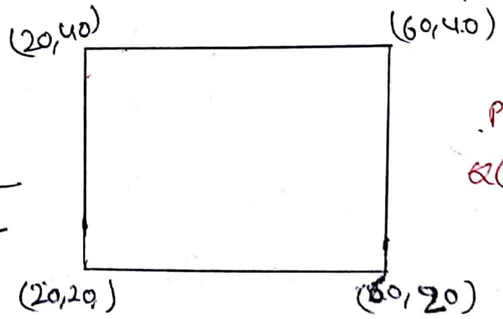


P - 1000

Q - 0010

AND

0000



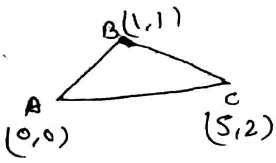
x y

P (40, 80)

Q (120, 30)

Q: Perform a 45° rotation of Triangle $A(0,0)$ $B(1,1)$ and $C(5,2)$

(i) about origin



$$P = \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$T_{R\theta} = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

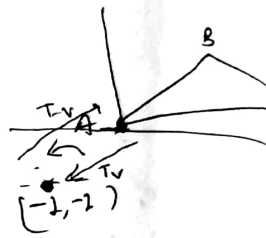
$$P' = T_{R\theta} \cdot P$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 0 & 0 & \frac{3\sqrt{2}}{2} \\ 0 & \sqrt{2} & \frac{7\sqrt{2}}{2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x'_A & x'_B & x'_C \\ y'_A & y'_B & y'_C \\ 1 & 1 & 1 \end{bmatrix}$$

$A' = (0,0)$ $B' = (0,\sqrt{2})$ $C' = \left(\frac{3\sqrt{2}}{2}, \frac{7}{\sqrt{2}}\right)$

(ii) about $P(-1,-1)$



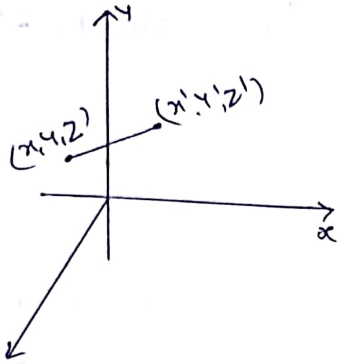
$$P' = T_V \, T_{R\theta} \, T_{-V} \cdot P$$

$$= \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$P' =$

3-D Transformation with Homogeneous Coordinates

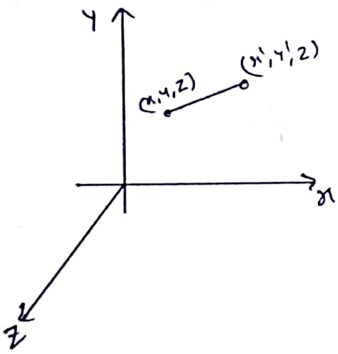
Translation



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

$$P' = T_t \cdot P$$

Scaling:-



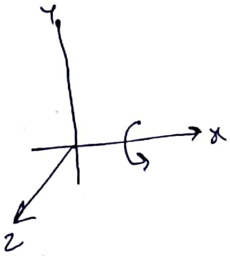
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = T_s \cdot P$$

Rotation:-

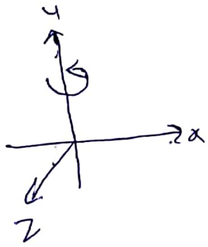
x-axis

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



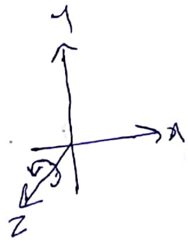
y-axis

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



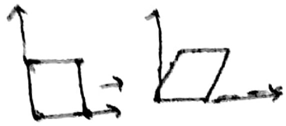
Z axis

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

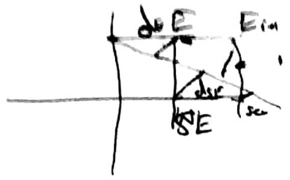


Shearing

$x' = x + Sh_x y$
$y' = y$



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



$d_E = f(x_i+1, y) = (x+1)^2 + y^2 - R^2$ (+ve)

$d_{SE} = f(x+1, y-1) = (x+1)^2 + (y-1)^2 - R^2$ (− ve)

if that is inside the circle ; < 0 −ve
outside ——— > 0 +ve

$d = d_{SE} + d_E = x$

if $d < 0$ — $\parallel E$

$d_{new} = (x+2)^2 + y^2 - R^2$

Δ

$= x^2 + 1 + 2x + y^2 - R^2$
$= x^2 + 1 + 2x + y^2 + 1 - 2y - R^2$
$= 2(x^2 + y^2 - R^2) + 2 + 4x + 1 - 2y$
$d = 3 + 4x - 2y$

$d = 3 - 2r$

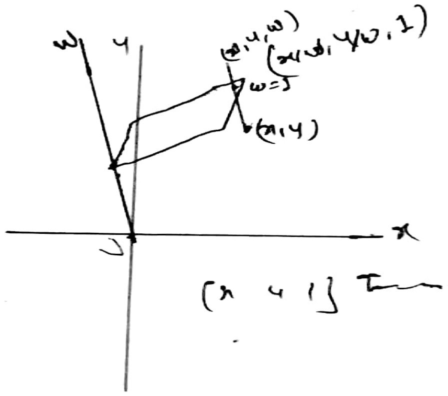
$d_{new} = d_E + d_{SE}$
$= (x+2)^2 + y^2 - R^2 + (x+2)^2 + (y-1)^2 - R^2$
$= x^2 + 4 + 4x + y^2 - R^2 + x^2 + 4 + 4x + y^2 + 1 - 2y - R^2$
$= 4 + 4x + 4 + 4x + 1 - 2y$
$= 8 + 8x + 1 - 2y$
$d_{new} = 9 + 8x - 2y$

$\Delta d_E = d_{new} - d_o = 9 + 8x - 2y - 3 - 4x + 2y$
$= 6 + 4x$

Homogenous Coordinates System

- Each position (x, y) is represented as $(x, y, 1)$
- All transformation can be represented as matrix multiplication
- Composite transformation becomes easier
- Unifying representation for transforming

(x, y, w) $\quad P_h = (4, 2, 2) = (8, 4, 4)$



$[x \;\; y \;\; 1]$ T

Translation: $T_t = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

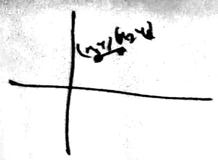
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$X = (x, y) \rightarrow X'(x', y')$

$(x \;\; y \;\; 1) \rightarrow (x' \;\; y' \;\; 1)$

Scaling $= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Rotation $= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$



$= T_t \times T_S \times \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{bmatrix}$

$=V_{ymax}, W_y \Rightarrow$

y_{min}

V_{ymin}

$00)$

$50)$

$200, 200)$

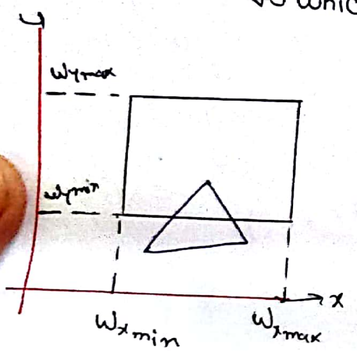
$(100, 100$

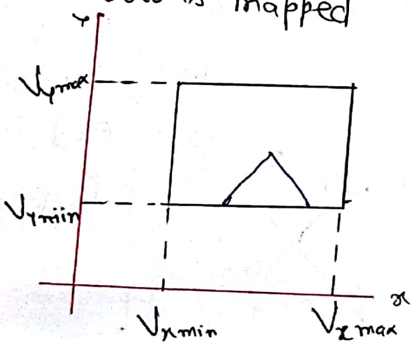
Window and View Port

Window Port: World coordinate area selected for display

→ Window is associated with the object rather than with image

View Port: An area on a display device to which window is mapped





window to view port transformation

$$f_x = \frac{V_{xmax} - V_{xmin}}{W_{xmax} - W_{xmin}}, \quad f_y = \frac{V_{ymax} - V_{ymin}}{W_{ymax} - W_{ymin}}$$

If ($f_x < f_y$)
then $f = f_x$
else $f = f_y$

Let $V_x = V_{xmax}$; $W_x = W_{xmax}$, $V_y = V_{ymax}$, W_y

$$f = \frac{V_x - V_{xmin}}{W_x - W_{xmin}}, \quad f = \frac{V_y - V_{ymin}}{W_y - W_{ymin}}$$

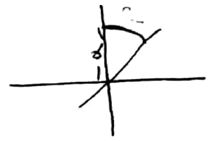
Ex Windowport (100, 100, 300, 300)
Viewport (50, 50, 150, 150)
Convert Windowport coordinate (200, 200) to view port
↓
(100, 100)

$$f = \frac{150-50}{300-100} = \frac{1}{2} = \frac{V_x - 50}{200-100}$$

$$\boxed{V_x = 100}$$

$$\frac{1}{2} = \frac{V_y - 50}{200-100} \Rightarrow V_y = 100$$

Bresenham Circle Algorithm



Circle equation - $x^2 + y^2 - R^2 = 0$

$F(x, y) = x_p^2 + y_p^2 - R^2 = 0$

$F(x_p, y_p) =$

$d_E = F(x_p+1, y_p) = (x_p+1)^2 + y_p^2 - R^2$ [+Ve]

$d_{SE} = F(x_p+1, y_p-1) = (x_p+1)^2 + (y_p-1)^2 - R^2$ [-Ve]

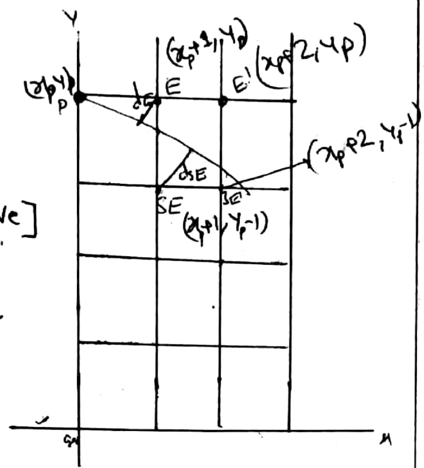
$d = d_E + d_{SE}$ // decision Parameter

If (d<0) - Select E // $d_{SE} > d_E$

$x_E = x_p + 1$, $y_E = y_p$

If (d>0) → Select SE // $d_E > d_{SE}$

$x_{SE} = x_p + 1$; $y_{SE} = y_p - 1$



$F(x,y) > 0$ → Outside

$F(x,y) < 0$ - Inside

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

$d = 3 + 4x - 2y$ // $d = 3 - 2R$, $x=0, y=R$

E

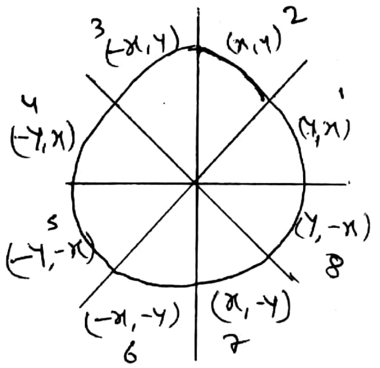
$d_{new} = (x_p+2)^2 + y_p^2 - R^2 + (x_p+2)^2 + (y_p-1)^2 - R^2$

$d^E_{new} = 9 + 8x - 2y$

$\Delta E = d^E_{new} - d = 6 + 4x$

$d^E = d + \Delta E$

$SE \Rightarrow d^{SE} = 4(x-y) + 10$



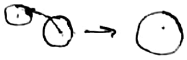
Algorithm:

```
x = 0; y = R;   P = 3 - 2r
Putpixel (x, y);
While ( y > x ) do
{  x++;
   If ( d<0 )
     d = d + 4*x + 6;
   else
     d = d + 4*(x-y) + 10; y--;
   end
   Putpixel (x, y);
endwhile
```

(x_c, y_c) Putpixel $(x_c + x, y_c + y)$ - 2

4 $(x_c + y, y_c + x)$ - 1

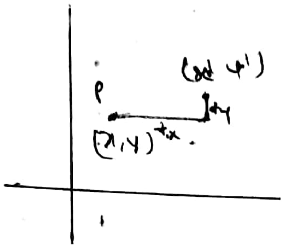
—— $(x_c + y, y_c - x)$ - 8



Transformation

2D Transformation

1) Translation



$x' = x + t_x$

$y' = y + t_y$

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

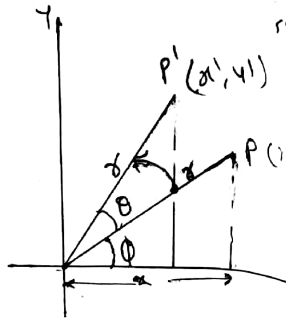
$[x' \ \ y'] = [x \ \ y] + [t_x \ \ t_y]$

$P = 2, 3$

$t_x = 4 \quad t_y = 3$

$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$

Rotation



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

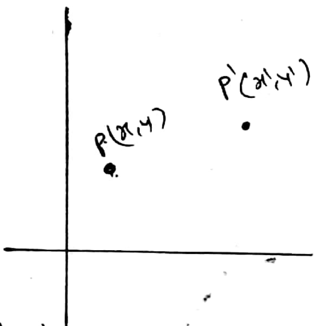
$y' = x\sin\theta + y\cos\theta$

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

2) Scaling

$x' = S_x x$

$y' = S_y y$



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

T_S

$P(2, 3)$

$S_x = 2 \quad S_y = 2$

line $T_S \cdot \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}$

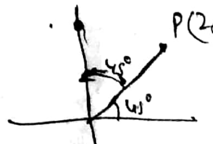
$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$

Proof-

$x' = r\cos(\theta + \phi) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$

$y' = r\sin(\theta + \phi) = r\cos\phi\sin\theta + r\sin\phi\cos\theta$

$x = r\cos\phi \quad , \quad y = r\sin\phi$



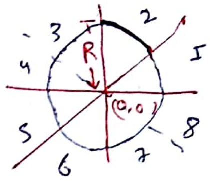
$x' = x\cos\theta - y\sin\theta$

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

$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos 45° & -\sin 45° \\ \sin 45° & +\cos 45° \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{bmatrix} \frac{2}{\sqrt{2}} - \frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} + \frac{2}{\sqrt{2}} \end{bmatrix}$

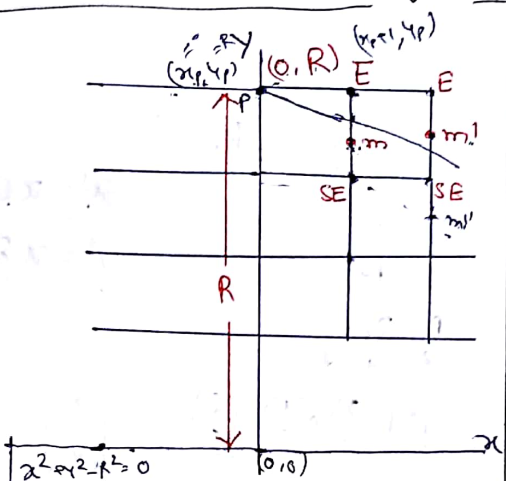
$y' = 2\sqrt{2}$

Mid Point Circle - Algorithm



Circle equation

$F(x,y) = x^2 + y^2 - R^2 = 0$



Evaluate F(m)

if $F(m) < 0$ (m inside circle)

Choose E

if $F(m) > 0$ (outside circle)

Choose SE

Decision: if Point (x_p, y_p)

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

$(y_p - \frac{1}{2})^2 = y_p^2 - 2 \cdot \frac{1}{2} y_p + \frac{1}{4} - r$

$d = (x_p+1)^2 + (y_p - \frac{1}{2})^2 - R^2 = 1 + \frac{1}{4} - R^2 \ldots R = \frac{5}{4} - R$

when E, $d < 0$

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

$d_{new} = (x_p+2)^2 + (y_p - \frac{1}{2})^2 - R^2$

$\Delta E = d_{new} - d = 2x_p + 3$

when SE selected:

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

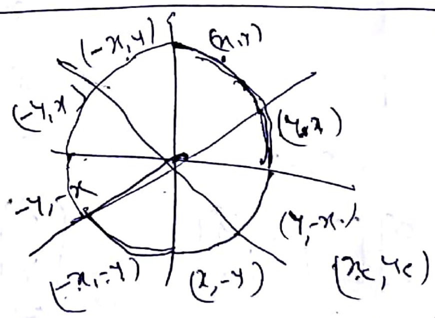
$\Delta SE = d_{new} - d = 2x_p - 2y_p + 5$

For a given point (x', y')

$F(x', y') = 0 \rightarrow$ On Circle

$F(x', y') > 0 \rightarrow$ Outside circle

F(m) $F(x', y') < 0 \rightarrow$ Inside Circle



Putpixel$(x_c + x, y_c + y)$

Algorithm

$x = 0$; $y = R$; $d = \frac{5}{4} - R$

Putpixel(x, y);

while(y > x) do

If (d < 0)

$d = d + \underbrace{2x + 3}_{\Delta E}$; $x = x + 1$;

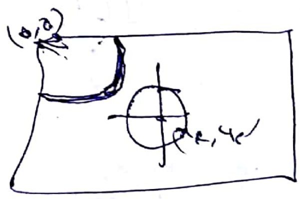
else

$d = d + \underbrace{2x - 2y + 5}_{\Delta SE}$; x++; y--;

end

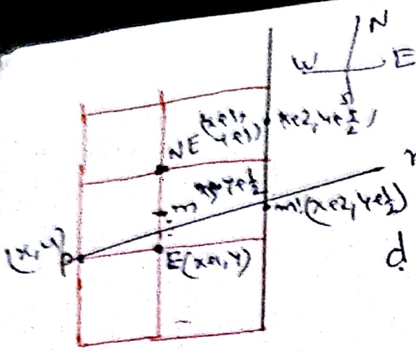
Putpixel (x, y);

end while → Putpixel(y, x);



Mid-Point Line Algorithm (Proof)

(x_1, y_1) (x_2, y_2)



$ax + by + c = 0 \quad -(i)$

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

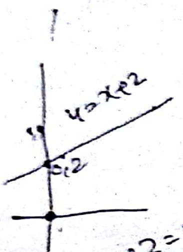
$d = a(x+1) + b(y+\frac{1}{2}) + c =$

$= ax + a + by + \frac{b}{2} + c = a + \frac{b}{2}$

$y = mx + c = \frac{dy}{dx} x + c'$

$x \cdot dy - y\,dx + c'dx = 0 \quad -(ii)$

$(\because a = dy,\ b = -dx,\ c = c'dx)$



$F \Rightarrow x - y + 2 = 0$

$F(0,0) = 0 - 0 + 2 > 0$

$F(0,2) = 0 - 2 + 2 = 0$

$F(0,3) = 0 - 3 + 2 = 0$
$-1 \le 0$

$F(x, y) > 0 \rightarrow$ below line

- $d < 0 \rightarrow E$
  $d > 0 \rightarrow NE$

$\rightarrow E \rightarrow (x+1, y)$

$d_{new} = F(x+2, y+\frac{1}{2}) = 2a + \frac{b}{2}$

$\Delta E = a = dy$

$\rightarrow NE\ (x+1, y+1)$

$\Delta NE = dy - dx$

Mid point Line Algorithm

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

$d = 2dy - dx,\ \Delta E = 2dy,$

$\Delta NE = 2(dy - dx);$

$x = x_1,\ y = y_1;$

Putpixel (x, y);

While $(x < x_2)$

if $(d \le 0)$

$d = d + \Delta E;\ x = x + 1$

else

$d = d + \Delta NE;\ x = x + 1;$

end

Putpixel (x, y);

end while

Digital Differential Analyzer (DDA) Line Algorithm

- This algorithm works on principal of obtaining the successive pixel values based on the differential equation.

(x_0, y_0) (x_1, y_1)

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

Case 1: $m < 1$

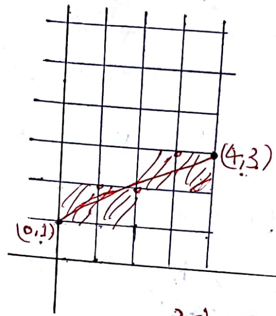
{

$x_n = x_0 + 1, \quad y_n = y_0 + m$

Plot (x_n, round(y_n))

$x_0 = x_n, \quad y_0 = y_n$

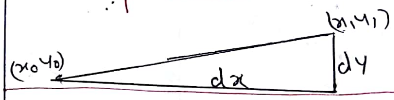
}



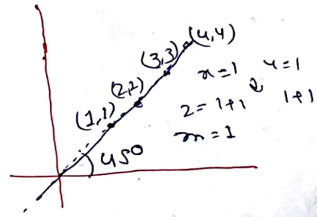
$m = \frac{3-1}{4-0} = \frac{2}{4} = 0.5$

Line $y = mx + c$
slope

$m < 1$



$m = 1$



Case 2: $m > 1$

$y_n = y_0 + 1, \quad x_n = x_0 + \frac{1}{m}$

Plot (round(x_n), y_n)

$y_0 = y_n, \quad x_0 = x_n$

Plot

x_n	round(y_n)	x_n	y_n
0	1	0	1
1	2	0+1 = 1	1+0.5 = 1.5
2	2	1+1=2	1.5+0.5 = 2
3	3	2+1=3	2+0.5 = 2.5
4	3	3+1=4	2.5+0.5 = 3

$m > 1$