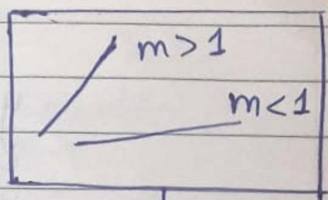
02/01/2018

Computer Graphics

2D on screen



Drawn L to R

#include <graphics.h>

Primitives: Line, Circle

LINE:

$$y = mx + c$$

for m>1

y changes more as compared to x.

$$x = y/m - c/m$$

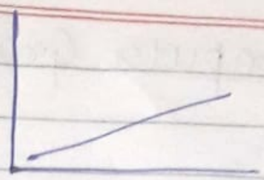
$$x_i = \frac{y_i}{m} - \frac{c}{m}$$

$$x_i + 1 = \frac{y_i + 1}{m} - \frac{c}{m} \qquad (y_{i+1} = y_i + 1)$$

$$x_i + 1 = \frac{y_i}{m} - \frac{c}{m} + \frac{1}{m}$$

$$\boxed{x_i + 1 = x_i + \frac{1}{m}} \qquad [y_{i+1} = y_i + 1]$$

for m < 1



Changes in x are more than those in y

$y = mx + c$

$y_i = mx_i + c$

$y_{i+1} = mx_{i+1} + c$

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

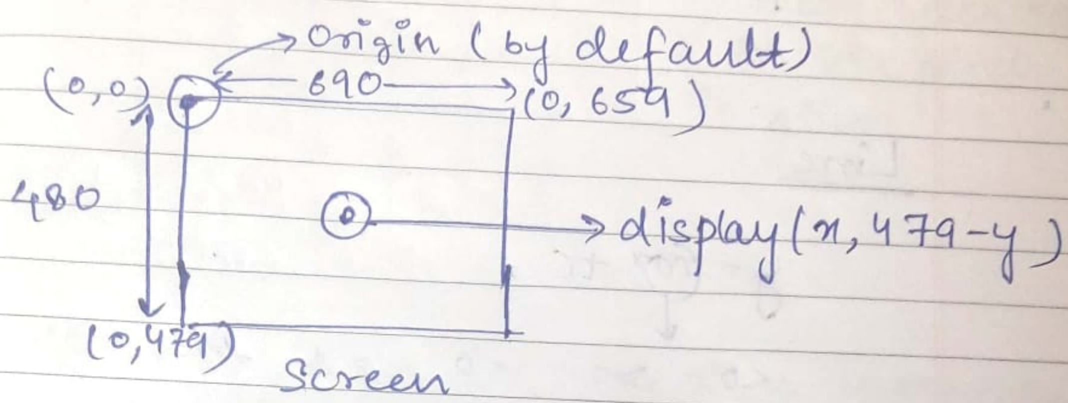
$y_i + 1 = mx_i + c + m$

$\boxed{y_i + 1 = y_i + m}$ $[x_{i+1} = x_i + 1]$

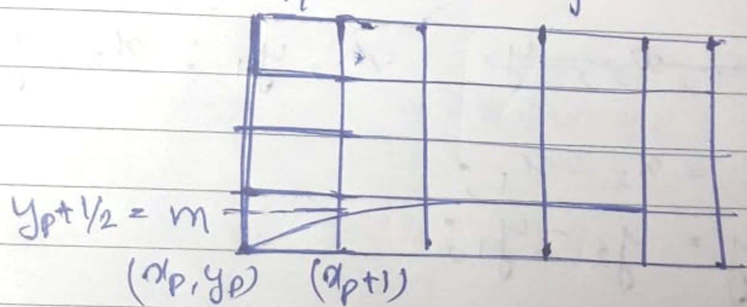
Algorithm to draw a line:

```
mline(x1, x2, y1, y2){
    dy = y2 - y1;
    dx = x2 - x1;
    m = dy/dx
  if(abs(dy) < abs(dx)) {
         x = x1, y = y1;
         display(x, y);
        while(x < x2){
              x++;
              y+ = m;
             display(x, y);
        }
  }

  if (abs(dy) ≥ abs(dx)) {
        x = x1, y = y1;
        display(x, y);
       while (y <= y2) {
```

```
y++;
x+ = 1/m;
display(x, y)
}
}
```



Another approach of line (Mid point Approach)



$ax+by+c = f(x.y)$

if $f(x,y) = 0$, then point lies on the line

$$f(x_p, y_p) = ax_p + by_p + c$$

deciding value

$$f(m) = a(x_p+1) + b(y_p + 1/2) + c$$

<u>Case 1:</u> if $f(m) = 0$, then lies on the line

<u>Case 2:</u> $f(m) > 0$

<u>Case 3:</u> $f(m) < 0$

04/01/2018

Primitive Drawing

→ Mathematical equation

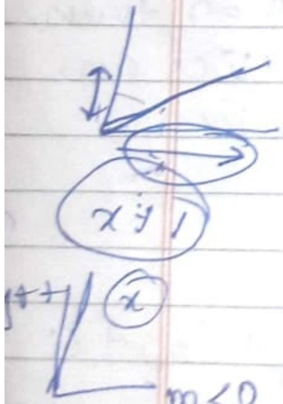
Line

$y = mx + c$

m: <0 >0 $=0$ $=1$ $=\infty$

Smallest unit: Pixel

Inputs: ~~x_1, x_2, y~~ x_1, y_1 ; x_2, y_2

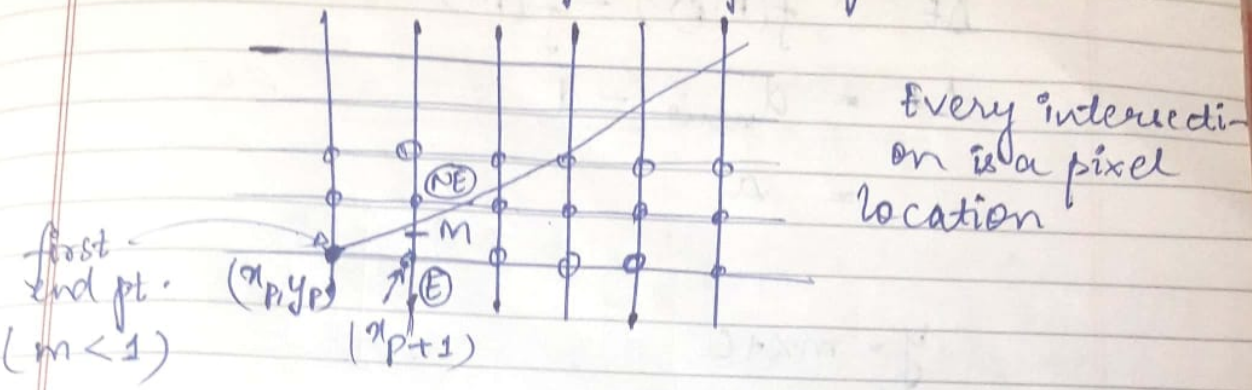



```
dx = x2 - x1;
dy = y2 - y1;

x = x1, y = y1;
putpixel (x, y, WHITE);
while (x <= x2)
{
    y += m;      → floating pt. & floating pt. computation takes more time.
    x++;
    put pixel (x, y, WHITE);
}
```

MidPoint Approach:

To avoid floating pt. computation

Screen is in the form of a grid



$\because$ $m<1$, $dy < dx$ $\Rightarrow$ x is incremented & accordingly y is computed (major movement in x).

$f(x,y) = ax + by + c$

<0 $\quad =0$ $\quad >0$

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

d (decision) $= f(M) = a(x_p + 1) + b(y_p + 1/2) + c$

keeps on updating depending on value of $f(M)$

if $d < 0$
selection is E

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

$d_{new} = a(x_p + 2) + b(y_p + 1/2) + c$

$\Delta E = f(M_E) - f(M)$

$\Delta E = d_{new} - d$

$= a$

$y = mx + c$

$\frac{dy}{dx}$ Replace m with dy/dx.

$dx \cdot y = dy \cdot x + dx \cdot c$

$f(x, y) = dy\, x - dx \cdot y + dx \cdot c = 0$

$\Downarrow$

(a)

$\Rightarrow \Delta E = a = dy$

else if $d > 0$

Selection is NE

$f(M_{NE}) = f(x_{p+2}, y_p + 3/2)$

$d_{new} = a(x_{p+2})$

$+ b(y_p + 3/2) + C$

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

$\Delta NE = f(M_{NE}) - f(M)$

$\Delta NE = a + b$

$\Rightarrow$ $\underline{a + b = dy - dx}$

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

$= ax_p + a + by_p + b/2 + C$

$= \underbrace{ax_p + by_p + C}_{\Downarrow} + a + b/2$

This pt. is on line ; so it will be 0.

$$\boxed{d_{in} = a + b/2}$$

To remove 1/2 from b/2, multiply it with a scalar qty, that is, 2.

$f(x, y) = 2(ax + by + c)$

$d_{in} = 2a + b = \underline{2dy - dx}$

If $d < 0$, $\Delta E = 2dy = 2a$.

else $\Delta NE = 2(dy - dx)$
$= 2(a + b)$

~~$d = f(x_p + 1,$~~

Changes in previous code:

```
d = ~~2*(dy-dx);~~  2*dy - dx;
while (x <= x2)
{
    if d<0
        d += 2*dy;
    else
      {d += 2*(dy-dx);
        y++;
      }
}
    x++;
    putpixel(x, y, WHITE);
}
```

Example:

$x_1 = 10$
$y_1 = 10$
$x_2 = 20$
$y_2 = 18$

$m < 1$

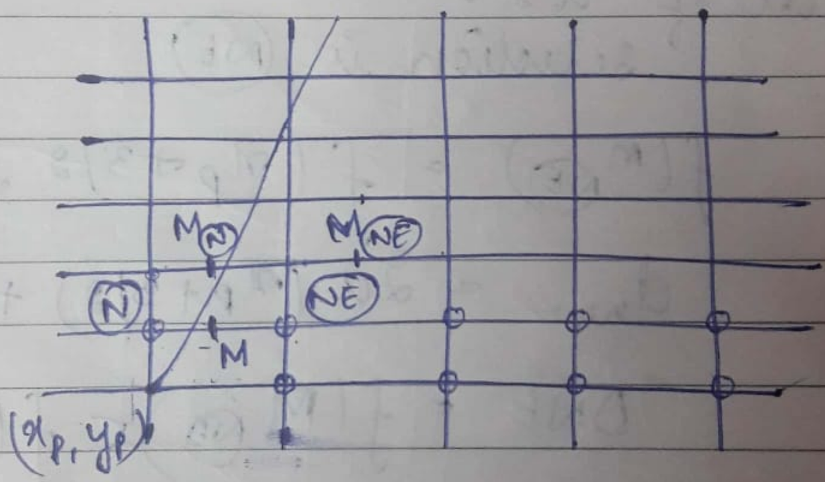
$\cancel{dy} = \cancel{8}$
$\cancel{dx} = \cancel{10}$
$x = 10\ (x_1)$
$y = 10\ (y_1)$

Check Table

x_1	y_1	d
10	10	6
11	11	2
12	12	-2
13	13	14

; complete the rest of the table

For $m>1$:



At (x_p, y_p):

$$f(x_p, y_p) = 2(ax_p + by_p + c)$$

$$d = f(M) = f(x_p + 1/2 + y_p + 1)$$

$$d = 2a(x_p + 1/2) + 2b(y_p + 1) + 2c$$

if $d < 0$

Selection is N

$f(M_{N}) = f(x_p + 1/2, y_p + 2)$

$d_{new} = 2a(x_p + 1/2) + 2b(y_p + 2) + 2c$

$\Delta N = f(M_{N}) - f(M)$

$\Delta N = d_{new} - d$

$\Delta N = 2b = -2dx.$

else if $d \geq 0$

selection is NE

$f(M_{NE}) = f(x_p + 3/2, y_p + 2)$

$d_{new} = 2a(x_p + 3/2) + 2b(y_p + 2) + 2c$

$\Delta NE = f(M_{NE}) - f(M)$

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

$\Delta NE = 2a + 2b = 2(dy - dx)$

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

$d = ax_p + a/2 + by_p + b + c$

$d = \underbrace{ax_p + by_p + c}_{=0} + a/2 + b$

$d = a/2 + b$

$f(x,y) = 2(ax + by + c)$

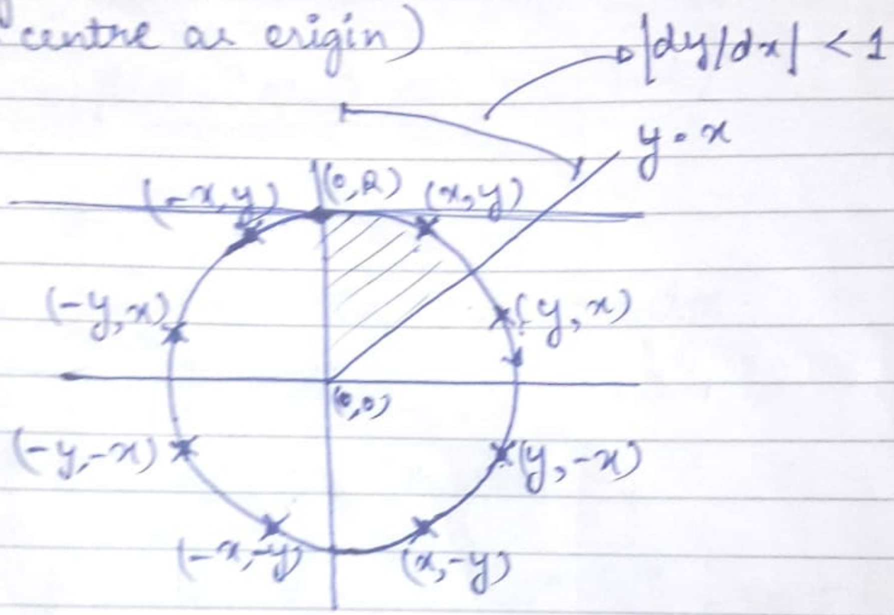
$d_{in} = a + 2b = dy - 2dx$

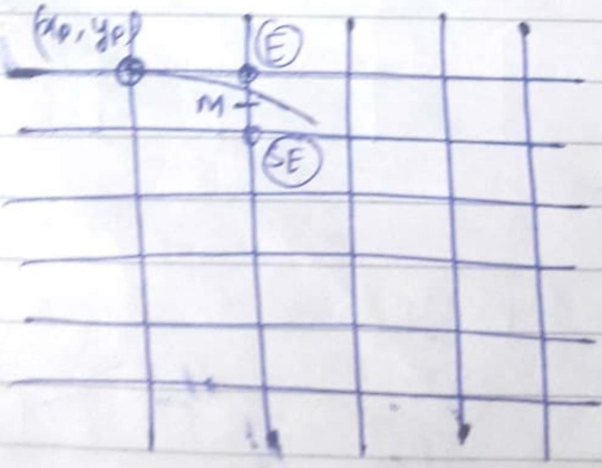
08/01/2018

Scan Conversion

Primitive Circle:

$x^2 + y^2 = R^2$
(with centre as origin)





At (x_p, y_p)

$$f(x_p, y_p) = x_p^2 + y_p^2 - R^2$$

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

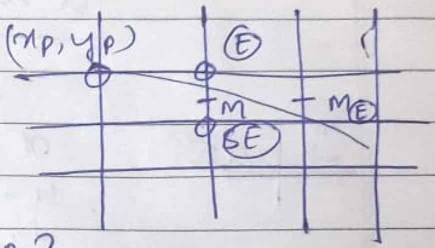
$= (x_p+1)^2 + (y_p-1/2)^2 - R^2$

<u>1^{st} differential:</u>

if $d<0$

selection is E (assumed)

$d_{new} = f(M_E)$



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

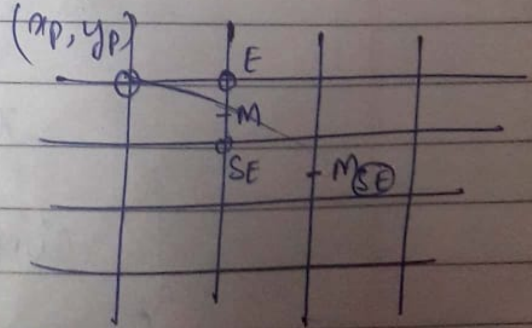
$= (x_p+2)^2 + (y_p-1/2)^2 - R^2$

$\Delta E = d_{new} - d = \boxed{3+2x_p}$ (2 degree circle reduced to degree 1)

else

selection is SE

$d_{new} = f(M_{SE})$



$= f(x_p+2, y_p-3/2)$

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

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

initialization:

$$d = \cancel{(x_p^2 + y_p^2 - R^2)}^{\to 0} + 2x_p + 1 - y_p + 1/4$$

$$d = 2x_p - y_p + 5/4$$

$\boxed{d = 5/4 - R}$ (starting from (0,R))

$\boxed{d \approx 1-R}$

Inputs:
$x = 0;$
$y = R;$
$d = 1-R;$

8 times putpixel

```
Circle Symmetry (x,y);
while ( x <= y)
{
  if d<0
     d+= 3+2*x;
  else
  { d+ = 2(x-y) +5;
    y--;
  }
  x++;      ( becoz of major movement in x)
circleSymmetry (x,y);
}
```

$R = 8$

x	y	d
0	8	-7
1	8	-4
2	8	1
3	7	-6
4	7	3
5	6	2
6	5	5

* Rather ~~for~~ going for 1st diff., we go for 2nd diff. becoz time taken for multiplication is more than ~~arithmetic~~ addition so arithmetic computation time can be reduced

Pre requisites for 2nd differential:

At (x_p, y_p), 1st differential are:

$\Delta E = 2x_p + 3$

$\Delta SE = 2(x_p - y_p) + 5$

If $d < 0$
selection is E

At E : $(x_p + 1, y_p)$

$\Delta E_{new} = 2x_p + 5$

$\Delta SE_{new} = 2(x_p - y_p) + 7$

$\boxed{\Delta^2 E^{(E)} = 2}$ $(\Delta E_{new} - \Delta E)$ for the selection of E

$\boxed{\Delta^2 SE^{(E)} = 2}$ $(\Delta SE_{new} - \Delta SE)$

else
selection is SE

At SE : $(x_p + 1, y_p - 1)$

$\Delta E_{new} = 2x_p + 5$

$\Delta SE_{new} = 2(x_p - y_p) + 9$

$\boxed{\Delta^2 E^{(SE)} = 2}$

$\boxed{\Delta^2 SE^{(SE)} = 4}$

Now, we have to initialise ΔE & ΔSE.

$$\left.\begin{aligned} d &= 1 - R \\ \Delta E &= 3 \\ \Delta SE &= 5 - 2R \end{aligned}\right\} \text{Starting from } (0, R)$$

Changes in previous code:

```
x = 0;
 y = R;
deltaE = 3;
deltaSE = 5-2R;

circle symmetry (x, y);
while (x <= y)   -> for 1st octant only
{
  if d<0
  {
        d+ = delta E;
        deltaE + = 2;
        deltaSE+ = 2;
  }

  else
  {
      d+ = deltaSE;
      delta E + = 2;
      deltaSE + = 4;   y--;
  }
  x++;
}  circle symmetry (x,y);
```

R = 8

x	y	d	deltaE	deltaSE
0	8	-7	3	-11
1	8	-4	5	-9
2	8	1	7	-7
3	7	-6	9	-3
4	7	3	11	-1
5	6	2	13	3
6	5	5	15	7

$\text{grad}(E) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j$. If $\partial f/dx = \partial f/dy$ then $|dy/dx| = 1$

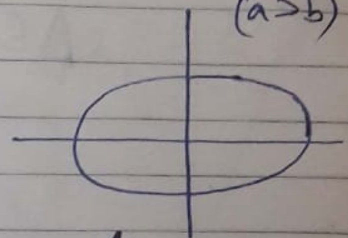
11/01/2018

Scan Conversion

Ellipse:

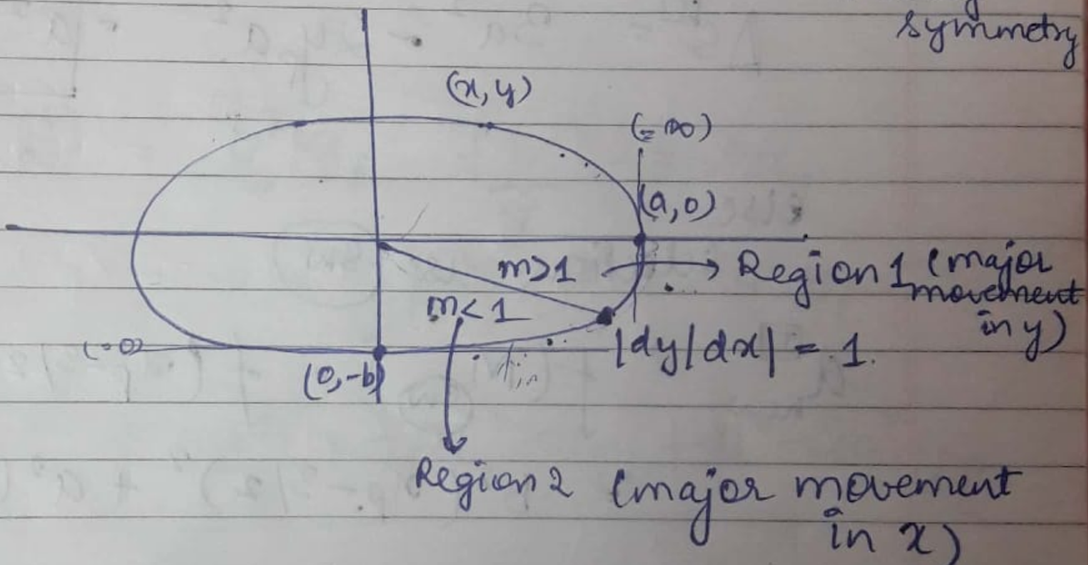
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

(a>b)



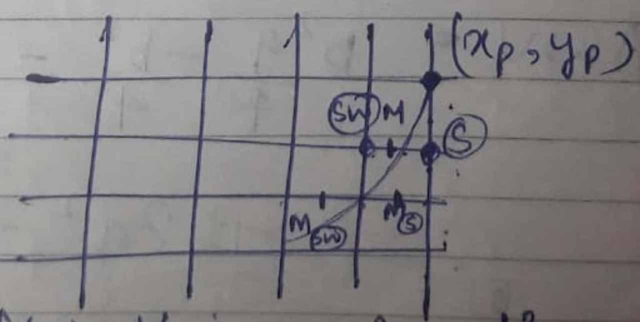
4-way symmetry

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



I^st Derivative:

Region 1:



$d^{\boxed{I}} = f(M) = f(x_p - 1/2, y_p - 1) = b^2(x_p - 1/2)^2 + a^2(y_p - 1)^2 - a^2b^2$

If $d < 0$

Selection is S

$d^{\boxed{I}}_{new} = f(M_S) = f(x_p - 1/2, y_p - 2)$

$d_{new}^{\boxed{I}} = b^2(x_p - 1/2)^2 + a^2(y_p - 2)^2 - a^2b^2$

$\Delta S = d_{new}^{\boxed{I}} - d^{\boxed{I}}$

$\Delta S = \cancel{a^2 y_p^2} + 4a^2 - 4y_p a^2 - \cancel{a^2 y_p^2}$
$- a^2 + 2y_p a^2$

$\Delta S^{\boxed{I}} = 3a^2 - 2y_p a^2 = \boxed{a^2(3 - 2y_p)}$

else
selection is SW

$d_{new}^{\boxed{I}} = f(M_{SW}) = f(x_p - 3/2, y_p - 2)$
$= b^2(x_p - 3/2)^2 + a^2(y_p - 2)^2 - a^2b^2$

$\Delta SW = d_{new}^{\boxed{I}} - d^{\boxed{I}}$

$\Delta SW = b^2\frac{9}{4} - \frac{b^2}{4} - b^2 3x_p + b^2 x_p$

$+ 3a^2 - 2y_p a^2$

$\Delta SW^{\boxed{I}} = 2b^2 - 2b^2 x_p + 3a^2 - 2y_p a^2$

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

Starting from $(a, 0)$ (Clockwise)

$$d_{in}^{\boxed{I}} = b^2(x_p - 1/2)^2 + a^2(y_p - 1)^2 - a^2b^2$$

$$d_{in}^{\boxed{I}} = b^2(a - 1/2)^2 + a^2 - a^2b^2$$

$$d_{in}^{\boxed{I}} = a^2b^2 + \frac{b^2}{4} - ab^2 + a^2 - a^2b^2$$

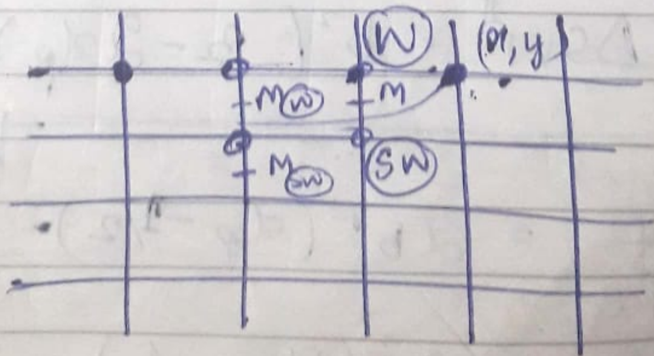
$$d_{in}^{\boxed{I}} =$$

$$d_{in}^{\boxed{I}} = b^2x_p^2 + \frac{b^2}{4} - b^2x_p + a^2y_p^2 + a^2 - 2a^2y_p - a^2b^2$$

$$d_{in}^{\boxed{I}} = \frac{b^2}{4} - b^2x_p + a^2 - 2a^2y_p$$

$$\boxed{d_{in}^{\boxed{I}} = \frac{b^2}{4} - b^2a + a^2}$$

Region II :



$$d^{\boxed{II}} = f(M) = f(x_p - 1, y_p - 1/2)$$

$$d^{\boxed{II}} = b^2(x_p - 1)^2 + a^2(y_p - 1/2)^2 - a^2b^2$$

Selection is W

$$d_{new}^{\boxed{II}} = f(M_W) = f(x_p - 2, y_p - 1/2)$$

$$d_{new}^{\boxed{II}} = b^2(x_p - 2)^2 + a^2(y_p - 1/2)^2 - a^2b^2$$

$$\Delta W^{II} = d_{new}^{\boxed{II}} - d^{\boxed{II}} = 3b^2 - 2b^2 x_p$$

$$\boxed{\Delta W^{II} = b^2(3 - 2x_p)}$$

Selection is SW

$$d_{new}^{\boxed{II}} = f(M_{SW}) = f(x_p - 2, y_p - 3/2)$$

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

$$\Delta SW^{II} = d_{new}^{\boxed{II}} - d^{\boxed{II}}$$

$$\boxed{\Delta SW^{II} = b^2(3 - 2x_p) + 2a^2(1 - y_p)}$$

$$\boxed{\begin{aligned} \frac{\partial f}{\partial x} &= 2b^2(x_p - 1/2) \\ \frac{\partial f}{\partial y} &= a^2(y_p - 1) \end{aligned}}$$

II Derivative :

Prerequisites → 1^{st} derivative

Region I :

At (x_p, y_p) : 1^{st} differential are :

$$\Delta S = a^2(3-2y_p)$$
$$\Delta SW = a^2(3-2y_p) + b^2(2-2x_p)$$

Selection is (S) :

At (S) : $(x_p, y_p - 1)$

$$\Delta S_{new} = a^2(3-2(y_p-1))$$
$$\Delta S_{new} = a^2(3-2y_p+2)$$
$$\Delta S_{new} = a^2(5-2y_p)$$
$$\Delta SW_{new} = a^2(3-2(y_p-1)) + b^2(2-2x_p)$$
$$\Delta SW_{new} = a^2(5-2y_p) + b^2(2-2x_p)$$

$$\boxed{\begin{aligned} \Delta^2 S_{I}^{S} &= \Delta S_{new} - \Delta S = 2a^2 \\ \Delta^2 SW_{I}^{S} &= \Delta SW_{new} - \Delta SW = 2a^2 \end{aligned}}$$

Selection is SW:

At SW : $(x_p - 1, y_p - 1)$

$$\Delta S_{new} = a^2(3 - 2(y_p - 1))$$
$$\Delta S_{new} = a^2(5 - 2y_p)$$

$$\Delta SW_{new} = a^2(3 - 2(y_p - 1)) + b^2(2 - 2(x_p - 1))$$

$$\Delta SW_{new} = a^2(5 - 2y_p) + b^2(4 - 2x_p)$$

$$\boxed{\begin{aligned} \Delta^2 S^{SW}_{I} &= \Delta S_{new} - \Delta S = 2a^2 \\ \Delta^2 SW^{SW}_{I} &= \Delta SW_{new} - \Delta SW = 2a^2 + 2b^2 \end{aligned}}$$

Region II:

At (x_p, y_p): 1st differential are:

$$\Delta W = b^2(3-2x_p)$$
$$\Delta SW = b^2(3-2x_p) + 2a^2(1-y_p)$$

Selection is W:

At W: $(x_p - 1, y_p)$

$$\Delta W_{new} = b^2(3-2(x_p-1))$$
$$\Delta W_{new} = b^2(5-2x_p)$$
$$\Delta SW_{new} = b^2(3-2(x_p-1)) + 2a^2(1-y_p)$$
$$\Delta SW_{new} = b^2(5-2x_p) + 2a^2(1-y_p)$$
$$\Delta^2 W^{W}_{II} = \Delta W_{new} - \Delta W$$

$$\boxed{\Delta^2 W^{W}_{II} = 2b^2 \qquad \Delta^2 SW^{W}_{II} = \Delta SW_{new} - \Delta SW = 2b^2}$$

Selection is SW:

At SW: $(x_p - 1, y_p - 1)$

$\Delta W_{new} = b^2(3 - 2(x_p - 1)) = b^2(5 - 2x_p)$

$\Delta SW_{new} = b^2(3 - 2(x_p - 1)) + 2a^2(2 - y_p + 1)$

$\Delta SW_{new} = b^2(5 - 2x_p) + 2a^2(2 - y_p)$

$$\boxed{\begin{aligned} \Delta^2 W^{SW}_{II} &= \Delta W_{new} - \Delta W = 2b^2 \\ \Delta^2 SW^{SW}_{II} &= \Delta SW_{new} - \Delta SW = 2b^2 + 2a^2 \end{aligned}}$$

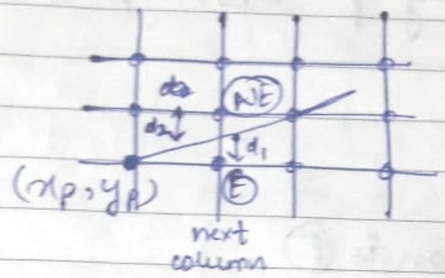
- Difference in measurable qty. for a primitive should either be +ve or -ve

15/01/2018

Bresenham's Approach

- Find a measurable qty
- Difference in measurable qty for decision making.

LINE:



$m < 1$

At (x_p, y_p): $y_p = mx_p + c$

→ For every primitive, there is a unique mathematical equation & unique measurable quantity.

→ Identify a measurable qty. for every primitive

→ Here, measurable qty is a distance.

At $(x_p + 1)$: $y = m(x_p + 1) + c$

$$d_1 = y - y_p \quad — ①$$

$$d_2 = (y_p + 1) - y \quad — ②$$

$$d = d_1 - d_2 = 2y - y_p - y_p - 1$$

$$\boxed{d = 2[m(x_p+1) + c] - 2y_p - 1}$$

→ floating pt. computation (because of m)

$$d.\Delta x = 2[\Delta y(x_p+1) + c.\Delta x] - 2y_p\Delta x - \Delta x$$

$$d = 2\Delta y x_p + 2\Delta y - 2y_p\Delta x - \Delta x + 2c\Delta x$$

$$d = 2\Delta y x_p - 2\Delta x y_p + [\underbrace{2\Delta y - \Delta x - 2c.\Delta x}_{\text{constant}}]$$

$$\boxed{d = 2\Delta y x_p - 2\Delta x y_p + C} \quad -③$$

if $d < 0$

Selection is E

$$d_{new} = 2\Delta y (x_p + 1) - 2\Delta x y_p + C$$

$$\boxed{\Delta E = d_{new} - d = 2\Delta y}$$

else

selection is NE

$$d_{new} = 2\Delta y (x_p + 1) - 2\Delta x (y_p + 1) + C$$

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

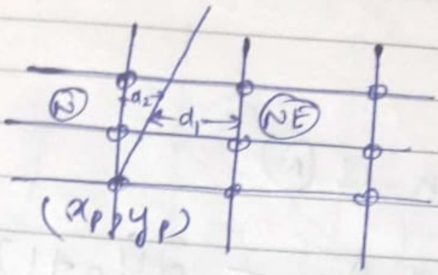
For d_{in} :

$$d = 2[m(x_p + 1) + c] - 2y_p - 1$$

$$d = 2\Delta x[\cancel{m x_p - y_p + c}^{\,0}] + (2\Delta y - \Delta x)$$

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

For $m > 1$:



$$d_1 = \cancel{x}\ (x_p + 1) - x$$

$$d_2 = \cancel{x_p - x}\ \ x - x_p$$

At (y_p+1) $\longrightarrow$ $x = \frac{1}{m}\left[(y_p+1) - C\right]$

$y = mx + C$

$\frac{y_p - C}{m} = x_p$ (At x_p, y_p)

$$d = d_1 - d_2 = x_p + 1 - x - x + x_p$$

$$d = 2x_p - 2x + 1$$

$$d = 2x_p - 2\left[\frac{1}{m}\left[(y_p+1) - c\right]\right] + 1$$

$$d \cdot \Delta y = 2x_p\Delta y - 2\cdot\left[(y_p+1)\Delta x - c\Delta x\right] + \Delta y$$

$$d = 2x_p\Delta y - 2y_p\Delta x + \left[-2\Delta x + 2c\Delta x + \Delta y\right]$$

$$\boxed{d = 2x_p\Delta y - 2y_p\Delta x + C}$$

if $d < 0$

selection is NE

$$d_{new} = 2(x_p+1)\Delta y - 2(y_p+1)\Delta x + C$$

16/01

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

else

selection is (N)

$$d_{new} = 2x_p \Delta y - 2(y_p + 1)\Delta x + C$$

$$\boxed{\Delta N = d_{new} - d = -2\Delta x}$$

For d_{in}: $\Delta y - 2\Delta x$

$$\cancel{d\Delta y = 2x_p \Delta y - 2y_p \Delta x - 2\Delta x + 2c\Delta x + \Delta y}$$

$$\cancel{d = 2\left[x_p - \frac{y_p}{m}\right.}$$

~~Calculate d_{in}.~~

$$d_i = 2x_p + 1 - 2x$$

$$= 2x_p + 1 - 2\left[\frac{1}{m}[y_p + 1] - c\right]$$

$$= 1 + 2\left[x_p - \cancel{\frac{1}{m}y_p + c} - \frac{1}{m}\right]$$

$$= 1 - \frac{2}{m} = 1 - \frac{2\Delta x}{\Delta y}$$

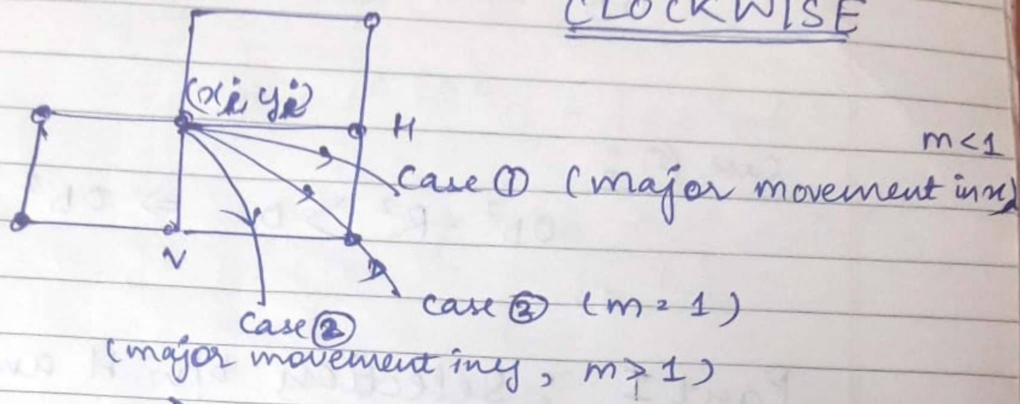
$$\boxed{d_{in} = \Delta y - 2\Delta x}$$

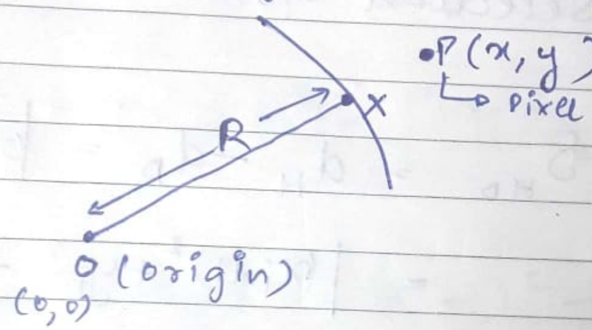
Scanned by CamScanner

16/01/2018

CIRCLE : $(x^2+y^2=R^2)$

CLOCKWISE





$$d_p = |OP^2 - OX^2|$$

$$d_p = |(x^2+y^2) - R^2|$$

Measurable qty for a circle.

$$d_H = |(x_i+1)^2 + y_i^2 - R^2|$$

$$d_D = |(x_i+1)^2 + (y_i-1)^2 - R^2|$$

$$d_V = |x_i^2 + (y_i-1)^2 - R^2|$$

Case ①:

$$OD^2 - R^2 < 0 \Rightarrow OD^2 < R^2$$

Case ②:

$$OD^2 - R^2 > 0 \Rightarrow OD^2 > R^2$$

Part I : Selection b/w H and D

$$\delta_{HD} = d_H - d_D = |OH^2 - R^2| - |OD^2 - R^2|$$

$$= |(x_i+1)^2 + y_i^2 - R^2| - |(x_i+1)^2 + (y_i-1)^2 - R^2|$$

$\because OD^2 < R^2$
$OH^2 > R^2$

$$\delta_{HD} = (x_i+1)^2 + y_i^2 - R^2 - [R^2 - (x_i+1)^2 - (y_i-1)^2]$$

$$\delta_{HD} = \underbrace{2[(x_i+1)^2 + (y_i-1)^2 - R^2]} + 2y_i - 1$$

$$\delta_{HD} = \Delta D_i + 2y_i - 1$$

Part II: Selection b/w V & D

$$\delta_{VD} = d_V - d_D$$

$$= |OV^2 - R^2| - |OD^2 - R^2|$$

$$\because OD^2 > R^2 \ \& \ OV^2 < R^2$$

$$\delta_{VD} = 2R^2 - OV^2 - OD^2$$

$$= 2R^2 - [x_i^2 + (y_i - 1)^2 + (x_i + 1)^2 + (y_i - 1)^2]$$

$$= 2[R^2 - (x_i+1)^2 + (y_i - 1)^2] + 2x_i + 1$$

(1) Pixel H is chosen

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

$$\Delta D_i = (x_i + 1)^2 + (y_i - 1)^2 - R^2 \quad \text{— (1)}$$

$$\Delta D_{i+1} = (x_i + 2)^2 + (y_i - 1)^2 - R^2 \quad \text{— (2)}$$

(2) − (1)

$$\boxed{\Delta D_{i+1} = \Delta D_i + 2x_i + 3}$$

(2) Pixel V is chosen

$$x_{i+1} \rightarrow x_i$$
$$y_{i+1} \rightarrow y_i - 1$$

$$\Delta D_{i+1} = (x_i + 1)^2 + (y_i - 2)^2 - R^2 \quad -(3)$$

(3) − (1)

$$\boxed{\Delta D_{i+1} = \Delta D_i - 2y_i + 3}$$

(3) Pixel D is chosen

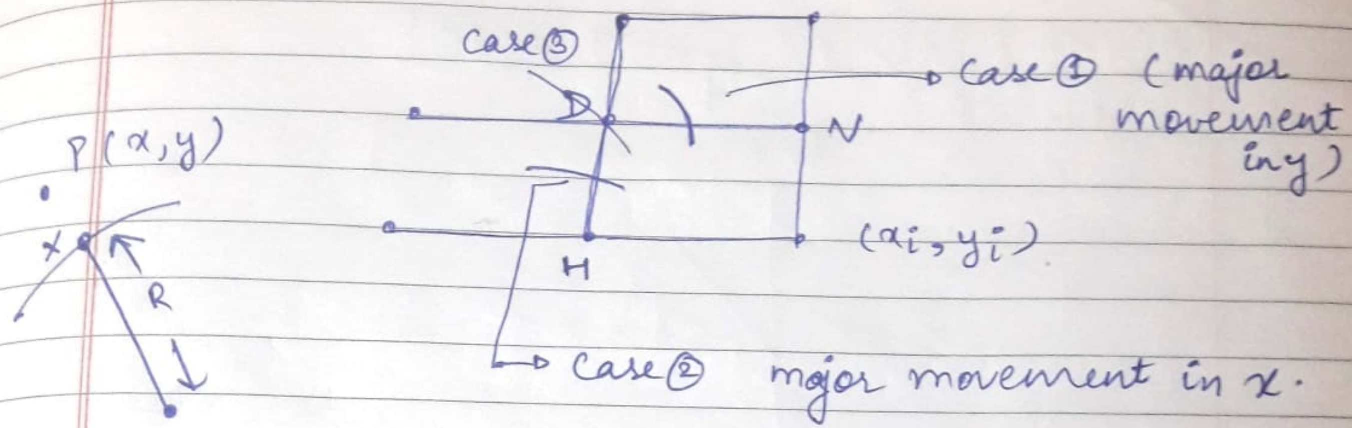
$$x_{i+1} \rightarrow x_i + 1$$
$$y_{i+1} \rightarrow y_i - 1$$

$$\Delta D_{i+1} = (x_i + 2)^2 + (y_i - 2)^2 - R^2$$

$$\boxed{\Delta D_{i+1} = \Delta D_i + 2(x_i - y_i) + 6}$$

$$\Delta D = 2(1 - R)$$ (Starting from 0, R)

ANTICLOCKWISE : (Starting from R,0)



$$d_p = |OP^2 - OX^2|$$

$$d_p = |(x^2 + y^2) - R^2|$$

$$d_V = |x_i^2 + (y_i + 1)^2 - R^2|$$

$$d_D = |(x_i - 1)^2 + (y_i + 1)^2 - R^2|$$

$$d_H = |(x_i - 1)^2 + y_i^2 - R^2|$$

complete

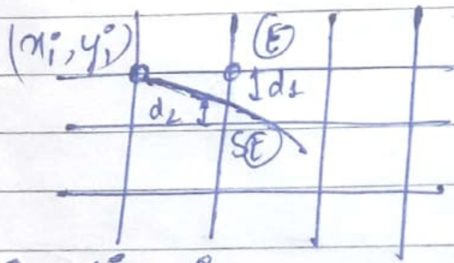
18/01/2018

Bresenhem's Approach

Circle (with 1st quadrant):

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

1st octant : major movement in x.



1st derivative:

For column position $(x_i + 1)$:

$$d_1 = y_P^2 - y^2 \quad (\text{w.r.t } E)$$

$$d_2 = y^2 - (y_P - 1)^2 \quad (\text{w.r.t. } SE)$$

$$y^2 = R^2 - (x_P + 1)^2$$

$D = d_1 - d_2 \longrightarrow$ Taking diff. in measurable qty.

$D = y_p^2 - y^2 - y^2 + (y_p - 1)^2$

$D = y_p^2 - 2y^2 + y_p^2 + 1 - 2y_p$

$D = 2y_p^2 - 2y^2 - 2y_p + 1$

$$\boxed{D = 2y_p^2 - 2(R^2 - (x_p+1)^2) - 2y_p + 1}$$

If $D < 0$

Selection is E

$x_{p+1} \rightarrow x_p + 1$

$y_{p+1} \rightarrow y_p$

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

$\cancel{D_{new} = 2y_p^2 - 2R^2}$

$\Delta E = D_{new} - D = 2(R^2 - (x_p+2)^2) - 2(R^2 - (x_p+1)^2)$

$\Delta E = 2[\cancel{R^2} - (x_p+2)^2 - \cancel{R^2} + (x_p+1)^2]$

$\Delta E = 2[-3 - 2x_p] = -2[3 + 2x_p]$

$$\boxed{\Delta E = 2[2x_p + 3]}$$

else if $D \geq 0$
selection is SE

$$x_{p+1} \rightarrow x_p + 1$$

$$y_{p+1} \rightarrow y_p - 1$$

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

$$\Delta SE = D_{new} - D$$

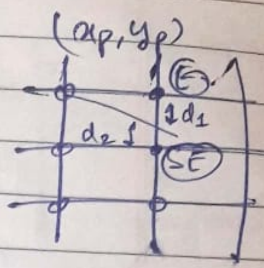
$$\boxed{\Delta SE = 2(2x_p + 3) - 4(y_p - 1)}$$

For D_{in} :

$$D_{(0,R)} = 2R^2 - 2(R^2 - 1) - 2R + 1$$

$$\boxed{D_{in} = 3 - 2R}$$

2nd Derivative:



$\Delta E = 2(2x_p+3)$

$\Delta SE = 2(2x_p+3) - 4(y_p-1)$

(P, R)

→ $\Delta E^i = 6$

→ $\Delta SE^i = 10 - 4R$

If $D<0$

selection is E

$x_{p+1} \rightarrow x_p+1$

$y_{p+1} \rightarrow y_p$

$\Delta E_{new} = 2(2x_p+5)$

$\Delta SE_{new} = 2(2x_p+5) - 4(y_p-1)$

$\Delta^2 E_{E} = \Delta E_{new} - \Delta E$

$\boxed{\Delta^2 E_{E} = 4}$

$\boxed{\Delta^2 SE_{E} = \Delta SE_{new} - \Delta SE = 4}$

if $D >= 0$

selection is SE

$x_{p+1} \longrightarrow x_p + 1$

$y_{p-1} \longrightarrow y_p - 1$

$\Delta E_{new} = 2(2x_p + 5)$

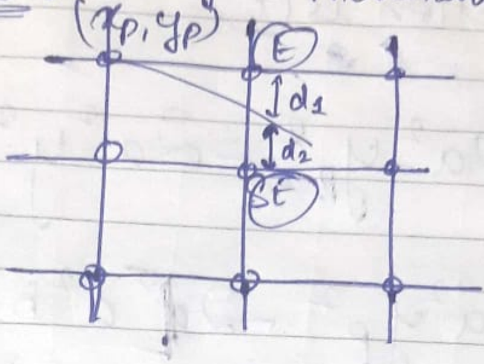
$\Delta SE_{new} = 2(2x_p + 5) - 4(y_p - 2)$

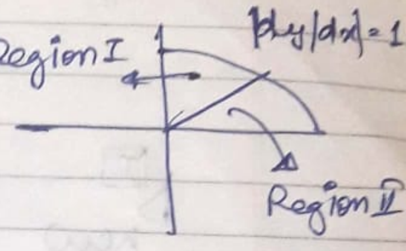
$$\boxed{\Delta^2 E_{SE} = \Delta E_{new} - \Delta E = 4}$$

$$\boxed{\Delta^2 SE_{SE} = \Delta SE_{new} - \Delta SE = 8}$$

Ellipse: ($b^2x^2 + a^2y^2 = a^2b^2$)

Region I: (major movement in x)





$$d_1 = a^2y_p^2 - a^2y^2$$
$$d_2 = a^2y^2 - a^2(y_p - 1)^2$$

$$a^2y^2 = a^2b^2 - b^2(x_p+1)^2$$

$$D^{\boxed{I}} = d_1 - d_2$$

$$D^{\boxed{I}} = -2a^2y^2 + 2a^2y_p^2 - a^2 + 2y_pa^2$$

$$D^{\boxed{I}} = 2a^2y_p^2 + 2y_pa^2 - a^2 \cancel{2a}$$
$$- 2(a^2b^2 - b^2(x_p+1)^2)$$

$$\cancel{D^{\boxed{I}} = a^2(2y_p^2 + 2y_p}$$

$$D^{\boxed{I}} = 2a^2y_p^2 - 2a^2y^2 + 2y_pa^2 - a^2$$

if $D < 0$

selection is E

$$D_{new}^{\boxed{I}} = 2a^2 y_p^2 - 2a^2 y^2 + 2y_p a^2 - a^2$$

$$D_{new}^{\boxed{I}} = 2a^2 y_p^2 - 2\left[a^2 b^2 - b^2 (x_p + 2)^2\right] + 2y_p a^2 - a^2$$

$$\Delta E^{\boxed{I}} = D_{new}^{\boxed{I}} - D^{\boxed{I}}$$

$$\Delta E^{\boxed{I}} = -2\left[a^2 b^2 - b^2 (x_p + 2)^2\right] + 2\left[a^2 b^2 - b^2 (x_p + 1)^2\right]$$

$$\Delta E^{\boxed{I}} = 2b^2 (x_p + 2)^2 - 2b^2 (x_p + 1)^2$$

$$\Delta E^{\boxed{I}} = 2b^2 (\cancel{x_p^2} + 4 + 4x_p - \cancel{x_p^2} - 2x_p - 1)$$

$$\boxed{\Delta E^{\boxed{I}} = 2b^2 (3 + 2x_p)}$$

else if $D >= 0$

selection is SE

$D_{new}^{\boxed{I}} = 2a^2y_p^2 - 2a^2y^2 + 2y_pa^2 - a^2$

$D_{new}^{\boxed{I}} = 2a^2(y_p-1)^2 - 2\left[a^2b^2 - b^2(x_p+2)^2\right]$
$+ 2(y_p-1)a^2 - a^2$

$\Delta SE^{\boxed{I}} = D_{new}^{\boxed{I}} - D^{\boxed{I}}$

$\Delta SE^{\boxed{I}} = 4a^2y_p^2 - 2\left[a^2b^2 - b^2(x_p+2)^2\right]$
$+ 2\left[a^2b^2 - b^2(x_p+1)^2\right]$

$\Delta SE^{\boxed{I}} = 2b^2\left[(x_p+2)^2 - (x_p+1)^2\right] + 4a^2y_p^2$

$$\boxed{\Delta SE^{\boxed{I}} = 2b^2\left[3 + 2x_p\right] + 4a^2\left[1 - y_p\right]} \cancel{+ 4a^2y_p^2}$$

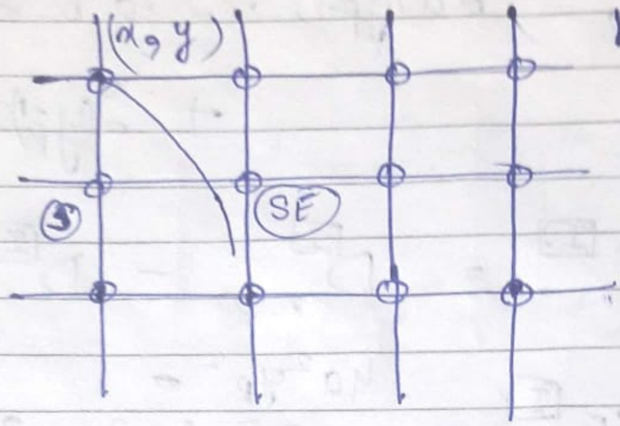
For $D_{in}^{\boxed{I}}$:

Starting from (0, b) :

$D_{in}^{\boxed{I}} = 2a^2b^2 + 2ba^2 - a^2 - 2(a^2b^2 - b^2)$

$$\Rightarrow \boxed{D_{in}^{\boxed{I}} = 2ba^2 - a^2 + 2b^2}$$

Region II: (major movement in y)



$b^2x^2 = a^2b^2 - a^2(y_p - 1)^2$

note

$$d_1 = -b^2x_p^2 + b^2x^2$$
$$d_2 = -b^2x^2 + b^2(x_p+1)^2$$

$$D^{\boxed{II}} = d_1 - d_2$$

$$D^{\boxed{II}} = -b^2x_p^2 - b^2(x_p+1)^2 + 2b^2x^2$$

$$D^{\boxed{II}} = 2b^2x^2 + b^2\left[(x_p+1)^2 - x_p^2\right]$$
$$D^{\boxed{II}} = 2\left[a^2b^2 - a^2(y_p-1)^2\right] - b^2\left[(x_p+1)^2 + (x_p)^2\right]$$

if $D < 0$, selection is S

$$x_{p+1} \rightarrow x_p$$
$$y_{p+1} \rightarrow y_{p-1}$$

$$D^{\boxed{II}}_{new} = 2\left[a^2b^2 - a^2(y_p-2)^2\right] - b^2\left[(x_p+1)^2 + (x_p)^2\right]$$

$$\Delta S^{\boxed{II}} = D^{\boxed{II}}_{new} - D^{\boxed{II}}$$

Scanned by CamScanner

$$\boxed{\Delta S^{\boxed{II}} = 2a^2(2y_p - 3)}$$

if $D > 0$, selection is SE

$$x_{p+1} \longrightarrow x_p + 1$$
$$y_{p+1} \longrightarrow y_p - 1$$

$$D_{new}^{\boxed{II}} = 2[a^2b^2 - a^2(y_p - 2)^2] - b^2[(x_p+2)^2 + (x_p+1)^2]$$

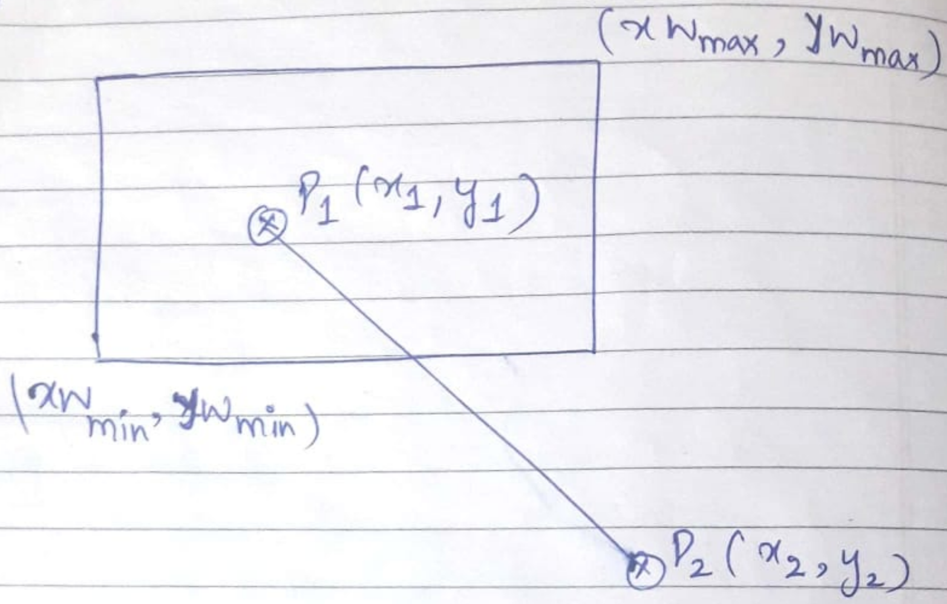
$$\Delta SE^{\boxed{II}} = D_{new}^{\boxed{II}} - D^{\boxed{II}}$$

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

$$\boxed{\Delta SE^{\boxed{II}} = 2a^2(2y_p - 3) - 4b^2(x_p + 1)}$$

CLIPPING

Rectangular window



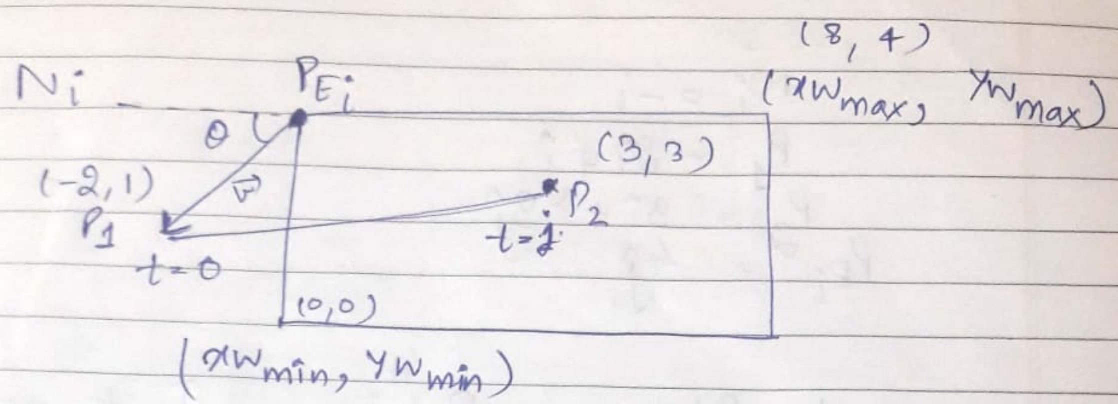
$min \leq x \leq max$
$min \leq y \leq max$

```
int inside ()
{
    Accordingly
      return (0) or (1)
}
```

// depending on whether it lies inside or outside.

0 0 (Both pts. out)
0 1
1 0
1 1 Both pts. in (no clipping reqd.)
line retained.

Cyrus – Beck Method :



Parametric eqn: $P(t) = P_1 + (P_2 - P_1)t$ —— (1)

Drop a vector from an edge & let it swing like a pendulum.

$$\vec{v} = [P(t) - P(E_i)]$$

$\vec{v} \cdot \vec{N_i}$ < 0 (pt. is inside, $\theta > 90°$)
$= 0$ (pt. is on edge)
> 0 (pt is outside, $\theta < 90°$)

$$\vec{v} \cdot \vec{N_i} = 0 \quad \text{(At intersection)}$$

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

$$[P_1 + (P_2 - P_1)t - P_{E_i}] \cdot \vec{N_i} = 0$$

$$N_i \cdot (P_1 - P_{E_i}) + \vec{N_i} \cdot (P_2 - P_1)t = 0$$

$$\boxed{t = \frac{N_i \cdot (P_1 - P_{E_i})}{-N_i \cdot (P_2 - P_1)}}$$

Left Edge :

$N_i = -i$

$P_1 - 2\hat{i} + \hat{j}$

$P_2 = 3\hat{i} + 3\hat{j}$

$P_{E_i} = 4\hat{j}$

$t = 2/5 \longrightarrow$ Put it in eqn (1) to get the intersection pt

$-N_i.(P_2 - P_1)$ $\neq 0$

< 0 — is potentially exiting

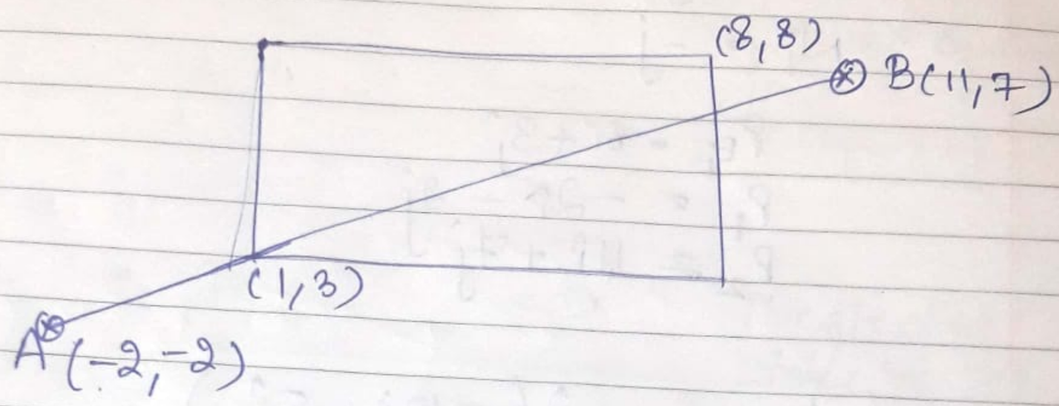
> 0 — is potentially entering

If $\boxed{0 \le t \le 1}$

→ then only we have to find the intersection pt

Otherwise, the edge & the line ~~can~~ do not intersect. So, there is no need to find the intersection pt.

Example 1:



Left Edge:

$\cancel{t_E}$

$N_i = -\hat{i}$

$P_{E_i} = (1,8) = \hat{i} + 8\hat{j}$

$P_1 = -2\hat{i} - 2\hat{j}$

$P_2 = 11\hat{i} + 7\hat{j}$

$$t_L = \frac{-\hat{i} \cdot (-2\hat{i} - 2\hat{j} - \hat{i} - 8\hat{j})}{+\hat{i} \cdot (11\hat{i} + 7\hat{j} + 2\hat{i} + 2\hat{j})}$$

$$t_L = \frac{-\hat{i}(-3\hat{i} - 10\hat{j})}{\hat{i}(13\hat{i} + 9\hat{j})}$$

$$t_L = \frac{3 + 0}{13 + 0} = \boxed{\frac{3}{13}}$$

Bottom Edge:

$N_i = -j$

$P_{E_i} = 8\hat{i}+3\hat{j}$

$P_1 = -2\hat{i} - 2\hat{j}$

$P_2 = 11\hat{i} + 7\hat{j}$

$$t_B = \frac{-\hat{j}\cdot(-10\hat{i} - 5\hat{j})}{\hat{j}\cdot(13\hat{i} + 9\hat{j})}$$

$$\boxed{t_B = 5/9}$$

Right Edge:

$N_i = +\hat{i}$

$P_{E_i} = 8\hat{i}+8\hat{j}$, $P_1 = -2\hat{i}-2\hat{j}$, $P_2 = 11\hat{i}+7\hat{j}$

$$t_R = \frac{\hat{i}(-10\hat{i} - 10\hat{j})}{-\hat{i}\cdot(13\hat{i} + 9\hat{j})}$$

$$t_R = \frac{-10}{-13} = \boxed{\frac{10}{13}}$$

Up Edge:

$N_i = +\hat{j}$

$P_{E_i} = 8\hat{i}+8\hat{j}$, $P_1 = -2\hat{i}-2\hat{j}$, $P_2 = 11\hat{i}+7\hat{j}$

$$t_T = \frac{\hat{j}(-10\hat{i} - 10\hat{j})}{-\hat{j}(13\hat{i}+9\hat{j})} = \frac{-10}{-9} = \boxed{10/9}$$

Scanned by CamScanner

		Denominator Values	
t_L	$= 3/13$	> 0	Max of these two
t_B	$= 5/9$	> 0	
t_R	$= 10/13$	< 0	
t_T	$= 10/9$	< 0	Neglected since $t > 1$.

$\max(t_L, t_B) = t_B$

Clipped line: line (t_B, t_R)

Potentially entering $\rightarrow \max(D_r > 0)$

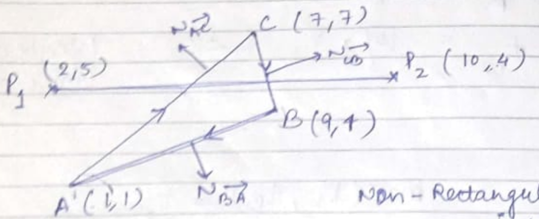
Potentially exiting $\rightarrow \min(D_r < 0)$

23/01/2018

Line Clipping

Cyrus – Beck :

Example 3 :



Non-Rectangular window ↓ Can be convex or concave

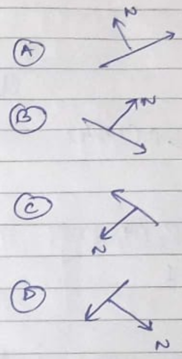
This window is convex as all the angles < 180°

We can traverse the window either in ↻ or ↺ dir^n.
When ↻ → RHS is always towards the window, LH → outward

Edges	Normal (N_i)	P_{E_i}	$P_1 - P_{E_i}$	$N_i \cdot (P_1 - P_{E_i})$	$(P_2 - P_1) \cdot N$	t	
$\overrightarrow{AC}$ $(6\hat{i}+6\hat{j})$	$-6\hat{i}+6\hat{j}$	$\hat{i}+\hat{j}$	$\hat{i}+4\hat{j}$	18	−54	1/3 ~~($P_{E_{exit}}$)~~ ($P_{E_{ent}}$)	Increasing order ↓
$\overrightarrow{CB}$ $(2\hat{i}-3\hat{j})$	$3\hat{i}+2\hat{j}$	$7\hat{i}+7\hat{j}$	$-5\hat{i}-2\hat{j}$	−19	22	19/22 ~~($P_{E_{ent}}$)~~ ($P_{E_{xit}}$)	
$\overrightarrow{BA}$ $(-8\hat{i}-3\hat{j})$	$3\hat{i}-8\hat{j}$	$9\hat{i}+4\hat{j}$	$-7\hat{i}+\hat{j}$	−29	32	29/32 ~~($P_{E_{ent}}$)~~ ($P_{E_{xit}}$)	

$P_2 - P_1 = 8\hat{i} - \hat{j}$

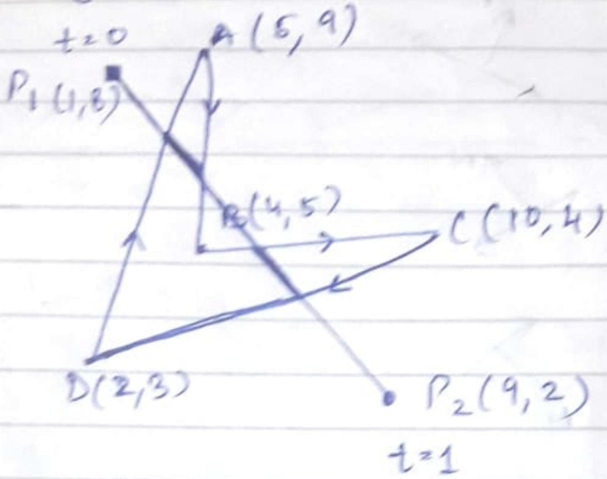
Edge	Normal
(A) + +	− +
(B) + −	+ +
(C) − +	− −
(D) − −	+ −



25/01/2018

Line Clipping

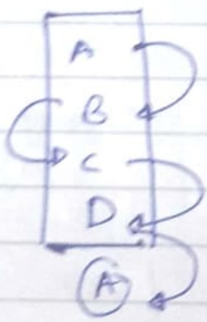
Non-Rectangular Window:



Clockwise

Concave window

arr[ ]



$P_1 = \hat{i} + 8\hat{j}$

$P_2 = 9\hat{i} + 2\hat{j}$

$P_2 - P_1 = 8\hat{i} - 6\hat{j}$

Edges	Normal (N_i)	P_{E_i}	$P_1 - P_{E_i}$	$N_i \cdot (P_1 - P_{E_i})$	$N_i \cdot (P_2 - P_1)$	t
$\vec{AB} = -\hat{i} - 4\hat{j}$	$4\hat{i} - \hat{j}$	$5\hat{i} + 9\hat{j}$	$-4\hat{i} - \hat{j}$	-15	38 (P_{Exit})	15/38
$\vec{BC} = 6\hat{i} - \hat{j}$	$\hat{i} + 6\hat{j}$	$4\hat{i} + 5\hat{j}$	$-3\hat{i} + 3\hat{j}$	15	-28 (P_{Ent})	15/28
$\vec{CD} = -8\hat{i} - \hat{j}$	$\hat{i} - 8\hat{j}$	$10\hat{i} + 4\hat{j}$	$-9\hat{i} + 4\hat{j}$	-41	56 (P_{Exit})	41/56
$\vec{DA} = 3\hat{i} + 6\hat{j}$	$-6\hat{i} + 3\hat{j}$	$2\hat{i} + 3\hat{j}$	$-\hat{i} + 5\hat{j}$	21	-66 (P_{Ent})	21/66

* **Convex:** Ent, Ent, Ext, Ext , Ent Ext Ext
Ext Ext Ent Ent , Exit Ent Ent
Ent Ent, Ext
Ext Ext Ent

concavity ⟶ increasing order

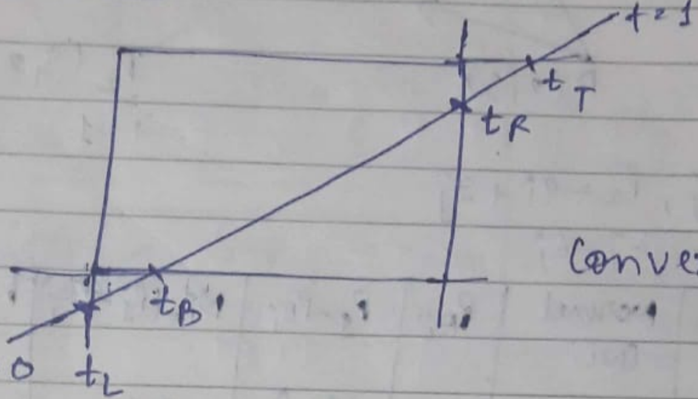
$$S : \left(\overset{t}{\underset{P_1}{0}} \quad \tfrac{21}{66} \quad \tfrac{15}{38} \quad \tfrac{15}{28} \quad \tfrac{41}{56} \quad \underset{P_2}{1} \right)$$

P_{Ent} P_{Exit} P_{Ent} P_{Exit}

To show clipped line, draw line b/w (ent / ext) pairs

line(21/66, 15/38)
line (15/28, 41/56)

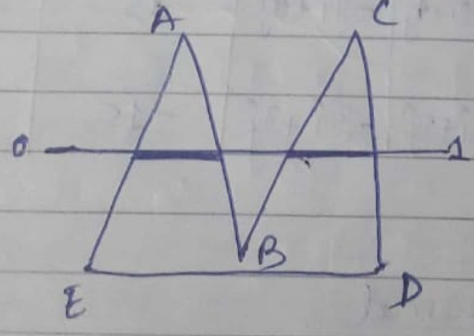
For Convex:



Convex Window

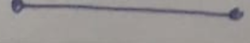
$$S : \left(0, \overset{t_L}{P_{Ent}}, \overset{t_B}{P_{Ent}}, \overset{t_R}{P_{Exit}}, \overset{t_T}{P_{Exit}}, 1\right)$$

⟵ line b/w ent & exit



⟹ Ent, Exit, Ent, Exit

If such pairs (Ent, Exit) are more than 1 b/w t=0 & t=1 . ⟹ then clipping in concave part.

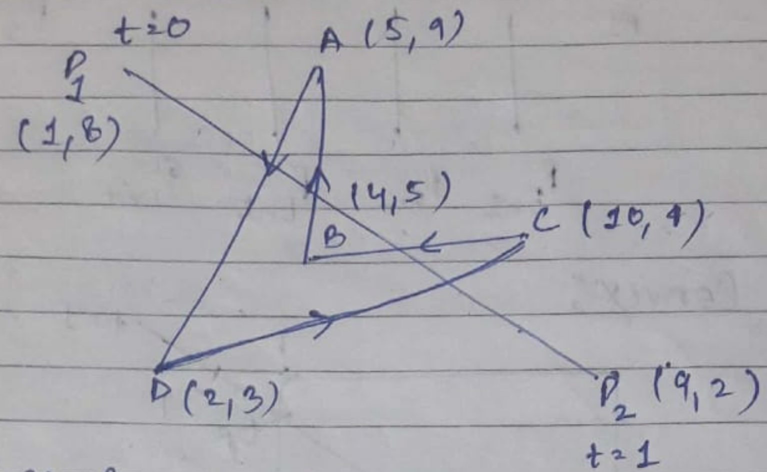
 ⟹ $t < 0$ or $t > 1$

If one enter/ exit → one clipping

If more than one enter/exit → more than 1 clipping

20

Anticlockwise:



$P_1 = \hat{i} + 8\hat{j}$, $P_2 = 9\hat{i} + 2\hat{j}$

$P_2 - P_1 = 8\hat{i} - 6\hat{j}$

Edges	Normal (N_i)	P_{E_i}	$P_1 - P_{E_i}$	$N_i \cdot (P_1 - P_{E_i})$	$N_i \cdot (P_2 - P_1)$	t
$\vec{AD} = -3\hat{i} - 6\hat{j}$	$-6\hat{i} + 3\hat{j}$	$5\hat{i} + 9\hat{j}$	$-4\hat{i} - \hat{j}$	21	-66 (P_{Ent})	21/66
$\vec{DC} = 8\hat{i} + \hat{j}$	$\hat{i} - 8\hat{j}$	$2\hat{i} + 3\hat{j}$	$-\hat{i} + 5\hat{j}$	-41	56 (P_{Exit})	41/56
$\vec{CB} = -6\hat{i} + \hat{j}$	$\hat{i} + 6\hat{j}$	$10\hat{i} + 4\hat{j}$	$-9\hat{i} + 4\hat{j}$	15	-28 (P_{Ent})	15/28
$\vec{BA} = \hat{i} + 4\hat{j}$	$4\hat{i} - \hat{j}$	$4\hat{i} + 5\hat{j}$	$-3\hat{i} + 3\hat{j}$	-15	38 (P_{Exit})	15/38

Edge	Normal
+ +	+ −
+ −	− −
− +	+ +
− −	− +

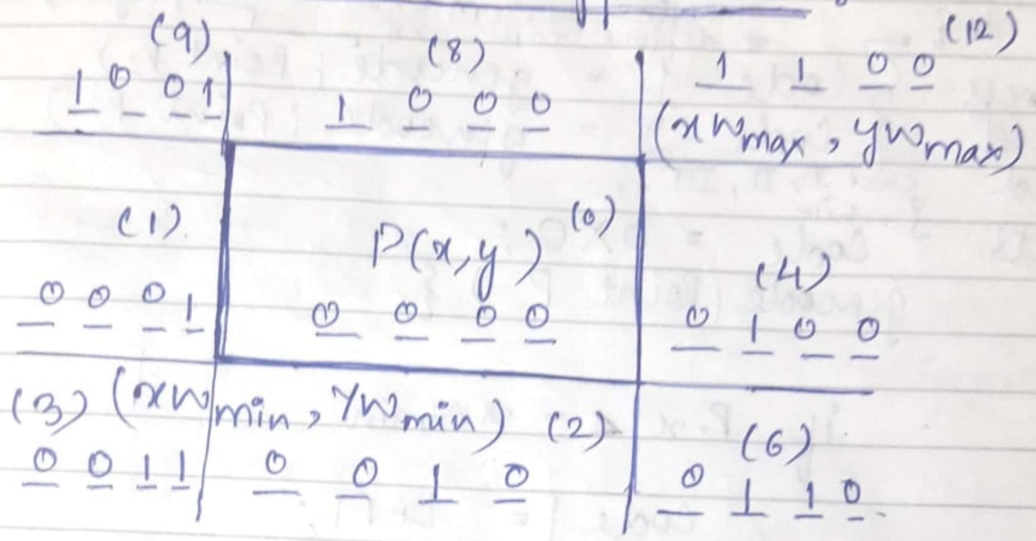
S: (0 21/66 15/38 15/28 41/56 1)

21/66 ↓ P_{Ent}, 15/38 ↓ P_{Exit}, 15/28 ↓ P_{Ent}, 41/56 ↓ P_{Exit}

29/01/2018

Line Clipping

Cohen - Sutherland Approach :



$$xw_{min} \leq x \leq xw_{max}$$
$$yw_{min} \leq x \leq yw_{max}$$

A rectangular window as 9 regions.
Point can be any where in these 9 regions.
Each region has a unique region code (address).

Left side → LSB 1.
Bottom side → 2nd bit 1
Right side → 3rd bit 1.
Upper side → MSB 1

Input: $P_1(x_1, y_1)$, $P_2(x_2, y_2)$

Method:

```
# define LEFT 0X1
# define RIGHT 0X4
# define BOTTOM 0X2
# define TOP 0X8
```

flag = 0; done = 0;

Address assigned to end points

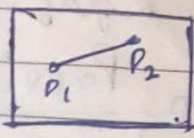
```
do{
    Code 1 = gencode ( Point P1)
    Code 2 = gencode ( Point P2)
    case I, II, III ;
} while
    Code = 0X0;

gencode ( Point P ) {

    if P.x <= xwmin
        code |= LEFT;

    if P.x >= xwmax
        code |= RIGHT;

    if P.y <= ywmin
        code |= BOTTOM;

    if P.y >= ywmax
        code |= TOP;

}
```

Case I :

Code (P_1) = 0000
code (P_2) = 0000

AND
OR

$\Downarrow$

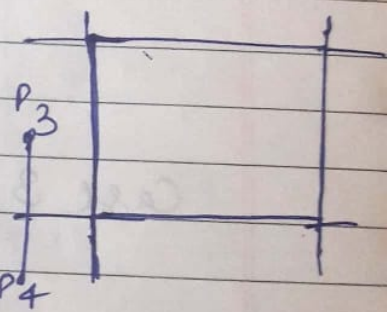


P_1 AND P_2 = 0
P_1 OR P_2 = 0 } True

$\boxed{\text{True}}$ ⟹ Line is inside, no clipping

Case II :

Code (P_3) = 0001
Code (P_4) = 0011



AND

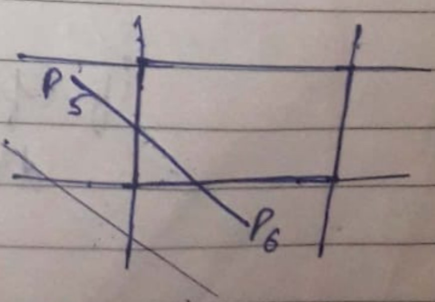
P_1 AND P_2 ≠ 0

$\Downarrow$

$\boxed{\text{NON-Zero}}$ ⟹ Line is outside

Case III :

Code (P_5) = 0001
Code(P_6) = 0010



$\Downarrow$

AND ⟶ 0
OR ⟶ NON-ZERO

P_1 AND P_2 = 0
P_1 OR P_2 ≠ 0

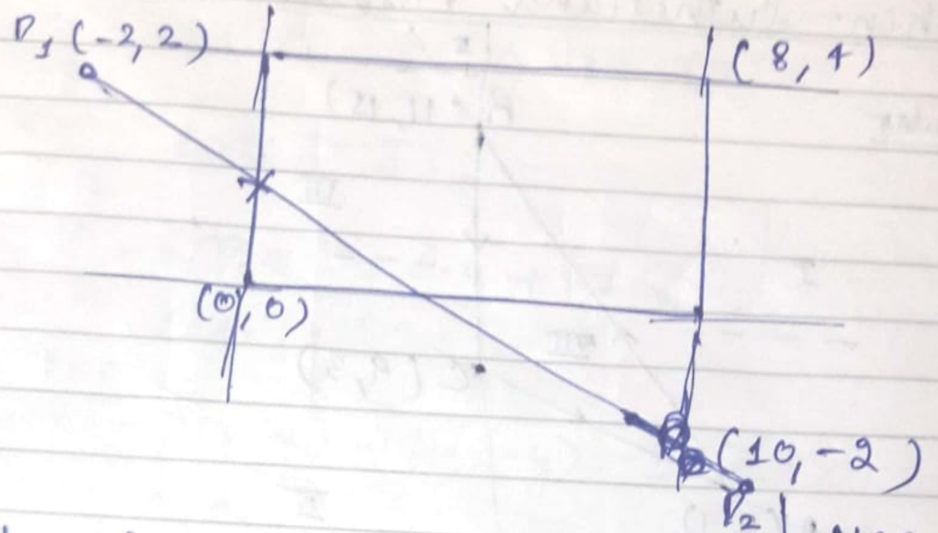
For case III:

First clip ~~B~~. Then

* Keep checking iteratively whether after clipping line falls under case I or case II

```
flag = 0; done = 0;
do
{
   Previous code:  code(P1) = gencode(Point P1)
                   code(2) = gencode(Point P2)
   case 1: if (code 1 & code 2 = = 0) &&
                        (code 1 | code2 = = 0)
           {
               /* line is visible */
               flag = 1; done = 1;
               exit();
           }

   case 2: if (code 1 & code 2 != 0)
           {
               /* line is not visible */
               exit(); done = 1;
           }

   Case 3: if (code 1 & code2 = = 0) && (code1 | code2 != 0)

               find intersection();

} while (done != 1)

if (flag)          // line is inside
   drawline(P1, P2)
```

EXAMPDE 1 :



Code (P_1) = 0001
Code (P_2) = 0110

case (3
LEFT = 0X1;
if (LEFT & code P_1 == 1)
$x_I = xw_{min}$;
$y_I = y_1 + \frac{(y_2-y_1)}{(x_2-x_1)} * (x_1 - xw_{min})$

$P_1 \leftarrow (x_I, y_I)$

Clipped at left edge

flag	done
0	0

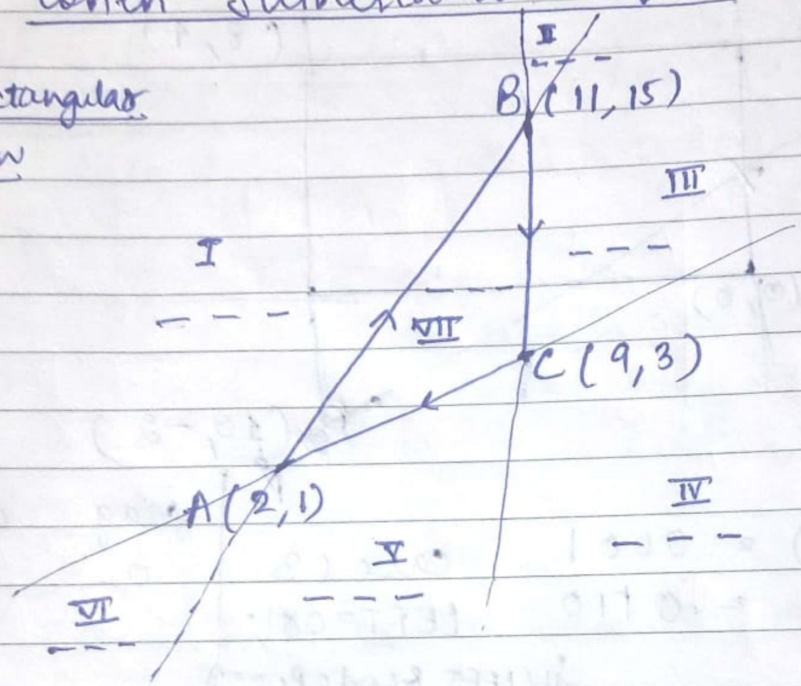
code (P1) = 0000
code (P_2) = 0110

30/01/2018

Line Clipping

Cohen-Sutherland Method:

Non-rectangular window



Inside Test

Equation	
AB f(x,y): 14x − 9y − 19	f(x,y): C(9, 3) >0
BC f(x,y): −6x + y + 51	f(x,y): A(2,1) >0
CA f(x,y): −2x + 7y − 3	f(x,y): B(11,15) >0

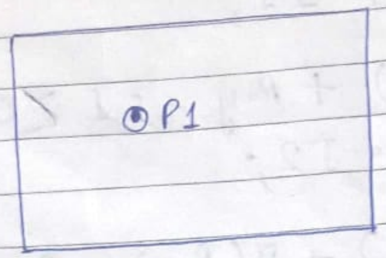
>0 → Point is inside the window.

```
# define I1  1
# define I2  2
# define I3  4
int code = 0;
int gencode (Point P)
{
    if ( 14(P.x) - 9(P.y) -19 < 0)
        code |= I1;
    if ( -6(P.x) + P.y +51 < 0)
        code |= I2;
    if ( -2(P.x) + 7(P.y) -3 < 0)
        code |= I3;
}
```

01/02/2018

Line Clipping

Nicholl - Lee - Nicholl :



Computation time faster than other algorithms but code length is large

Step 1:
Locate one of the end point of a line.

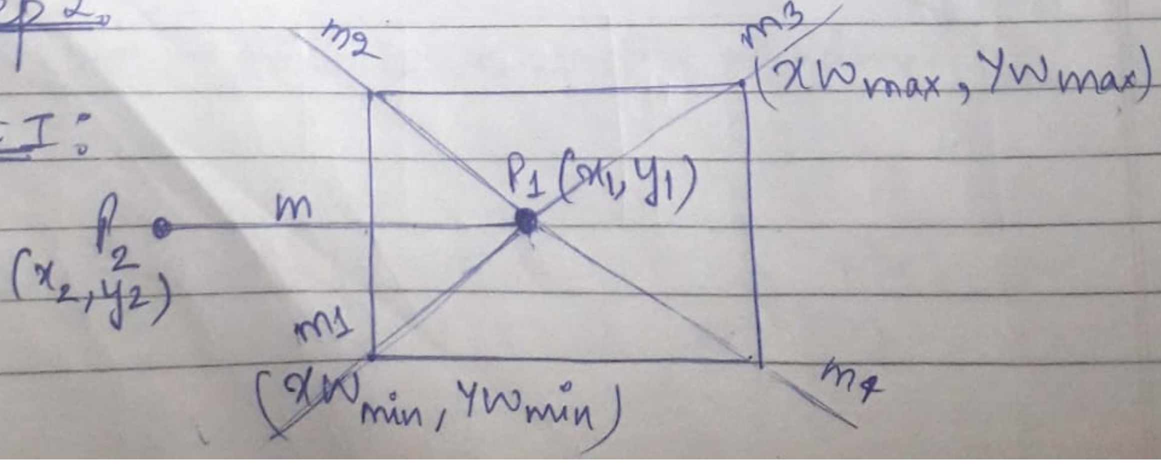
Part I :
End point is inside the rectangular window.

Part II :
End point is on the edge side.
(Left, Right, Top, Bottom)

Part III :
End point is on the corner.
(left Top, left bottom, Right top, right bottom)

Step 2 :

Part I :



If line ($P_1 - P_2$) is rotated, then its slope m' will between any of the slopes m_1, m_2, m_3, m_4.

Slope (m)

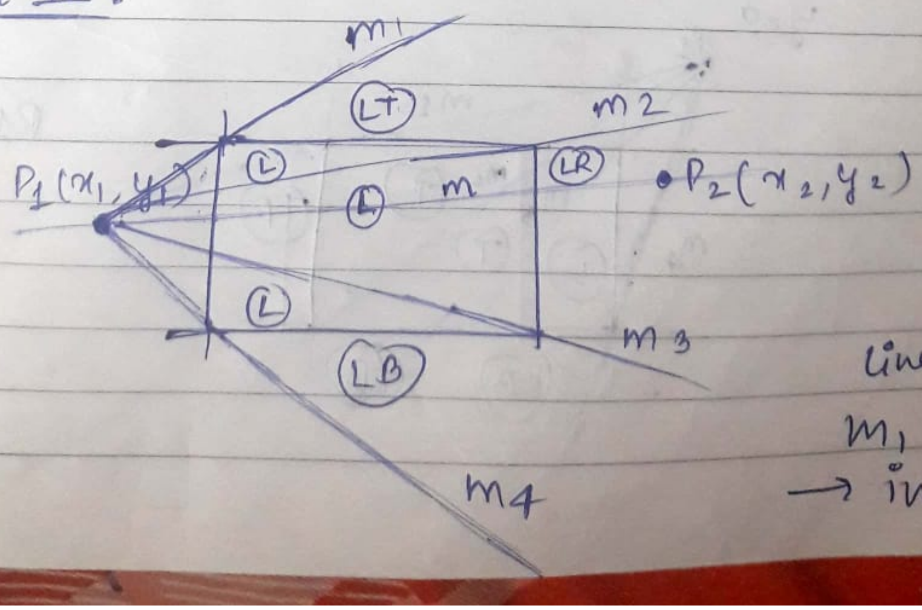
locating 2nd end pt w.r.t. slopes { $m_1 \le m < m_2$ $m_2 \le m < m_3$ $m_3 \le m < m_4$ $m_4 \le m < m_1$

(Y) Clipping (N) No clipping

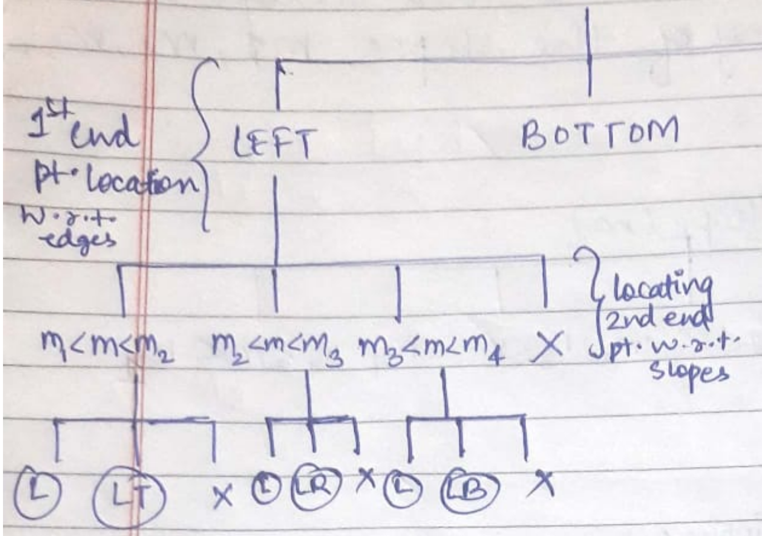
```
if ((m > m2) && (m < m1) && (P1.x1 > P2.x2))
{
    if (P2.x2 < xwmin)
        clipping
    ~~No cli~~
}
```

// Similarly for other cases.

Part II :

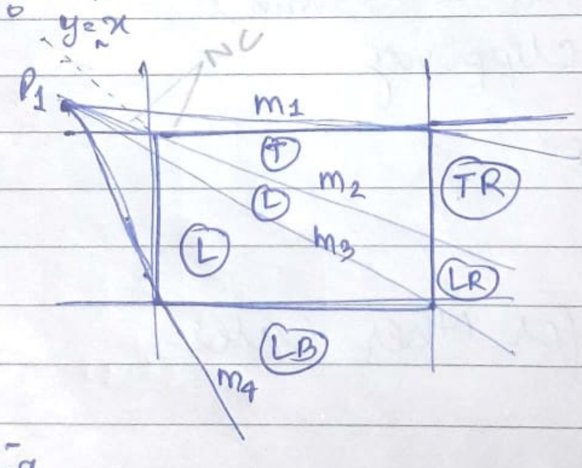


Line b/w m_1 & m_4 → invisible

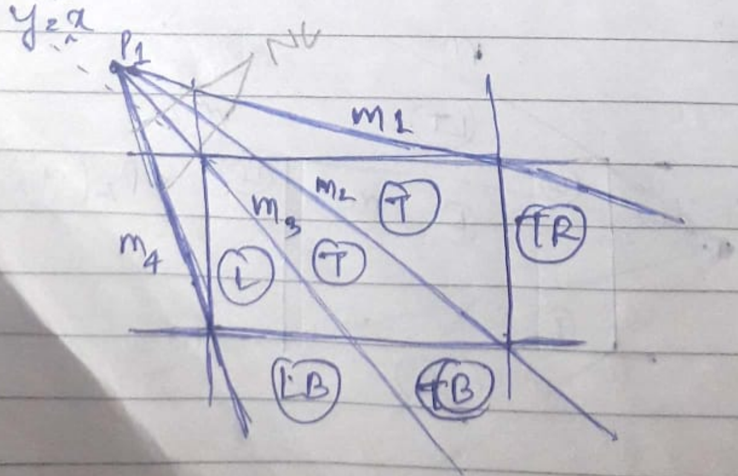


// Similarly for other cases.

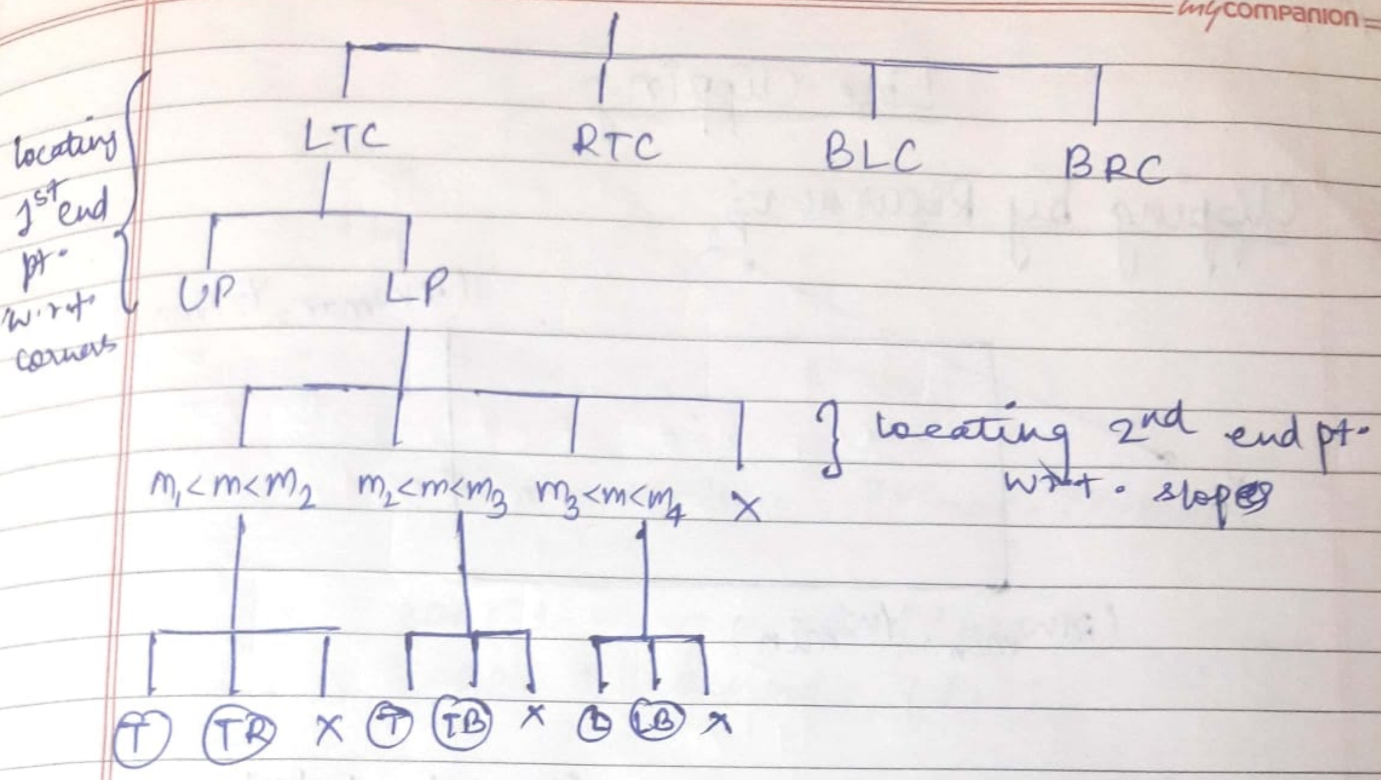
Part III :



P1 in lower part of y=x



P1 above y=x.

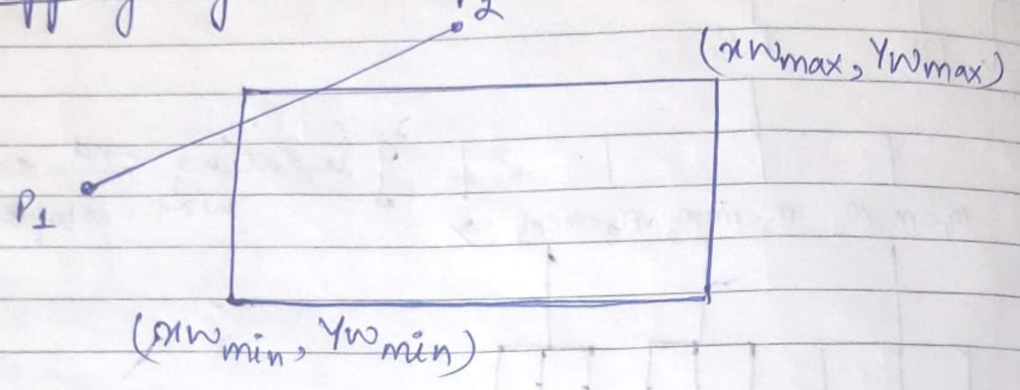


// Similarly for other cases.

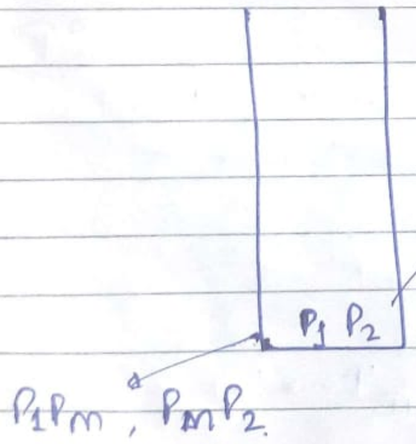
05/02/2018

Line Clipping

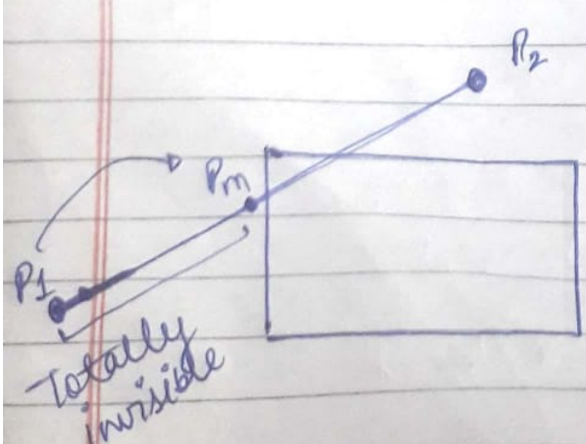
Clipping by Recursion: (Mid Point Subdivision)

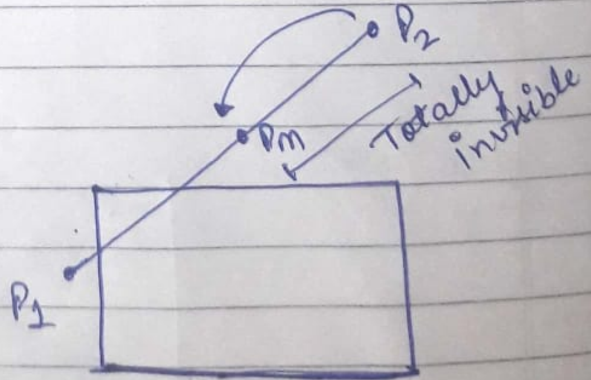


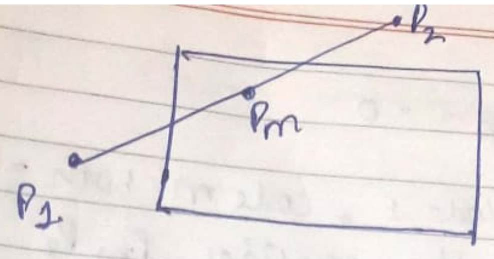
mid point of a line can be easily calculated.



Cohen Sutherland:
- → totally visible
- → totally invisibe
- → partially visible







Both are visible

Input: $P_1(x_1, y_1)$, $P_2(x_2, y_2)$

xw_{min}, xy_{min}, xw_{max}, yw_{max}.

code 1 = gencode (P1);
code 2 = gencode (P2);

Step ①

if code1 , code 2 both = 0
then draw line $P_1 P_2$

Step ②

if code 1 & code 2 <> 0
/* line invisible */

Step ③

$P_m = (P_1 + P_2)/2$;

code m = gencode (P_m);

Step ④

if code m <> 0
then
if code 1 & code m <> 0 then
$P_1 \leftarrow P_m$; goto step①

else if code 2 & code m <> 0 then
$P_2 \leftarrow P_m$, goto step①

Step ⑤

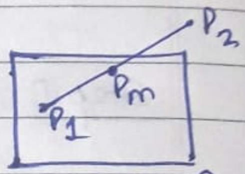
If code m = = 0

if code 1, code m both = 0.
then consider $P_m P_2$

~~else if code~~
else if code 2 ; code m both = 0
then consider $P_1 P_m$.

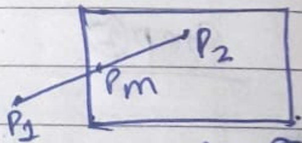
Step ⑥

consider $P_1 P_m$



Step ⑥

do { $P_{m_1} = (P_1 + P_m)/2$;
code m_1 = gencode(P_{m_1});
if code m_1 <> 0 then $P_1 = P_m$, else
$P_m = P_{m_1}$
}



Step ⑦

while ($P_m.x$ <> xw_{min} &&
$P_m.x$ <> xw_{max} &&
$P_m.y$ <> yw_{min} &&
$P_m.y$ <> yw_{max})

$P_1 \leftarrow P_m$.

Step ⑦

consider $P_m P_2$

do {

$P_{m_2} = (P_2 + P_m)/2$; code m_2 = gencode(P_{m_2})

if code m_2 <> 0 then $P_2 = P_{m_2}$, else
$P_m = P_{m_2}$

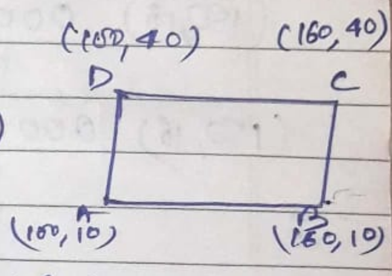
}

While (same condition as step ⑥)

$P_2 \leftarrow P_m$

EXAMPLE :

$P_1 (120, 5)$, $P_2 (180, 30)$



Window ABCD

A(100, 10) , B(160, 10), C(160, 40), D(100, 40)

P_1	Code 1	P_2	Code 2	P_m	Code m	Remarks
(120,5)	0010	(180,30)	0100	150,18	0000	Consider $P_1 P_m$ & Consider $P_m P_2$.
Consider $P_1 P_m$						
(120,5)	0010	(150,18)	0000	(135,12)	0000	
(120,5)	0010	(135,12)	0000	(128,9)	0010	
(128,9)	0010	(135,12)	0000	(132,11)	0000	
(128,9)	0010	(132,11)	0000	(130,10)	0001	
(128,9)	0010	(130,10) P_m				

Consider $P_m P_2$

~~(135,12) 0000 150~~

P_1 code 1 P_2 code 2 P_m Codem Remarks.

(150,18) 0000 (180,30) 0100 (165,24) 0100

(150, 18) 0000 (165,24) 0100

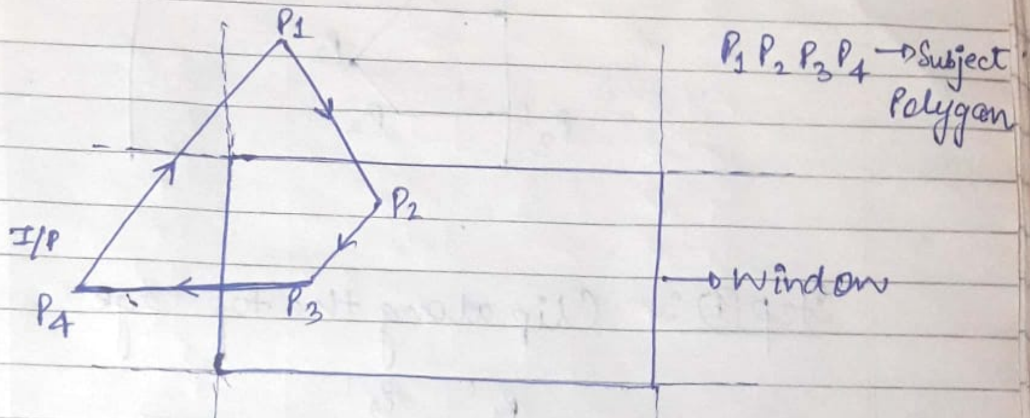
Scanned by CamScanner

Polygon Clipping → Window → Rectangular, Non-Rectan; Polygon → Convex, Concave.

06/02/2018

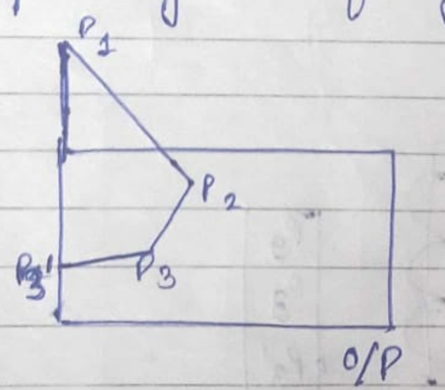
CLIPPING

Polygon Clipping

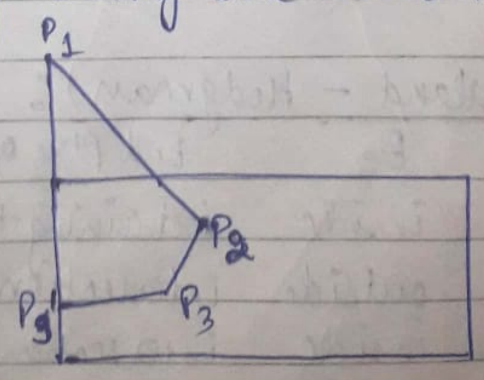


Clipping against rectangular window

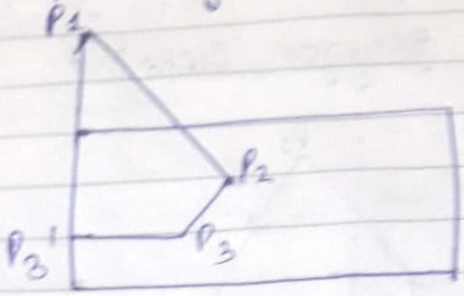
Step ① : Clip along the left edge



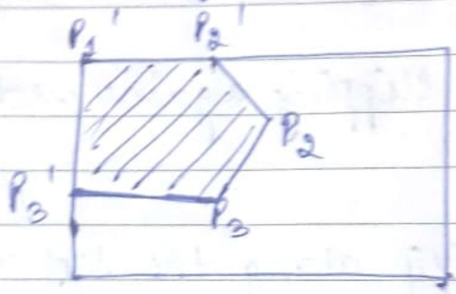
Step ② : Clip along the bottom edge



Step ③: Clip along the right edge



Step ④: Clip along the top edge



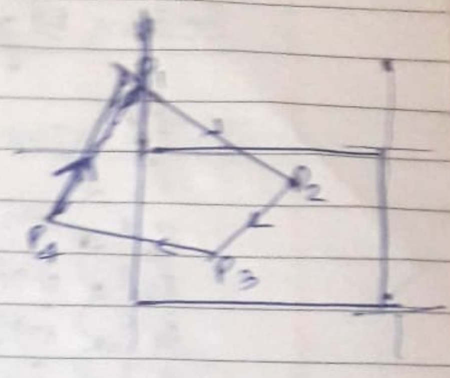
Rules (Sutherland – Hodgman):

E_1	E_2	End pts. of edge (E_1 & E_2)
inside	inside	Retaining the 2nd End Pt. (E_2)
inside	outside	Intersection pt. is computed & stored
outside	inside	intersection pt & 2nd end pt to be stored
outside	outside	NIL

EXAMPLE:

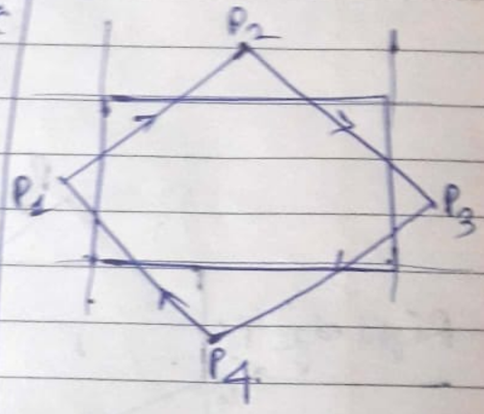
Near left edge

I/P	After Processing
P_1	P_1'
P_2	P_2'
P_3	P_2
P_4	P_3'
P_1	P_1'

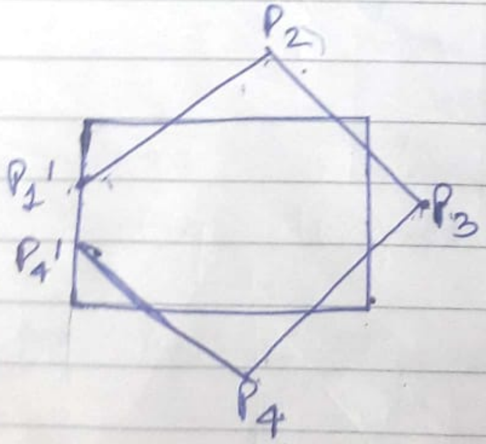


EXAMPLE:

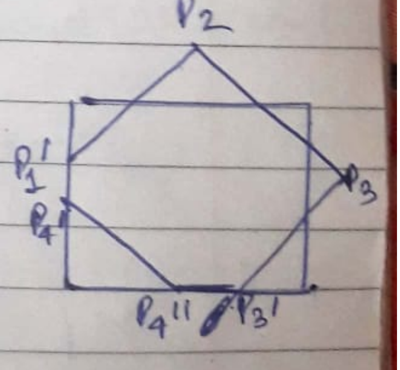
	I/P	Left edge		Bottom edge
	P_1	P_1'	$i \to i$	P_2
$O \to i$	P_2	P_2	$i \to i$	P_3
$i \to i$	P_3	P_3	$i \to O$	P_3'
$i \to i$	P_4	P_4	$O \to i$	P_4''
$i \to O$	P_1	P_4'	$i \to i$	P_4'
		P_1'		P_1'
				P_2



After clipping at left edge:

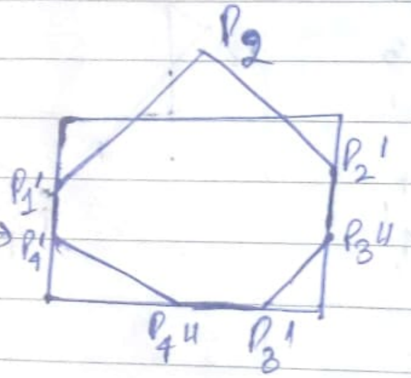


Bottom edge.

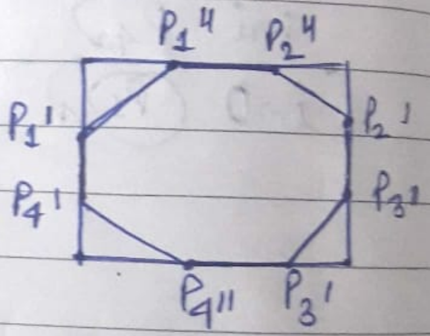


Bottom Edge		Right Edge		Top Edge
P_2	$i \to o$	P_2'	$i \to i$	P_3''
P_3	$o \to i$	P_3''	$i \to i$	P_3'
P_3'	$i \to i$	P_3'	$i \to i$	P_4''
P_4''	$i \to i$	P_4''	$i \to i$	P_4'
P_4'	$i \to i$	P_4'	$i \to i$	P_1'
P_1'	$i \to i$	P_1'	$i \to o$	P_1''
P_2		P_2	$o \to i$	P_2''
		P_2'		P_2'
		P_2		P_3''

Right edge $\longrightarrow$



Top edge $\longleftrightarrow$

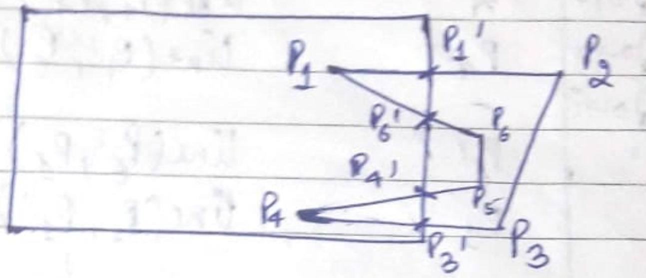


08/02/2018

Polygon Clipping

Sutherland-Hodgman Method

EXAMPLE 1 :



Subject polygon: concave polygon

Input : (a) window Coordinates
(b) Subject Polygon coordinates

Method :

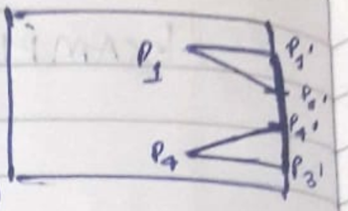
Rules

(a)	$i \rightarrow o$	Store intersection point (I)
(b)	$o \rightarrow i$	Store intersection point (I), Second vertex (P_2)
(c)	$i \rightarrow i$	Store second vertex (P_2)
(d)	$o \rightarrow o$	Nothing is stored

Output : Set of Clipped polygons.

Scanned by CamScanner

i/p		Right edge	
P_1	$i \to o$	P_1'	line($P_1'P_3'$) × (Not Reqd.)
P_2	$o \to o$	—	
P_3	$o \to i$	P_3'	line(P_3', P_4)
P_4	$i \to o$	P_4	line(P_4, P_4')
P_5	$o \to o$	P_4'	line(P_4', P_6') × (Not Reqd.)
P_6	$o \to i$	—	
P_1		P_6'	line(P_6', P_1)
		P_1	line(P_1, P_1')



We should remove (P_1', P_3') & (P_4', P_6').
In place of these two, (P_1', P_6') & (P_4', P_3') should be there.

Step 1:

	i/o	traversed
P_1'	1	~~0~~ 1
P_3'	0	~~0~~ 1
P_4	0	~~0~~ 1
P_4'	1	~~0~~ 1
P_6'	0	~~0~~ 1
P_1	0	~~0~~ 1
P_1'	0	~~0~~ 1

c(if i/o ①)

Step 2:

P_1', P_3', P_4', P_6' $\xrightarrow[\text{descending}]{\text{sort}(y)}$ P_1', P_6', P_4', P_3'

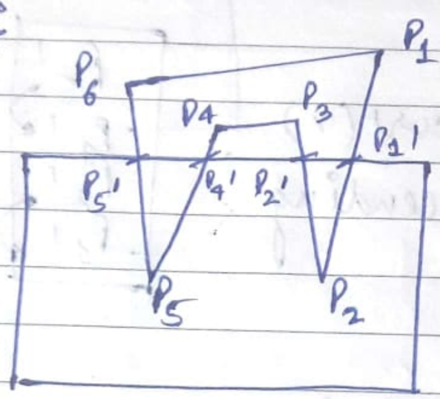
c(if i/o ①)

(x coordinate of each intersection pt. is same, so sort them acc. to y coordinate).

line (P_1', P_6')
line (P_6', P_1)

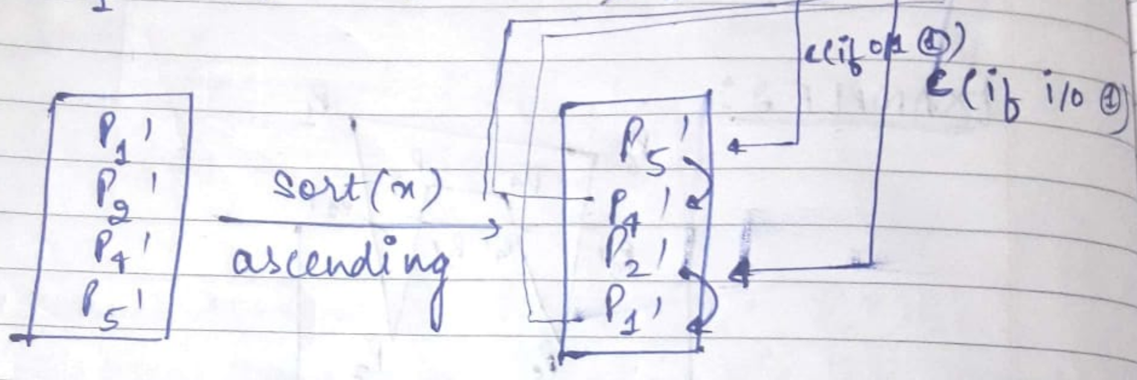
line (P_1, P_1')
line (P_3', P_4)
line (P_4, P_4')
line (P_4', P_3')

EXAMPLE 2:



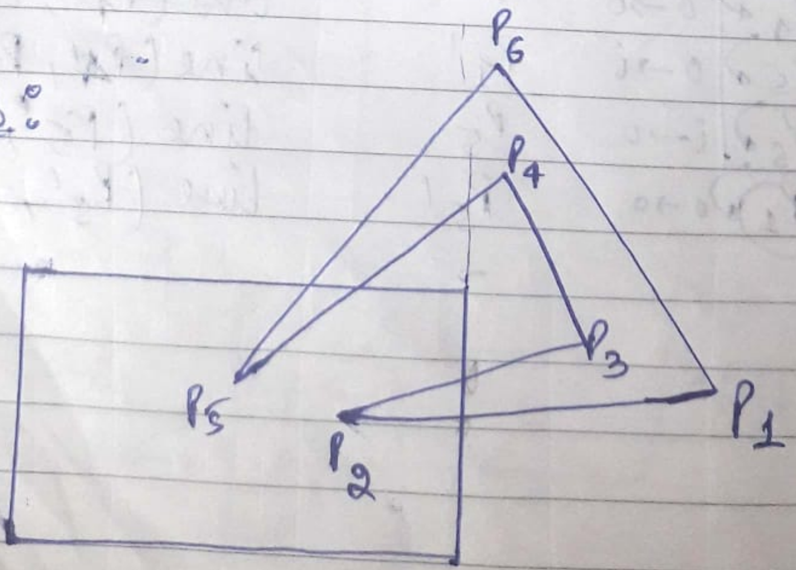
i/P		Top edge	
P_1	$o \to i$	P_1'	line (P_1', P_2)
P_2	$i \to o$	P_2	line (P_2, P_2')
P_3		P_2'	line(P_2', P_4') × (Not Reqd.)
P_4	$o \to o$	–	~~line (P_4', P_5)~~
P_5	$o \to i$	P_4'	line (P_4', P_5)
P_6	$i \to o$	P_5	line (P_5, P_5') ~~(Not Reqd)~~
P_1	$o \to o$	P_5'	line (P_5', P_1') (Not Reqd)
		–	

	~~i/o~~	i/o	visit (traversed)
P_1'	0	0	~~0~~ 1
P_2	~~1~~ 0	0	~~0~~ 1
P_2'	1	1	~~0~~ 1
P_4'	0	0	~~0~~ 1
P_5	1	0	~~0~~ 1
P_5'	0	1	~~0~~ 1
P_1'	0	0	~~0~~ 1



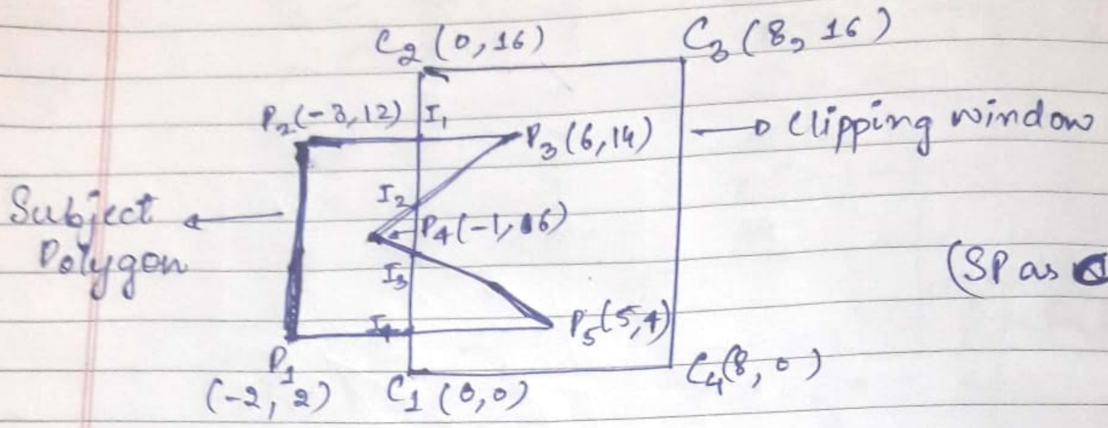
line (P_1', P_2)
line (P_2, P_2')
line (P_2', P_1')
line (P_4', P_5)
line (P_5, P_5')
line (P_5', P_4')

Example 3:



12/02/2018

Polygon Clipping
WEILER – ATHERTON



(SP as CW)

→ walk over the subject polygon.
→ Illumination starts on entry ⌐→ traversal is on clipping window
→ whenever you exit, take a right turn (traversing in clockwise direction).

$$t = \frac{N.(C_1 - P_{E_i})}{-N.(C_2 - C_1)}$$

Edge (clipping window)

For edge C_1C_2:

C_1C_2 / Edge of SP		Normal	Entry/Exit (1) (0)	t	Intersection Pt.
P_1P_2	$-\hat{i}+10\hat{j}$	$-10\hat{i}-\hat{j}$	Exit	$-9/8$	×
P_2P_3	$9\hat{i}+2\hat{j}$	$-2\hat{i}+9\hat{j}$	Entry	$19/24$	I_1
P_3P_4	$-7\hat{i}-8\hat{j}$	$+8\hat{i}-7\hat{j}$	Exit	$30/112$	I_2
P_4P_5	$6\hat{i}-2\hat{j}$	$2\hat{i}+6\hat{j}$	Entry	$1/12$	I_3
P_5P_1	$-7\hat{i}-2\hat{j}$	$2\hat{i}-7\hat{j}$	Exit	$9/56$	I_4

Step ①:

SP P_1 P_2 P_3 P_4 P_5

CW C_1 C_2 C_3 C_4

Step ②: ⓐ Evolve Intersection pt.
ⓑ Entry/Exit
→ Determino

Step ③:

SP (updated): P_1 P_2 $\boxed{1}$ I_1 P_3 $\boxed{0}$ I_2 P_4 $\boxed{1}$ I_3 P_5 $\boxed{0}$ I_4

CW (acc. to t values): C_1 I_4 I_3 I_2 I_1 C_2 C_3 C_4

$\boxed{1}$ $\boxed{0}$ $\boxed{1}$ $\boxed{0}$

Step ④:

line (I_1, P_3)
line (P_3, I_2)
line (I_2, I_1)
] Clipped polygon 1

Entry list {I_1, I_3}

line (I_3, P_5)
line (P_5, I_4)
line (I_4, I_3)
] Clipped polygon 2

Start traversing, whenever a match with entry list found, start illuminating.

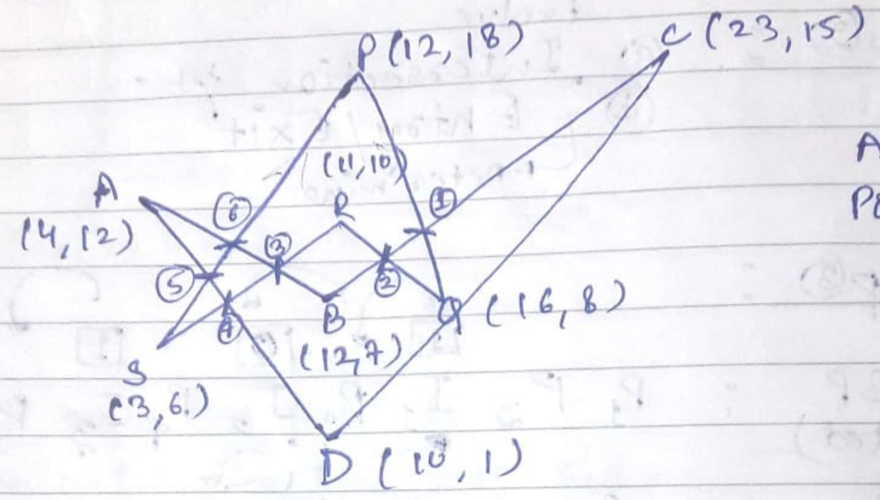
* Note: Right turn → switching from SP to CW

15/02/2018

Polygon Clipping

Weiler Atherton

Non-Rectangular ~~##~~ Clipping Window



ABCD (CW)
PQRS (SP)

Step ①:

SP : P Q R S
CW : A B C D

Step ②:

Intersection point computation

SP : Edge 1 (PQ) $4\hat{i} - 10\hat{j}$
(clockwise traversal).

$Q - P = 4\hat{i} - 10\hat{j}$

$$t = \frac{N\cdot(P-E_i)}{-N\cdot(Q-P)}$$

Edge (CW)	Normal	$P-E_i$	Denominator	t	
AB $(8\hat{i}-5\hat{j})$	$5\hat{i}+8\hat{j}$	$8\hat{i}+6\hat{j}$	60	22/15	× (>1)
BC $(11\hat{i}+8\hat{j})$	$-8\hat{i}+11\hat{j}$	$11\hat{j}$	142	121/142 $(o\to i)$	✓
CD $(-13\hat{i}-14\hat{j})$	$14\hat{i}-13\hat{j}$	~~$25\hat{i}+22\hat{j}$~~ $-11\hat{i}+3\hat{j}$	-186	193/186	× (>1)
DA $(-6\hat{i}+11\hat{j})$	$-11\hat{i}-6\hat{j}$	~~$18\hat{i}+7\hat{j}$~~ $2\hat{i}+17\hat{j}$	-16	$\frac{124}{16} > 1$	×

Sp

P	121/142	Q	
-1	0	-1	

SP: Edge 2 (RS) : $-8\hat{i}-4\hat{j}$

$R - S = -8\hat{i} - 4\hat{j}$

Edge (CW)	Normal (N)	$R-E_i$	Denominator	t
AB $(8\hat{i}-5\hat{j})$	$5\hat{i}+8\hat{j}$	$7\hat{i}-2\hat{j}$	72 (Ent) $(o\to i)$	0.263 (19/72) (3)
BC $(11\hat{i}+8\hat{j})$	$-8\hat{i}+11\hat{j}$	$-\hat{i}+3\hat{j}$	-20 $(i\to o)$	$-41/20$ ×
CD $(-13\hat{i}-14\hat{j})$	$14\hat{i}-13\hat{j}$	$-12\hat{i}-5\hat{j}$	60 $(o\to i)$	$-103/60$ ×
DA $(-6\hat{i}+11\hat{j})$	$-11\hat{i}-6\hat{j}$	$\hat{i}+9\hat{j}$	-112 (Exit) $(i\to o)$	65/112 (0.58) (4)

SP : to be clipped
CW : against which to be clipped

in SP ⟵ [i → o (1)
o → i (0)

SP Tagging Traversal

SP									
P	① 151/143	Q	②	R	③	④	S	⑤	⑥
-1	0	-1	1	-1	0	1	-1	0	1
~~0~~1	~~0~~ 1	~~0~~1	~~0~~ 1	~~0~~1	1 ~~0~~	~~0~~1	~~0~~1	~~0~~1	~~0~~1

$\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$ $\boxed{5}$ $\boxed{6}$

CW

A	⑥	③	B	②	①	C	D	④	⑤
-1	0	1	-1	0	1	-1	-1	0	1
0	1 ~~0~~	~~0~~1	0	1 ~~0~~	~~0~~1	0	0	~~0~~1	~~0~~1

$\boxed{1}$, $\boxed{2}$ - - -
↓
control transfers

<u>O/P:</u> line (①, Q)
line (Q, ②)
line (②, ①)
line (③, ④)
line (④, ⑤)
line (⑤, ⑥)
line (⑥, ③)

* Search for tagging value 0 in SP. (Before that, make traversal value 1)
* -1 in SP ⟶ continue to next pt.
* 1 in SP ⟶ Switch control to CW list.

19/02/2018

~~Polygon Clipping~~ Polygon Filling

Polygon
- → Non-horizontal (no horizontal edges) (case I)
- → Non-vertical (No vertical edges) (case II)

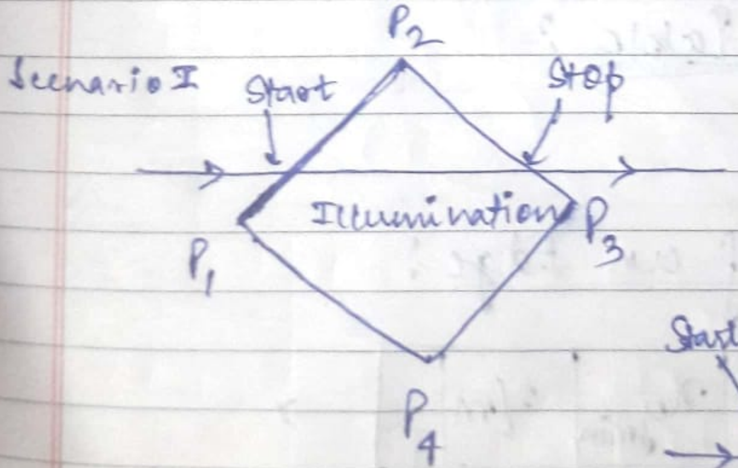
Scan line in filling the polygon.

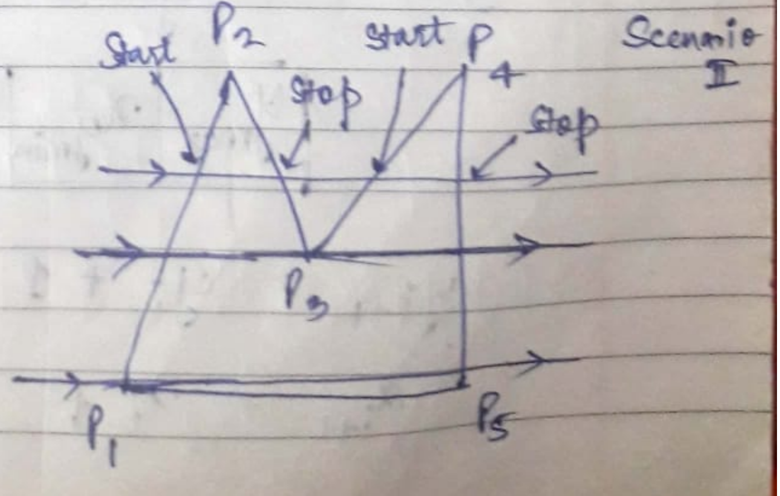
- → Horizontal scanning
- → vertical Scanning

① Horizontal Scanning: (non-horizontal edges)

Start Scanning: (either L→R or R→L)

Illumination starts when an edge is encountered & stops when another edge is encountered.

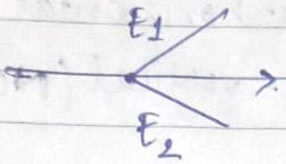




→ Vertices are formed from 2 consecutive edges

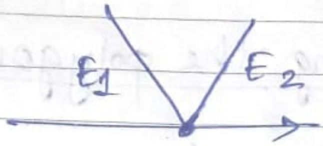
→ when a scan line passes through a vertex

Single encounter w.r.t. 2 edges

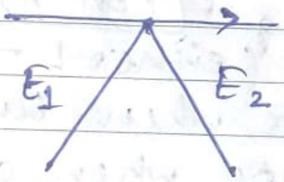
(a) 

$E_1 \cdot y_{min} = E_2 \cdot y_{max}$ (either side of scan line)

Double encounter w.r.t. 2 edges

(b) 

$E_1 \cdot y_{min} = E_2 \cdot y_{min}$

(c) 

$E_1 \cdot y_{max} = E_2 \cdot y_{max}$

Global Edge Table :

Structure of an Edge :

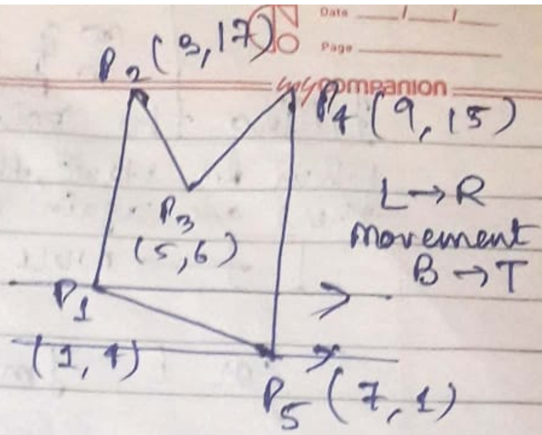
y_{max}	$x_{y_{min}}$	$1/m$	$\cdot \rightarrow$

$y_{i+1} = y_i + 1$ (B → T) (L → R)

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

$P_1 \; P_5$

1	7	-2	→



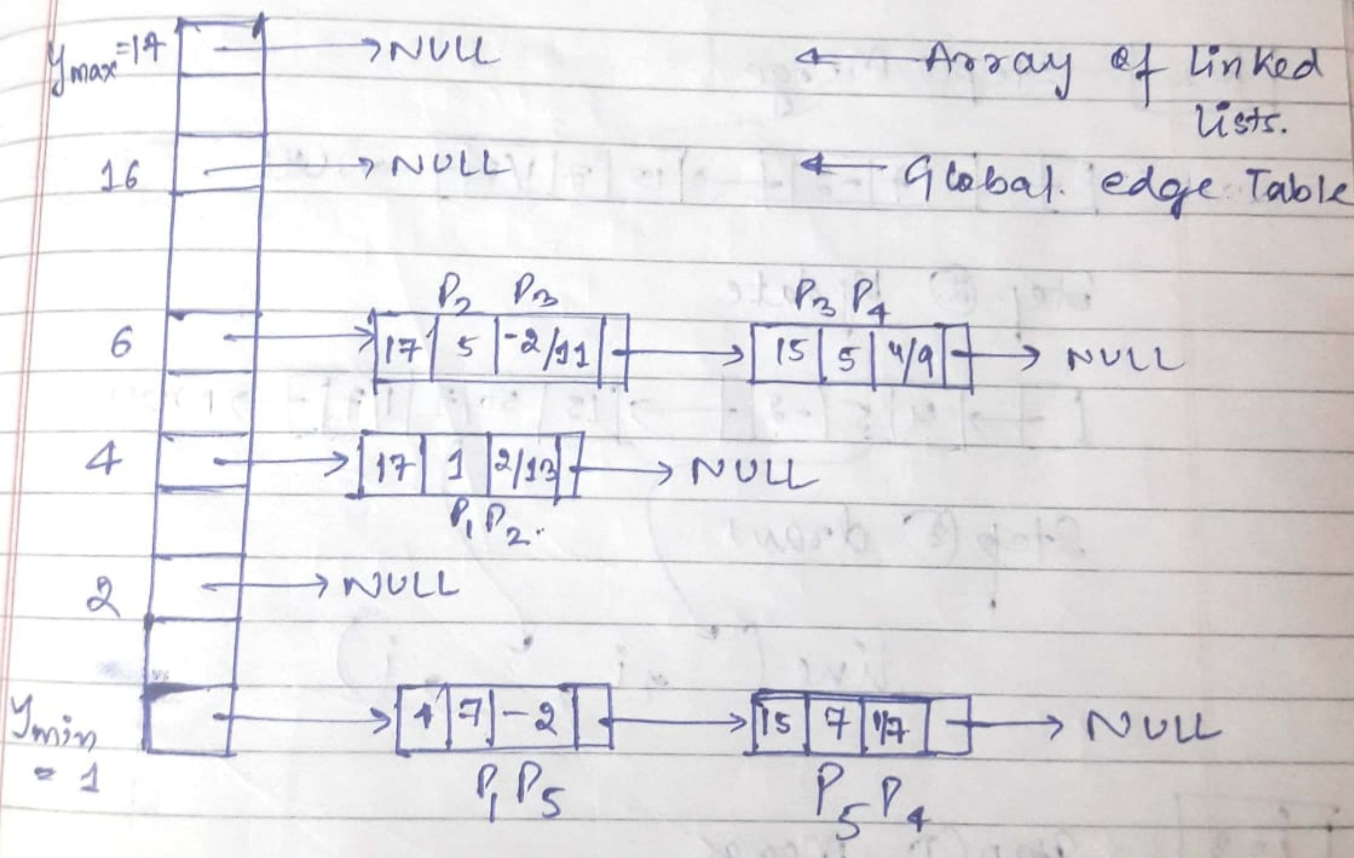
L→R movement B→T

T → B (and R → L):

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

$$x_{i+1} = x_i - 1/m$$

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



Global Edge Table

Steps:
① Merge ② update (if the nodes are previous) $x_{i+1} = x_i \pm 1/m$
③ if i = ymax of any node, ~~DELETE THAT~~ NODE
④ Draw

Filling the polygon (Scan line):

when we start, there are no edges & when we end. there are again no edges.

[ ] → NULL (start)

for i = ymin to ymax

i = 1

[ ] → [4 | 7 | -2] → [15 | 7 | 1/7] → NULL

line(, i, , i)

i = 2

Step ① Merge

[ ] → [4 | 7 | -2] → [15 | 7 | 1/7] → NULL

Step ② Update

[ ] → [4 | 5 | -2] → [15 | 50/7 | 1/7] → NULL

Step ③ draw

line(, i, , i)

i = 3

Step ① Merge

Step ② Update

[ ] → [4 | 3 | -2] → [15 | 51/7 | 1/7] → NULL

Step ③ draw

line(, i, , i)

$\boxed{i = 4}$ Step ① Merge

(sorted acc. using x) (L → R)

[ ] → [17 | 1 | 2/13 |] → [4 | 3 | -2 |] → [15 | 51/7 | 1/7 |] → NULL

Step ② update

[ ] → [17 | 1 | 2/13 |] → [~~4 | 1 | -2~~ |] → [15 | 52/17 | 1/7 |] → NULL

(Delete)
($i = y_{max}$)

[ ] → [17 | 1 | 2/13 |] → [15 | 52/17 | 1/7 |] → NULL

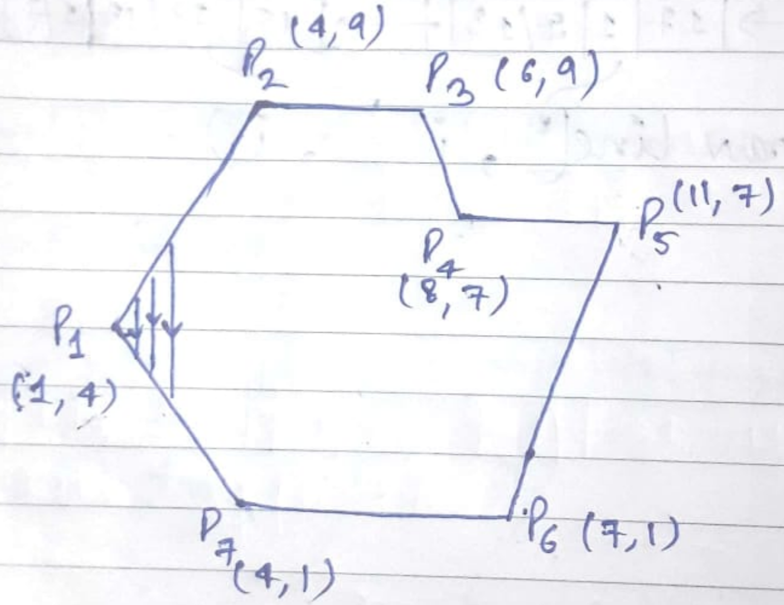
Step ③ draw line(, i , , i)

20/02/2018

Polygon Filling

Scan Conversion
- → Horizontal Scanning
- → vertical Scanning

Vertical Scanning:
(No vertical edges)



Top to Bottom Movement from L→R

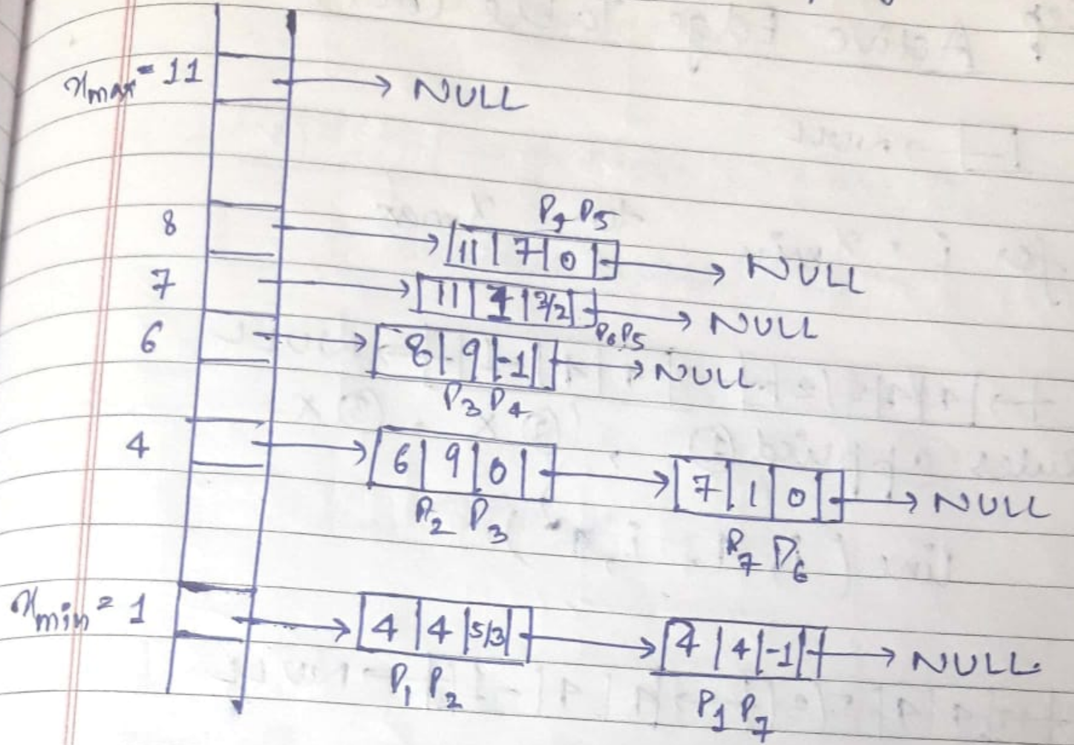
Step ①

Edge-Structure

x_{max}	$y\,x_{min}$	m	→

Edge node

Step ②: Construction of global edge table



Step ③: Filling of polygon (Active Edge Table)

Rules (for processing Active Edge Table):

① Merge (Active edge list at an instant with global edge table).

② Update (only those edges which are old one).

$$x_{i+1} \rightarrow x_i + 1 \; ; \; y_{i+1} = y_i \pm m$$

③ Delete (only that edge whose first member element matches at that instant).

④ Draw the scan line (vertical).

Active Edge Table (AET)

[ ] → NULL

for i = x_{min} to x_{max}

i = 1   [ ] → [4 | 4 | 5/3 |] → [4 | 4 | -1 |] → NULL

Rules applied (1), (2) ✗, (3) ✗

line(i, 4, i, 4)

i = 2   [ ] → [4 | 4 | 5/3 |] → [4 | 4 | -1 |] → NULL

Rules applied (1)

(tm)

[ ] → [4 | 17/3 | 5/3 |] → [4 | 3 | -1 |] → NULL

Rule applied (2)

(3) ✗

line(i, 17/3, i, 3)

i = 3   [ ] → [4 | 17/3 | 5/3 |] → [4 | 3 | -1 |] → NULL

Rule applied (1)

[ ] → [4 | 22/3 | 5/3 |] → [4 | 2 | -1 |] → NULL

Rule applied (2)

(3) ✗

line(i, 22/3, i, 2)

(Sorted according y) (T→B)

i=4 [ ] → [6|9|0|] → [4|22/3|5/3|] → [4|2|-1|] → [7|1|0|] → NULL

Rule applied ①

[ ] → [6|9|0|] → [4|9|~~5/3~~|] → [4|~~1~~|~~-1~~|] → [7|1|0|] → NULL

Rule applied ②

[ ] → [6|9|0|] → [7|1|0|] → NULL

Rule applied ③

line(i, 9, i, 1)

i=5 [ ] → [6|9|0|] → [7|1|0|] → NULL

① merge ② update ③ X

line(i, 9, i, 1)

i=6 [ ] → [6|9|0|] → [8|9|-1|] → [7|1|0|] → NULL

① Merge

[ ] → ~~[6|9|0|]~~ → [8|9|-1|] → [7|1|0|] → NULL

② update Delete

③ Deletion: [ ] → [8|9|-1|] → [7|1|0|] → NULL

line(i, 9, i, 1)

① merge

i=7 [ ] → [8|9|-1|] → [7|1|0|] → [11|1|3/2|] → NULL

② update [ ] → [8|8|-1|] → ~~[7|1|0|]~~ → [11|1|3/2|] → NULL

③ Delete [ ] → [8|8|-1|] → [11|1|3/2|] → NULL

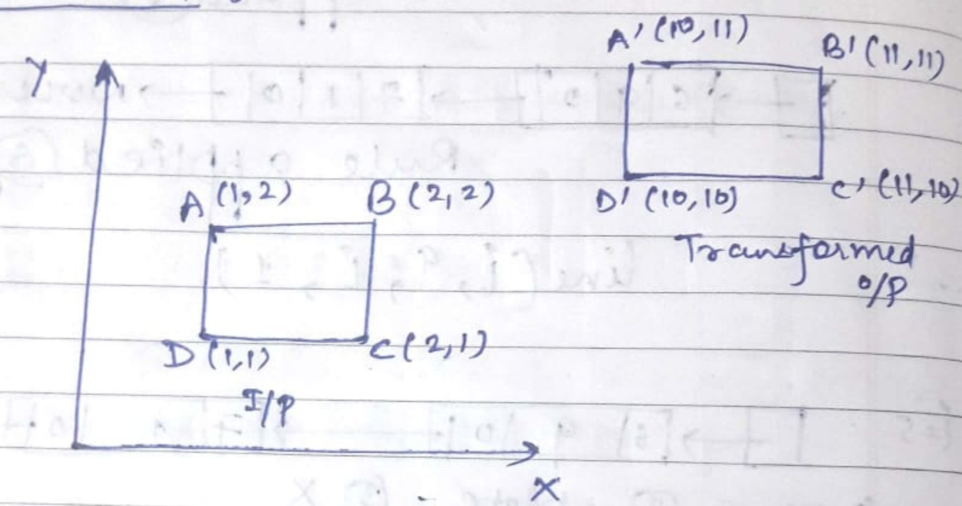
line(i, 8, i, 1)

22/02/2018

2D TRANSFORMATION

Translation Scaling Rotation

TRANSLATION :

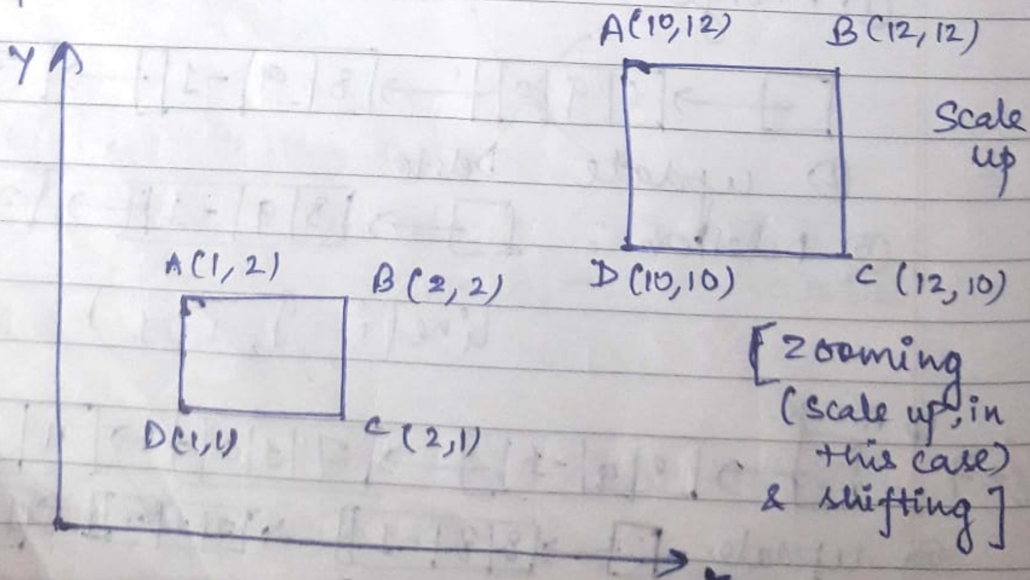


$x' = x + \Delta x$

$y' = y + \Delta y$

+ : Transformation operator

SCALING :



S_x : Scaling factor in x- direction
S_y : Scaling factor in Y- direction

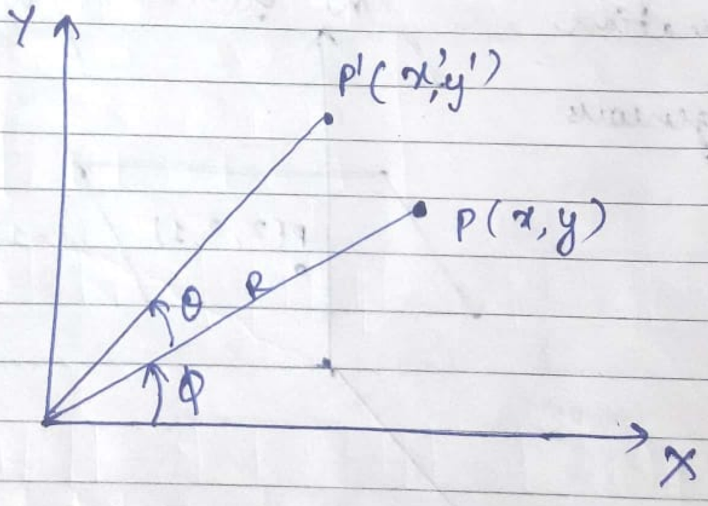
$$X' = (X-1) S_x + \Delta x + 1$$

$$Y' = (Y-1) S_y + \Delta y + 1$$

Transformation operator : $\pm$, $*$ → Scaling

$\pm$ → Translation

ROTATION:
(can be about any pt.)



$x = R\cos\phi$
$y = R\sin\phi$

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

$$x' = R\cos\theta\cos\phi - R\sin\theta\sin\phi$$

~~y' = R~~

$$\Rightarrow \boxed{x' = x\cos\theta - y\sin\theta}$$

$y' = R\sin\theta\cos\phi + R\cos\theta\sin\phi$

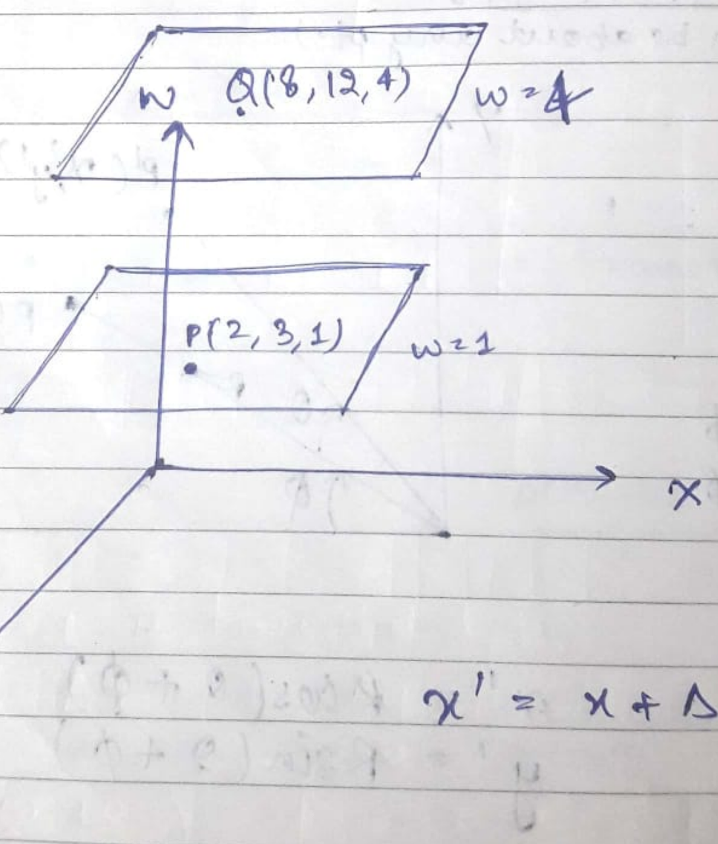
$$\boxed{y' = x\sin\theta + y\cos\theta}$$

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

<u>Transformation operator</u> : *

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

We need to make the transformation operator homogeneous



$x' = x + \Delta x$

Instead of taking 2-D, represent the pt. in a homogeneous coordinate system.
$\rightarrow [x/w, y/w, 1]$

Translation:

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

Every pt. in 2D can be taken as a pt. in 3D with $w=1$ in a plane $w=1$.

Scaling:

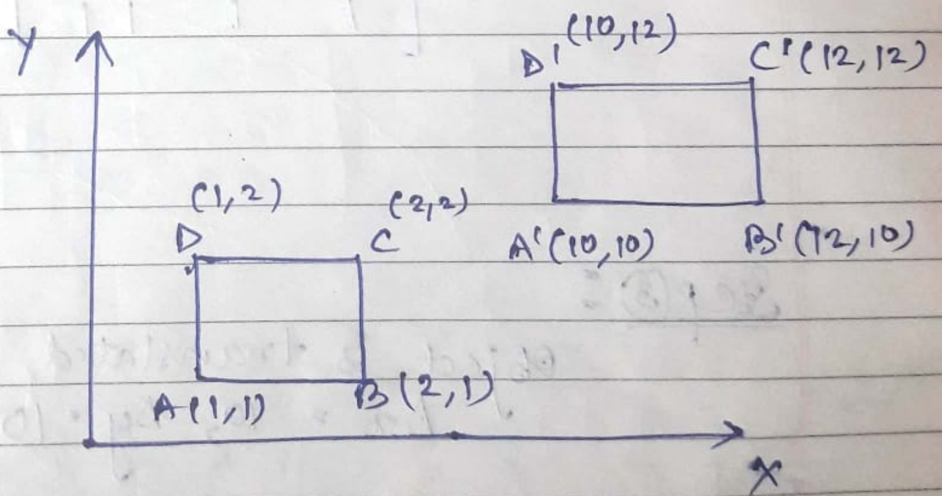
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Rotation:

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

Homogenous transformation operator : *

EXAMPLE 1:



Step ①: A to (0,0)

Object is brought to origin

bcz scaling is w.r.t the origin

$$\underset{\text{(Translation) matrix}}{T_1} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 1 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Scanned by CamScanner

$$T = \begin{array}{c} \begin{array}{cccc} A' & B' & C' & D' \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Step ②:

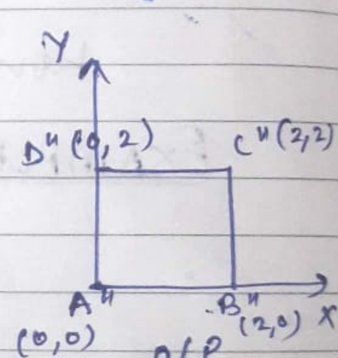
Object to be scaled up

$S_x = 2,\ S_y = 2$

Scaling matrix

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{array}{c} \begin{array}{cccc} A' & B' & C' & D' \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$= \begin{array}{c} \begin{array}{cccc} A'' & B'' & C'' & D'' \end{array} \\ \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$



O/P

Step ③:

Object is translated

$\Delta x = 10,\ \Delta y = 10$

$$\begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Translation Matrix (T_2)

$$\Rightarrow \begin{array}{cccc} A''' & B''' & C''' & D''' \end{array}$$
$$\begin{bmatrix} 10 & 12 & 12 & 10 \\ 10 & 10 & 12 & 12 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$C = \underbrace{\begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{pmatrix}}_{T_2} \times \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{S} \times \underbrace{\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}}_{T_1}$$

$\longleftarrow$

$$C = \begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 2 & 0 & -2 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

$\longrightarrow$ ~~S x~~ $T_1 \times S$

$$\underset{\text{(composite matrix)}}{C} = \begin{pmatrix} 2 & 0 & 8 \\ 0 & 2 & 8 \\ 0 & 0 & 1 \end{pmatrix}$$

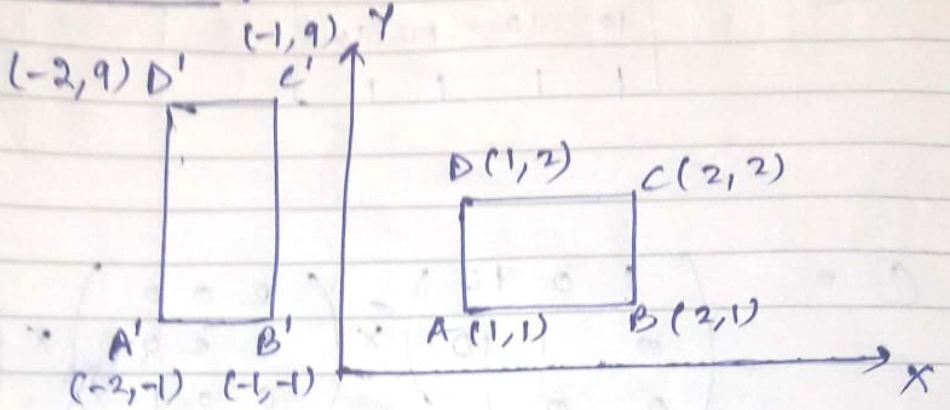
$$(C) \longrightarrow \begin{pmatrix} 2 & 0 & 8 \\ 0 & 2 & 8 \\ 0 & 0 & 1 \end{pmatrix} * \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 1 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \longleftarrow \text{(I/P)}$$

$\Downarrow$

$$O/P = \begin{array}{c} \begin{array}{cccc} A''' & B''' & C''' & D''' \end{array} \\ \begin{bmatrix} 10 & 12 & 12 & 10 \\ 10 & 10 & 12 & 12 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$\Rightarrow \boxed{O/P = C * I/P}$

EXAMPLE 2 :



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

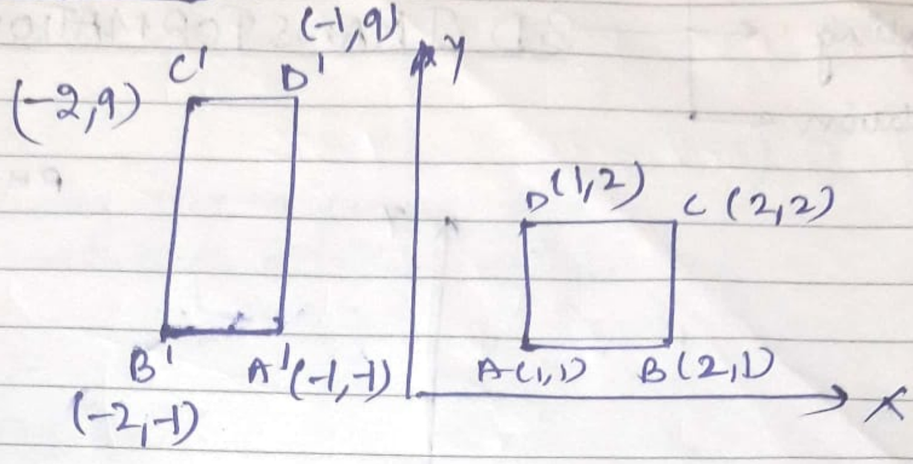
$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C = T_2 * S * T_1$$

$$C = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 10 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

EXAMPLE 3:

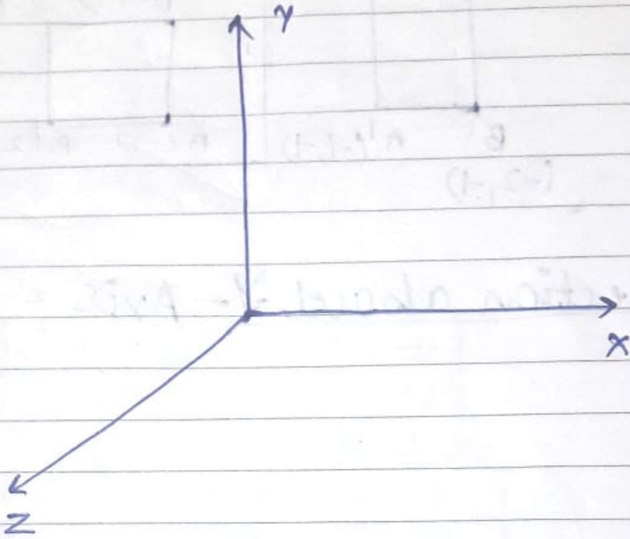


Reflection about Y-Axis = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

13/03/2018

Rotation
Scaling ← **3D TRANSFORMATION**
Translation

RHCS (Right Handed coordinate System)

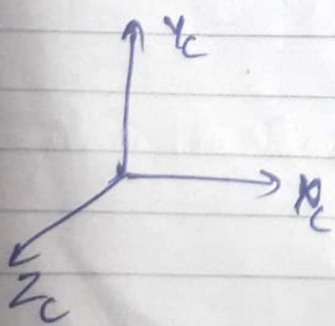


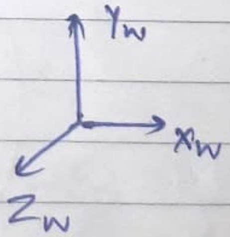
Camera Analogy:

Camera

object

(Camera Coord. system)

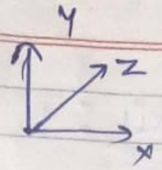




world coordinate system (for the object) — taken as RHCS

Object is referred w.r.t. world coordinate system

LHCS



→ z is along the viewing dir^n
→ eye is always at the origin

Translation :

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

Scaling:

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

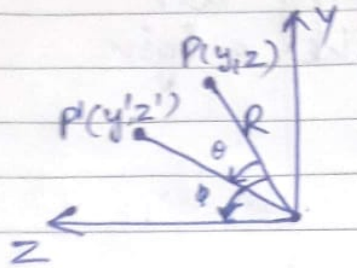
Rotation :

- Rotation about X-axis
- Rotation about Y-axis
- Rotation about Z-axis

Any rotation in anticlockwise dir^n is considered as +ve.

Rotation about X-axis :

Anticlockwise : $Y \rightarrow Z$



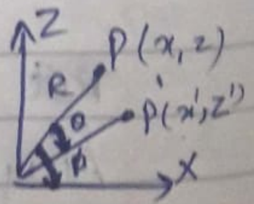
~~$y' = R\cos\phi, \; z' = R$~~

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

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

Rotation about Y-axis :

Anticlockwise rotation : $Z \rightarrow X$



$$y' = y$$

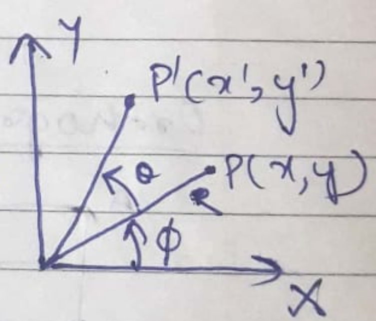
$$\cancel{x = R\cos\phi}, \; y = R\sin\phi$$
$$\cancel{x = R\cos}$$

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

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

Rotation about Z-axis:

Anticlockwise rotation: $X \rightarrow Y$



$$x' = R\cos(\theta + \phi)$$
$$x' = R\cos\theta\cos\phi - R\sin\theta\sin\phi$$
$$\Rightarrow x' = x\cos\theta - y\sin\theta$$

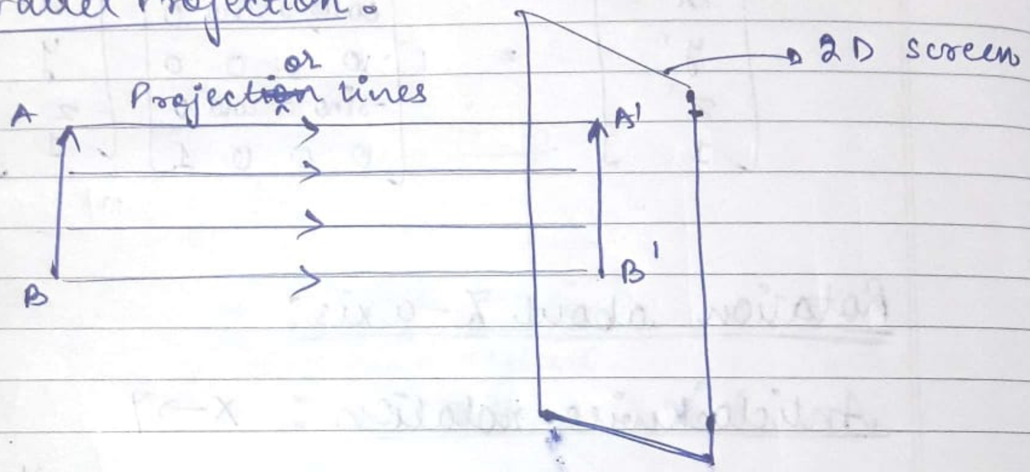
$$y' = R\sin(\theta + \phi)$$
$$y' = R\sin\theta\cos\phi + R\cos\theta\sin\phi$$
$$\Rightarrow y' = x\sin\theta + y\cos\theta$$

$$z' = z$$

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

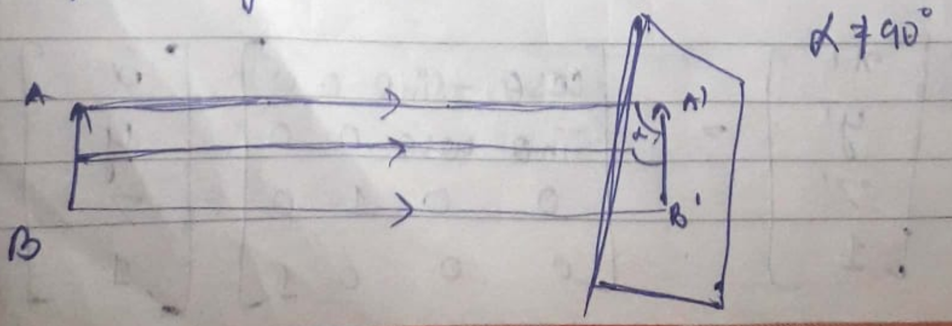
- Projection
  - Parallel
    - Orthographic
      - Isometric
      - Dimetric
      - Trimetric
    - Oblique
      - Cabinet
      - Cavalier
  - Non-Parallel

Parallel Projection :



Orthographic Projection :

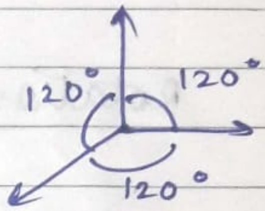
Oblique Projection :



Isometric Projection :

Parallel projection or lines falling on the screen at an angle of 90°. These lines make equal angle with all the 3 coordinate axes.

If these coordinate axis are projected on a plane, then these axis will make 120° angle with each other.



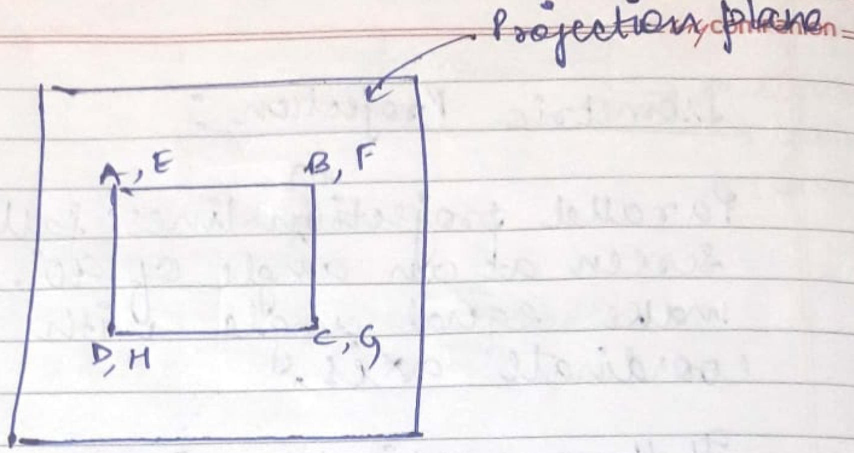
Dimetric Projection:

Orthogonal Projection + Projector Lines make equal angles only with 2-axis.

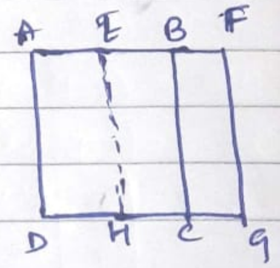
Trimetric Projection:

orthogonal Projection + Projector lines don't make equal angles with any axis.

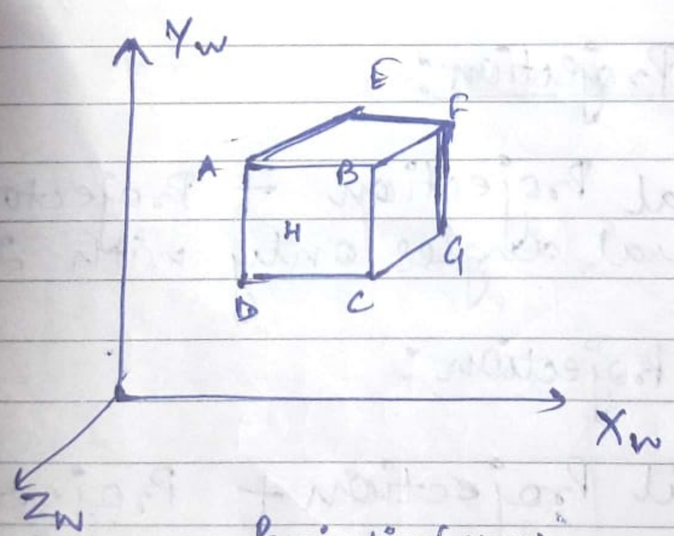
15/03/20



$\boxed{\text{Cube}}$



Rotation about y-axis :



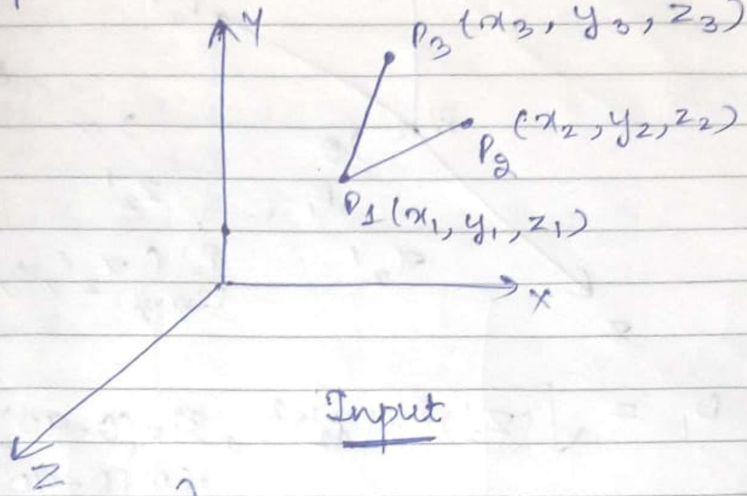
Projection(x-y)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A \; B \; C \; D - - - H \\ \\ \\ \\ \end{bmatrix}$$

15/03/2018

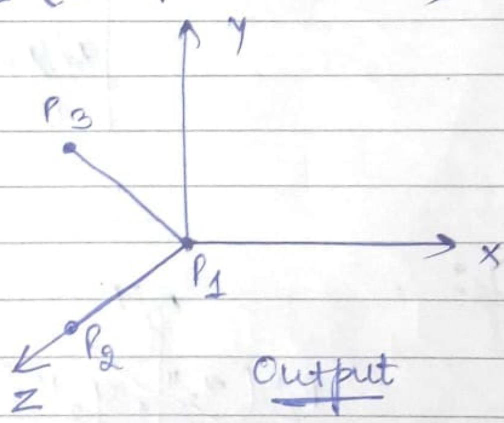
3D TRANSFORMATION

Example:



Input

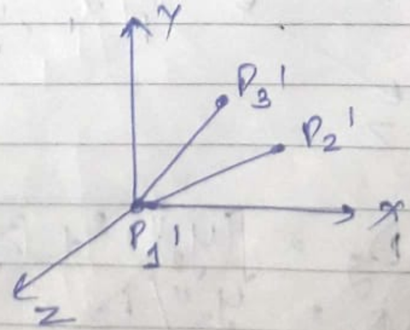
C (composite matrix, to be derived)



P_1P_2 along z-axis & P_1P_3 ~~along~~ is on y-z plane.

Output

Step ①:



$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P_1' = (0, 0, 0)$

$P_2' = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \Rightarrow (x_2', y_2', z_2')$

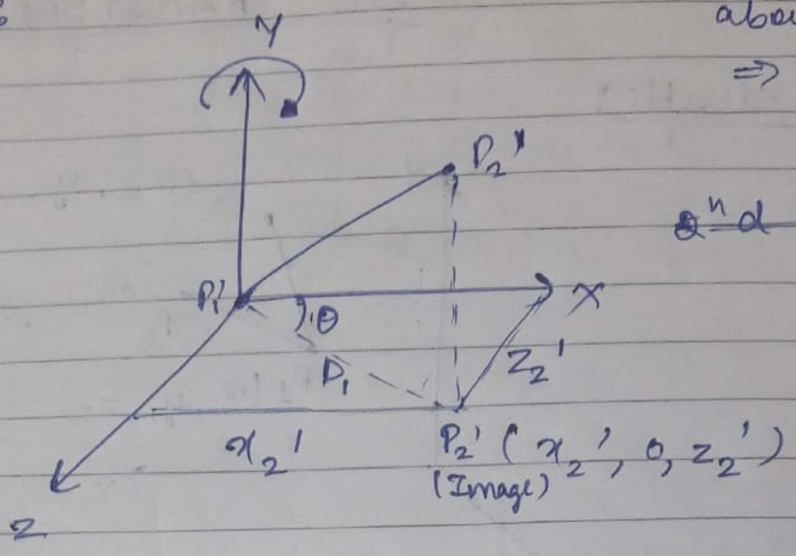
$P_3' = (x_3 - x_1, y_3 - y_1, z_3 - z_1) \Rightarrow (x_3', y_3', z_3')$

(clockwise)

1st Rotation:
about y-axis
$\Rightarrow P_1' P_2'$ on
yz plane

~~2nd Rota~~

Step ②:



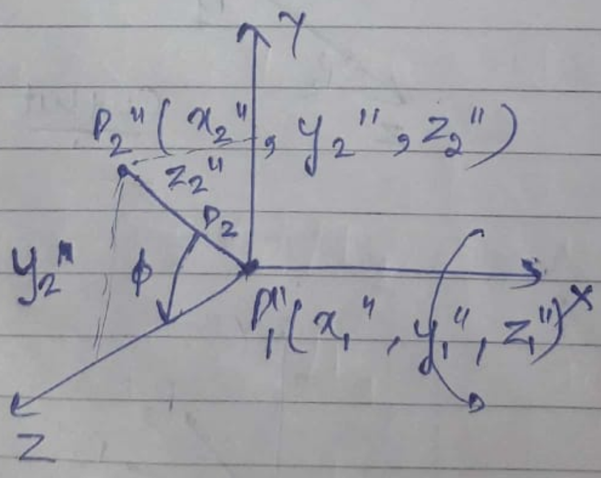
$D_1 = \sqrt{(z_2')^2 + (x_2')^2}$, $\sin(\theta - 90) = \cos\theta = x_2'/D_1$

$\cos(\theta - 90) = \sin\theta = z_2'/D_1$

$$R_{y,-(90-\theta)} = \begin{bmatrix} z_2'/D_1 & 0 & -x_2'/D_1 & 0 \\ 0 & 1 & 0 & 0 \\ x_2'/D_1 & 0 & z_2'/D_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step ③:

2nd Rotation:
Rotation about x-axis
$\Downarrow$
$P_1'' P_2''$ along z-axis
(Anticlockwise)



$D_2 = \sqrt{(y_2'')^2 + (z_2'')^2}$

$\sin\phi = y_2''/D_2$

$\cos\phi = z_2''/D_2$

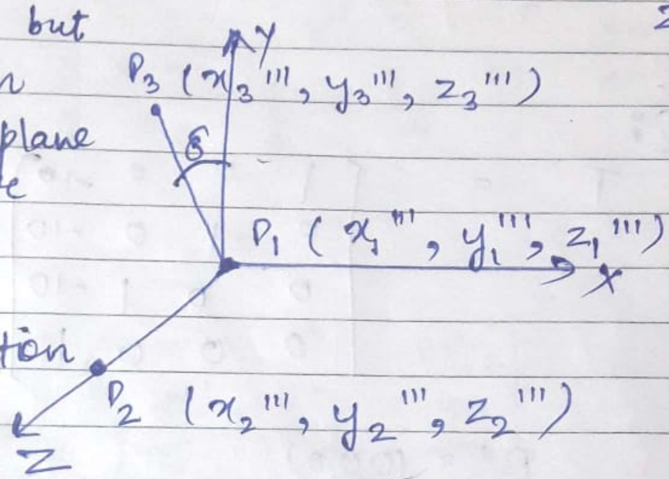
$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & z_2''/D_2 & -y_2''/D_2 & 0 \\ 0 & y_2''/D_2 & z_2''/D_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step ④ :

Rotation about Z-axis (anticlockwise)

Hold Z-axis & rotate about it.

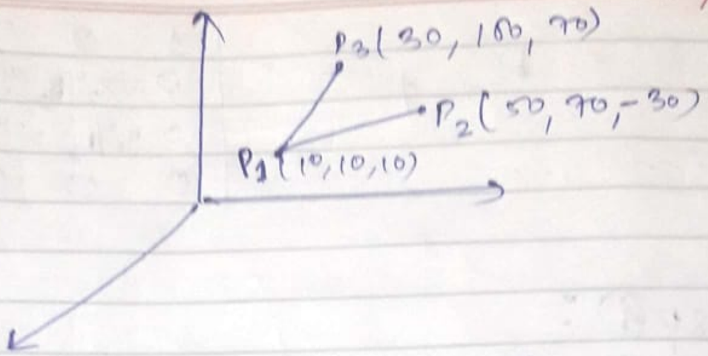
P_1P_2 now on Z-axis but P_1P_3 can now be on left or right of YZ plane (x for P_3 can be +ve also, -ve also)



Say δ angle rotation about Z-axis to bring P_3 in YZ plane. If already aligned, 0.

$$\boxed{C = R_{z,\delta} * R_{x,\phi} * R_{y,-\theta} * T}$$

Example:



Step ①:

$$T = \begin{bmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P_1' = (0,0,0)$
$P_2' = (40, 60, -40)$
$P_3' = (20, 90, 60)$

Step ②:

$$D_1 = \sqrt{1600 + 1600} = \sqrt{3200} = 10\sqrt{32}$$
$$= 40\sqrt{2}$$

$$\sin(\theta - 90^\circ) = -\cos\theta = -x_2'/D_1 = -40/40\sqrt{2} = -1/\sqrt{2}$$

$$\cos(\theta - 90^\circ) = \sin\theta = z_2'/D_1 = -40/40\sqrt{2} = -1/\sqrt{2}$$

$$R_y, -\theta = \begin{bmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{y,-\theta} = \begin{bmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 40 & 20 \\ 0 & 60 & 90 \\ 0 & -40 & 60 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\Downarrow$$

$$\Rightarrow \begin{bmatrix} 0 & 0 & -80/\sqrt{2} \\ 0 & 60 & 90 \\ 0 & 80/\sqrt{2} & -40/\sqrt{2} \\ 1 & 1 & 1 \end{bmatrix}$$

$P_1''(0,0,0)$

$P_2''(0, 60, 80/\sqrt{2}) \Rightarrow P_2''(0, 60, 40\sqrt{2})$

$P_3''(-80/\sqrt{2}, 90, -40/\sqrt{2})$

$\Rightarrow (-40\sqrt{2}, 90, -20\sqrt{2})$

Step (3):

$\cancel{D_2 = \sqrt{3600 + 8100} = \sqrt{11700}}$

$\cancel{= 10\sqrt{117}}$

$D_2 = \sqrt{3600 + 3200} = \sqrt{6800}$

$= 10\sqrt{68}$

$= 20\sqrt{17}$

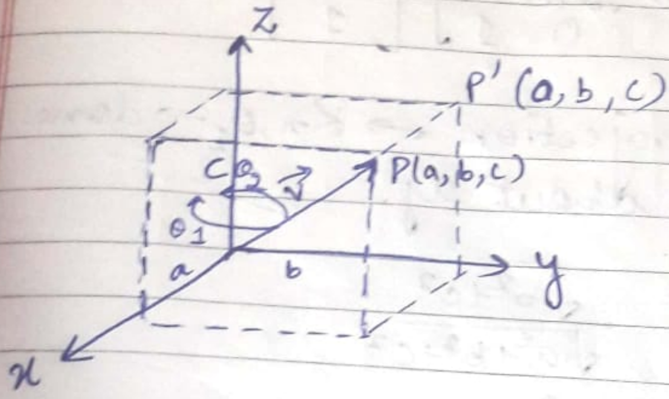
$\sin\phi = 60/20\sqrt{17} = 3/\sqrt{17}$

$\cos\phi = 40\sqrt{2}/20\sqrt{17} = \frac{2\sqrt{2}}{\sqrt{17}}$

$R_{x,\phi}$

19/03/2018

3D TRANSFORMATION (vector)

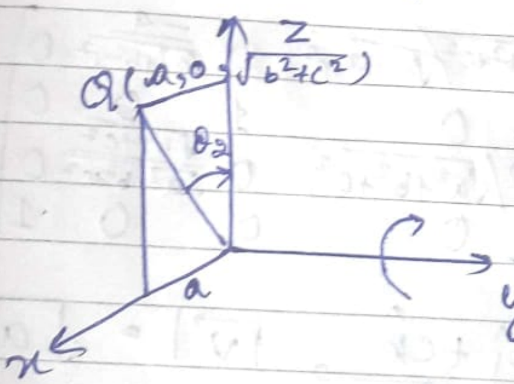


$A_{\vec{v}}^{k}$: Composite transformation to align the vector $\vec{v}$ along k vector

$$\vec{V} = a\hat{i} + b\hat{j} + c\hat{k}$$

$OP' \rightarrow$ Image of OP on yz plane

Projection in yz plane if Rotation about x.



$$D = \sqrt{a^2+b^2+c^2}$$

if Rotation about x-axis anticlockwise

$$\cos\theta_1 = c / \sqrt{b^2+c^2}$$
$$\sin\theta_1 = b / \sqrt{b^2+c^2}$$

$$R_{x,\theta_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/\sqrt{b^2+c^2} & -b/\sqrt{b^2+c^2} & 0 \\ 0 & b/\sqrt{b^2+c^2} & c/\sqrt{b^2+c^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{\sqrt{b^2+c^2}} & \frac{-b}{\sqrt{b^2+c^2}} & 0 \\ 0 & \frac{b}{\sqrt{b^2+c^2}} & \frac{c}{\sqrt{b^2+c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}$$

First part of projection $\rightarrow R_{x,\theta_1} \rightarrow$ done

Now, rotation about y.

$$\cos(-\theta_2) = \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}}$$

$$\sin(-\theta_2) = \frac{-a}{\sqrt{a^2+b^2+c^2}}$$

$$R_{y,-\theta_2} = \begin{bmatrix} \sqrt{b^2+c^2}/\sqrt{a^2+b^2+c^2} & 0 & \frac{-a}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 1 & 0 & 0 \\ a/\sqrt{a^2+b^2+c^2} & 0 & \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\vec{v} = a\hat{i} + b\hat{j} + c\hat{k}$, $|\vec{v}| = \sqrt{a^2+b^2+c^2}$

Let $\lambda = \sqrt{b^2+c^2}$

$$\Rightarrow \begin{bmatrix} \lambda/|v| & 0 & -a/|v| & 0 \\ 0 & 1 & 0 & 0 \\ a/|v| & 0 & \lambda/|v| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Leftarrow R_{y,-\theta_2}$$

$$R_{x,\theta_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/\lambda & -b/\lambda & 0 \\ 0 & b/\lambda & c/\lambda & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

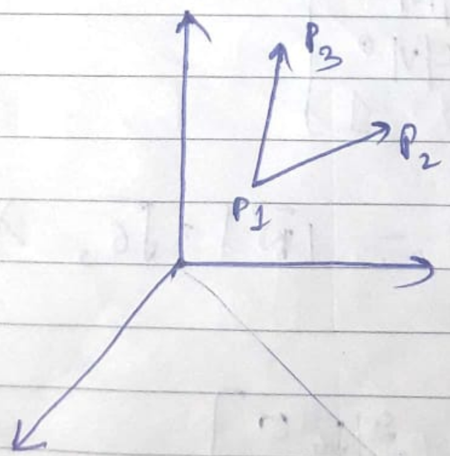
$$A_{\vec{v}}^{k} = \begin{bmatrix} \lambda/|v| & 0 & -a/|v| & 0 \\ 0 & 1 & 0 & 0 \\ a/|v| & 0 & \lambda/|v| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/\lambda & -b/\lambda & 0 \\ 0 & b/\lambda & c/\lambda & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{\vec{v}}^{k} = \begin{bmatrix} \lambda/|v| & -ab/\lambda|v| & -ac/\lambda|v| & 0 \\ 0 & c/\lambda & -b/\lambda & 0 \\ a/|v| & b/|v| & c/|v| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question 1 Align vector $\vec{v} = \hat{i} + \hat{j} + \hat{k}$ with the vector K.

$|\vec{v}| = \sqrt{3}, \quad \lambda = \sqrt{2}$

$$A_{\vec{v}}^{k} = \begin{bmatrix} \sqrt{2}/3 & -1/\sqrt{6} & -1/\sqrt{6} & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Now, we can directly use the transformation matrix to align P_1P_2 along z-axis

Alignment of K to vector $\vec{v}$: Reverse of Previous

$$A_K^{\vec{v}} = \left[A_{\vec{v}}^{K}\right]^{-1}$$

$$= \left[R_{y,-\theta_2} \times R_{x,\theta_1}\right]^{-1}$$

$$= \left[R_{x,\theta_1}\right]^{-1} \times \left[R_{y,-\theta_2}\right]^{-1}$$

$$= R_{x,-\theta_1} * R_{y,\theta_2}$$

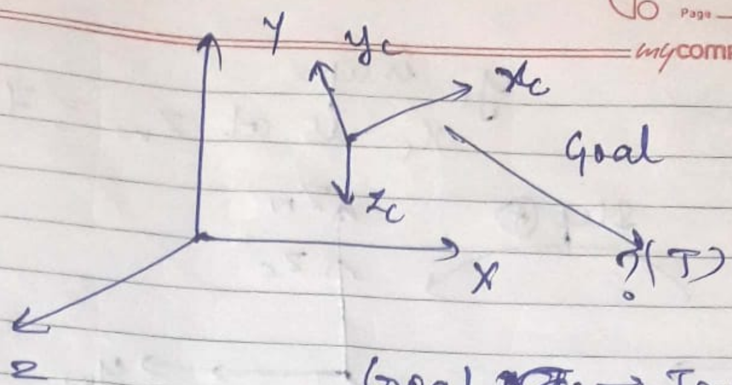
<u>Question 2</u> Align vector $\vec{v} = \hat{i}+\hat{j}+\hat{k}$ with the vector $\vec{N} = 2\hat{i}-\hat{j}-\hat{k}$

$$A_{\vec{v}}^{\vec{N}} = A_K^{\vec{N}} \times A_{\vec{v}}^{K}$$

$$\begin{bmatrix} \lambda/|v| & 0 & a/|v| & 0 \\ -ab/\lambda|v| & c/\lambda & b/|v| & 0 \\ -ac/\lambda|v| & -b/\lambda & c/|v| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Leftarrow A_K^{\vec{v}}$$

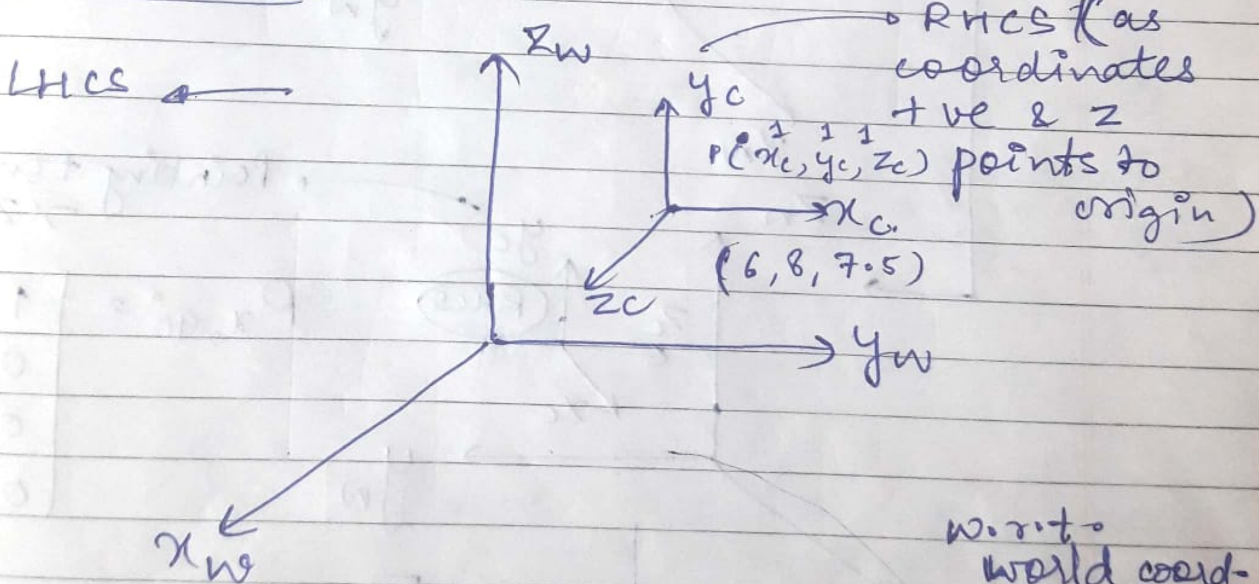
$$\vec{N} = 2\hat{i}-\hat{j}-\hat{k}, \quad |\vec{N}| = \sqrt{6}, \quad \lambda_N = \sqrt{2}$$

$$A_K^{\vec{N}} = \begin{bmatrix} 1/\sqrt{3} & 0 & 2/\sqrt{6} & 0 \\ 2/\sqrt{12} & -1/\sqrt{2} & -1/\sqrt{6} & 0 \\ 2/\sqrt{12} & 1/\sqrt{2} & -1/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Goal → Transformation matrix to convert coordinate axis to x_c, y_c, z_c
↓
To see how an object lookes in different coordinate axes.

EXERCISE (2):



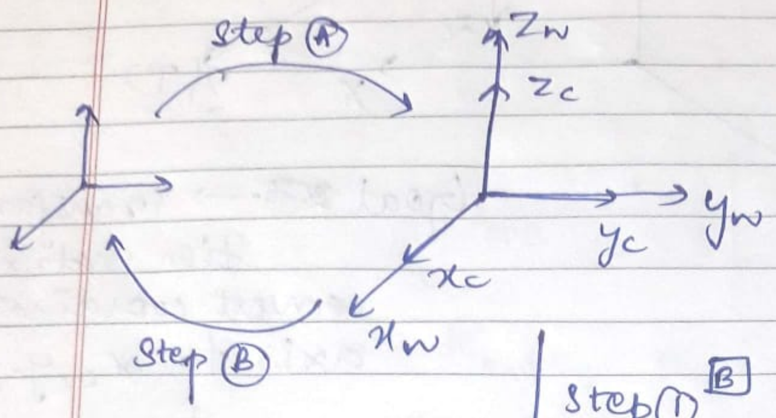
w.r.t. world coord-system

A camera is positioned at (6, 8, 7.5). A Point P is (1, 1, 1) w.r.t. camera coordinate ~~axis~~ system. What will be point P w.r.t world coordinate system. Given view direction is towards origin of WCS.

Scanned by CamScanner

y_c is up

x_c is at $z_w = 7.5$



Step (A) & Step (B) are inverse. Finding step (B) can give for Step (A)

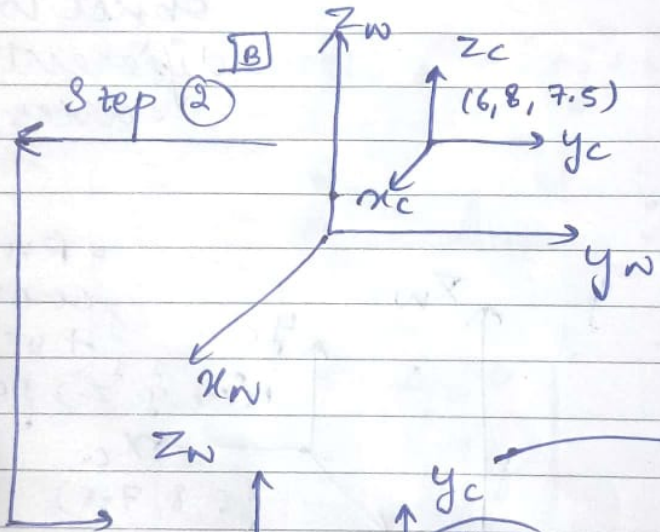
(1) Translation

(2) Rotate along x_c so that ~~z~~ pts. towards $-y$

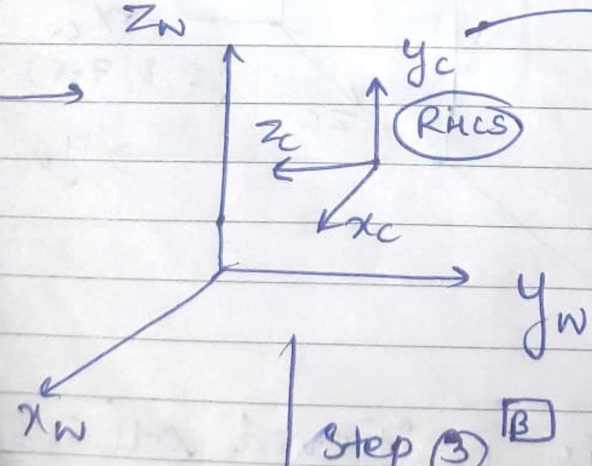
(3) Rotate to point z_c towards z axis (edge)

(4) Rotate to point z_c towards origin

Step (1) [B]



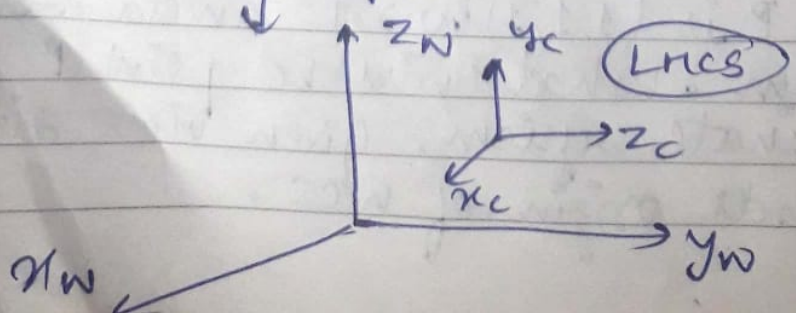
$$\begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 7.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

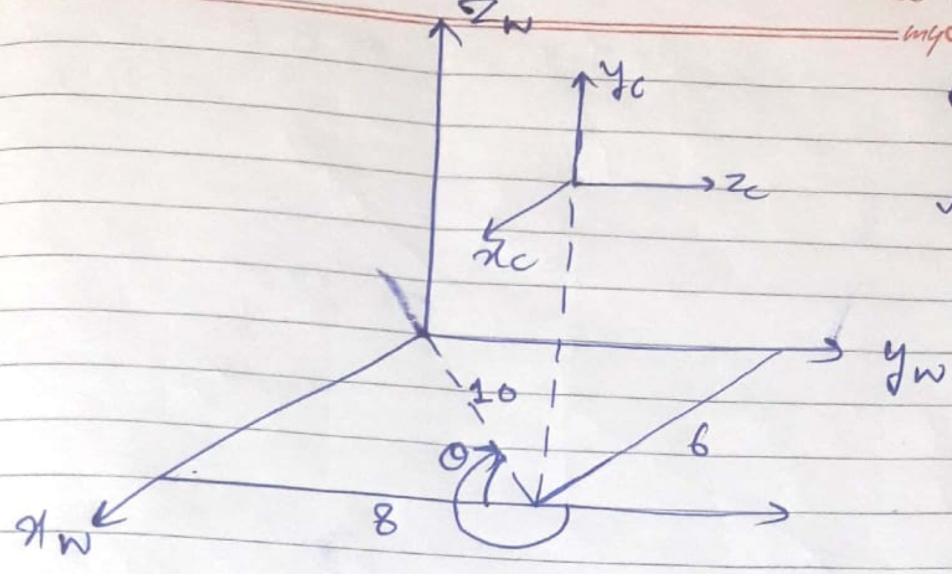


Pointing towards xz plane

$$R_{x,90^\circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

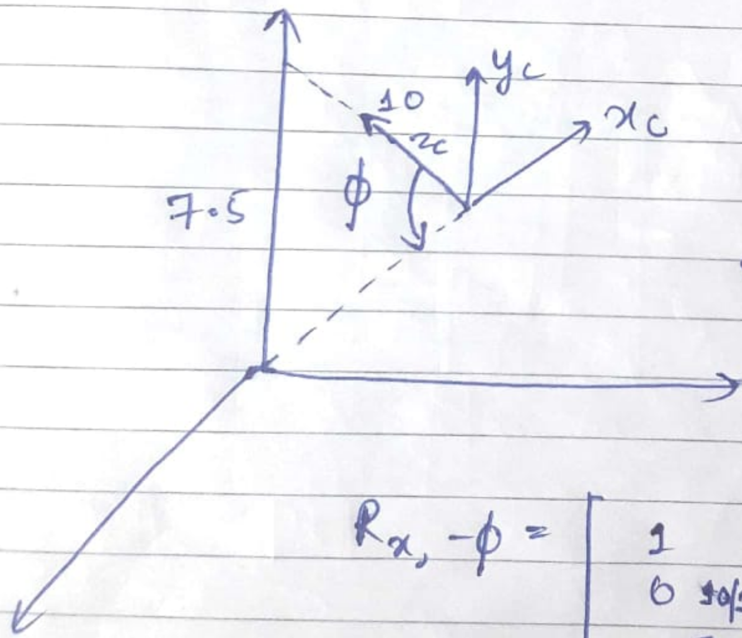
Step (3) [B]





Rotation is clockwise

$$R_{y, -(180+\theta)} = \begin{bmatrix} -8/10 & 0 & 6/10 & 0 \\ 0 & 1 & 0 & 0 \\ -6/10 & 0 & -8/10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotation is clockwise

$\sqrt{(7.5)^2 + (10)^2} = \underline{12.5}$

$$R_{x, -\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10/12.5 & 7.5/12.5 & 0 \\ 0 & 7.5/12.5 & 10/12.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

New viewing dir^n is towards origin

$\cos(-\phi) = 10/12.5$
$\sin(-\phi) = -7.5/12.5$

Camera → Close images : blurred
Too far images also blurred
There is a ~~vis~~ volume for clear vision.

Answer: $\frac{-13}{\sqrt{3}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{6}}$

Date ___/___/___

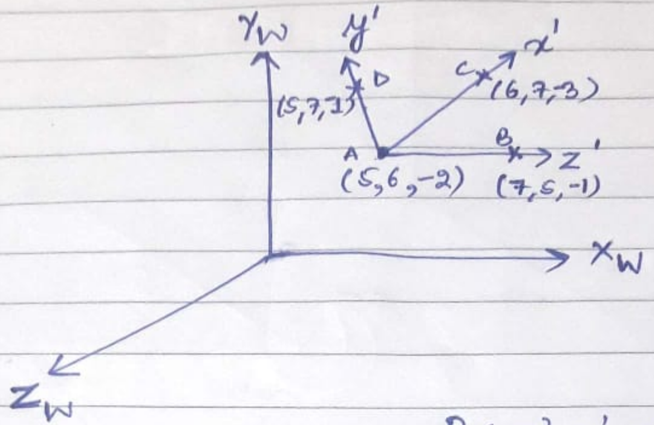
Page ___

myCOMPANION

02/04/2018

3D TRANSFORMATION

(Assignment 2)



$P(x', y', z') = \{1, 2, 3\}$

$P_W \rightarrow ?$

Step ①:

(Translation) $T = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Coordinates after translation:

$A' \rightarrow (0,0,0)$

$B' \rightarrow (2,-1,1)$

$C' \rightarrow (1,1,-1)$

$D' \rightarrow (0,1,1)$

$P' \rightarrow (-4,-4,5)$

Step ②: Align z' axis

$\vec{Z'} = 2\hat{i} - \hat{j} + \hat{k}$, $|z'| = \sqrt{6}$, $\lambda = \sqrt{2}$

$$A_{\vec{Z'}}^{\vec{Z_W}} = \begin{bmatrix} \lambda/|z'| & -ab/\lambda|z'| & -ac/\lambda|z'| & 0 \\ 0 & c/\lambda & -b/\lambda & 0 \\ a/|z'| & b/|z'| & c/|z'| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{\vec{Z'}}^{\vec{Z_w}} = \begin{bmatrix} \sqrt{2}/\sqrt{6} & 2/\sqrt{12} & -2/\sqrt{12} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 2/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 2/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>New Values</u> : $\Rightarrow$

$$\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 2/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{array}{c} \begin{array}{cccc} A' & B' & C' & D' \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$\Rightarrow \begin{bmatrix} 0 & 0 & \sqrt{3} & 0 \\ 0 & 0 & 0 & \sqrt{2} \\ 0 & \sqrt{6} & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(All the axes are aligned)

<u>Final Coordinates of P</u> :

$$\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 2/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \overset{P'}{\begin{bmatrix} -4 \\ -4 \\ 5 \\ 1 \end{bmatrix}}$$

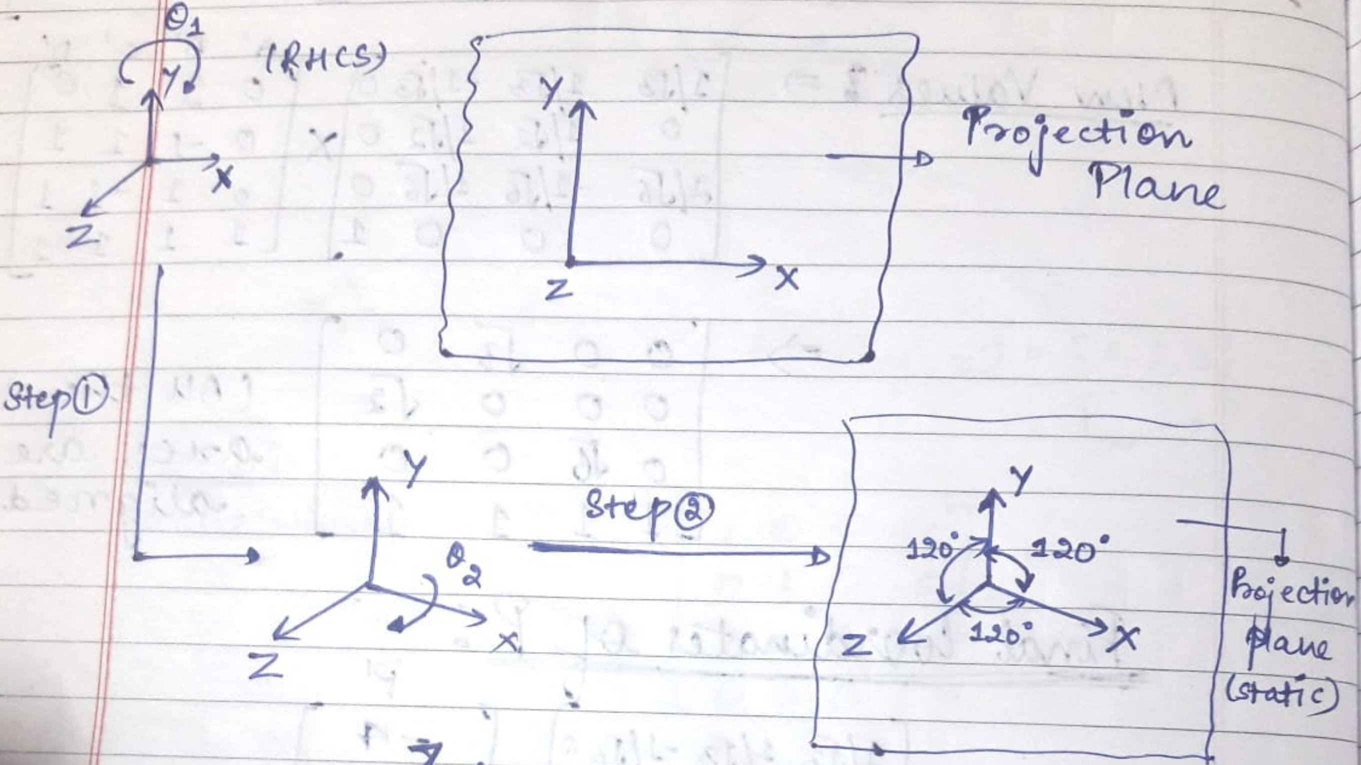
$$\Rightarrow \begin{bmatrix} -18/\sqrt{3} \\ 1/\sqrt{2} \\ 1/\sqrt{6} \\ 1 \end{bmatrix}$$

$$\boxed{P_W \longrightarrow \left(-13/\sqrt{3},\ 1/\sqrt{2},\ 1/\sqrt{6}\right)}$$

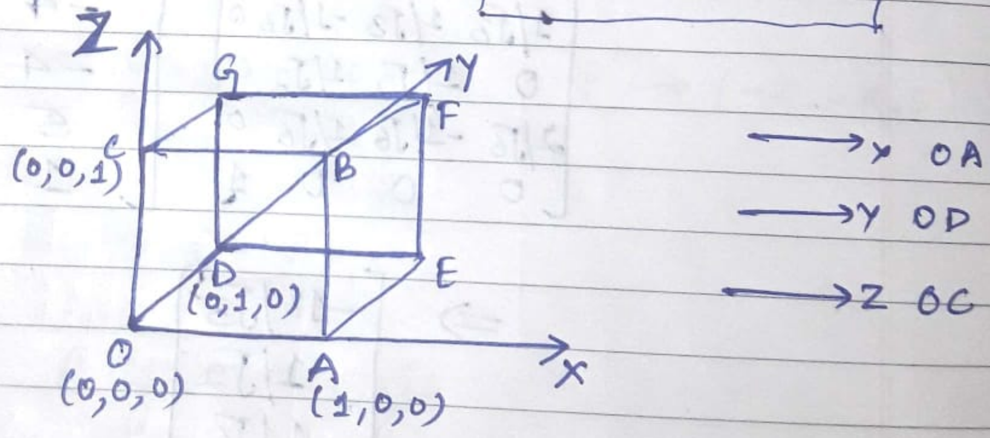
Scanned by CamScanner

03/04/2018

Isometric View



Unit Cube



After 2 rotations:

$$A' = (x_x, y_x, 0)$$

$$D' = (x_y, y_y, 0)$$

$$C' = (x_z, y_z, 0)$$

Scanned by CamScanner

$f_x \quad |OA| = 1$

$f_x' \quad |OA'| = \sqrt{x_x^2 + \cancel{y_y^2}\, y_x^2}$

Forshortening factor along x-axis $= \dfrac{f_x'}{f_x}$

<u>Property for Isometric projection</u> :

$$\boxed{\frac{f_x'}{f_x} = \frac{f_y'}{f_y} = \frac{f_z'}{f_z}}$$

$$R_{y,-\theta_1} = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{x,\theta_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = Proj_{x-y} * R_{x,\theta_2} * R_{y,-\theta_1}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ -\sin\theta_1\sin\theta_2 & \cos\theta_2 & -\sin\theta_2\cos\theta_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A' = (\cos\theta_1, -\sin\theta_1 \sin\theta_2, 0)$
$D' = (0, \cos\theta_2, 0)$
$C' = (-\sin\theta_1, -\sin\theta_2 \cos\theta_1)$

<u>Eqn (1):</u>

$$\frac{fx'}{fx} = \frac{fy'}{fy}$$

$fy' = |OD'| = \sqrt{\cos^2\theta}$
$fy = |OD| = 1$

$$\cos^2\theta_1 + \sin^2\theta_1 \sin^2\theta_2 = \cos^2\theta_2$$

<u>Eqn (2):</u>

$$\frac{fy'}{fy} = \frac{fz'}{fz}$$

$$\cos^2\theta_2 = \sin^2\theta_1 + \sin^2\theta_2 \cos^2\theta_1 .$$

<u>Add (1) & (2):</u>

$$1 + \sin^2\theta_2 \cos^2\theta_1 + \sin^2\theta_1 \sin^2\theta_2 = 2\cos^2\theta_2$$

$$1 + \sin^2\theta_2 (\cos^2\theta_1 + \sin^2\theta_1) = 2\cos^2\theta_2$$

$$1 + \sin^2\theta_2 = 2\cos^2\theta_2$$

$$\sin^2\theta_2 = -1 + 2\cos^2\theta_2$$

$$\cancel{1 - \cos^2\theta_2 = -1}$$

$$\cancel{2 = \cos^2\theta_2}$$

$$\cancel{\cos\theta_2 =}$$

$$1 - \cos^2\theta_2 = 2\cos^2\theta_2 - 1$$

$$2 = 3\cos^2\theta_2$$

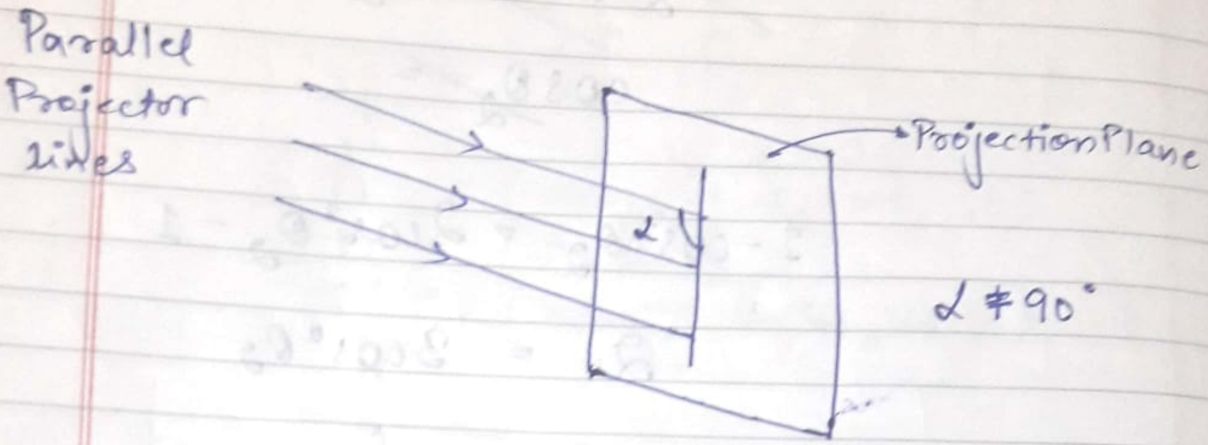
$$\cos\theta_2 = \pm\sqrt{\frac{2}{3}}$$

$$\boxed{\theta_2 = \pm 35.26}$$

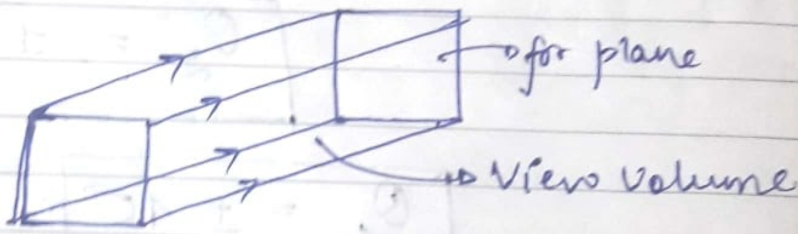
$$\boxed{\theta_1 = \pm 45^\circ}$$

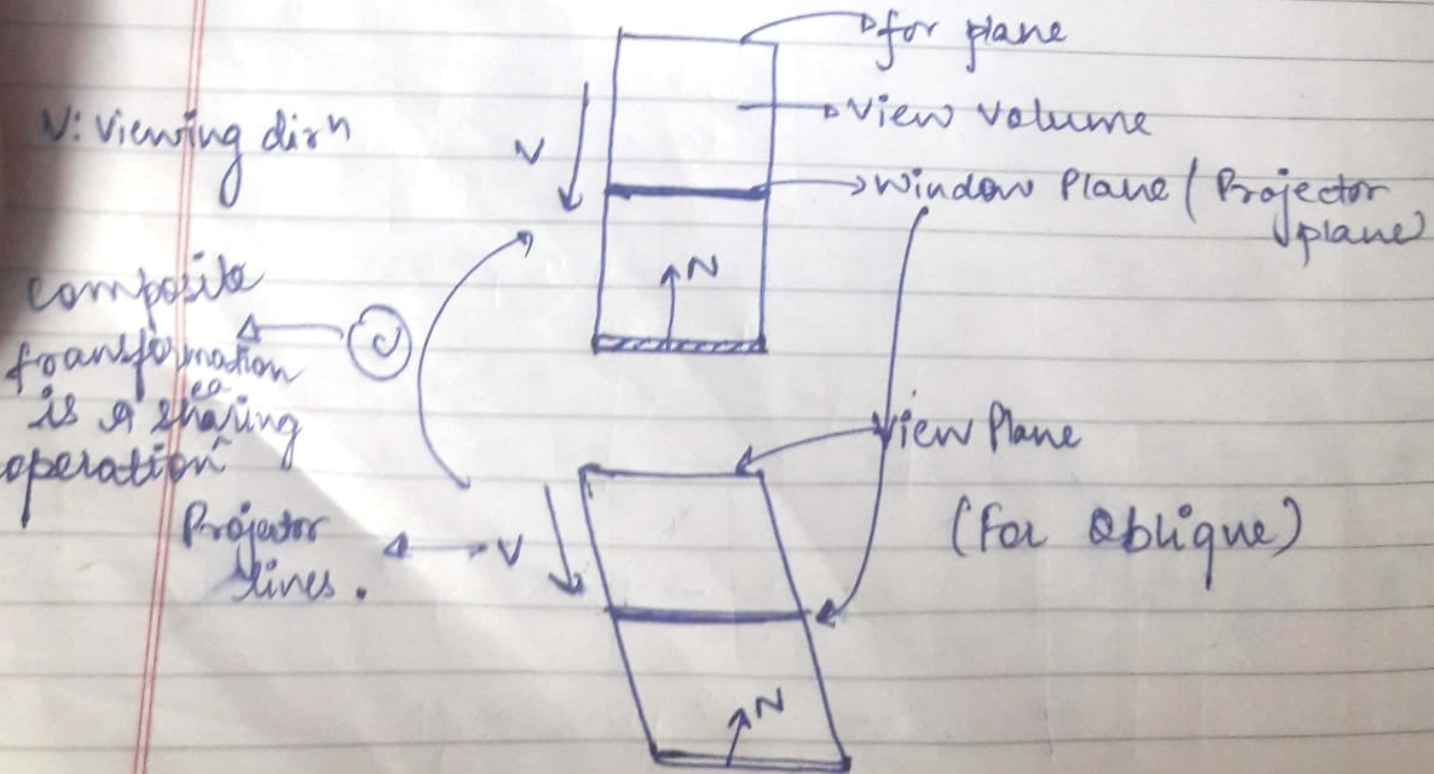
Scanned by CamScanner

OBLIQUE PROJECTION

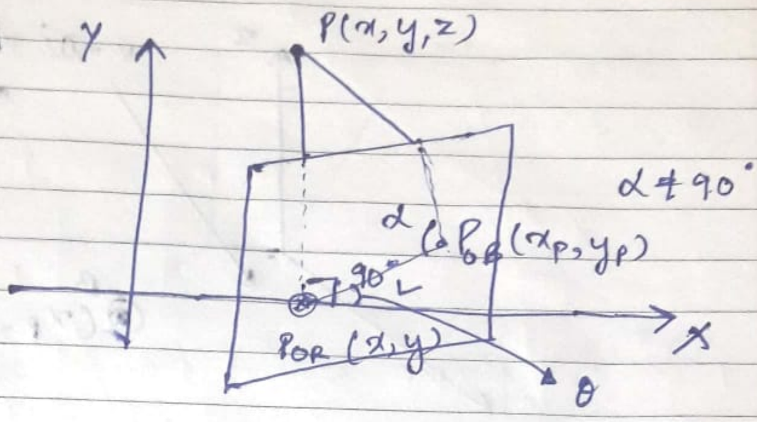


In 3D.:





$$C = Proj(x-y) \times \overrightarrow{A_v^N}$$



$$x_p = x + L\cos\theta$$
$$y_p = y + L\sin\theta$$

$$\boxed{\tan\alpha = \frac{Z}{L}}$$

$$L = \frac{Z}{\tan\alpha}$$

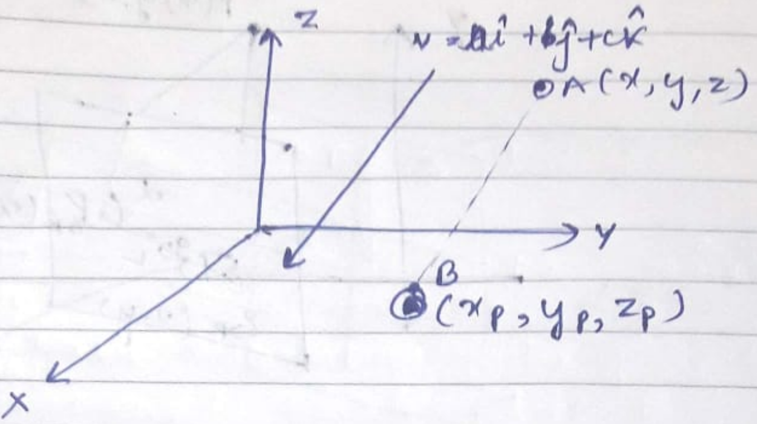
$$L = ZL_1 \quad (\text{let } L_1 = 1/\tan\alpha)$$

$$x_p = x + L_1\cos\theta Z$$
$$y_p = y + L_1\sin\theta Z$$

$$\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & L_1\cos\theta & 0 \\ 0 & 1 & L_1\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

EXAMPLE 1:

Derive the equation of



$$\vec{AB} = (x_p - x)\hat{i} + (y_p - y)\hat{j} + (z_p - z)\hat{k}$$

$$\because \vec{AB} \parallel \vec{v}$$

$$(x_p - x)\hat{i} + (y_p - y)\hat{j} + (z_p - z)\hat{k} = t(a\hat{i} + b\hat{j} + c\hat{k})$$

$$\Rightarrow (x_p - x) = at$$
$$(y_p - y) = bt$$
$$(z_p - z) = ct$$

z_p is 0

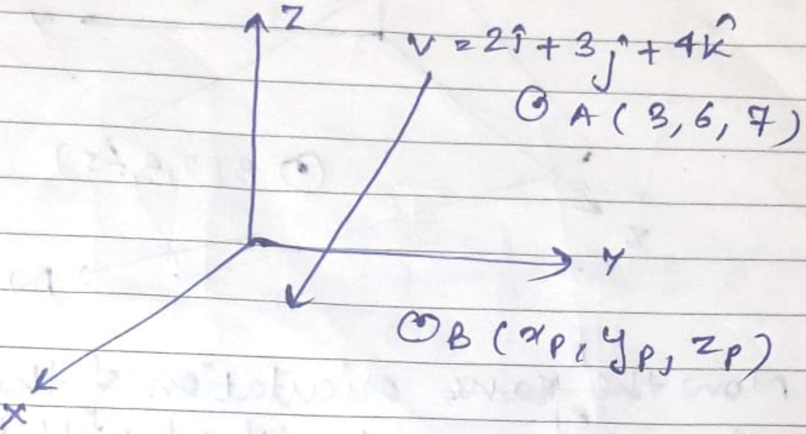
$$\Rightarrow -z = ct$$
$$\Rightarrow \boxed{t = -z/c}$$

$$x_p = x - \left(\frac{a}{c}\right)z$$

$$y_p = y - \left(\frac{b}{c}\right)z$$

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -a/c & 0 \\ 0 & 1 & -b/c & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Example 2:



Find B.

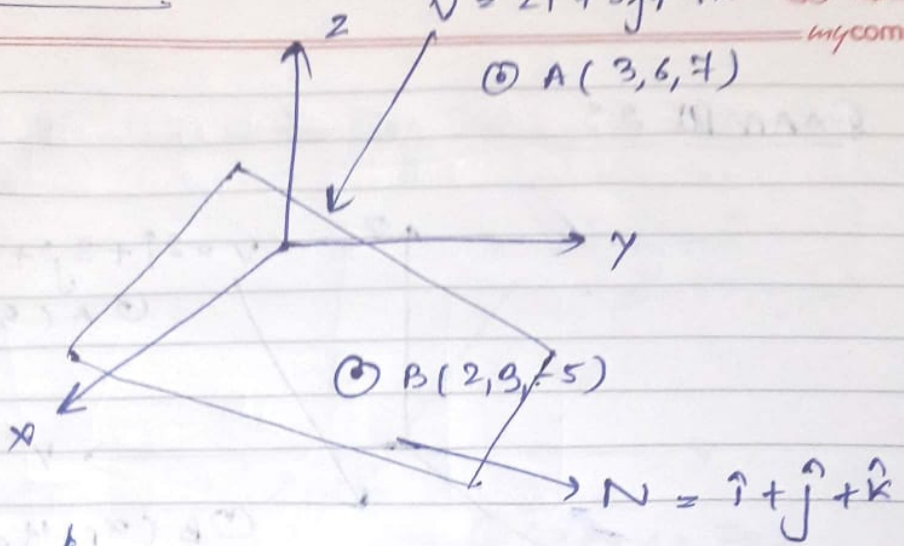
$$\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & -1/2 & 0 \\ 0 & 1 & -3/4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 7 \\ 1 \end{bmatrix}$$

$(x_p, y_p, z_p) \longrightarrow (-0.5, 0.75, 0)$

$B \rightarrow (-1/2, 3/4, 0)$

Now if $Z = -30$, what is B?

EXAMPLE 3 :



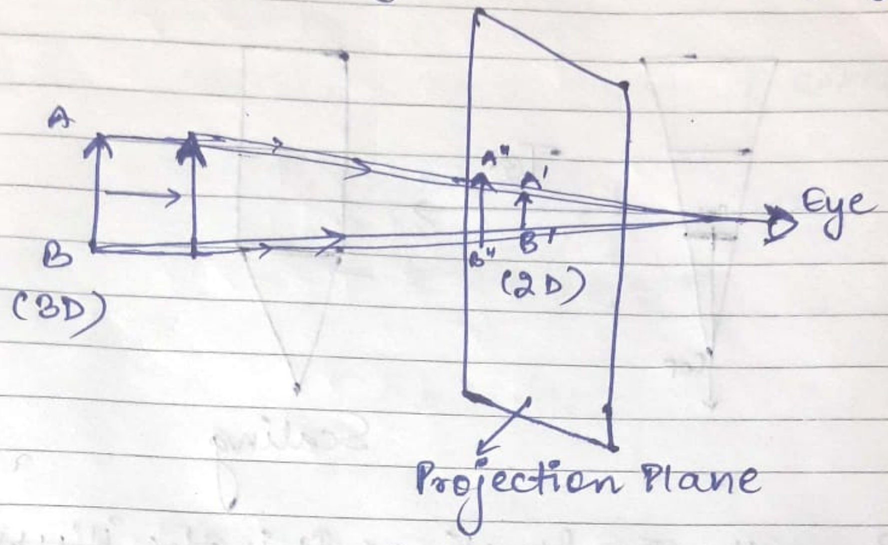
Now ~~the plane~~ orientation of the projection plane is changed. What will be B'?

$$\left((x_p - 3)\hat{i} + (y_p - 6)\hat{j} + (z_p - 7)\hat{k}\right) \cdot (\hat{i} + \hat{j} + \hat{k})$$
$$= (2\hat{i} + 3\hat{j} + 4\hat{k})(\hat{i} + \hat{j} + \hat{k})$$

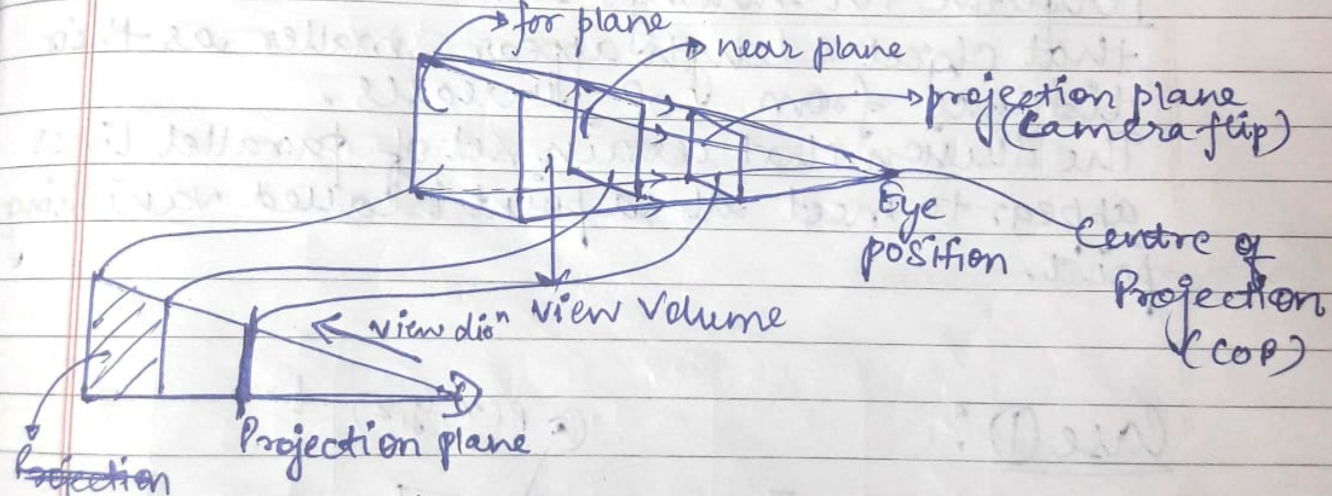
10/07/2018

Perspective Projection

(non-parallel projector lines, converging to a point)

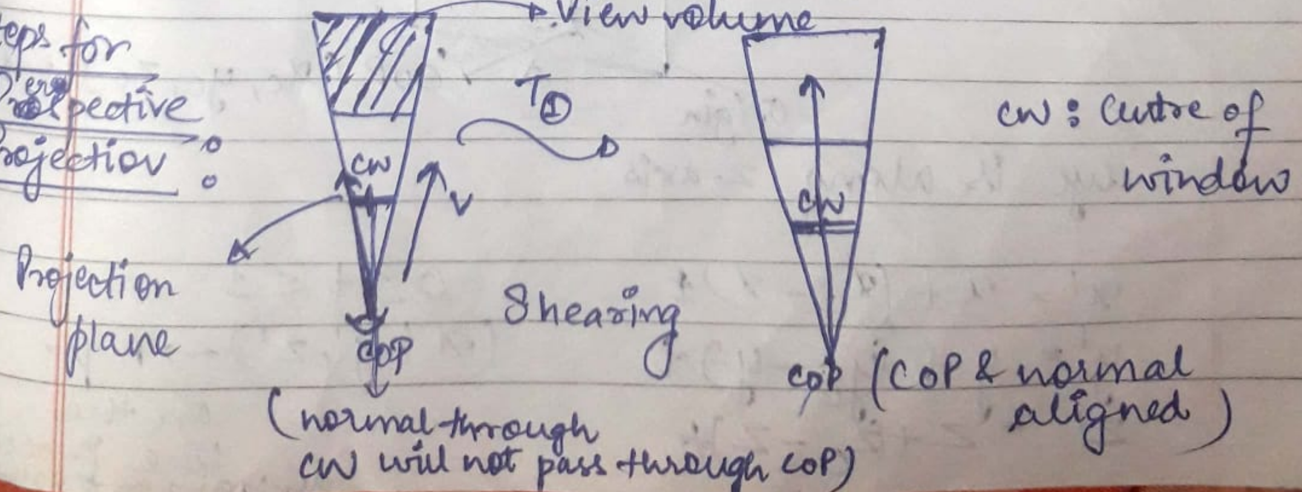


2)

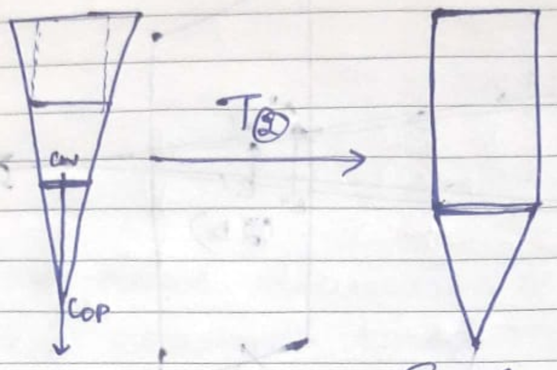


Trapezium $\xrightarrow{\text{transformed}}$ Rectangle

Steps for Perspective Projection :



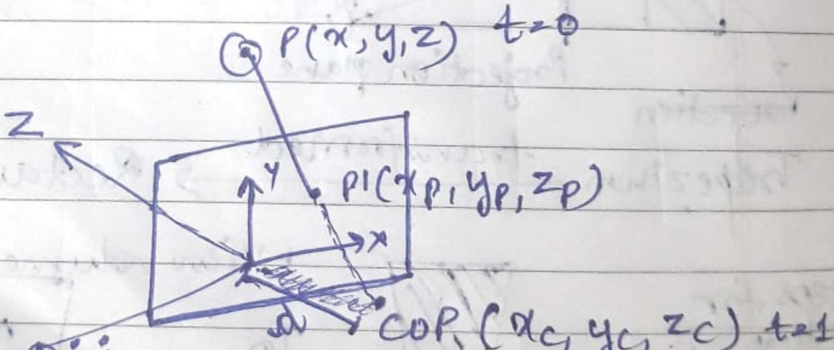
After this transformation, the trapezium becomes an isosceles trapezium.



Scaling

Perspective For Shorting: It is the ~~illusion~~ illusion that object & length appear smaller as their distance from COP increases.
The illusion that certain set of parallel lines appear to meet at a point is called vanishing point.

Case ①:



eye is along z-axis

$x' = x + (x_c - x)t$ $\quad 0 \leq t \leq 1$

$y' = y + (y_c - y)t$ $\quad (x', y', z') \to$ A pt on the line P→COP depending on t value

$z' = z + (z_c - z)t$

For a projection point at (x_p, y_p, z_p)

$\therefore z_p = 0$

$x_c = 0, y_c = 0, z_c = -d$ (camera is on z-axis)

$$x_p = x + (-x)t$$
$$y_p = y + (-y)t$$
$$z_p = z + (-d - z)t$$

$$\boxed{t = z/d+z}$$

$$x_p = x - \frac{xz}{d+z} = \frac{xd}{d+z}$$

$$y_p = y - \frac{yz}{d+z} = \frac{yd}{d+z}$$

Let $h = d+z$ (homogenous factor)

So, $x_h = x_p * h$
$y_h = y_p * h$

$$x_h = xd$$
$$y_h = yd$$
$$h = d+z$$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Case (2): Projection plane is moved near to the object at a distance k from the origin, that is, projection plane is at $z = k$

$$x_p = x + (-x)t$$
$$y_p = y + (-y)t$$
$$z_p = k = z + (-d - z)t$$

$$t = \frac{(z-k)}{d+z}$$

$$x_p = x - \frac{x(z-k)}{d+z} = \frac{x(d+k)}{d+z}$$

$$y_p = y - \frac{y(z-k)}{d+z} = \frac{y(d+k)}{d+z}$$

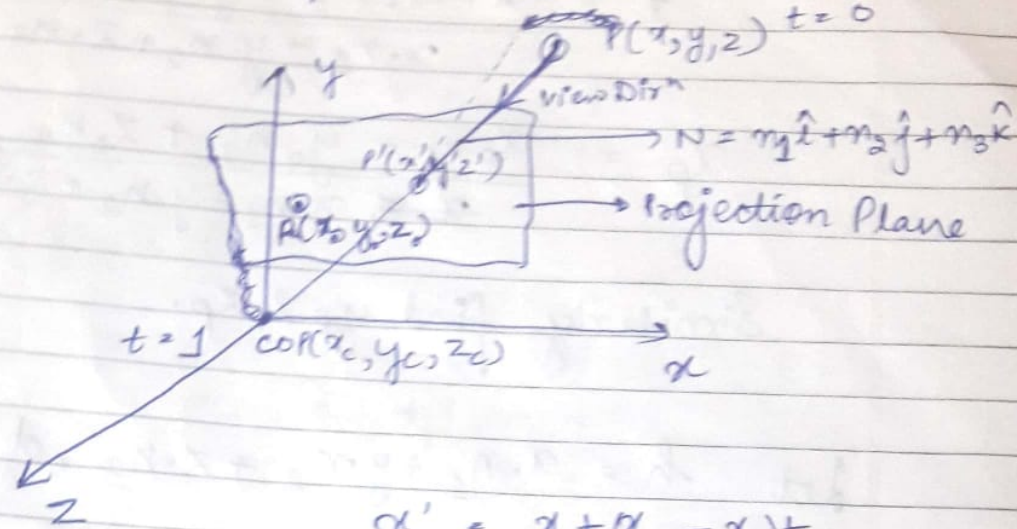
$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} d+k & 0 & 0 & 0 \\ 0 & d+k & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

12/04/18

Perspective View

Type 1:

COP at origin



$$x' = x + (x_c - x)t$$
$$y' = y + (y_c - y)t$$
$$z' = z + (z_c - z)t$$

If COP is at origin

$$\left.\begin{aligned} x_p &= x - xt \\ y_p &= y - yt \\ z_p &= z - zt \end{aligned}\right\} \text{— (1)}$$

$$\overrightarrow{P'R} = (x_p - x_0)\hat{i} + (y_p - y_0)\hat{j} + (z_p - z_0)\hat{k}$$

$$\therefore \overrightarrow{P'R} \perp \overrightarrow{N}$$

$$(x_p - x_0).n_1 + (y_p - y_0).n_2 + (z_p - z_0).n_3 = 0$$

$$(x - xt - x_0).n_1 + (y - yt - y_0).n_2 + (z - zt - z_0).n_3 = 0$$

$$-[x.n_1 + y.n_2 + z.n_3]t + [(x - x_0).n_1 + (y - y_0).n_2 + (z - z_0).n_3] = 0$$

Scanned by CamScanner

$$t = \frac{(x-x_0).n_1 + (y-y_0).n_2 + (z-z_0).n_3}{x.n_1 + y.n_2 + z.n_3}$$

$$x_p = \frac{x_0 n_1 + y_0 n_2 + z_0 n_3}{x.n_1 + y.n_2 + z.n_3} \times x$$

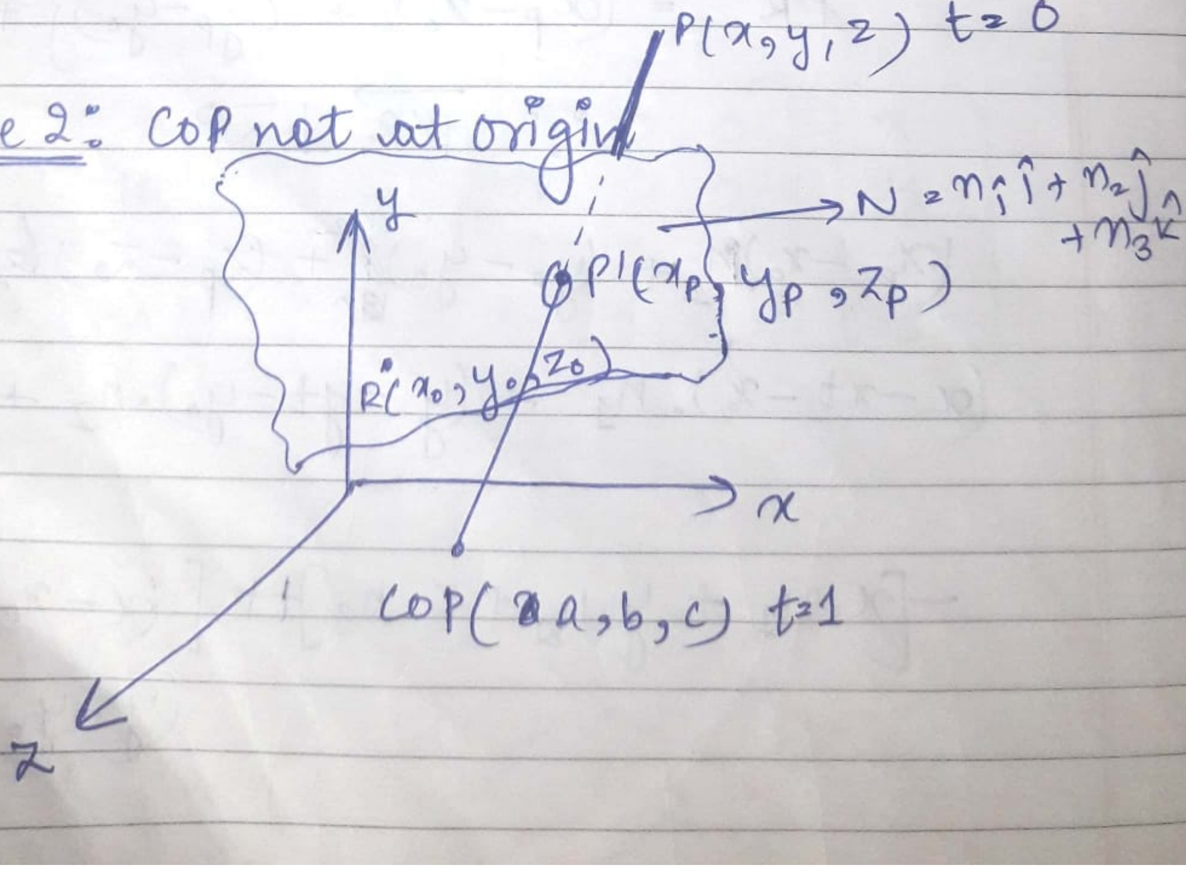
Similarly find y_p & z_p.

Let $h = x.n_1 + y n_2 + z.n_3$, $d = x_0 n_1 + y_0 n_2 + z_0 n_3$

$$x_h = x_p . h$$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_1 & n_2 & n_3 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Type 2: COP not at origin



Parametric Eqn P - coP

$$x' = x + (a-x)t$$
$$y' = y + (b-y)t$$
$$z' = z + (c-z)t$$

For a point P'

$$x_p = x + (a-x)t$$
$$y_p = y + (b-y)t$$
$$z_p = z + (c-z)t$$

$$\overrightarrow{P'R} \cdot \overrightarrow{N} = 0 \quad (\because \overrightarrow{P'R} \perp \overrightarrow{N})$$

$$(x_p - x_0) \cdot n_1 + (y_p - y_0) \cdot n_2 + (z_p - z_0) \cdot n_3 = 0$$

$$t = \frac{(x - x_0) \cdot n_1 + (y - y_0) \cdot n_2 + (z - z_0) \cdot n_3}{(x-a) \cdot n_1 + (y-b) \cdot n_2 + (z-c) \cdot n_3}$$

$$= \frac{(x \cdot n_1 + y \cdot n_2 + z \cdot n_3) - (x_0 \cdot n_1 + y_0 \cdot n_2 + z_0 \cdot n_3)}{(x \cdot n_1 + y \cdot n_2 + z \cdot n_3) - \underbrace{(a \cdot n_1 + b \cdot n_2 + c \cdot n_3)}_{d_0}}$$

$$x_p = x + (a-x) \frac{\left[(x_0 n_1 + y \cdot n_2 + z \cdot n_3) - (x_0 \cdot n_1 + y_0 \cdot n_2 + z_0 \cdot n_3)\right]}{\left[(x \cdot n_1 + y \cdot n_2 + z \cdot n_3) - (a \cdot n_1 + b \cdot n_2 + c \cdot n_3)\right]}$$

Find x_p, y_p, z_p & complete the matrix.

16th April, 2018

HIDDEN SURFACES

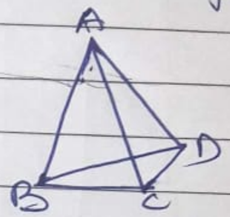
Backface Detection Method:

Tetrahedron

A(1,1,1)
B(3,1,1)
C(2,1,3)
D(2,2,2)

Torch light is at P(2,3,4)
Find out which surfaces are illuminated & which are not illuminated (Backface)

Tetrahedron has 4 faces
ABC, BCD, CDA, ABD



Step ① Recognition of faces

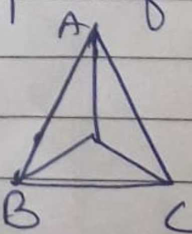
Step ② find normal outward ($N_{outward}$)

Edge	$N_{outward}$	Mean position (centroid)
ABC	$\vec{AB} \times \vec{AC} = -4\hat{j}$	$2\hat{i} + \hat{j} + 5/3\hat{k}$
BCD	$2\hat{i} + \hat{j} + \hat{k}$	$7/3\hat{i} + 4/3\hat{j} + 2\hat{k}$
ACD	$2\hat{i} - \hat{j} - \hat{k}$	$5/3\hat{i} + 4/3\hat{j} + 2\hat{k}$
ABD	$2\hat{j} - 2\hat{k}$	$2\hat{i} + 4/3\hat{j} + 4/3\hat{k}$

Step ③ (Centroid) mean position

Step ④ $(\vec{P-M})$ (torch light pos to mean pos of every vector)

Edge	$(\vec{P-M})$
ABC	$-2\hat{j} - 7/3\hat{k}$
BCD	$1/3\hat{i} - 5/3\hat{j} - 2\hat{k}$
ACD	$-1/3\hat{i} - 5/3\hat{j} - 2\hat{k}$
ABD	$-5/3\hat{j} - 8/3\hat{k}$



when $\vec{N}_{outward}$ & $\vec{P-M}$ in same dir^n, that face won't be visible.

Step ⑤:

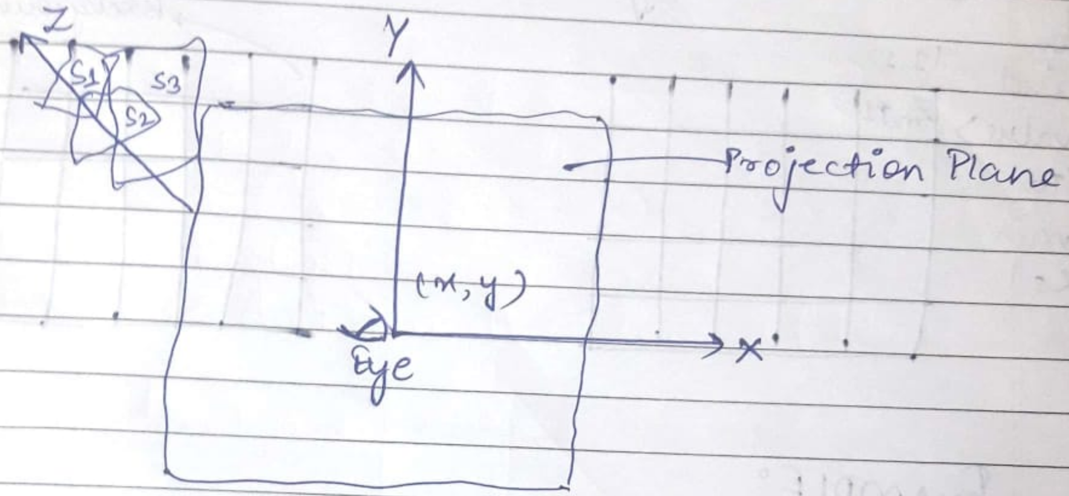
Check if dirⁿ of $\overrightarrow{(P-M)}$ & $\overrightarrow{N}$ outward is same by using if $\overrightarrow{N} \cdot \overrightarrow{(P-M)}$ < 0, visible

> 0, not visible

17/04/2018.

HIDDEN SURFACES

① Backface Detection (3D object)
② Z-Buffer image (2D)



set of surfaces

$S : \{ S_1, S_2, \text{------} S_n \}$

$S_1 \rightarrow ax + by + cz + d = 0$

These are coordinates → $S_1 (x, y, z_1)$

$S_2 (x, y, z_2)$

$S_3 (x, y, z_3)$

↓

Only min z. values will appear

$z = \frac{-1}{c} [ax + by + d]$

$Z_{(x,y)} = \frac{-1}{c} [ax + by + d]$

Step ①:

$\boxed{Z_{(x+1, y)} = Z_{(x,y)} - \frac{a}{c}}$

$\{ x_{min}, y_{min}, z_{min}; x_{max}, y_{max}; z_{max} \}$

(using S)

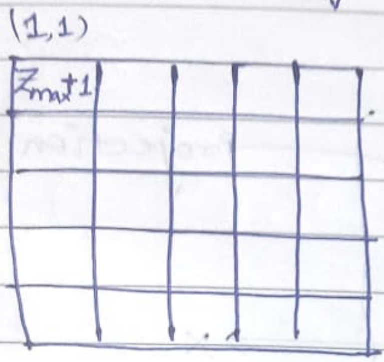
Step ②:

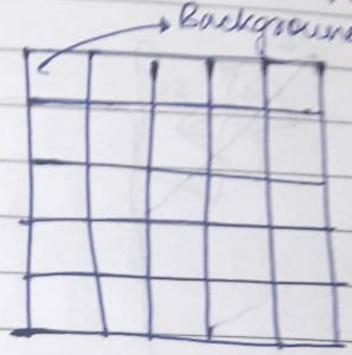
2D Buffer
(Z-values)

$\boxed{M_1}$ Z-Buffer

$\boxed{M_2}$ Color Buffer → initialised with b'grd color

Every location initialised with value > Z_{max} (To find min. of Zs)





EXAMPLE:

Tetrahedron

A (1, 1, −1)
B (3, 1, −1)
C (2, 1, −3)
D (2, 2, −2)

Surfaces	color
ACD	RED(1)
CBD	BLUE(2)
BAD	CYAN(3)
ACB	GREEN(4)
Background	BLACK(0)

Step ①:

$x_{min} = 1$, $y_{min} = 1$, $z_{min} = -3$

$x_{max} = 3$, $y_{max} = 2$, $z_{max} = -1$

classmate
Date
Page

Step ②: Find $eq^n(s)$ of the planes.

Plane ACD:

$\vec{AC} = \hat{i} - 2\hat{k}$, $\vec{AD} = \hat{i} + \hat{j} - \hat{k}$

$$\vec{AC} \times \vec{AD} = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ 1 & 0 & -2 \\ 1 & 1 & -1 \end{bmatrix} = 2\hat{i} - \hat{j} + \hat{k}$$

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

$\Rightarrow \boxed{Z_{ACD} = -2x + y}$

Plane CBD:

$\vec{CB} = \hat{i} + 2\hat{k}$, $\vec{CD} = \hat{j} + \hat{k}$

$$\vec{CB} \times \vec{CD} = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix} = -2x - y + z$$

$\to a = -2, b = -1, c = 1$

$ax + by + cz = d$

$-2(2) - 1(1) + 1(-3) = d$

$\Rightarrow \boxed{d = -8}$

$\Rightarrow -2x - y + z = -8$

$\Rightarrow -2x - y + z + 8 = 0$

$\Rightarrow \boxed{Z_{CBD} = 2x + y - 8}$

Plane BAD:

$\vec{BA} = -2\hat{i}$, $\vec{BD} = -\hat{i} + \hat{j} - \hat{k}$

$$\vec{BA} \times \vec{BD} = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ -2 & 0 & 0 \\ -1 & 1 & -1 \end{bmatrix} = -2\hat{j} - 2\hat{k} \quad (d = 0)$$

$\Rightarrow -2y - 2z = 0$

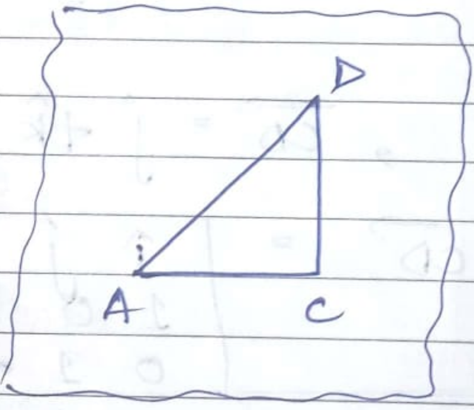
$\Rightarrow \boxed{Z_{BAD} = -y}$

Plane ACB:

Step②: (Image)

ACD (Image):

A(1,1), C(2,1), D(2,2)



Inside Test: To determine if a pt. is inside or outside

≥ 0, if inside < 0, if outside

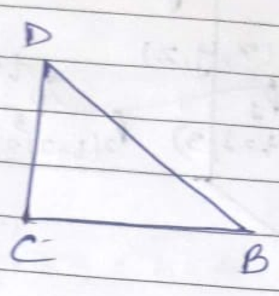
line eq AC $y = 1$ $f(x,y)_D : y - 1 > 0$

CD $x = 2$ $f(x,y) : 2 - x > 0$

AD $y = x$ $f(x,y) : x - y > 0$

CBD (Image)

C(2,1) B(3,1), D(2,2)



line eq CB $y=1$ $f(x,y)_D : y-1>0$

line eq BD $x+y=4$ $f(x,y)_C : 4-x-y>0$

line eq DC : $x=2$ $f(x,y)_B : x-2>0$

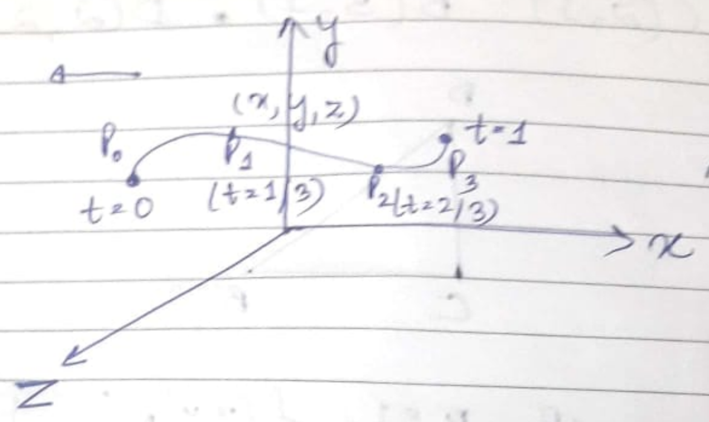
x	y	Z	Inside Test	
1	1	$Z_{ACD} = -1$ $Z_{CBD} = -5$	T F	ACD (RED)
1	2	$Z_{ACD} = 0$ $Z_{CBD} = -4$	F F	Background
2	1			
2	2			
3	1			
3	2			

→ font designing
→ Automobile
→ ship building

19/04/2018

3D CURVES

For Cubic ←→



n=1 line
n=2 quadratic
n=3 cubic curve
n=4 quadratic space curve

Parametric form (curve segment)

$$P(t) = \{ x(t), y(t), z(t) \} \quad 0 \le t \le 1$$

$a_0, a_1 \ldots$ → constants

$$x(t) = a_0x_0 + a_1x_1t + a_2x_2t^2 + \cdots + a_nx_nt^n$$

$$y(t) = \vdots$$

$$z(t) = \vdots$$

For cubic curves ($n=3$), there are 4 control points. They are responsible for giving ~~the~~ shape to the curve, That is they control the shape of the curve.

Cubic curve segment :

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

$$P(t) = \vec{a} t^3 + \vec{b} t^2 + \vec{c} t + \vec{d}$$ Parametric Form

$$P_0 = \vec{d}$$

$$P_1 = \frac{\vec{a}}{27} + \frac{1}{9}\vec{b} + \frac{1}{3}\vec{c} + \vec{d}$$

$$P_2 = \frac{8\vec{a}}{27} + \frac{4\vec{b}}{9} + \frac{2\vec{c}}{3} + \vec{d}$$

$$P_3 = \vec{a} + \vec{b} + \vec{c} + \vec{d}$$

$$\begin{pmatrix} \vec{a} \\ \vec{b} \\ \vec{c} \\ \vec{d} \end{pmatrix} = \begin{pmatrix} -9/2 & 27/2 & -27/2 & 9/2 \\ 9 & -45/2 & 18 & -9/2 \\ -11/2 & 9 & -9/2 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

[By inverse of coefficients]

$$P(t) = \left(-\frac{9}{2}P_0 + \frac{27}{2}P_1 - \frac{27}{2}P_2 + \frac{9}{2}P_3\right)t^3$$

$$+ \left(9P_0 - \frac{45}{2}P_1 + 18P_2 - \frac{9}{2}P_3\right)t^2$$

$$+ \left(-\frac{11}{2}P_0 + 9P_1 - \frac{9}{2}P_2 + P_3\right)t$$

$$+ P_0$$

Scanned by CamScanner

for $t = 0$ to 1 skip of 0.0001

$$P(t) = \left[-\frac{9}{2}t^3 + 9t^2 - \frac{11}{2}t + 1\right]P_0 +$$

$$\left[\frac{27}{2}t^3 - \frac{45}{2}t^2 + 9t\right]P_1 +$$

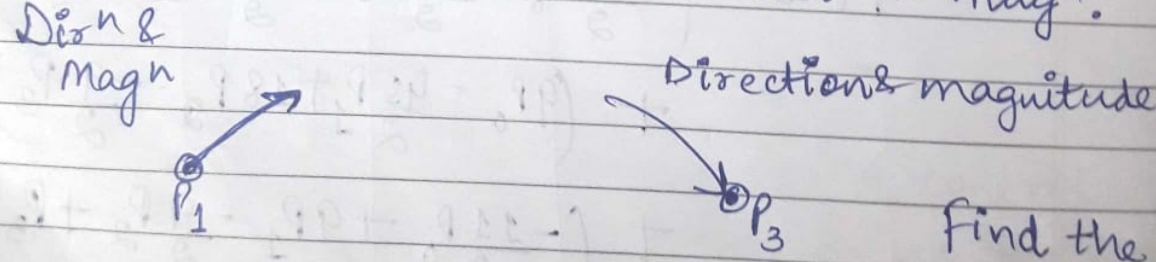
$$\left[-\frac{27}{2}t^3 + 18t^2 - \frac{9}{2}t\right]P_2 +$$

$$\left[\frac{9}{2}t^3 - \frac{9}{2}t^2 + t\right]P_3$$

→ Used in font designing.

Problem 1: (Sol^n given by Hermite).

A stone is thrown from a point P_0 with dir^n & magnitude. The stone is received at Point P_3. It has a location as well as dir^n & mag^n.



Find the parametric form of eq^n of curve.

$P_0(t=0) = \vec{d}$

$P_3(t=1) = \vec{a} + \vec{b} + \vec{c} + \vec{d}$

$P'(t) = 3\vec{a}t^2 + 2\vec{b}t + \vec{c}$

$P'_0(t=0) = \vec{c}$

$P'_3(t=1) = 3\vec{a} + 2\vec{b} + \vec{c}$

Scanned by CamScanner

$$P_3 = \vec{a} + \vec{b} + \vec{c} + P_0$$

$$P_3 = \vec{a} + \vec{b} + P_0' + P_0$$

$$\vec{a} = P_3 - \vec{b} - P_0' - P_0$$

P_1 P_2

P_0

P_3

HERMITE

$$\begin{pmatrix} \vec{a} \\ \vec{b} \\ \vec{c} \\ \vec{d} \end{pmatrix} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_3 \\ P_0' \\ P_3' \end{pmatrix}$$

for $t=0$ to 1 step 0.0001

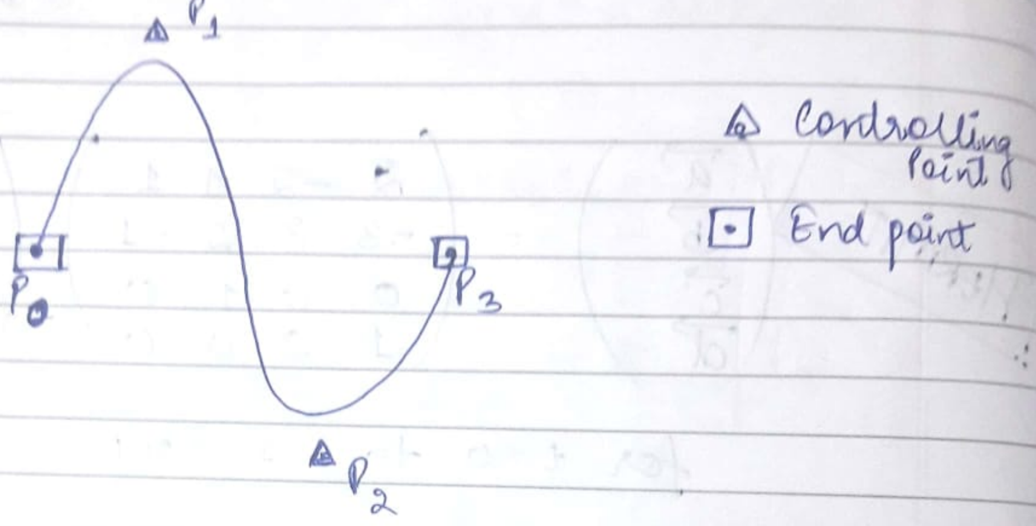
$P(t) =$ ______.

This font can be drawn with 3 curves

13/04/2018

BEZIER CURVE
(inventor → Pierre Bezier)

Cubic Curve :



Example :

$P_0 (m_0)$ $P_1 (m_1)$

$\boxed{P}$

$P_3 (m_3)$ $P_2 (m_2)$

$$P = \frac{m_0 P_0 + m_1 P_1 + m_2 P_2 + m_3 P_3}{m_0 + m_1 + m_2 + m_3}$$

(centre of mass)

- If point masses are fixed, the centre of mass is fixed.
- If point masses are variable, the centre of mass changes in value.

Beizer $\longrightarrow$ Put masses as bionomial func(s)

$m_0 = (1-t)^3$, $m_1 = 3t(1-t^2)$, $m_2 = 3t^2(1-t)$

$m_4 = t^3$

(~~$t \leq 0$~~ $0 \leq t \leq 1$)

$$P(t) = \sum_{i=0}^{3} P_i \cdot \underbrace{B_{i,n}(t)}_{\text{Blending Function}}$$

mapped

$$= P_0 \cdot B_{0,3}(t) + P_1 B_{1,3}(t) + P_2 \cdot B_{2,3}(t) + P_3 \cdot B_{3,3}(t)$$

mapped

$$P(t) = P_0 (1-t)^3 + P_1 \cdot 3t(1-t^2) + P_2 \cdot 3t^2(1-t) + P_3 \cdot t^3$$

$$= P_0 [-t^3 + 3t^2 - 3t + 1] P_0 +$$

$$[3t^3 - 6t^2 + 3t] P_1 +$$

$$[-3t^3 + 3t^2] P_2 +$$

$$[t^3] P_3$$

$$\boxed{P_B(t) = T \cdot M_B \cdot G_B}$$

for $t = 0$ to 1 skip $= 0.0001$

$$P_B(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ -3 & 6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{(M_B)} \underbrace{\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}}_{G_B}$$

<u>Rel^n b/w Beizer & Hermite :-</u>

(Transformation from Hermite to Beize)

$$P_H(t) = \underbrace{\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{M_H} \underbrace{\begin{bmatrix} P_0 \\ P_3 \\ P_0' \\ P_3' \end{bmatrix}}_{G_H}$$

$$G_H = M_{(H-B)}\, G_B$$

Transformation Matrix ~~[illegible]~~ for Hermite to Bezier

$$\cancel{G_H = \begin{bmatrix} P_0 \\ P_3 \\ P_0' \\ P_3' \end{bmatrix} =}$$

<u>Beizer</u>: $P'(t) = [-3t^2 + 6t - 3]P_0 + [9t^2 - 12t + 3]P_1$
$+ [-9t^2 + 6t]P_1$
$+ 3t^2 P_3$

$P'(t=0) = -3P_0 + 3P_1$

$P'(t=1) = -3P_2 + 3P_3$

$$G_H = \begin{bmatrix} P_0 \\ P_3 \\ P_0' \\ P_3' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$G_H = M_{H-B} \cdot G_B$$

$$P_H(t) = T \cdot M_H \cdot G_H$$

$$= T \cdot M_H \cdot (M_{H-B} \cdot G_B)$$

$$= T \cdot (M_H \cdot M_{H-B}) \cdot G_B$$

$$= T \cdot (M_B) \cdot G_B = P_B(t)$$

To prove this

$$\underset{M_H}{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}} \underset{M_{H-B}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix}} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(M_B) (verified)

Question1 Find eqn of Bezier curve which passes through pt (0,0) & (−2,1) and is controlled through pts (7,5) & (2,0).

$$P_B(t) = T \cdot M_B \cdot G_B$$

I eqn:

$$\begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 0 \\ -2 & 1 \end{bmatrix}$$

II Eqn:

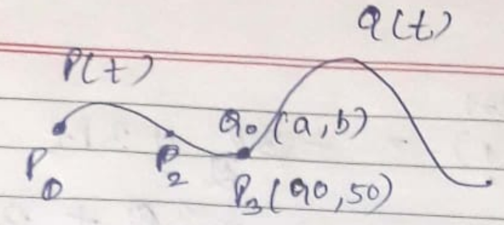
$$\begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 7 & 5 \\ -2 & 1 \end{bmatrix}$$

Question2 Cubic Beizer curve segment is described by pts $P_0(20, 20)$, $P_1(40, 80)$, $P_2(80, 80)$, $P_3(90, 50)$

Another curve segment is described as

$Q_0(a, b)$, $Q_1(c, 20)$, $Q_2(150, 20)$, $Q_3(180, 20)$

Determine the value of a, b, c if the 2 curves are joined smoothly.



(Tangent at end pt. of joining → equal)

⇒ Tangent at P_3, Q_0 is same.

∴ Equate $P'(t=1)$ with $Q'(t=0)$

$$P'(t=1) = -3P_2 + 3P_3.$$

$$Q'(t=0) = -3Q_0 + 3Q_1$$

$$\boxed{a = 90,\ b = 50}$$

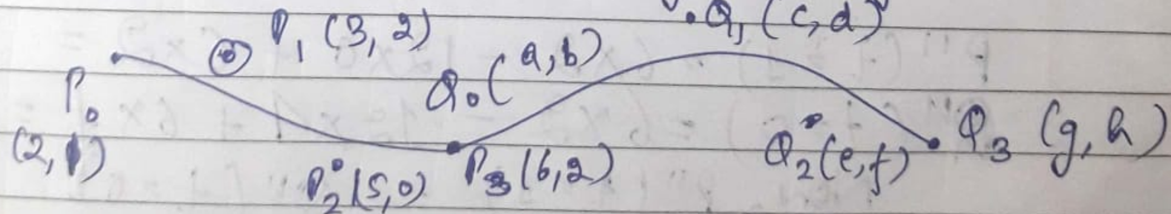
$$-3 \times 80 + 3 \times 90 = -3 \times 90 + 3 \times C$$

$\cancel{-120 = -120 + 3C}$: $30 = -270 + 3C$

$\cancel{30 = -120 + 3C}$ ⇒ $3C = 300$

$$\boxed{C = 100}$$

Question 3 Given 2 curve segments $P(t)$ & $Q(t)$. ~~where P~~ The 2 curve segments join smoothly.



Find a, b, c, d, e, f, g, h.

$$\boxed{a = 6,\ b = 2}$$

$P'(t=1) = -3P_2 + 3P_3 = -3\times5 + 3\times6$

$Q'(t=0) = -3Q_0 + 3Q_1 = -3\times6 + 3\times c$

$\Rightarrow -3\times5 + 3\times6 = -3\times6 + 3\times C$

$\Rightarrow 3 = -18 + 3C$

$\Rightarrow 21 = 3C$

$\Rightarrow \boxed{C = 7}$.

$-3\times0 + 3\times2 = -3\times2 + 3\times d$

$\Rightarrow \boxed{d = 4}$

$P''(t) = [-6t+6]P_0 + [18t-12]P_1$
$[-18t+6]P_2 + 6t\,P_3$

$\cancel{Q''(t) =}$

$P''(t=1) = 6P_1 - 12P_2 + 6P_3$

$P''(t=1) = 6\times3 - 12\times5 + 6\times6 = -6$

$Q''(t=0) = 6Q_0 - 12Q_1 + 6Q_2$

$Q''(t=0) = 6\times6 - 12\times7 + 6\times e = -48 + 6e$

Now, $P''(t=1) = Q''(t=0)$

$\Rightarrow -6 = -48 + 6e$

$\Rightarrow 6e = 42 \Rightarrow \boxed{e = 7}$

$P''(t=1) = 6\times2 - 12\times0 + 6\times2 = 24$.

$Q''(t=0) = 6\times2 - 12\times4 + 6\times f = -36 + 6f$

Now, $P''(t=1) = Q''(t=0)$

$\Rightarrow 24 = -36 + 6f$.

$\Rightarrow 60 = 6f$

$\Rightarrow \boxed{f = 10}$.

To find (g, h), use 3rd derivative $[P'''(t)$ & $Q'''(t)]$.

24/04/2018

B-Spline

Natural Spline:

Hermite

Bezier

Disadvantages:

→ (1) Polynomial coefficients are dependent on all points (control & end points).

→ Any movement of these points changes the shape of the entire curve.

→ local control /end points (if $n+1$ control /end points then changing any one changes all parts of curve)

→ (2) Computation time needed is large, Since it involves inverting of matrix.

Bases: Local control /end points (Thus, the name B-Spline)

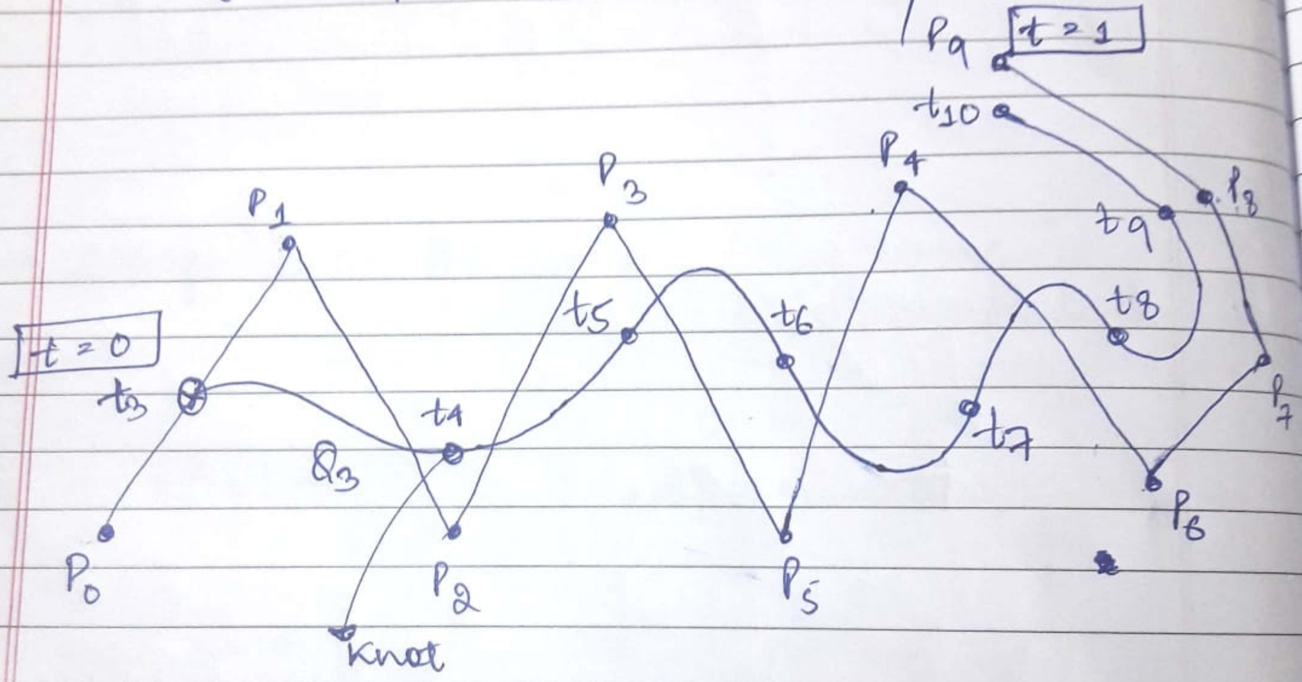
→ for $n+1$, inverse needed of $n+1 \times n+1$ matrix every time we change even 1 point.

if local control pts., only some (local) control pts. need to be changed.

Scanned by CamScanner

Global variables → affect whole f^ns
Local variables → " only one f^n.

Similarly, we want to change only a part of curve. To advocate this disadvantage, we take a subset of all & say they affect only one part. (Local control & end points)



B-spline:

$$P(t) = \sum_{i=0}^{n} P_i * \text{Blending } f^n.$$

$$0 \le t \le 1$$

$$\{t_0, t_1, t_2, \ldots t_n\}$$

If we see as cubic curve, 4 pts. needed.
P_0, P_1, P_2, P_3.

Scanned by CamScanner

$P(t) = \sum_{i=0}^{n} P_i * \text{Blending fn}$

$t_{min} < t \leq t_{max}$

Curve segment

Q_3	$P_0\ P_1\ P_2\ P_3$
Q_4	$P_1\ P_2\ P_3\ P_4$
Q_5	$P_2\ P_3\ P_4\ P_5$
Q_6	$P_3\ P_4\ P_5\ P_6$

$t_3 - t_4$

$t_{max} - t_{min}$

In closed curve, every control point controls 4 curves.

eg: t_3 comes in Q_3, Q_4, Q_5, Q_6

if there are $(n+1)$ control pts., then there will be $(n-2)$ curve segments.

Blending fn — Inherent property of Recursion

$t_3, t_4, t_5 \text{ - - - -} \rightarrow$ Knots

Must satisfy

$$P(t) = \sum_{i=0}^{n} P_i * B_{i,d}(t)$$

Cubic $d-1 = 3$

$d-1$ is a degree

$t_{min} \leq t \leq t_{max}$

$2 \leq d \leq n+1$

if Knot values uniform ↓ Uniform B-spline

if non-uniform ↓ Non-uniform B-spline

Cox-De-Boor Recursion func^n

$$B_{i,d}(t) = \left\{\frac{t - t_i}{t_{i+d-1} - t_i}\right\} B_{i,d-1}(t) + \left\{\frac{t_{i+d} - t}{t_{i+d} - t_{i+1}}\right\} B_{i+1,d-1}(t)$$

Scanned by CamScanner

where $B_{i,1}(t) = 1$ if $t_i \le t < t_{i+1}$

else $B_{i,1}(t) = 0$

Cubic B-spline $\longrightarrow d = 4$

$P(t) = P_0 * B_{0,4}(t) + P_1 * B_{1,4}(t) +$
$P_2 * B_{2,4}(t) + P_3 * B_{3,4}(t)$

Sampling :

$t_{max} - t_{min}$ is divided into $\underbrace{n + d}_{7}$ subinterval

($n = 3$, $d = 4$)

$t : \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$

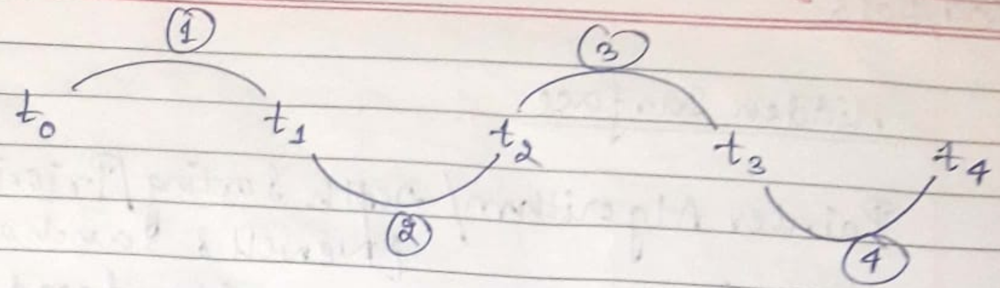
$t_0 \downarrow 0$, $t_1 \downarrow 1$, $t_7 \downarrow 7$

$B_{0,4}(t) = \left\{\frac{t-0}{3-0}\right\} B_{0,3}(t) + \left\{\frac{4-t}{3}\right\} B_{1,3}(t)$

$B_{1,4}(t)$

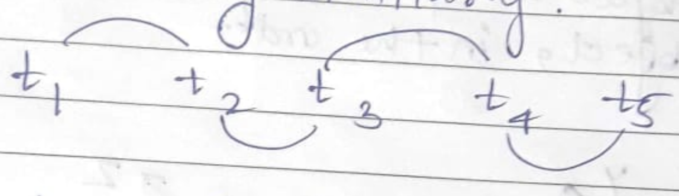
$B_{2,4}(t)$

Scanned by CamScanner



In all 4, variation from $t_3 - t_4$ present in all segments ∴ we draw that curve.

Proceeding similarly.



t_2 t_3 t_4 t_5 t_6

t_3 t_4 t_5 t_6 t_7

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Question 4 points (1) $P_0 (a, b)$ (2) $P_1 (20, 50)$ (3) $P_2 (40, 40)$ (4) $P_3 (70, c)$ are available for drawing a B-Spline curve. Compute the value of a, b, c s.t. curve starts at point $(21, 43)$ & terminates with the slope $(-1/2)$.

$\left.\begin{array}{l} t \text{ starts at } 3 \\ \text{ends at } 4 \end{array}\right\}$ as 3, 4 was common in all.

24/04/2018

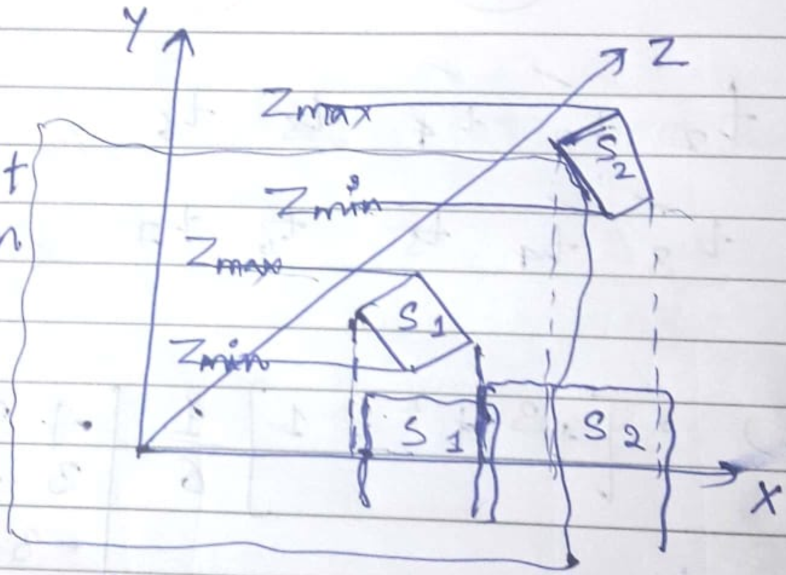
Hidden Surface

Painter Algorithm/ Depth Sorting/Priority
(Newell & Sancha)

Standard practice of a painter

→ Distant object are painted first ↓
→ Nearer object, in the order

Surfaces
↓
They don't overlap on z-axis



$S : \{S_1, ---- S_n\}$

Sort (Depth

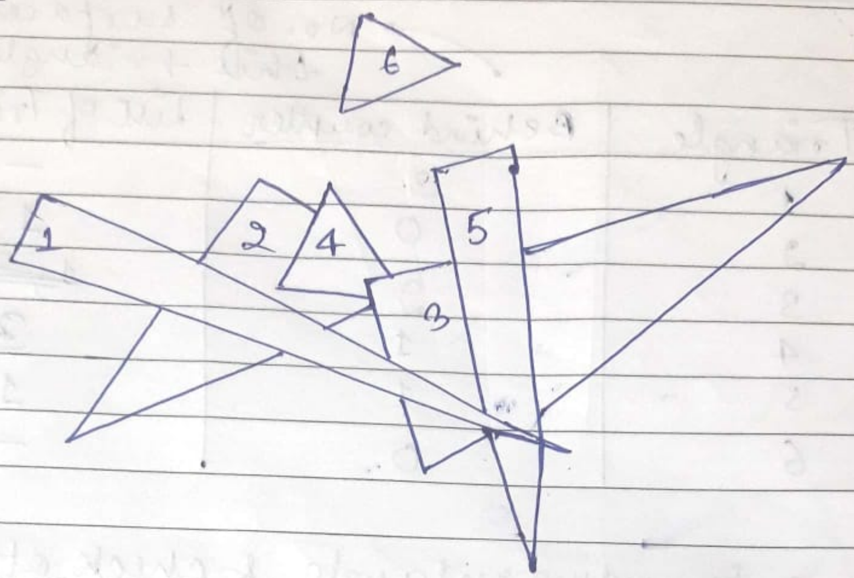
↓

Cases

	I	II	
z - extent overlap	×	×	×
y - extent overlap	×	×	✓
x - extent overlap	×	✓	×

Page

Ex ①:



Set of surfaces which don't have z-extent overlap

$S : \{S_1, S_2, S_3, S_4, S_5, S_6\}$

(x, y, z)

Coordinates of all surfaces

Given: x, y, z of all surfaces

To find Behind counter

Step ①: Draw Bounding rectangles & see whose bounding rectangle are overlapping.

Solution: Construction of Table

No. of surfaces behind this triangle

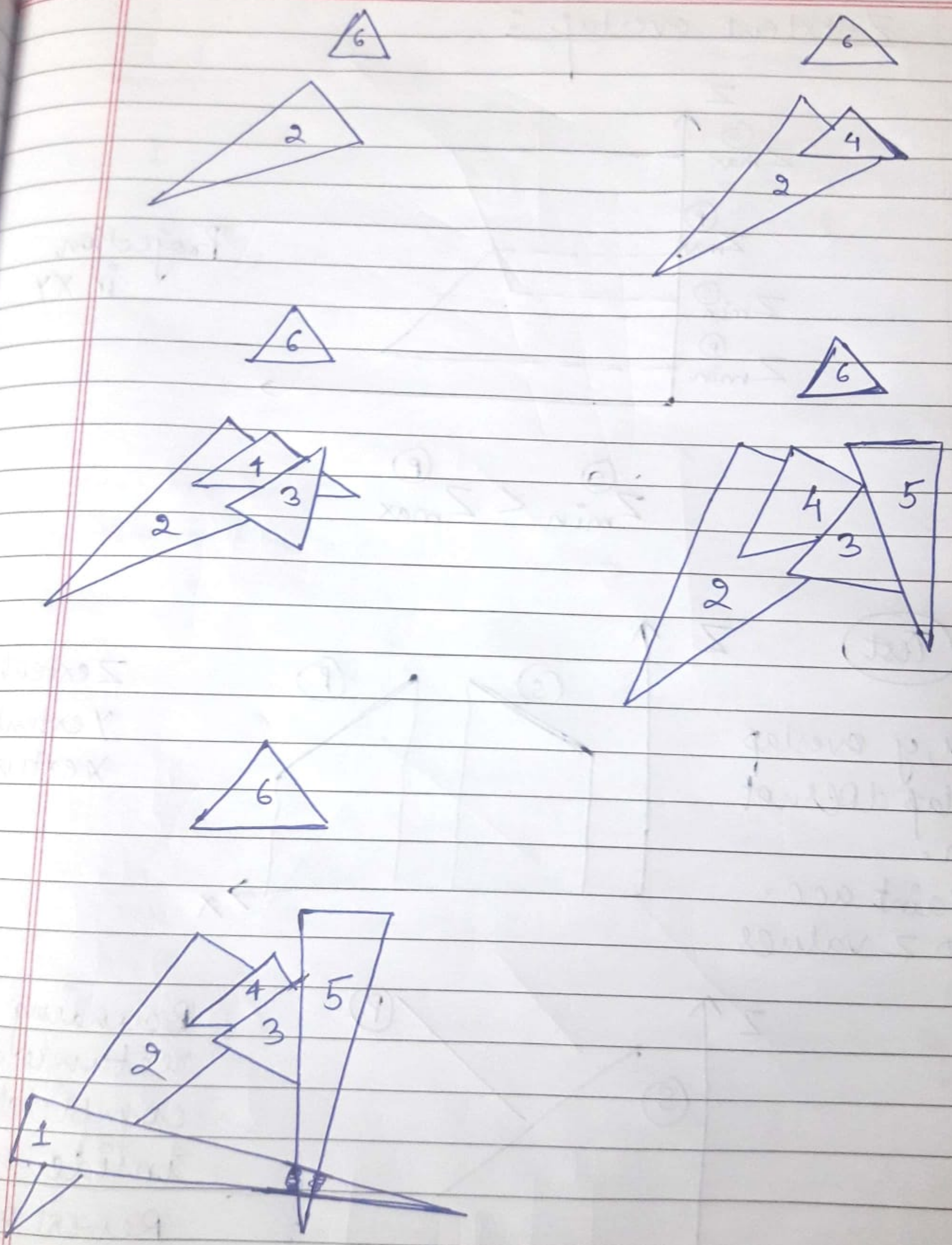
Triangle	Behind counter	List of Triangle in front
1	3	—
2	0	1, 3, 4
3	2	~~1, 3, 4~~ 1, 5
4	1	3.
5	1	1
6	0	—

Take a bounding rectangle & check other bounding rectangles for no. of overlappings.

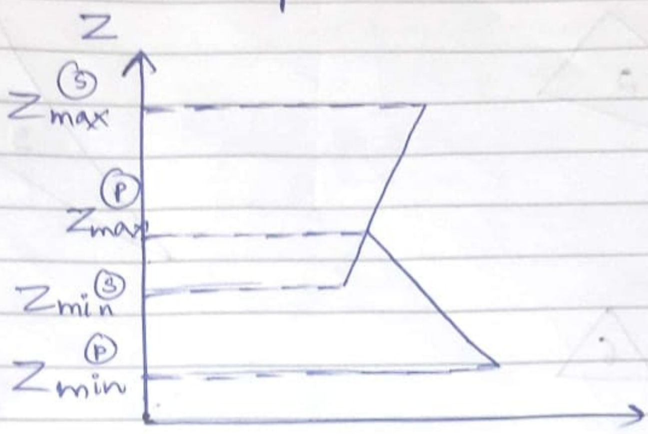
Contents of Table (Processed)

Triangle	Front list	Behind counter (Pass 1)	Pass 2	3	4	5
1	—	3	2	2	1 →	0
2	1, 3, 4	0 ✓				
3	1, 5	2	1 →	0 ✓		
4	3	1	0 ✓			
5	1	1	1	1 →	0 ✓	
6	—	0 ✓				

Take all with behind counter = 0 & draw. Then decrement the behind counter of all in their front list.


6
2
6
4
2
6
4
3
2
6
4
5
3
2
6
4
5
3
2
1

Z-extent overlap :

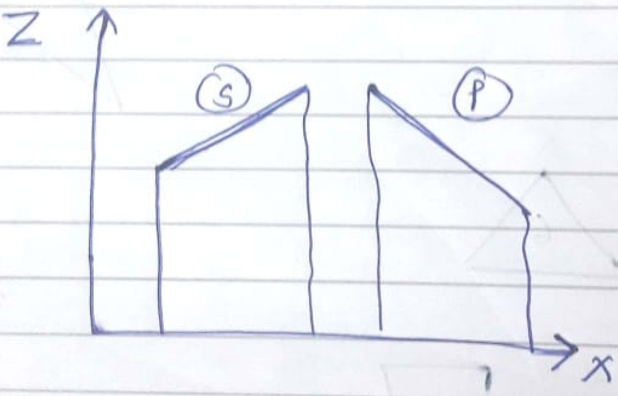


Projection in XY

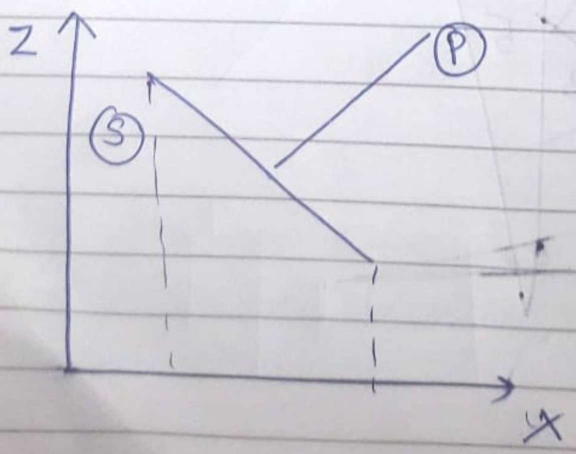
$\therefore Z_{min}^{(S)} < Z_{max}^{(P)}$

Test

if no x,y overlap Z overlap doesnot matter.
∴ Paint acc. to z values

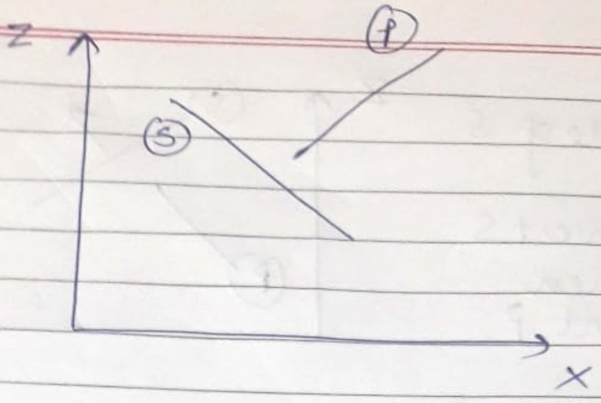


Z extent ✓
Y extent ✗
X extent ✗



Bounding rectangle are completely inside..

Revert order
Scans
ignore P → No need to draw P.

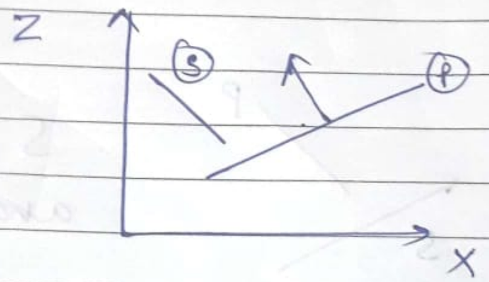


Partially obscured

↓

There are no. of cases

Cases :



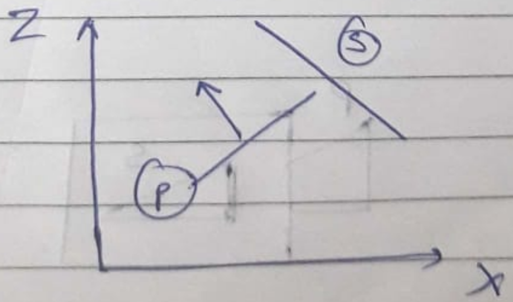
"Polygon S is outside" of Polygon P

if Polygon S is outside of Polygon P relative to view plane

$$f(x,y) : Ax + By + Cz + D \longrightarrow P$$

(1) if putting point P in plane eq^n gives >0, then S is outside the plane P.

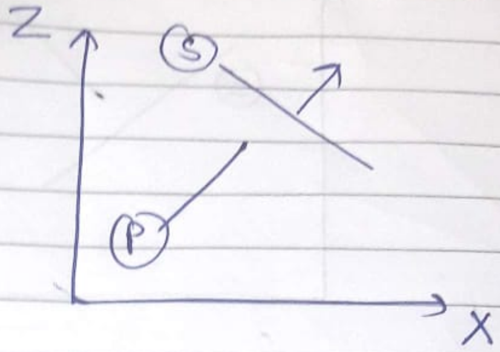
(2) Take normal of P. If both vertices of S on same side on normal then S is outside of S.



If case S "outside" of P fails then test to see if P is inside of S (again w.r.t. view plane).

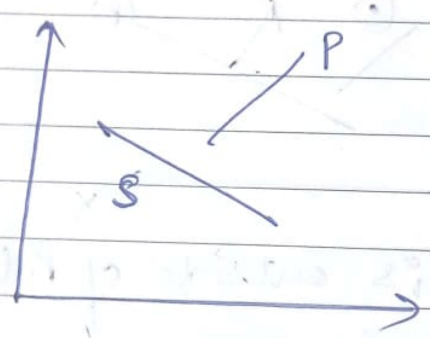
P is inside of S

so, scan converts S first & then paint P.

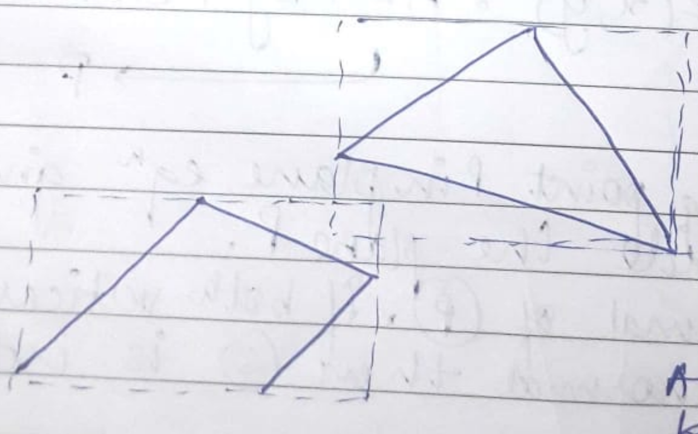


Bounding rect. overlap but not figure

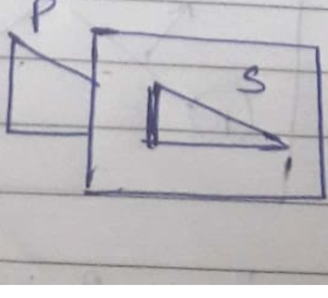
Case :



S is not inside of P and P is not inside of S



if there is overlapping with actual surface S and P
Apply polygon clipping of P & reapply whole procedure again.



stion.

Triangle 1 T_{11} (0, 7, 5)

T_{12} (6, 7, 5)

T_{13} (0, 1, 5)

Triangle 2 T_{21} (0, 6, 7)

T_{22} (0, 1, 2)

T_{23} (5, 1, 7)

Date

Page