Q. If pixels are accessed from the frame buffer, with an average access time of 20 ns and the resolution of the screen is 1024 x 1024, will there be a flickering effect seen on the screen?

→ Solⁿ:

Resolution = 1024 x 1024

access time = 20 ns = 20×10^{-9} s

To access 1 resolution we need 20×10^{-9} s

For accessing 1024 x 1024 resolution we need

$= 1024 \times 1024 \times 20 \times 10^{-9}$

$T = 0.020975$ s

$\therefore F = \frac{1}{T} = \frac{1}{0.020975}$

$\therefore F = 47.68$ fps

∴ There will be flickering effect since the standard value of frequency is 50 fps.

Q. How long will it take to load a 640 X 480 frame buffer with 12 bits per pixel if 10^5 bits can be transferred per second?

→ Solⁿ:

Resolution = 640 X 480

Frame buffer = 12 bits

Scanned by CamScanner

So, Total number of buffer = $640 \times 480 \times 12$
$= 3686400$

$10^5 \longrightarrow 1$ second

$1s \longrightarrow \frac{1}{10^5}$

$3686400 \longrightarrow \frac{1s}{10^5} \times 3686400$

$= 36.864$ s

$\therefore$ It takes 36.864s time.

Q. A system has a resolution of 640×480, how many pixels can be accessed per second in this slm by a video controller that refreshes the screen at the rate of 60 frames per second? What will be the access time per pixel in this system?

→ Soln:

Resolution = 640×480

In 1 frame there are 640×480 pixels.
In 1 sec, the slm refreshes 60 fps
$= 640 \times 480 \times 60$ pixels.

The access time per pixel is $\frac{1}{640 \times 480 \times 60}$

$= 54.25$ ns

Scanned by CamScanner

Q. A raster scan system has 12 inch by 10 inch screen with a resolution of 100 pixels per inch in each direction, if the video controller refreshes the screen at a rate of 50 fps, how many pixels can be accessed per second? What is the access time per pixel of the screen?

→ Solⁿ:

Given:-

The size of slm = 12 X 10

Resolution = 1200 x 1000

Frequency = 50 fps

Size of pixel = 1200 x 1000 x 50 = 6×10^7

So, the access time per pixel = $\frac{1}{1200 \times 1000 \times 50}$

$= \frac{1}{6 \times 10^7}$

$= 16$ ns

Q. A mobile phone has resolution of 1920 X 1080 pixels, with 423 ppi. What is the size of the mobile screen?

→ Solⁿ:

one inch = 423 ppi

~~X 1920 X 1080~~

Resolution = 1920 x 1080

Then;

size = 4.53 X 2.55

Scanned by CamScanner

Then;

size of mobile screen $= \sqrt{4.5^2 + 2.5^2}$

$= 5.14$

Q. Compute the resolution/sq inch of 2x2 inch that has 512x512 pixels in both directions.

→ Sol^n:

Pixel = 512x512

size = 2x2

$\therefore$ resolution $= \frac{512}{2} \times \frac{512}{2}$

$= 256 \times 256$.

Q. Compute the size of 640x480 image at 240 pixels per inch.

→ Sol^n:

resolution size = 640 x 480.

pixel = 240

$\therefore$ size $= \frac{640}{240} \times \frac{480}{240}$

$= 2.667 \times 2$.

Scanned by CamScanner

Q. If we want to resize a 1024 x 768 image to one that is 640 pixels wide with the same aspect ratio, what would be the height of the image?

→ Solⁿ:

size = 1024 x 768

pixel = 640

Then;

$$\frac{1024}{768} = \frac{640}{H}$$

or $H = 480$. pixel.

Scanned by CamScanner

* Line Drawing Algorithms
1. DDA
2. BLA

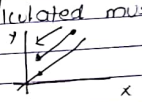
DDA	BLA
- Floating point based.	Integer point based.
- Slower (needs to be convert)	Faster.

* Digital Differential Analyzer (DDA)
- a scan conversion algorithm based on calculating either $\Delta x \Delta y$.
- we sample at unit interval in one coordinate and determine corresponding integer value nearest the line path in the other coordinate.
- it makes use of floating point arithmetic thus making it slow and errors may get accumulated.

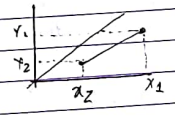
* Line with positive slope (moving from left to right)
1. Line with slope ≤ 1.
Here we sample at unit 'x' interval ($\Delta x = 1$) & each successive 'y' value is computed as
$y_{k+1} = y_k + m$.
the 'y' value calculated must be rounded off to the nearest integer.

2. Line with slope > 1.
Here we sample at unit 'y' interval ($\Delta y = 1$) and each successive 'x' value is computed as $x_{k+1} = x_k + 1/m$
the 'x' value calculated must be rounded off to nearest integer.

* Line with +ve slope (moving from right to left)
3. Line with slope > 1.
Here we sample at unity 'y' interval ($\Delta y = -1$) and each successive 'x' value is computed as
$x_{k+1} = x_k - \frac{1}{m}$ the 'x' value calculated must be rounded off to nearest integer.



4. Line with slope ≤ 1
Here we sample at unit 'x' interval ($\Delta x = -1$) & each successive 'y' value is computed as $y_{k+1} = y_k - m$
the 'y' value calculated must be rounded off to nearest integer.



* Line with -ve slope (moving from left to right)
5. Line with slope ≤ 1
Here we sample at unit 'x' interval ($\Delta x = -1$) and each successive 'y' value is

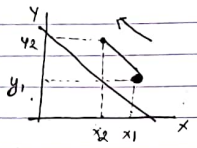
Scanned by CamScanner

computed as $y_{k+1} = y_k + m$ the (y) value calculated must be rounded off to the nearest integer.

6. Line with slope > 1

Here we sample at unit (y) interval ($\Delta y = -1$) and each successive (x) value is computed as

$x_{k+1} = x_k - \frac{1}{m}$ the (x) value calculated must be rounded off to the nearest integer.



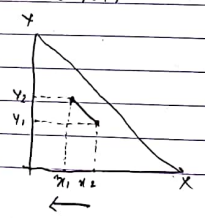
7. Line with slope > 1 (moving from right to left)

$\Delta y = 1$ " "

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

8. Line with slope ≤ 1 (From rt to left)

$\Delta x = -1$ $y_{k+1} = y_k - m$



Q. Digitize a line with end points A (2,1) B (7,3) using DDA.

→ Soln:

we have,

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3-1}{7-2} = \frac{2}{5} = 0.4$$

x_{k+1} ($x_k + 1$)	y_{k+1} ($y_k + m$)	rounded value
3	$y_{k+1} = 1 + 0.4 = 1.4$	1
4	$1.4 + 0.4 = 1.8$	2
5	$1.8 + 0.4 = 2.2$	2
6	2.6	3
7	3	3

Q. A (4,8) & B (7,4)

$$m = \frac{-4}{3} = -1.33$$

x_{k+1}	y_{k+1} ($y_{k+1} + m$)	rounded value
5	$8 - 1.33 = 6.77$	7
6	$6.77 - 1.33 = 5.34$	5
7	$5.34 - 1.33 = 4.01$	4

Scanned by CamScanner

Q. A (3, 8) B (4, 5) [right to left)

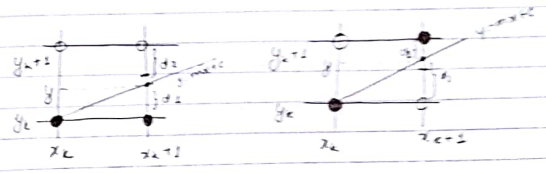
slope (m) = $\frac{5-8}{4-3} = \frac{-3}{-4} = \frac{3}{4} = 0.75$

y_{k+1}	x_{k+1}	rounded value
8		
6	4+1.33 = 5.33	5
7	5.33+1.33 = 6.66	7
8	6.66+1.33 = 7.99	8

$(x_{k+1} = x_k - 1)$

x_{k+1}	y_{k+1}	rounded value
7	8 - 0.75 = 7.25	7
6	7.25 - 0.75 = 6.5	7
5	6.5 - 0.75 = 5.75	6
4	5.75 - 0.75 = 5	5

Bresenham's Line Drawing Algorithm for line with slope ≤ 1



$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$
$d_1 - d_2 < 0$

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k + 1$
$d_1 - d_2 > 0$

We have equation of line is $y = m(x_k + 1) + c$ $(x = x_k + 1)$

$d_1 = y - y_k = m(x_k + 1) - y_k$

$d_2 = y_k + 1 - y = y_k + 1 - m(x_k + 1) - c$

Now,

$d_1 - d_2 = 2m(x_k + 1) + 2c - 2y_k - 1$

we have, $m = \frac{\Delta y}{\Delta x}$

so, $(d_1 - d_2)\Delta x = 2\Delta y \cdot x_k + 2\Delta y + 2c\Delta x - 2y_k \cdot \Delta x - \Delta x$

or, $P_k = 2\Delta y\, x_k - 2\Delta x\, y_k + b$ (for kth term)

where,

$b = 2\Delta y + 2c\Delta x - \Delta x$

$P_k = \Delta x(d_1 - d_2)$ decision parameter

Then, ~~from~~ for k+1th term

$P_{k+1} = 2\Delta y\, x_{k+1} - 2\Delta x\, y_{k+1} + b$

Scanned by CamScanner

$P_{k+1} = P_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$

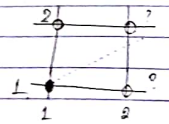
Also,

$P_{k+1} = P_k + 2\Delta y$ (for $y_{k+1} = y_k$)

and

$P_{k+1} = P_k + 2\Delta y - 2\Delta x$ (for $y_{k+1} - y_k = 1$)

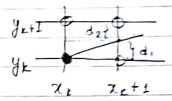
Accurate and easier raster line drawing Algorithm developed by Bresenham. For line with slope ≤ 1, we sample at unit 'x' interval in 'x' direction and determine which of the 2 possible positions is closer to the ideal line path at each sample step i.e. if we have plotted (1,1) next pixel to plot is either (2,1) or (2,2). Bresenham's algorithm gives solution to this problem by testing the sign of integer parameter whose value is proportional to the difference between the separation of 2 pixel positions from actual line path for line with slope less than equal to 1 pixel positions are determined by sampling at unit 'x' interval starting at (x_0, y_0) from left end. The idea is to find the value of 'y' closest to the line path.



For any k^{th} step, if pixel at (x_k, y_k) has been plotted/displayed then we need to decide which next pixel to plot in the x_{k+1} column i.e. either (x_{k+1}, y_k) or (x_{k+1}, y_k+1). At x_{k+1} label vertical pixel separations from line path as d_1, d_2. Now y coordinate at x_k+1 will be $y = m(x_k+1) + b$.

So, $d_1 = y - y_k = m(x_k+1) + b - y_k$.

(d_1 is the diff of lower pixel from ideal location)



$d_2 = y_{k+1} - y = y_k + 1 - m(x_k+1) - b$

(d_2 is the diff of ideal location from upper pixel)

Thus, the difference $d_1 - d_2 = 2m(x_k+1) - 2y_k + 2b - 1$

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

$\Delta x(d_1 - d_2) = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$ where, $c = 2\Delta y + \Delta x\, 2b - \Delta x$.

or,

$P_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$ — (I)

where P_k is the decision parameter at kth step.

At $(k+1)^{th}$ step the decision parameter will be

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

Subtracting (I) & (II) & also since $x_{k+1} - x_k = 1$

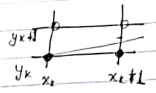
$$\boxed{P_{k+1} = P_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)} \quad \text{— (III)}$$

case 1

if pixel at y_k is closer to the line path then $d_1 < d_2$ (P_k is -ve) and $y_{k+1} = y_k$, next pixel to plot

Scanned by CamScanner

is (x_k+1, y_k), eq (iii) becomes

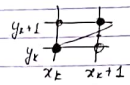


$P_{k+1} = P_k + 2\Delta y$ //

case II

if pixel of y_k+1 is closer to the line path then $d_2 < d_1$ i.e. P_k is +ve, next pixel to plot is (x_k+1, y_k+1) i.e. $y_{k+1} = y_k+1$, eq^n (iii) becomes

$P_{k+1} = P_k + 2\Delta y - 2\Delta x$



[NOTE: if asked initial decision parameter P_0 = ? upto]

We have;

$d_1 - d_2 = 2m(x_k+1) - 2y_k + 2c - 1$

if line passes thru (x_0, y_0)

$d_1 - d_2 = 2m(x_0+1) - 2y_0 + 2c - 1$

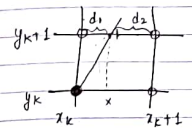
$d_1 - d_2 = (2mx_0 - 2y_0 + 2c) + 2m - 1$

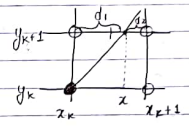
$d_1 - d_2 = 2m - 1$ $(\because 2mx_0 - 2y_0 + 2c = 0)$

or, $\Delta x(d_1 - d_2) = 2\Delta y - \Delta x$

$\boxed{\therefore P_0 = 2\Delta y - \Delta x}$

*. BLA for $|m| > 1$.





Here;

$y_{k+1} = y_k + 1$ $\qquad$ $y_{k+1} = y_k + 1$

$x_{k+1} = x_k$ $\qquad$ $x_{k+1} = x_k + 1$

$y = mx + c$

or, $y_k + 1 = mx + c$

or, $x = \dfrac{y_k + 1 - c}{m}$

$d_1 = x - x_k$

or, $d_1 = \dfrac{y_k + 1 - c}{m} - x_k$

or, $md_1 = y_k + 1 - c - mx_k$

$d_2 = x_k + 1 - x$

or, $d_2 = x_k + 1 - \dfrac{y_k + 1 - c}{m}$

or,

Scanned by CamScanner

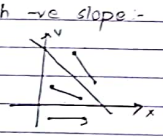
* Summary of Bresenhams Algorithm for $|m|<1$

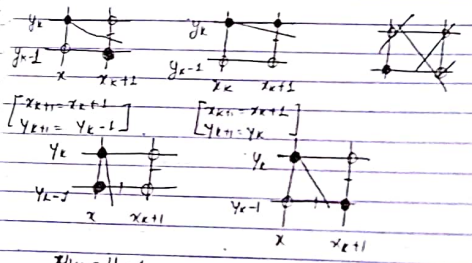
1. given 2 end points, plot left end point (x_0, y_0)
2. calculate $\Delta x, \Delta y, 2\Delta y - 2\Delta x, 2\Delta y$
3. obtain the initial decision parameter as

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

4. for each of the succeeding column if $P_k < 0$ then pixel to plot is (x_k+1, y) and $P_{k+1} = P_k + 2\Delta y$
   if $P_k \geq 0$ then pixel to plot is (x_k+1, y_k+1) &
   $P_{k+1} = P_k + 2\Delta y - 2\Delta x$
5. Repeat step 4 Δx times

Lines with -ve slope -





$y_{k+1} = y_{k-1}$
$x_{k+1} = x_k$

$y_{k+1} = y_{k-1}$
$x_{k+1} = x_k + 1$

8. Digitize a line with end points A (2,1) B(5,2) using BLA

→ Sol^n: $m = \frac{1}{3}(<1)$ $\Delta y = 1$ $\Delta x = 3$

$P_0 = 2 - 3 = -1$. $(x_0, y_0) = (2,1)$

k	P_k	x_{k+1}	y_{k+1}
k=0	$P_0 = -1$	3	1
k=1	$P_1 = -1+2 = 1$	4	2
k=2	$P_2 = 1+2-6 = -3$	5	2

9. A(20,10) B(30,13)

→ Sol^n:

$m = \frac{3}{10}(<1)$, $\Delta y = 3$ $\Delta x = 10$

$P_0 = 2\times3 - 10 = -4$ $(x_0, y_0) = (20,10)$

k	P_k	x_{k+1}	y_{k+1}
k=0	$P_0 = -4$	21	10
k=1	$P_1 = -4+6 = 2$	22	11
k=2	$P_2 = 2+6-20 = -12$	23	11
k=3	$P_3 = -12+6 = -6$	24	11
k=4	$P_4 = -6+6 = 0$	25	12
k=5	$P_5 = 0+6-20 = -14$	26	12
k=6	$P_6 = -14+6 = -8$	27	12

Scanned by CamScanner

$k=7$	$p_7 = -8+6 = -2$	28	12
$k=8$	$p_8 = -2+6 = 4$	29	13
$k=9$	$p_9 = 4+6-20 = -10$	30	13
~~$k=10$~~	~~$p_{10} = -10+6 = -4$~~	~~30~~	~~13~~

$\therefore$ end point $= (30, 13)$

Q. $A(10, 5)$ $B(20, 13)$

$m = \frac{8}{10} = \frac{4}{5}$ (<1) $\Delta y = 8$ $\Delta x = 10$

$P_0 = 2\times 8 - 10 = 16 - 10 = 6$

k	P_k	x_{k+1}	y_{k+1}
$k=0$	$p_0 = 6$	11	6
$k=1$	$p_1 = 6+16-20 = 2$	12	7
$k=2$	$p_2 = 2+16-20 = -2$	13	7
$k=3$	$p_3 = -2+16 = 14$	14	8
$k=4$	$p_4 = 14+16-20 = 10$	15	9
$k=5$	$p_5 = 10+16-20 = 6$	16	10
$k=6$	$p_6 = 6+16-20 = 2$	17	11
$k=7$	$p_7 = 2+16-20 = -2$	18	11
$k=8$	$p_8 = -2+16 = 14$	19	12
$k=9$	$p_9 = 14+16-20 = 10$	20	13

$\therefore$ end points $= (20, 13)$

Q. $A(1, 1)$ $B(5, 8)$

$m = \frac{7}{4}$ (>1) $\Delta y = 7$ $\Delta x = 4$

$P_0 = 2\times 4 - 7 = 8 - 7 = 1$ $x_0, y_0 = (1, 1)$

k	P_k	x_{k+1}	y_{k+1}
$k=0$	$P_0 = 1$	2	2
$k=1$	$P_1 = 1+8-14 = -5$	2	3
$k=2$	$P_2 = -5+8 = 3$	3	4
$k=3$	$P_3 = 3+8-14 = -3$	3	5
$k=4$	$P_4 = -3+8 = 5$	4	6
$k=5$	$P_5 = 5+8-14 = -1$	4	7
$k=6$	$P_6 = -1+8 = 7$	5	8

$\therefore$ end points $= (5, 8)$

Scanned by CamScanner

k=7
k=8
k=9
~~k=10~~

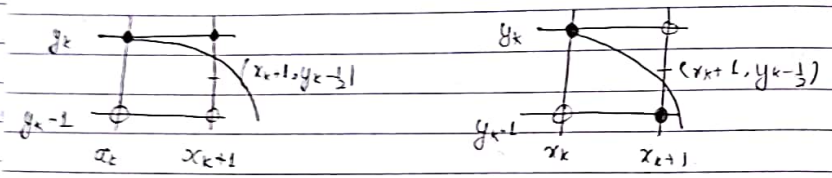
∴

Q.

k

k=0
k=1
k=2
k=3
k=4
k=5
k=6
k=7
k=8
k=9

* Mid point circle Algorithm



$x_{k+1} = x_k+1$ $\qquad$ $x_{k+1} = x_k+1$

$y_{k+1} = y_k$ $\qquad$ $y_{k+1} = y_k - 1$

Now;

$P_k = f_{circle}(x_k+1, y_k-\frac{1}{2})$

$P_k = (x_k+1)^2 + (y_k-\frac{1}{2})^2 - r^2$ — (I)

and,

$P_{k+1} = ((x_k+1)+1)^2 + (y_{k+1}-\frac{1}{2})^2 - r^2$ $\quad (\because x_k+1 = x_{k+1})$

or, $P_{k+1} = (x_k+1)^2 + 2(x_k+1) + 1 + (y_{k+1}-\frac{1}{2})^2 - r^2$ — (II)

From (I) & (II)

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

Two cases:

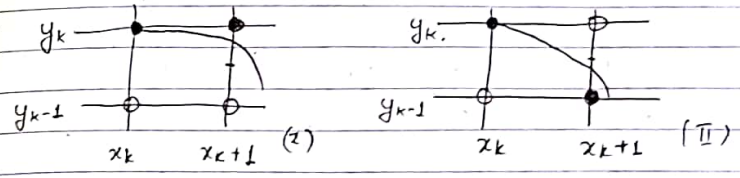
In case 1,

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

In case 2:

$P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1$

* Mid point Circle Algorithm :-



$x_{k+1} = x_k+1$ $\qquad$ $x_{k+1} = x_k+1$

$y_{k+1} = y_k$ $\qquad$ $y_{k+1} = y_k - 1$

For k^{th} step,

$P_k = f_{circle}(x_k+1, y_k-\frac{1}{2})$

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

For $(k+1)^{th}$ step,

$p_{k+1} = [(x_k+1)+1]^2 + (y_{k+1}-\frac{1}{2})^2 - r^2$ $\quad [\because x_{k+1} = x_k+1]$

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

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

From fig I

$P_{k+1} = P_k + 2(x_k+1) + 1$ — ①

Scanned by CamScanner

from fig II;

$$P_{k+1} = P_k + 2(x_k+1) + (y_k - \tfrac{3}{2})^2 - (y_k - \tfrac{1}{2})^2 + 1$$

$$= P_k + 2(x_k+1) + \left[y_k - 3y_k + \tfrac{9}{4} - y_k^2 + y_k - \tfrac{1}{4} \right] + 1$$

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

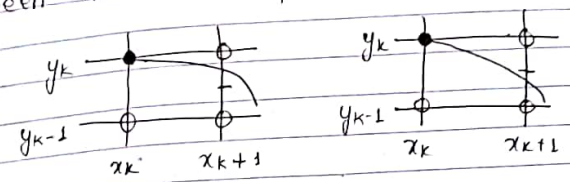
$$= P_k + 2(x_{k+1}) - 2(y_k - 1) + 1$$

$$\therefore P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1 \quad // \qquad [\because y_{k+1} = y_k - 1]$$

The equation of a circle is given by $x^2+y^2=r^2$, to apply the midpoint method, we define a circle function as $F_{circle}(x,y) = x^2+y^2-r^2$ and $f_{circle}(x,y) < 0$ if (x,y) is inside circle boundary, $=0$ if (x,y) is on circle boundary, >0 if (x,y) is outside circle boundary.

This circle function serves as the decision parameter.

Select the next pixel along the circle path according to the sign of this circle function evaluated at the mid point between 2 candidate pixels. Start at $(0,y)$ take unit step in (x) direction (sample in x direction $x_{k+1} = x_k+1$). Assuming the pixel at (x_k, y_k) has been plotted at previous step, we determine next position (x_{k+1}, y_{k+1}) as either (x_{k+1}, y_k) or (x_{k+1}, y_k-1) along circle path by evaluating the decision parameter (circle function). The decision parameter is the circle function evaluated at the mid point between these two pixels.



The decision parameter at k^{th} step ~~$P_k = F_{circle}(x_{k+1}, y_k - \frac{1}{2})$~~

$$P_k = F_{circle}(x_{k+1}, y_k - \tfrac{1}{2}) = (x_{k+1})^2 + (y_k - \tfrac{1}{2})^2 - r^2 \quad \text{— (I)}$$

At next sampling position $(x_{k+1}+1 = x_k+2)$ the decision parameter is evaluated as

$$P_{k+1} = F_{circle}(x_{k+1}+1, y_{k+1} - \tfrac{1}{2}) = [(x_{k+1})+1]^2 + (y_{k+1} - \tfrac{1}{2})^2 - r^2$$

$$= (x_k+2)^2 + 2(x_k+1) + 1 + (y_{k+1} - \tfrac{1}{2})^2 - r^2 \quad \text{— (II)}$$

Subtracting (I) and (II)

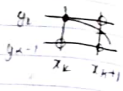
$$P_{k+1} = P_k + 2(x_{k+1}) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1 \quad \text{— (III)}$$

where y_{k+1} is either y_k or $y_k - 1$ depending on the sign of P_k.

cases.

If $P_k < 0$ then mid point is inside the circle boundary so pixel on scan line 'y_k' is closer to the circle boundary and $y_{k+1} = y_k$

from eq^n (ii)



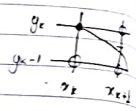
$P_{k+1} = P_k + 2x_{k+1} + 1$ where, $x_{k+1} = x_k + 1$

$2x_{k+1} = 2x_k + 2$

Case II

if $P_k \geq 0$ then midpoint is outside or on the boundary of circle so we select pixel on scan line $(y_k - 1)$ then $y_{k+1} = y_k - 1$

from eq^n (iii)



$P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1$ where,

$2y_{k+1} = 2y_k - 2$

$2y_{k+1} = 2x_k + 2$

The initial decision parameter $P_0 = ?$ can be obtained by evaluating the circle function at start point $(x_0, y_0) = (0, r)$. next pixel to plot is either $(1, r)$ or $(1, r-1)$. So, midpoint coordinate position is $(1, r - 1/2)$

So, $f_{circle}(1, r-\frac{1}{2}) = 1 + (r - \frac{1}{2})^2 - r^2$

$= 1 + r^2 - r + \frac{1}{4} - r^2$

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

If the radius is specified as an integer we can round off P_0 to $P_0 = 1 - r$.

Midpoint Circle Algorithm:-

1. Input radius 'r' circle center (x_c, y_c) and obtain the first point on the circumference of circle centered on origin as $(x_0, y_0) = (0, r)$
2. Calculate initial decision parameter value as $P_0 = 5/4 - r$ or $1 - r$.
3. At each x_k position, starting at $k = 0$ perform tests if $P_k < 0$ then next point to plot is $(x_k + 1, y_k)$ and $P_{k+1} = P_k + 2x_{k+1} + 1$ else next point to plot is $(x_k + 1, y_k - 1)$ and $P_{k+1} = P_k + 2x_{k+1} - 2y_{k+1} + 1$

   where, $2x_{k+1} = 2x_k + 2$

   $2y_{k+1} = 2y_k - 2$
4. Determine symmetry points in other seven octants
5. move each calculated pixel position (x, y) onto circular path centered on (x_c, y_c) and plot $x = x + x_c$ & $y = y + y_c$.
6. Repeat steps 3 through 5 until $x \geq y$.

Q4.

Scanned by CamScanner

Q. Digitize a circle with a radius of 5 pixels.
starting pixel $P_0 = 1-r$ $(x_0, y_0) = (0, 5)$

→

k	P_k	x_{k+1}	y_{k+1}
0	$P_0 = 1-5 = -4$	1	5
1	$P_1 = -4+2+1 = -1$	2	5
2	$P_2 = -1+4+1 = 4$	3	4
3	$P_3 = 4+6-8+1 = 3$	4	3

//

Q. Digitize a circle with radius of 12 pixels.
starting pixel $P_0 = 1-r$ $(x_0, y_0) = (0, r) = (0, 12)$

→ $P_0 = 1-r = 1-12 = -11$

k	P_k	x_{k+1}	y_{k+1}
0	$P_0 = -11$	1	12
1	$P_1 = -11+2+1 = -8$	2	12
2	$P_2 = -8+4+1 = -3$	3	12
3	$P_3 = -3+6+1 = 4$	4	11
4	$P_4 = 4+8-22+1 = -9$	5	11
5	$P_5 = -9+10+1 = 2$	6	10
6	$P_6 = 2+12-20+1 = -5$	7	10
7	$P_7 = -5+14+1 = 10$	8	9
8	$P_8 = 10+16-18+1 = 9$	9	8

//

P_1	P_2	P_2	P_1	P_3
0	1	2	4	8	35

W.T

$P_1 = (0-0)+(4-1) = 0+3 = 3$

$P_2 = 1-1 = 0$

$P_3 = 8-2 = 6$

*. Digitize a circle with 8 pixels and centered at (5,3)
Let starting pixel $P_0 = 1-r = 1-8 = -7$
$(x_0, y_0) = (0, 8)$

k	P_k	x_{k+1}	y_{k+1}
0	$P_0 = -7$	1	8
1	$P_1 = -7+2\times1+1 = -4$	2	8
2	$P_2 = -4+4+1 = 1$	3	7
3	$P_3 = 1+6-14+1 = -6$	4	7
4	$P_4 = -6+8+1 = 3$	5	6
5	$P_5 = 3+10-12+1 = 2$	6	5

$\therefore x_{k+1} = 6+5 = 11$

$y_{k+1} = 5+3 = 8$

since the circle is centered at (5,3) then,

$x = x+x_c = 5+6 = 11$

$y = y+y_c = 5+3 = 8$

So;

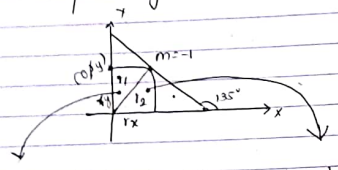
k	P_k	x_{k+1}	y_{k+1}
0	P_0	$1+5 = 6$	$8+3 = 11$
1	P_1	$2+5 = 7$	$8+3 = 11$

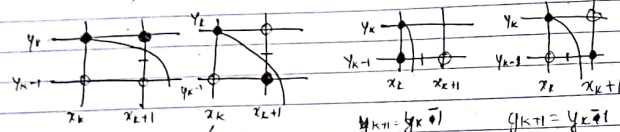
Scanned by CamScanner

k	P_k	x_{k+1}	y_{k+1}
2	P_2	3+5=8	7+3=10
3	P_3	4+5=9	7+3=10
4	P_4	5+5=10	6+3=9
5	P_5	6+5=11	5+3=8

$\therefore (x, y) = (11, 8)$

★ Mid-point Ellipse Algorithm





$x_{k+1} = x_k + 1$ $\quad$ $y_{k+1} = y_k - 1$ $\quad$ $y_{k+1} = y_k - 1$ $\quad$ $y_{k+1} = y_k - 1$

$y_{k+1} = y_k$ $\quad$ $x_{k+1} = x_k + 1$ $\quad$ $x_{k+1} = x_k$ $\quad$ $x_{k+1} = x_k + 1$

→ $f_{ellipse}(x, y) = x^2 r_y^2 + y^2 \cdot r_x^2 - r_x^2 r_y^2$

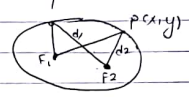
$f_{ellipse}(x_k + 1, y_k - r_2)$

$P_1 = (x_k + 1)^2 r_y^2 + \left(y_k - \frac{1}{2}\right)^2 r_x^2 - r_x^2 r_y^2$

Region 1

An ellipse is defined as a set of points such that the sum of distances from the two fixed point is same for all points. If the distance to two fixed point from any point $P(x, y)$ on ellipse are d_1, d_2 then the general equation of ellipse is $d_1 + d_2 = \text{const}$. Or expressing distance d_1, d_2 in terms of focal coordinate $f_1(x_1, y_1)$ & $f_2(x_2, y_2)$ we have $\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} = \text{const}$

Mid point ellipse method is applied throughout the first quadrant in 2 parts (according to the slope of ellipse)



Ellipse equation is given by $\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} = 1$ or

$f_{ellipse}(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$

< 0 if (x, y) is inside ellipse boundary
$= 0$ " " on " "
> 0 " " outside " "

This ellipse function serves as the decision parameter we select the next pixel along the ellipse path according to the sign of ellipse function ~~evaluated~~ at the mid point between a candidate pixels.

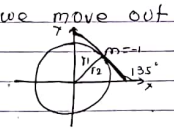
Scanned by CamScanner

Start at $(0, r_y)$ take unit steps in 'x' direction until we reach boundary region 1 and 2 then switch to unit steps in 'y' direction for remainder of curve in first quadrant. At each step test the value of the slope of curve. The ellipse slope is given by

$$2r_y^2 \cdot x + 2r_x^2 y \frac{dy}{dx} = 0 \quad \text{or} \quad \frac{dy}{dx} = -\frac{2r_y^2 x}{2r_x^2 y} \quad \text{at boundary}$$

between region 1 and 2 $\frac{dy}{dx} = -1$ so,

$2r_y^2 x = 2r_x^2 y$ and we move out of region 1 when $2r_y^2 x \geq 2r_x^2 y$



Assuming position (x_k, y_k) has been selected at previous step we determine next position (x_{k+1}, y_{k+1}) as either (x_k+1, y_k) or (x_k+1, y_k-1) along elliptic path be evaluating the decision parameter

$$P1_k = f_{ellipse}(x_k+1, y_k - 1/2)$$
$$= r_y^2(x_k+1)^2 + r_x^2(y_k - \tfrac{1}{2})^2 - r_x^2 r_y^2 \quad \text{—— (1)}$$

at next sampling position $(x_{k+1} = x_k+2)$ the decision parameter for region 1 is evaluated as

$$P1_{k+1} = f_{ellipse}(x_{k+1}+1, y_{k+1} - \tfrac{1}{2}) = r_y^2[(x_{k+1})+1]^2 + r_x^2(y_{k+1} - \tfrac{1}{2})^2 - r_x^2 r_y^2 \quad \text{—— (ii)}$$

Subtracting (i) and (ii)

$$P1_{k+1} = P1_k + 2r_y^2(x_k+1) + r_y^2 + r_x^2[(y_{k+1} - \tfrac{1}{2})^2 - (y_k - \tfrac{1}{2})^2] \quad \text{—— (iii)}$$

where, y_{k+1} is either y_k or $y_k - 1$ depending on sign of $P1_k$.

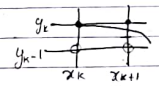
Case 1:
if $P1_k < 0$ then midpoint is inside ellipse so pixel on scaline (y_k) is closer to ellipse boundary and $y_{k+1} = y_k$ so increment will be $2r_y^2 x_{k+1} + r_y^2$

From eq^n (iii)

$$P1_{k+1} = P1_k + 2r_y^2 x_{k+1} + r_y^2 \quad \text{—— a}$$

where, $x_{k+1} = x_k + 1$

$$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2$$



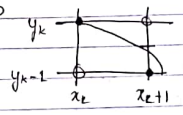
case 2:
if $P1_k \geq 0$ then midpoint is outside or on the ellipse boundary so we select pixel on scaline $(y_k - 1)$ and $y_{k+1} = y_k - 1$ the increment will be $2r_y^2 x_{k+1} - 2r_x^2 y_{k+1}$

From eq^n (iii)

$$P1_{k+1} = P1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2 \quad \text{—— b}$$

where,

$$2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$$
$$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2$$

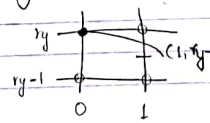


Initial decision parameter for region 1 $P1_0 = ?$

The starting position is $(0, r_y)$ next pixel to plot is either $(1, r_y)$ or $(1, r_y - 1)$ so midpoint coordinate is $(1, r_y - \frac{1}{2})$

$$f_{ellipse}(1, r_y - \tfrac{1}{2}) = r_y^2 + r_x^2(r_y - \tfrac{1}{2})^2 - r_y^2 r_x^2$$

thus, $P1_0 = r_y^2 + \frac{1}{4} r_x^2 - r_x^2 r_y$



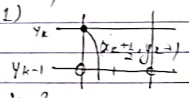
Scanned by CamScanner

For Region 2:

Sample at unit steps in 'y' interval, the midpoint is taken between horizontal pixels at each step now. Assuming (x_k, y_k) has been plotted, next pixel to plot is either (x_k, y_k-1) or (x_k+1, y_k-1) and midpoint coordinate is $(x_k+\frac{1}{2}, y_k-1)$

Fellipse $(x_k+\frac{1}{2}, y_k-1)$ or $P2_k = r_y^2\left(x_k+\frac{1}{2}\right)^2 + r_x^2(y_k-1)^2 - r_x^2 r_y^2$ —(iv)

At next sampling position, the next pixel to plot will either be $(x_{k+1}, y_{k+1}-1)$ or $(x_{k+1}+1, y_{k+1}-1)$



thus,

Fellipse $(x_{k+1}+y_2, y_{k+1}-1)$

as $P2_{k+1} = r_y^2(x_{k+1}+\frac{1}{2})^2 + r_x^2(y_{k+1}-1)^2 - r_x^2 r_y^2$

$= r_y^2(x_{k+1}+\frac{1}{2})^2 + r_x^2[(y_{k+1})-1]^2 - r_x^2 r_y^2$ —(v)

Subtracting (iv) & v

$P2_{k+1} = P2_k - 2r_x^2(y_k-1) + r_x^2 + r_y^2[(x_{k+1}+\frac{1}{2})^2 - (x_k+\frac{1}{2})^2]$ —(vi)

where, x_{k+1} is either x_k or x_k+1 depending on the sign of $P2_k$.

Case 1

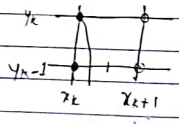
if $P2_k > 0$ then midpoint is outside the boundary of ellipse and we select pixel at 'x_k' and

$P2_{k+1} = P2_k - 2r_x^2(y_k-1) + r_x^2$

$= P2_k - 2r_x^2 y_{k+1} + r_x^2$

where, $y_{k+1} = y_k - 1$

$2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$



Case 2

if $P2_k \le 0$ then midpoint is inside or on ellipse boundary so we select pixel at 'x_k+1' i.e. $x_{k+1} = x_k+1$

From eq^n (vi)

$P2_{k+1} = P2_k - 2r_x^2(y_k-1) + r_x^2 + r_y^2[(x_{k+1}+\frac{1}{2})^2 - (x_k+\frac{1}{2})^2]$

$= \quad '' \quad '' \quad '' \quad [(x_k+1)+\frac{1}{2}]^2 - (x_k+\frac{1}{2})^2]$

$= \quad '' \quad '' \quad '' \quad [(x_k+3/2)^2 - (x_k+\frac{1}{2})^2]$

$= \quad '' \quad '' \quad '' \quad [x_k^2+3x_k+\frac{9}{4} - x_k^2 - x - \frac{1}{4}]$

$= \quad '' \quad '' \quad [2x_k+2]$

$= P2_k - 2r_x^2 y_{k+1} + r_x^2 + 2r_y^2 x_{k+1}$,,

where,

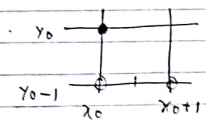
$2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$

$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2$

For region 2,

the initial position $(x_0 y_0)$ is taken as the last position selected in region 1 and the initial decision parameter in region 2 is;

$P2_0 = \text{Fellipse}(x_0+\frac{1}{2}, y_0-1) = r_y^2(x_0+\frac{1}{2})^2 + r_x^2(y_0-1)^2 - r_x^2 r_y^2$



Scanned by CamScanner

Midpoint Ellipse Algorithm

1. Input r_x, r_y and obtain first point on ellipse centered on origin as $(x_0, y_0) = (0, r_y)$
2. Calculate the initial value of decision parameter in region 1 as $P1_0 = r_y^2 + \frac{1}{4} r_x^2 - r_x^2 r_y$
3. At each x_k position in region 1, starting at $k=0$ perform tests:
   if $P1_k < 0$ then next point along ellipse centered on $(0,0)$ is (x_{k+1}, y_k) and
   $P1_{k+1} = P1_k + 2r_y^2 x_{k+1} + r_y^2$ else next pixel along ellipse is $(x_{k+1}, y_k - 1)$ with $P1_{k+1} = P1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$ and continue until $2r_y^2 x \geq 2r_x^2 y$.
4. Calculate the initial value of decision parameter in region 2 using the last point (x_0, y_0) calculate in region 1 as $P2_0 = r_y^2 (x_0 + \frac{1}{2})^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$
5. At each y_k position in region 2, starting at $k=0$ perform tests: if $P2_k > 0$ then next point along ellipse centered on $(0,0)$ is $(x_k, y_k - 1)$ and
   $P2_{k+1} = P2_k - 2r_x^2 y_{k+1} - r_x^2$ else next point along ellipse path is $(x_k + 1, y_k - 1)$ and
   $P2_{k+1} = P2_k - 2r_x^2 y_{k+1} + r_x^2 + 2r_y^2 x_{k+1}$.
6. Determine symmetry points in other 3 quadrants.
7. Repeat steps for region 2 until the value of 'y' becomes zero.

Q Generate an ellipse with $r_x = 8$ pixels $r_y = 6$ pixels $(x_0, y_0) = (0, 6)$

region 1: $P1_0 = r_y^2 + \frac{1}{4} r_x^2 - r_x^2 r_y = -332$

k	P_k	(x_{k+1}, y_{k+1})	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-332	(1, 6)	72	768
1				

Sol^n:

$P1_0 = r_y^2 + \frac{1}{4} r_x^2 - r_x^2 r_y = -332$

k	P_k	(x_{k+1}, y_{k+1})	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-332	(1, 6)	72	768
1	-224	(2, 6)	144	768
2	-44	(3, 6)	216	768
3	208	(4, 5)	288	640
4	-108	(5, 5)	360	640
5	288	(6, 4)	432	512
6	244	(7, 3)	504	$\geq$ 384

$\therefore (x_0, y_0) = (7, 3)$

$P2_0 = r_y^2 (x_0 + \frac{1}{2})^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$

$= 36 \left(\frac{7+1}{2}\right)^2 + 64 (3-1)^2 - 64 \times 36$

$= -23$

Scanned by CamScanner

k	p_{2k}	(x_{k+1}, y_{k+1})
0	-23	(8, 2)
1	361	(8, 1)
2	169	(8, 0) //

Q. Generate an ellipse with $r_x = 6$ $r_y = 8$

→ Sol^n:

$(x_0, y_0) = (0, 8)$

$P_{10} = r_y^2 + \frac{1}{4} r_x^2 - r_x^2 r_y$

$= -215$

k	P_{1k}	(x_{k+1}, y_{k+1})	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-215	(1, 8)	128	576
1	-23	(2, 8)	256	576
2	297	(3, 7)	384	504
3	241	(4, 6)	512	432

$\therefore (x_0, y_0) = (4, 6)$

Now,

$P_{20} = r_y^2 \left(x_0 + \frac{1}{2}\right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$

$= 64 \left(4 + \frac{1}{2}\right)^2 + 36 (6-1)^2 - 64 \times 36$

$= -108$

k	P_{2k}	(x_{k+1}, y_{k+1})
0	-108	(5, 5)
1	208	(5, 4)
2	-116	(6, 3)
3	472	(6, 2)
4	292	(6, 1)
5	184	(6, 0) //

Scanned by CamScanner

2D Transformations

* Translation :-

Repositioning an object along a straight line path from one coordinate location to another. And translational distance t_x, t_y to original coordinate position (x, y) to move a point to a new position (x', y') where (t_x, t_y) are translation vectors.

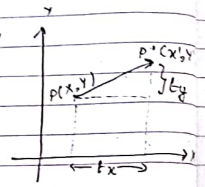
$x' = x + t_x \qquad y' = y + t_y$

writing the equation in a matrix form by using column vector to represent coordinate points and translation vectors.

$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$

$P' = P + T$

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



* Scaling :-

It changes the size of an object (magnify / reduce in size). Polygons are scaled by multiplying coordinate values (x, y) of each vertex by scaling factors s_x, s_y to produce the final transformation coordinate (x', y').

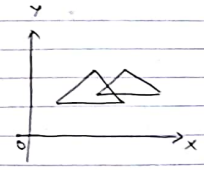
$x' = s_x \cdot x$

$y' = s_y \cdot y$ where s_x, s_y are scaling factor in x', y' direction

$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$

so, $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

or, $P' = S \times P$.



Here, the reference point is taken as the origin.

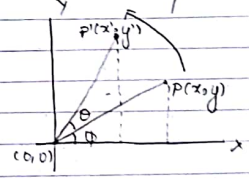
- values greater than 1 for s_x, s_y produce enlargement
- values less than 1 for s_x, s_y reduce the size of object.

$s_x = s_y = 1$, uniform scaling if $s_x = s_y$; non uniform scaling if $s_x \neq s_y$.

* Rotation :-

Repositioning an object along a circular path in x, y plane.

Specify rotation angle 'θ' and position (x_r, y_r) of rotation point about which the object is to be rotated. '+' value for 'θ' specify counter-clockwise rotation about a point and '-ve' values for 'θ' define clockwise rotation about a point. If (x, y) is the original point, 'r' the constant distance from origin 'ϕ' the original angular displacement from 'x' axis' - Now, the point (x, y) is rotated by angle 'θ' in counter clockwise.



Scanned by CamScanner

Expressing transformed coordinates in terms of 'ϕ' and 'θ' as $x' = r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$.

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

we know that original coordinate of a point in polar coordinate are $x = r\cos\phi$, $y = r\sin\phi$

Substituting x & y

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

using column vector representation for coordinate points in matrix form $P' = R \cdot P$.

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

Assumptions:-
- Default direction of rotation:- counter clockwise
- about origin
- by 'θ' angle.

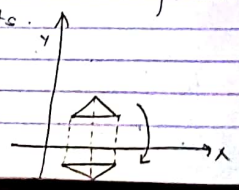
*. Reflection:-

Generates a mirror reflection of an object.

1) Reflection of an object about 'x' axis or line $y=0$ keeps 'x' value the same but flips the 'y' value of coordinate points.

$x' = x$ $\quad y' = -y$

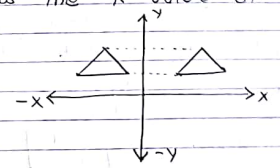
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



2. Reflection of an object about 'y' axis or line $x=0$ keep 'y' value the same but flips the 'x' value of coordinate points.

$x' = -x$ and, $y' = y$

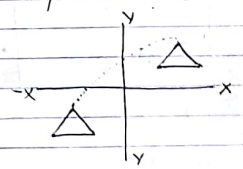
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



3. Reflection of an object about origin. flips the 'x', 'y' value of coordinate points

$x' = -x$ & $y' = -y$

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



4. Reflection of an object about line $y=x$.

Steps

i) The line along with the object is rotated so that it gets aligned with one of the major coordinate axes (i.e. x or y axis).

ii) Now, take reflection of the object about that ~~one~~ major coordinate axis.

iii) The line (along with the object) is rotated back to its original position.

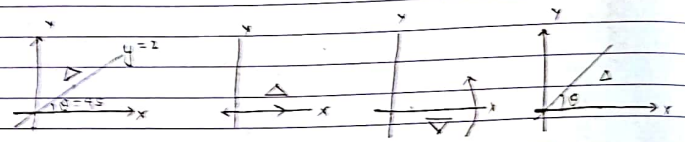
Scanned by CamScanner

The composite matrix (CM) = $R_{\theta}' \cdot R_f \cdot R_{\theta}$

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

$$= \begin{bmatrix} \cos 45 & -\sin 45 \\ \sin 45 & \cos 45 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos 45 & \sin 45 \\ -\sin 45 & \cos 45 \end{bmatrix}$$

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



*. Reflection of an object about line $y=-x$.

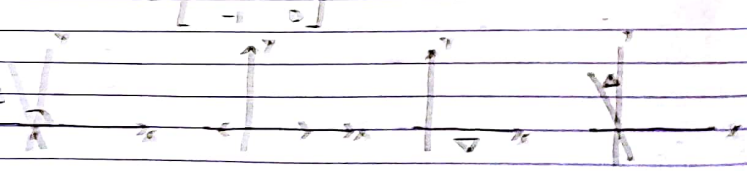
Steps :-

The composite matrix (CM) = $R_{\theta}' \cdot R_f \cdot R_{\theta}$

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

$$= \begin{bmatrix} \cos 135 & -\sin 135 \\ \sin 135 & \cos 135 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos 135 & \sin 135 \\ -\sin 135 & \cos 135 \end{bmatrix}$$

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



Q. Reflect a triangle with vertices A(-4,1) B(-2,1) C(-4,2) about line $y=-x$.

→ Sol^n:

~~Step~~ Steps :-

composite matrix (CM) = $R_{\theta=135}$ R_{f_x} $R_{\theta=135}$

$$= \begin{bmatrix} \cos 135 & -\sin 135 \\ \sin 135 & \cos 135 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos 135 & \sin 135 \\ -\sin 135 & \cos 135 \end{bmatrix}$$

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

Now,

$P' = CM \cdot P$

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

Scanned by CamScanner

Q. A point $P(0,3)$ is required to be shifted down by 2 units, rotated by 45 degrees in anticlockwise direction and scaled by twice its original size. what will be the final position of the point?

→ Sol^n:

Composite matrix (CM) = $\underset{S_x=S_y=2}{S} \; \underset{\theta=45^\circ \; ccw}{R} \; T_{(0,-2)}$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

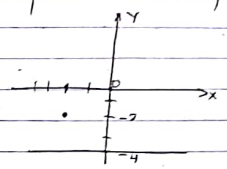
$$\alpha \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Q. A point $P(-2,-2)$ is required to be reflected about line $y=-4$, what will be final position of the point?

Sol^n:

→ Steps:-



$$CM = T_{(0,-4)} \, Rf_x \, T_{(0,4)}$$

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

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

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -8 \\ 0 & 0 & 1 \end{bmatrix}$$

Now, final position $= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -8 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ -2 \\ 1 \end{bmatrix}$

$$= \begin{bmatrix} -2 \\ -6 \\ 1 \end{bmatrix}$$

Scanned by CamScanner

*. Pivot point Rotation:-

To rotate an object about any fixed point $P(x_f, y_f)$

Steps:-

1. The fixed point along with object is translated to the coordinate origin.
2. The object is rotated by the desired angle.
3. The fixed point (along with object) is translated back to it's original location.

The composite matrix $CM = T_{(x_f, y_f)} \; R_\theta \; T_{(-x_f, -y_f)}$

$$= \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$CM = \begin{bmatrix} \cos\theta & -\sin\theta & x_f \\ \sin\theta & \cos\theta & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & -x_f\cos\theta + y_f\sin\theta + x_f \\ \sin\theta & \cos\theta & -x_f\sin\theta - y_f\cos\theta + y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_f(1-\cos\theta) + y_f\sin\theta \\ \sin\theta & \cos\theta & y_f(1-\cos\theta) - x_f\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Q. Rotate a triangle with vertices $A(3,2)$, $B(7,2)$, $C(5,4)$

1. By $45°$ in ccw dir^n.
2. By $45°$ in ccw dir^n about $P(5,3)$

→ Sol^n:

1. $P' = R_\theta \cdot P$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 7 & 5 \\ 2 & 2 & 4 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 7 & 5 \\ 2 & 2 & 4 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3/\sqrt{2} - 2/\sqrt{2} & 7/\sqrt{2} - 2/\sqrt{2} & 5/\sqrt{2} - 4/\sqrt{2} \\ 3/\sqrt{2} + 2/\sqrt{2} & 7/\sqrt{2} + 2/\sqrt{2} & 5/\sqrt{2} + 4/\sqrt{2} \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & 5/\sqrt{2} & 1/\sqrt{2} \\ 5/\sqrt{2} & 9/\sqrt{2} & 9/\sqrt{2} \\ 1 & 1 & 1 \end{bmatrix}$$

2. $CM = T_{(5,3)} \; R_\theta \; T_{(-5,-3)}$

$$= \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

$$= \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 5 \\ 1/\sqrt{2} & 1/\sqrt{2} & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & -5/\sqrt{2} + 3/\sqrt{2} + 5 \\ 1/\sqrt{2} & 1/\sqrt{2} & -5/\sqrt{2} - 3/\sqrt{2} + 3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 3.5857 \\ 1/\sqrt{2} & 1/\sqrt{2} & -6.192 \\ 0 & 0 & 1 \end{bmatrix}$$

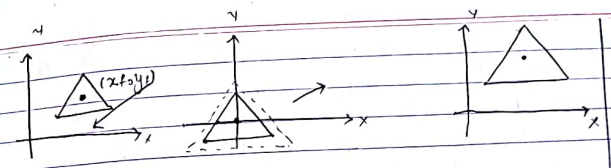
*. Fixed point scaling

Steps:-

1. The fixed point along with the object is translated to coordinate origin.

2. The object is scaled w.r.t. origin.

3. The point is translated back to its original location along with the object.



Composite matrix = $T_{(x_f, y_f)}$ S $T_{(-x_f, -y_f)}$

$$= \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & x_f \\ 0 & S_y & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & -S_x \cdot x_f + x_f \\ 0 & S_y & -S_y \cdot y_f + y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & x_f (1 - S_x) \\ 0 & S_y & y_f (1 - S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

$$= \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 5 \\ 1/\sqrt{2} & 1/\sqrt{2} & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

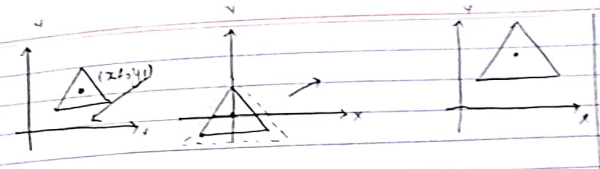
$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & -5/\sqrt{2} + 3/\sqrt{2} + 5 \\ 1/\sqrt{2} & 1/\sqrt{2} & -5/\sqrt{2} - 3/\sqrt{2} + 3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 3.5857 \\ 1/\sqrt{2} & 1/\sqrt{2} & -6.192 \\ 0 & 0 & 1 \end{bmatrix}$$

* Fixed point scaling

Steps:-

1. The fixed point along with the object is translated to coordinate origin.
2. The object is scaled w.r.t. origin.
3. The point is translated back to its original location along with the object.



Composite matrix = $T_{(x_f, y_f)}$ S $T_{(-x_f, -y_f)}$

$$= \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & x_f \\ 0 & S_y & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & -S_x \cdot x_f + x_f \\ 0 & S_y & -S_y \cdot y_f + y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & x_f (1 - S_x) \\ 0 & S_y & y_f (1 - S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

*. A triangle with vertices A (5,4) B (7,1) C (3,1) is required to be

1. scaled by twice the original size
2. scaled by twice the original size about any fixed point P (5,3).

→ Soln:

1. steps:-

1. Translate the triangle to origin from point P(5,3).
2. Scale the triangle by twice.
3. Again translate the triangle to its original position.

Then,

Composite Matrix (CM) = $T_{(5,3)} \; S_{Sx=Sy=2} \; T_{(-5,-3)}$

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

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

$$= \begin{bmatrix} 2 & 0 & -10+5 \\ 0 & 2 & -6+3 \\ 0 & 0 & 1 \end{bmatrix}$$

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

4. Now $P' = C_m \cdot P$

$$= \begin{bmatrix} 2 & 0 & -5 \\ 0 & 2 & -3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 7 & 3 \\ 4 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 10-5 & 14-5 & 6-5 \\ 8-3 & 4-3 & 4-3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 9 & 1 \\ 5 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

1→ Soln

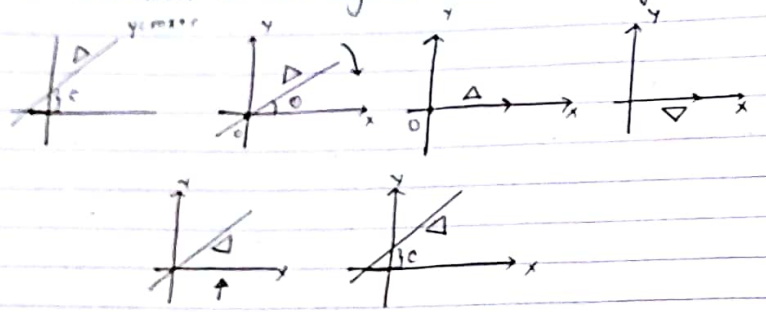
$P' = S_{x,y} \cdot P$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 7 & 3 \\ 4 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 10 & 14 & 6 \\ 8 & 4 & 4 \\ 1 & 1 & 1 \end{bmatrix}.$$

Scanned by CamScanner

* Reflection of an object about line $y = mx + c$



Steps:

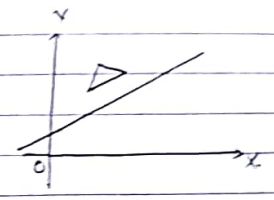
1. Translate the line by $t_x = 0$, $t_y = -c$ as 'c' is the y intercept at point $(0, c)$ so that the line passes thru origin.
2. Rotate the line by angle 'θ' about origin where $\theta = \tan^{-1} m$.
3. Take reflection of the object against x or y depending on the direction of rotation.
4. Again rotate in reverse direction.
5. Retranslate the line by $t_x = 0$ $t_y = c$.

Composite Matrix (CM) = $T'_{(t_x, t_y)} R'_\theta R_f R_\theta T_{(-t_x, -t_y)}$

Q. Reflect a triangle with vertices A(1,3) B(1,6) C(3,3) about the line $y = 2x + 1$.

Solⁿ:

→ Steps:



1. First traslate the line along with object to the origin.
2. Then, rotate align the line in x-axis
3. Reflect the object.
4. Again, rotate the line in it's original position.
5. Translate the line in its original position

we have;

$y = 2x + 1$

Then; $m = 2$

so, $\theta = \tan^{-1}(2)$

$= 63.43$

Now,

Composite matrix = $\underset{(0,1)}{T'} \; \underset{63.43 \text{ CW}}{R'} \; R_f \; \underset{63.43 \text{ CW}}{R} \; T_{(0,-1)}$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 63.43 & -\sin 63.43 & 0 \\ +\sin 63.43 & \cos 63.43 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos 63.43 & +\sin 63.43 & 0 \\ -\sin 63.43 & \cos 63.43 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & -0.8943 & 0 \\ 0.8943 & 0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8943 & 0 \\ -0.8943 & 0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & -0.8943 & 0 \\ 0.8943 & 0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8943 & -0.8943 \\ -0.8943 & 0.4472 & -0.4472 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.4472 & -0.8943 & 0 \\ -0.8943 & -0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8943 & -0.8943 \\ -0.8943 & 0.4472 & -0.4472 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1999 + 0.7997 & 0.3999 - 0.3999 & -0.3999 + 0.3999 \\ -0.3999 + 0.1999 & -0.7999 - 0.1999 & +0.7999 + 0.1999 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.9996 & 0 & 0 \\ -0.2 & -0.9999 & 0.9999 \\ 0 & 0 & 1 \end{bmatrix}$$

Now;

$P' = CM.P.$

$$= \begin{bmatrix} 0.4472 & -0.8943 & 0 \\ 0.8943 & 0.4472 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8943 & 0 \\ -0.8943 & 0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.4472 & 0.8943 & 0 \\ 0.8943 & -0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8943 & 0 \\ -0.8943 & 0.4472 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1999 - 0.7997 & 0.3999 + 0.3999 & 0 \\ 0.3999 + 0.3999 & 0.7999 - 0.1999 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5998 & 0.7998 & 0 \\ 0.7998 & 0.6 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5998 & 0.7998 & -0.7998 \\ 0.7998 & 0.6 & -0.671 \\ 0 & 0 & 1 \end{bmatrix}$$

$P' = CM.P$

$$= \begin{bmatrix} -0.5998 & 0.7998 & -0.7998 \\ 0.7998 & 0.6 & +0.4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 3 \\ 8 & 6 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5998 + 6.3994 - 0.7998 & -0.5998 + 4.7998 - 0.7998 & -1.7994 \end{bmatrix}$$

Scanned by CamScanner

Q. A mirror is placed such that it passes thru points (10,0) (0,10). Find the reflected view of a triangle ABC with coordinates A(5,50) B(20,40) C(10,70).

→ Sol^n:

The mirror passes thru points (10,0) & (0,10)

so;

$$y - 0 = \frac{10-0}{0-10}(x-10)$$

or, $y = -x + 10$ $\therefore$ $\theta = 135^\circ$

Steps:-

1. Translate the line along with the object to origin.
2. Then, rotate the line and align in x-axis.
3. Reflect the object.
4. Again rotate the line in its origin position.
5. Translate the line again in the starting point.

Then,

Composite matrix $(CM) = \underset{(0,10)}{T'}\ \underset{135^\circ\ CCW}{R'}\ R_f\ \underset{135^\circ\ CW}{R}\ T_{(0,-10)}$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 135^\circ & -\sin 135^\circ & 0 \\ \sin 135^\circ & -\cos 135^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 135^\circ & \sin 135^\circ & 0 \\ -\sin 135^\circ & \cos 135^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}-\frac{1}{2} & -\frac{1}{2}-\frac{1}{2} & 0 \\ -\frac{1}{2}-\frac{1}{2} & \frac{1}{2}-\frac{1}{2} & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

Scanned by CamScanner

$\therefore \; P' = C_M \cdot P$

$$= \begin{bmatrix} 0 & -1 & 10 \\ -1 & 0 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 20 & 10 \\ 60 & 40 & 70 \\ 1 & 1 & 1 \end{bmatrix}$$

7. Show that 2D reflection thru x-axis followed by 2D reflection thru line $y=-x$ is equivalent to pure rotation about origin (90° in ccw).

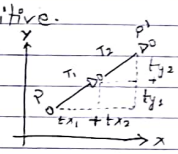
→ $$\left[Rf_{y=-x} \cdot Rf_x \right] = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \\ \sin 90^\circ & \cos 90^\circ \end{bmatrix}$$

1. Composite Transformation :-

With the matrix representation of transformation equations it is possible to set up a matrix for any sequence of transformation as a composite transformation matrix by calculating the matrix product of individual transformation.

For column matrix representation of coordinate positions we form composite transformations by multiplying matrices in order from right to left.

* Two successive translations are additive.



let two successive translations vectors (t_{x_1}, t_{y_1}) and (t_{x_2}, t_{y_2}) be applied to a point P then;

$$P' = T(t_{x_2}, t_{y_2}) \cdot \{ T(t_{x_1}, t_{y_1}) \cdot P \}$$
$$= \{ T(t_{x_2}, t_{y_2}) \cdot T(t_{x_1}, t_{y_1}) \} \cdot P$$

Here, the composite transformation matrix for this sequence of translation is;

$$\begin{bmatrix} 1 & 0 & t_{x_2} \\ 0 & 1 & t_{y_2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_{x_1} \\ 0 & 1 & t_{y_1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{x_1} + t_{x_2} \\ 0 & 1 & t_{y_1} + t_{y_2} \\ 0 & 0 & 1 \end{bmatrix}$$

i.e. $T(t_{x_2}, t_{y_2}) \cdot T(t_{x_1}, t_{y_1}) = T(t_{x_1} + t_{x_2}, t_{y_1} + t_{y_2})$

Scanned by CamScanner

* Two successive scalings are multiplicative

~~Let two successive~~

Let (S_{x_1}, S_{y_1}), (S_{x_2}, S_{y_2}) be two successive vectors applied to point P, then

$$P' = S(S_{x_2}, S_{y_2}) \cdot \{S(S_{x_1}, S_{y_1}) \cdot P\}$$
$$P' = \{S(S_{x_2}, S_{y_2}) \cdot (S_{x_1}, S_{y_1})\} \cdot P$$

Here the composite transformation matrix for this sequence of scaling is:

$$\begin{bmatrix} S_{x_2} & 0 & 0 \\ 0 & S_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{x_1} & 0 & 0 \\ 0 & S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_{x_2} S_{x_1} & 0 & 0 \\ 0 & S_{y_2} S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

i.e. $S(S_{x_2}, S_{y_2}) \cdot S(S_{x_1}, S_{y_1}) = S(S_{x_1} \cdot S_{x_2}, S_{y_1} \cdot S_{y_2})$

* Two successive rotations are additive

Let $R(\theta_1)$ $R(\theta_2)$ be 2 successive rotations applied to a point P then the composite rotation matrix thus produced is:

$$P' = \{R(\theta_2) \cdot R(\theta_1) \cdot P\}$$
$$P' = \{R(\theta_2) \cdot R(\theta_1)\} \cdot P$$

Hence, the composite transformation matrix for this sequence of scalings is

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2 & -\sin\theta_1\cos\theta_2 - \cos\theta_1\sin\theta_2 \\ \sin\theta_2\cos\theta_1 + \cos\theta_2\sin\theta_1 & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) \\ \sin(\theta_2+\theta_1) & \cos(\theta_1+\theta_2) \end{bmatrix} = R(\theta_1+\theta_2)$$

$$R(\theta_2) \cdot R(\theta_1) = R(\theta_1+\theta_2).$$

* Show that 2D scales and rotations don't commute in general. Under what restriction on scaling transformation is commutation of scales and rotations guaranteed?

→ Sol^n:

$$R \cdot S = S \cdot R$$

$$\Rightarrow \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} S_x\cos\theta & -S_y\sin\theta \\ S_x\sin\theta & S_y\cos\theta \end{bmatrix} = \begin{bmatrix} S_x\cos\theta & -S_x\sin\theta \\ S_y\sin\theta & S_y\cos\theta \end{bmatrix}$$

Here, S_x must be equal to S_y for $R \cdot S = S \cdot R$ i.e. for uniform scaling.

Scanned by CamScanner

Q. Scaling followed by rotation is equivalent to shearing

→ $R \cdot S = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$

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

Shearing $= \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix}$

Then, $R \cdot S =$ Shearing

$\begin{bmatrix} s_x\cos\theta & -s_y\sin\theta \\ s_x\sin\theta & s_y\cos\theta \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix}$

Equating both sides;

$s_x\cos\theta = 1$ — (i)

$-s_y\sin\theta = sh_x$ — (ii)

$s_x\sin\theta = sh_y$ — (iii)

$s_y\cos\theta = 1$ — (iv)

$\therefore$ $sh_x = -\tan\theta$

$\therefore$ $sh_y = \tan\theta$

Q. Find the final coordinates of a triangle after reflecting it about line $y = x+1$ and then scaling it by twice it's original shape.

→ Sol^n:

Given :- $y = x+1$

$m = 1$

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

$= 45°$

Then, composite matrix $(CM) = T'_{(0,1)} \; R'_{45° \, ccw} \; R_f \; R_{45° \, cw} \; T_{(0,-1)}$

$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\begin{bmatrix} \cos 45° & \sin 45° & 0 \\ -\sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$

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

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

Scanned by CamScanner

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

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

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

Now:

$CM' = S_{sx=sy=2} \cdot CM$

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

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

$P' = CM' \cdot P$

Q Rotate an object about a fixed point $P(x_f, y_f)$ then translate by 2 units in y direction

Sol^n:

→

$NC = (T_{(x_f, y_f)} R_\theta T_{(-x_f, -y_f)}) P$

$P' = T_{(0,-2)} * NC.$

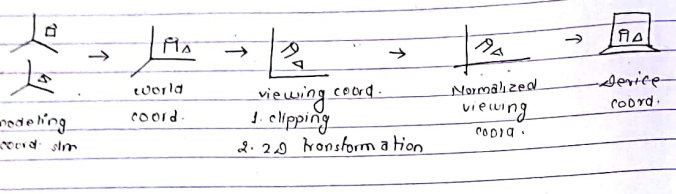
OR:

$P' = T_{(0,-2)} \{ (T' \cdot R_\theta \cdot T) \} \cdot P \}$

$P' = \{ T_{(0,-2)} T' \cdot R_\theta \cdot T \} \cdot P.$

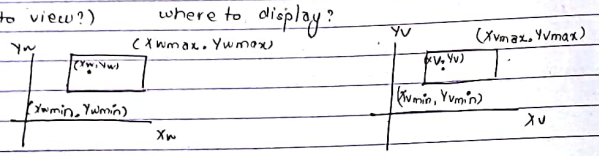
Scanned by CamScanner

2D Viewing pipeline :-



Window to Viewport Transformation :-

(what to view?) → where to display?



A point (x_w, y_w) in window is mapped into position (x_v, y_v) in view port.

To maintain same relative placement in view port as in window we require :-

$$\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} = \frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}}$$

$$\frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} = \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}}$$

$$x_v = x_{vmin} + \left(\frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}\right)(x_w - x_{wmin})$$

$$x_v = x_{vmin} + (x_w - x_{wmin}) \times S_x$$

$$\text{lly, } y_v = y_{vmin} + (y_w - y_{wmin}) \times S_y.$$

where, S_x & S_y are scaling factors.

* **Window :-**
  Rectangular area specified in world coord. s/m.

view port :-
  Rectangular area on the display device to which a window is mapped.

Window defines what is to be viewed & view port defines where it is to be displayed.

OR.....

Step 1 :- The object together with its window is translated until the lower left corner of window is at origin. Here, $T_x = -x_{wmin}$ & $T_y = -y_{wmin}$

Step 2 :- The object and window are scaled until window has the dimension of view port.

$$S_x = (x_{vmax} - x_{vmin}) / (x_{wmax} - x_{wmin})$$

$$S_y = (y_{vmax} - y_{vmin}) / (y_{wmax} - y_{wmin})$$

Scanned by CamScanner

Step 3:- Again translate to move the viewport to its own original position. Here, $T_x = X_{vmin}$ & $T_y = Y_{vmin}$.

Representing the steps using composite transformation
$= T_v \cdot S_{wv} \cdot T_w$

where;
T_w = translate window to origin
S_{wv} = scale window to viewport.
T_v translate view port to (x_{vmin}, y_{vmin})

$$C_M = \begin{bmatrix} 1 & 0 & X_{vmin} \\ 0 & 1 & Y_{vmin} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_{wmin} \\ 0 & 1 & -Y_{wmin} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & X_{vmin} \\ 0 & S_y & Y_{vmin} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_{wmin} \\ 0 & 1 & -Y_{wmin} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & -X_{wmin}S_x + X_{vmin} \\ 0 & S_y & -Y_{wmin}S_y + Y_{vmin} \\ 0 & 0 & 1 \end{bmatrix}$$

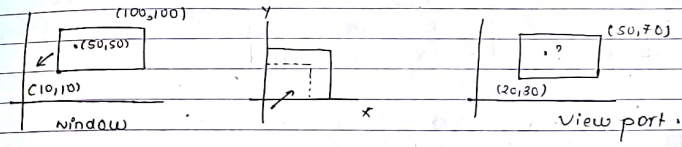
$$P' = C_M \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & -X_{wmin} \cdot S_x + X_{vmin} \\ 0 & S_y & -Y_{wmin} \cdot S_y + Y_{vmin} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x X_w - X_{wmin} S_x + X_{vmin} \\ S_y Y_w - Y_{wmin} S_y + Y_{vmin} \\ 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} S_x (X_w - X_{wmin}) + X_{vmin} \\ S_y (Y_w - Y_{wmin}) + Y_{vmin} \\ 1 \end{bmatrix}$$

Q.



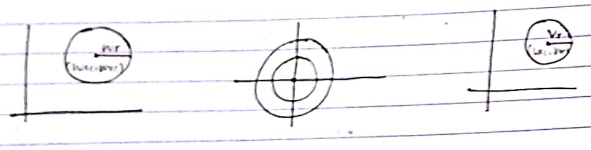
→ Sol^n:

$X_v =$
$Y_v =$

Scanned by CamScanner

Q. Find the window to viewport transformation matrix with a window of circular shape having its center at (W_{xc}, W_{yc}) and a radius as W_r. viewport of circular shape with center as (V_{xc}, V_{yc}) and radius as V_r.

→ Soln:



$$M_{wv} = \begin{bmatrix} 1 & 0 & V_{xc} \\ 0 & 1 & V_{yc} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_r/W_r & 0 & 0 \\ 0 & V_r/W_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -W_{xc} \\ 0 & 1 & -W_{yc} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} V_r/W_r & 0 & V_{xc} \\ 0 & V_r/W_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -W_{xc} \\ 0 & 1 & -W_{yc} \\ 0 & 0 & 1 \end{bmatrix}$$

2D Clipping :-

- used to cut off the lines which are outside the clip window so that the lines within the window are displayed.
- This is done by examining each line segment of the display against the window boundary to determine its visibility.

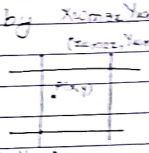
Types:-
(i) Point (ii) Line (iii) polygon

(i) Point clipping :-

Lower most left corner is defined by x_{wmin}, y_{wmin}.

Right most upper corner is defined by x_{wmax}, y_{wmax}

Point clipping against the window boundary requires testing the coordinate points with respect to these boundaries



$x_{wmin} \le x \le x_{wmax}$ $y_{wmin} \le y \le y_{wmax}$

if any of these 4 inequalities is not satisfied then the point is clipped else the point is visible.

Scanned by CamScanner

ii) Line clipping algorithm (Cohen Sutherlands line clipping algorithm)

- Based on coding scheme, makes use of bit operations to perform the tests.
- for each end point of a line a 4 bit binary code is used.

The lower order bit (bit 1) is set to 1 if the end points is at the left side of window else set it to '0'.
Bit 2 is set to 1 if the end point lies at the right side else set to '0'.
Bit 3 is " " " bottom " " "
Bit 4 is " " upper " " "

region codes →

1001	1000	1010
0001	0000	0010
0101	0100	0110

x x x x
T B R L
bit 4 3 2 1

Algorithm :-

Step 1 : Establish region codes for all line end points
Bit 1 is set to '1' if $x < x_{wmin}$ else set to '0'
Bit 2 " " $x > x_{wmax}$ " " "
Bit 3 " " " $y < y_{wmin}$ " " "
Bit 4 " " " $y > y_{wmax}$ " " "

Step 2 : Determine which lines are completely inside the window and which are completely outside using the following tests.

a) if both the end points of a line have region codes "0000" the line is completely inside the window (fully visible)

b) if the logical ~~and~~ AND operation of the region codes of the 2 end points of a line is NOT "0000" then the line is completely invisible.
(the same bit position ~~of~~ in the region codes of 2 end points of a line have 1's then the line is completely outside the clip window)

Step 3 : if Both steps in step 2 fail then the line isnot completely inside nor outside the clip window (case of partial visibility). So, we find out the intersection points of the end point of lines with the window boundaries.
we have $m = (y_2 - y_1) / (x_2 - x_1)$

a. if bit 1 is 1 the line intersects with the left window boundary $y_i = y_1 + m(x - x_1)$ where $x = x_{wmin}$
b. if bit 2 " " " right " $y_i = y_1 + m(x - x_1)$ where $x = x_{wmax}$
c. if bit 3 " " bottom " " $x_i = x_1 + \frac{1}{m}(y - y_1)$
where $y = y_{wmin}$.
d. if bit 4 " " upper " " $x_i = x_1 + \frac{1}{m}(y - y_1)$
" $y = y_{wmax}$.

Step 4 : Repeat step 1 to 3 until line is completely accepted or rejected

Scanned by CamScanner

Q. Clip a line with end points A (5,20) B (50,50) against the clip window having its lowermost left corner at P_1 (10,10) and uppermost right corner at P_2 (100,100).

→ Sol^n:

Step 1:- establish the region codes for 2 end points of the line.

Region code for end point A = 0001

Region code for end point B = 0000

For A;		For B;
5 < 10	bit1 = 1	50 :< 10 = 0
5 :> 100	bit2 = 0	50 :> 100 = 0
20 :< 10	bit3 = 0	50 :< 10 = 0
20 :> 100	bit4 = 0	50 :> 100 = 0

So the line is not completely visible as the region codes of the line has "0001"s

Step 2:- Perform logical AND operation of the region codes of the line.

region code of end point A 0001

" B 0000

AND = 0000

since it yields 0000, it is the case of partial visibility, so we find the intersection points of the line with the window boundaries.

Step 3:- Find the intersection points of the line with window boundaries.

$m = \frac{2}{3}$.

In case of the end point A, it is at the left side of window

so, $x = 10$.

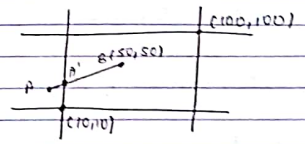
$y_i = y_1 + m(x - x_1)$

or, $y_i = 20 + \frac{2}{3}(10 - 5)$

$y_1 = 23$.

hence the intersection point is (10,23).

So, the line end points after performing clipping operation are A' (10,23) B (50,50)



Scanned by CamScanner

Q. End point coordinates $A(70,20)$ & $B(100,10)$ (x_1, y_1; x_2, y_2)
lower left corner $(50,10)$
upper right corner $(80,40)$

→ Soln:

Step 1: establish the region codes for end points.

For A;		For B;			
$70 \not< 50$	= 0	$100 \not< 50$	= 0	bit 1	(L)
$70 \not> 80$	= 0	$100 > 80$	= 1	bit 2	(R)
$20 \not< 10$	= 0	$10 < 10$	= 0	bit 3	(B)
$20 \not> 40$	= 0	$10 \not> 40$	= 0	bit 4	(T)

Step 2:- AND operation of region code?

$$\begin{array}{rl} A = & 0000 \\ B = & 0010 \\ \hline \text{AND} & 0000 \end{array}$$

So, it is the case of partial visibility.

Step 3:- Now the intersection point is;

$$m = \frac{10-20}{100-70} = \frac{-10}{30} = -\frac{1}{3}$$

In case of B, it is at the right.

So; $x = 80$

$$x_i = x_1 + \frac{1}{m}(y - y_1)$$

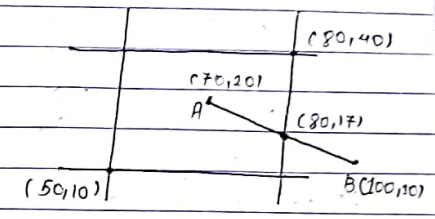
$= 100 + 3(\quad -10$

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

$$= 10 - \frac{1}{3}(80 - 100)$$

$$= 10 + \frac{20}{3}$$

$$= \frac{50}{3} \simeq 17$$



$\therefore B' = (80, 17)$ //

*- 3D Transformation:-

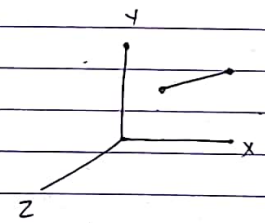
1. Translation:-

$x' = x + tx$

$y' = y + ty$

$z' = z + tz$

$1 = 1$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

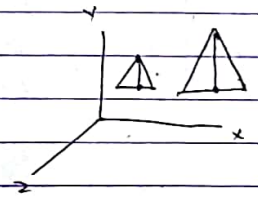
2. Scaling:-

$x' = sx \cdot x$

$y' = sy \cdot y$

$z' = \cancel{sy \cdot z}$ $\quad sz \cdot z$

$1 = 1$



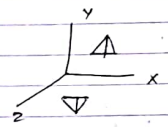
Scanned by CamScanner

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

fixed point scaling:

$C_M = T_{(t_x, t_y, t_z)} \; S \; T_{(-t_x, -t_y, -t_z)}$

3. Rotation :-



rotation about z-axis :-

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

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

rotation about 'x'-axis.

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

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

rotation about 'y'-axis'.

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

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

Scanned by CamScanner

1. Reflection (i. about yz plane)

$x' = -x$
$y' = y$
$z' = z$

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

2. about xy plane

$x' = x$
$y' = y$
$z' = -z$

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

3. about xz plane

$x' = x$
$y' = -y$
$z' = z$

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

*. Reflection of an object about a line parallel to any one of the major coordinate axes

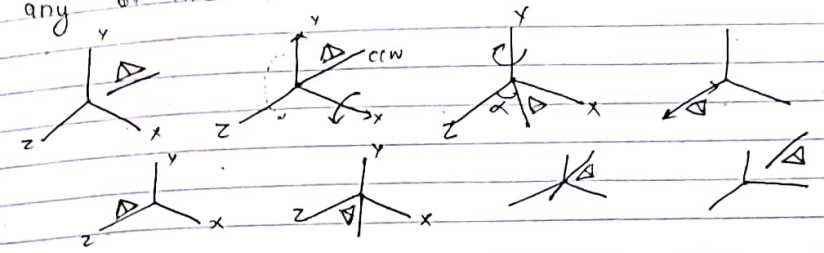
Step 1:- Line together with object is alligned to x-axis.

Step 2:- Reflect the object about xz plane.

Step 3:- Re-translate the ~~line~~ object and the line together to it's original position.

$$CM = T_{(tx, ty, tz)} \; R_{fxz} \; T_{(-tx, -ty, -tz)}$$

*. Reflection of an object about a line NOT parallel to any of the coordinate axes.



$$CM = T_{(tx, ty, tz)} \cdot R_{x'} \, R_{y'} \, R_f \cdot R_y \, R_x \, T_{(-tx, -ty, -tz)}$$

*. Rotation / Reflection of an object about arbitary axis in 3D.

Steps 1: Translate line along with the object to the origin

Here $T = \begin{bmatrix} 1 & 0 & 0 & -tx \\ 0 & 1 & 0 & -ty \\ 0 & 0 & 0 & -tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Steps 2: Rotate about x, y axes to align the arbitary axis to the positive z-axis.

a. Rotate about x-axis by angle 'θ'.

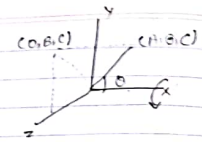
We know the axis (ABC) is translated to origin having its shadow on yz plane. so A = 0 and the shadow is (0, B, C)

Length of axis = $U = \sqrt{A^2 + B^2 + C^2}$ and length

Scanned by CamScanner

d. shadow is $v = \sqrt{B^2+C^2}$

Now rotate the arbitary line about x-axis in anticlockwise direction by 'θ' angle we have $\sin\theta = B/V$ and $\cos\theta = \frac{C}{V}$



Rotation of an object about x-axis in clockwise dirn is given by:

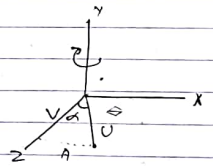
$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & -B/V & 0 \\ 0 & B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b. Now, rotate the arbitary line about y axis in clockwise direction so that it will be aligned with z axis.

Here, $\sin\alpha = A/U$ $\cos\alpha = V/U$

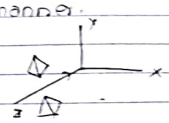
$$R_y = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} V/U & 0 & -A/U & 0 \\ 0 & 1 & 0 & 0 \\ A/U & 0 & V/U & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

~~c. Now rotate~~



c. Now rotate the object about z axis or reflect the object about yz plane in the desired manner.

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



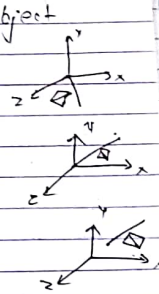
Step 3. Perform inverse rotations about y axis in anticlockwise dirn then rotate about x-axis in clockwise dirn.

Here,

$$y' = \begin{bmatrix} V/U & 0 & A/U & 0 \\ 0 & 1 & 0 & 0 \\ -A/U & 0 & V/U & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & B/V & 0 \\ 0 & -B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 4. Finally translate the line along with object back to its original location.

Here; $$T' = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

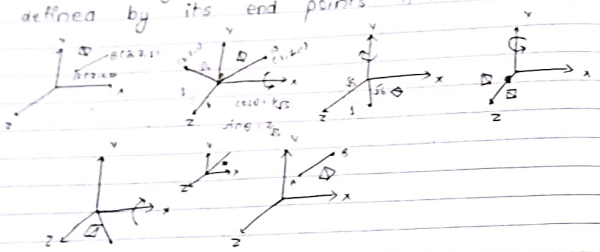


The composite transformation matrix

$(M = T_{(t_x, t_y, t_z)} R_x' R_y' R_z R_y R_x \cdot T_{(-t_x, -t_y, -t_z)})$

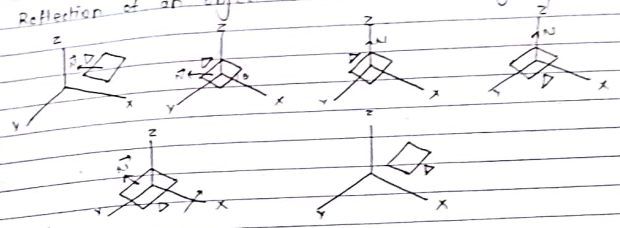
Scanned by CamScanner

* How would you reflect an object about arbitary axis defined by its end points A (2,1,0) B (3,3,1)



$C_M =$ $T_{(2,1,0)}\ R_{x\,CW}\ R_{y\,CW}\ R_z\ R_{y\,CW}\ R_{x\,CW}\ T_{(-2,-1,0)}$

1. Reflection of an object about an arbitary plane



Step 1 : Translate the plane along with object so that it passes through origin.

2 : Allign the normal with one of the major coordinate axes. (for this we need to rotate it with the two other axes).

3 : Take reflection of the object about plane

4 : Rotate the normal to bring the plane to initial position (Dealign the normal)

5 : Translate the plane to original location.

$\therefore\ C_M = T' R_x' R_y' R_z R_y R_x T$

Scanned by CamScanner

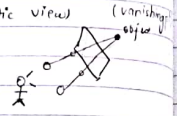
* 3D projection:-

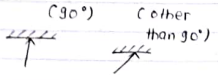
projection

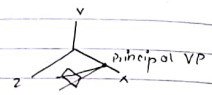
(छायाको तरिका) Parallel — Perspective (Realistic view) (vanishing point)

Parallel: ortho (90°), oblique (other than 90°)

Perspective: 1pt, 2pt, 3pt



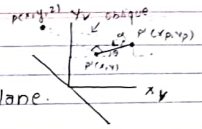




Transforming points in coordinate sim of 'n' dimension into points in a coordinate sim of dimension less than 'n'.

ortho parallel:- z component लाई 0 बनाउने

* oblique parallel projection:- obtained by projecting points along parallel lines that are not ⊥ to projection plane.
Often specified with 2 angles 'θ', 'α'. A point P (x, y, z) is projected to position (xp, yp) on the view plane. Orthographic projection coordinates on the plane are (x, y). oblique projection line from (x, y, z) to (xp, yp) makes an angle 'α' with the line on the projection plane that joins (xp, yp) and (x, y). This line of length 'L' and is at an 'θ' with horizontal direction in projection plane.



Expressing projection coordinates in terms of x, y, L and θ as:
$x_p = x + L\cos\theta$ $\quad y_p = y + L\sin\theta$

L depends upon angle 'α' and z coordinate to be projected $\tan\alpha = \frac{z}{L}$ so, $L = \frac{z}{\tan\alpha} = zL_1$ where, L_1 is inverse of $\tan\alpha$.

So, oblique parallel equations are:-
$x_p = x + z\,(L_1\cos\theta)$
$y_p = y + z\,(L_1\sin\theta)$

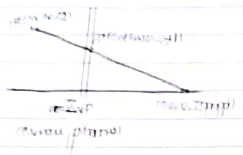
Transformation matrix for producing any parallel projection onto $x_v y_v$ plane is

$$M_{parallel} = \begin{bmatrix} 1 & 0 & L_1\cos\theta & 0 \\ 0 & 1 & L_1\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

orthographic projection is obtained when $L_1 = 0$ (when $\alpha = 90°$)
oblique parallel projection is obtained with non zero values of L_1.

Scanned by CamScanner

* Perspective – (Realistic view)



For $U = 0$ to 1

$x' = x - xU$

$y' = y - yU$

$z' = z - (z - Z_{prp})U$; $U = \dfrac{Z_{vp} - z}{Z_{prp} - z}$

$x' = x - x\dfrac{(Z_{vp} - z)}{(Z_{prp} - z)}$

$x' = \dfrac{x(Z_{prp} - Z_{vp})}{(Z_{prp} - z)}$

$y' = \dfrac{y(Z_{prp} - Z_{vp})}{(Z_{prp} - z)}$

Perspective is a realistic view, where points are transformed along projection line that meet projection reference point.

Suppose perspective reference point is set at position Z_{prp} along Z_v axis and the viewplane is at Z_{vp}. Equations describing coordinates along this perspective projection line in a parametric form is

$x' = x - xU$, $y' = y - yU$ and $z' = z - (z - Z_{prp})U$. Parameter 'U' takes values from 0 to 1 and a coordinate position (x', y', z') represents any point along the projection line. When $U = 0$ we are at $P(x, y, z)$. At the other point of line $U = 1$ and we have projection reference coordinates $(0, 0, Z_{prp})$. On the viewplane $z' = Z_{vp}$ & we can solve the z' equation for parameter U at this position along projection line:

$$U = \frac{Z_{vp} - z'}{Z_{prp} - z}$$

Substituting this value of U in equations for x', y' we obtain perspective transformation equations

$$x_p = x\left(\frac{Z_{prp} - Z_{vp}}{Z_{prp} - z}\right) = x\left(\frac{d_p}{Z_{prp} - z}\right)$$

$$y_p = y\left(\frac{d_p}{Z_{prp} - z}\right)$$

where $d_p = Z_{prp} - Z_{vp}$ is the distance of view plane from projection reference point.

Using 3D Homogeneous coordinate representation, we can write perspective projection in matrix form as:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -Z_{vp}/d_p & Z_{vp}(Z_{prp}/d_p) \\ 0 & 0 & -1/d_p & Z_{prp}/d_p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Here homogeneous factor is

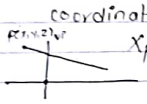
$$h = \frac{Z_{prp} - z}{d_p}$$

Scanned by CamScanner

Projection coordinates on the viewplane are calculated from homogeneous coordinates as $x_p = \frac{x_h}{h}$, $y_p = \frac{y_h}{h}$

Special cases :-

If view plane is taken as $z_{pv} = 0$, projection coordinates are



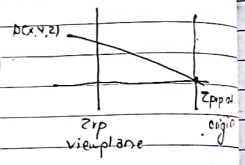
$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) = x\left(\frac{1}{1 - \frac{z}{z_{prp}}}\right) \qquad y_p = y\left(\frac{1}{1 - \frac{z}{z_{prp}}}\right)$$

If projection reference point is taken to be at viewing coordinate origin

Here,

$z_{prp} = 0$ and $x_p = x\left(\frac{z_{vp}}{z}\right) = x\left(\frac{1}{z/z_{vp}}\right)$

$y_p = y\left(\frac{1}{z/z_{vp}}\right)$



Scanned by CamScanner