PAGE:
DATE: / /

CHAPTER: 01
INTRODUCTION AND APPLICATION

1. Resolution = 1280 x 1024
fps = 60

(a) Calculate total pixel accessed per second.
(b) Acess time per pixel

⇒ Soln:

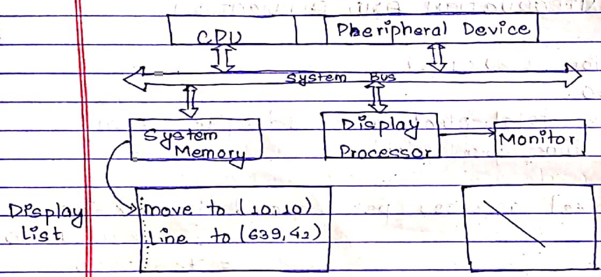
Total pixel accessed per second = Resoln x fps
= 1280 x 1024 x 60
= 78643200

Access time per pixel = $\frac{1}{1280 \times 1024 \times 60}$
= 0.127 ns

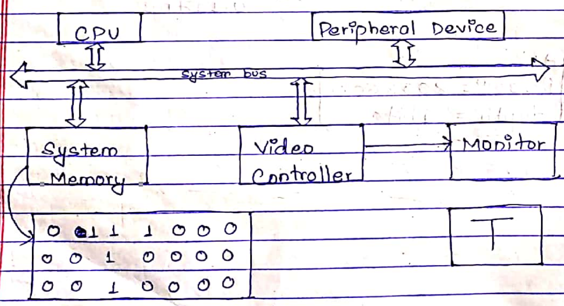
SVGA:

$\boxed{\begin{array}{l} \text{Resol}^n = 800 \times 600 \\ \text{depth} = 4 \text{ bits} \end{array}}$

Random / Vector Scan Display



Raster Scan Display



Vector Scan Display	Raster Scan Display
1. It can draw characters and lines.	1. It can draw areas field with colours and patterns.
2. They are expensive.	2. They are cheaper.
3. The resolution is higher.	3. The resolution is lower.
4. Refresh state depends on picture complexity.	4. Refresh state does not depend on picture complexity.
5. Editing is easy.	5. Editing is ~~com~~ difficult.
6. Scan conversion is not required.	6. Scan conversion is required.
7. Electron beams is moved betⁿ end points of graphic primitives.	7. Electron bem is moved all across the screen from top to bottom.

PAGE :
DATE : / /

CHAPTER : 02
SCAN CONVERSION

1. ~~All rising lines have~~

DDA (Digital Dibberential Analyzer Algorithm)
Assume, we follow L → R convention

Case I : +ve slope

(a) $|m| < 1$

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

$\Delta x > \Delta y$

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

$\Delta x = 1$

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

$\therefore m = \Delta y$

$y_{k+1} = y_k + \Delta y$

$y_{k+1} = y_k + m$

$$\boxed{\begin{array}{l} y_{k+1} \longrightarrow y_k + m \\ x_{k+1} \longrightarrow x_k + 1 \end{array}}$$

(b) $|m| > 1$

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

$\Delta y > \Delta x$

or, $y_{k+1} = y_k + 1$

$\therefore \Delta y = 1$

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

$\therefore \frac{1}{m} = \Delta x$

$x_{k+1} = x_k + \Delta x$

$x_{k+1} = x_k + 1/m$

$$\boxed{\begin{array}{l} y_{k+1} \Rightarrow y_k + 1 \\ x_{k+1} \longrightarrow x_k + 1/m \end{array}}$$

PAGE :
DATE : / /

Case II : -ve slope

(a) $|m| < 1$, $+m = \frac{\Delta y}{\Delta x}$

$|\Delta x| > |\Delta y|$

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

$y_{k+1} = y_k + \Delta y$

$\Delta y = m \Delta x$

$\Delta y = m$

$y_{k+1} = y_k + m$

$$\boxed{\begin{array}{l} y_{k+1} \longrightarrow y_k + m \\ x_{k+1} \longrightarrow x_k + 1 \end{array}}$$

(b) $|m| > 1$

$|\Delta x| < |\Delta y|$

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

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

$$\boxed{\begin{array}{l} y_{k+1} \longrightarrow y_k - 1 \\ x_{k+1} \longrightarrow x_k - 1/m \end{array}}$$

Digitalize the line of given points A(1,5) and B(6,12) by using DDA diagram

→ Here,

Slope$(m) = \frac{12-5}{6-1} = 1.4$

$\Delta x = x_2 - x_1 = 6 - 1 = 5$

$\Delta y = y_2 - y_1 = 12 - 5 = 7$

$\therefore$ Steps $(dy) = 7$

Calculated Points	K	(x_k, y_k)	$x_{k+1} = x_k + 1/m$	(x_{k+1}, y_{k+1})
(1, 5)	0			
(1.7, 5)	1	(1, 5)	1+0.714	(

PAGE :
DATE : / /

Digitize the line of given points A(7,15) and B(3,11) by using DDA diagram.

⇒ Soln:

$$\text{slope}(m) = \frac{11-15}{3-7} = \frac{-4}{-4} = \frac{4}{5}$$

$$\text{Slope}(m) = \frac{15-11}{7-3} = \frac{4}{4} = 1$$

$\Delta x = x_2 - x_1 = 15 - 11 =$

$\Delta x = x_2 - x_1 = 15 - 11 = 4$

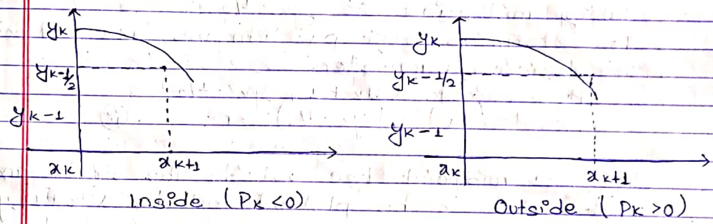
$\Delta y = y_2 - y_1 = 7 - 3 = 4$

Slope (m) =

PAGE :
DATE : / /

Mid point Circle ~~Theor~~ Algorithm

8 point Symmetry



Using concept of BSA and 8 point symmetry, we calculate points that ~~w~~use to be drawn in circle.

Eqn of circle,

$(x - x_c)^2 + (y - y_c)^2 = r^2$

For centre at origin,

$x^2 + y^2 = r^2$ —— (i)

If eqn (i) $= 0$, on the boundary of circle

If eqn (i) > 0, outside of circle

If eqn (i) < 0, inside of circle)

By eight point symmetry we know that we only have to calculate first octant of the circle i.e $(0, r)$ upto $x < y$ (45°)

For first octant $|m| < 1$

So, $|\Delta x| > |\Delta y|$ i.e, sampling occurs at x-axis also +ve sampling.

PAGE :
DATE : / /

i.e. $x_{k+1} = x_k + 1$
& $y_{k+1} = y_k$ or $y_k - 1$

Taking mid point i.e $y_k - 1/2$
To determine next position we take mid point as decision parameter.

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

$P_k = (x_k + 1)^2 + (y_k - 1/2)^2 - r^2$ ——— (ii)

If $P_k > 0$ midpoint lies outside the circle and next point to be plotted will be $(x_k + 1, y_k - 1)$

If $P_k < 0$ midpoint lies ~~outside~~ inside the circle and next point to be plotted will be $(x_k + 1, y_k)$

$P_{k+1} = f_{circle}(x_{k+1} + 1, y_{k+1} - 1/2)$

$= [(x_k + 1) + 1]^2 + (y_{k+1} - 1/2)^2 - r^2$

$P_{k+1} - P_k = \{[(x_k + 1) + 1]^2 + (y_{k+1} - 1/2)^2 - r^2\} - [(x_k + 1)^2 + (y_k - 1/2)^2 - r^2]$

or, $P_{k+1} - P_k = \cancel{(x_k + 1)^2} + 2(x_k + 1) + 1 + (y_{k+1} - 1/2)^2 - \cancel{r^2} - \cancel{(x_k + 1)^2} - (y_k - 1/2)^2 + \cancel{r^2}$

or, $P_{k+1} - P_k = 2x_k + 2 + 1 + (y_{k+1})^2 - y_{k+1} + \cancel{1/4} - y_k^2 + y_k - \cancel{1/4}$

$\therefore P_{k+1} - P_k = 2x_k + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 3$ ——— (iii)

Case I: For $P_k < 0$; $y_{k+1} = y_k$

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

or, $P_{k+1} - P_k = 2x_k + 3$

$\therefore P_{k+1} = P_k + 2x_k + 3$

PAGE :
DATE : / /

Case II: For $P_k > 0$; $y_{k+1} = y_k - 1$

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

or, $P_{k+1} - P_k = 2x_k + \cancel{y_k^2} - 2y_k + 1 - \cancel{y_k^2} - \cancel{y_k} + 1 + \cancel{y_k} + 3$

$\therefore P_{k+1} = P_k + 2x_k - 2y_k + 5$

For initial decision parameter, we calculate circle function at $(x_0, y_0) = (0, r)$

$P_0 = f_{circle}(0 + 1, r - 1/2)$

$= f_{circle}(1, r - 1/2)$

$= 1^2 + (r - 1/2)^2 - r^2$

$= 1 + \cancel{r^2} - 2 \cdot r \cdot 1/2 + 1/4 - \cancel{r^2}$

$= 1 - r + 1/4$

$= 5/4 - r \approx \boxed{1 - r}$

Q. Digitize first octant of a circle with $r = 10$ and center at origin.

⇒ Soln:

Initial points $(x_0, y_0) = (0, r) = (0, 10)$

$P_0 = 1 - r = 1 - 10 = -9$

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

$P_k \geq 0$; $y_{k+1} = y_k - 1$

$P_{k+1} = P_k + 2x_k - 2y_k + 5$

For $P_k < 0$; $y_{k+1} = y_k$

$P_{k+1} = P_k + 2x_k + 3$

Scanned with CamScanner

K	(x_k, y_k)	P_k	(x_{k+1}, y_{k+1})	P_{k+1}
0	(0,10)	-9	(1,10)	-6
1	(1,10)	-6	(2,10)	-1
2	(2,10)	-1	(3,10)	6
3	(3,10)	6	(4,9)	-3
4	(4,9)	-3	$\overset{5,9}{(\cancel{6,8})}$	8
5	(5,9)	8	(6,8)	5
6	(6,8)	5	(7,7)	

$P_1 = P_0 + 2x_0 + 3$
$= -9 + 0 + 3 = -6$

$P_2 = P_1 + 2x_1 + 3$
$= -6 + 2 + 3 = -1$

$P_3 = P_2 + 2x_2 + 3$
$= -1 + 2\times2 + 3 = 6$

$P_4 = P_3 + 2x_3 + 5 - 2y_3$
$= 6 + 6 + 5 - 20$
$= -3$

$P_5 = P_4 + 2x_4 + 3$
$= -3 + 8 + 3$
$= 8$

$P_6 = P_5 + 2x_5 \overset{+5}{} - 2y_5$
$= 8 + 10 + 5 - 18$
$= 5$

Q. Digitize complete circle with eqn $(x-2)^2 + (y-3)^2 = 25$

⇒ Soln:

Initial points $(x_0, y_0) = (0, r) = (0, 5)$

$P_0 = 1 - r = 1 - 5 = -4$

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

$P \geq 0$; $y_{k+1} = y_k - 1$

$P_{k+1} = P_k + 2x_k + 5 - 2y_k$

$P < 0$; $y_{k+1} = y_k$

$P_{k+1} = P_k + 2x_k + 3$

$P_1 = P_0 + 2x_0 + 3$
$= -4 + 2\times0 + 3$
$= -1$

$P_2 = P_1 + 2x_1 + 3$
$= -1 + 2\times1 + 3$
$= -1 + 2 + 3$
$= 4$

$P_3 = P_2 + 2x_2 + 5 - 2y_2$
$= 4 + 2\times2 + 5 - 2\times5$
$= 4 + 4 + 5 - 10$
$= 3$

K	(x_k, y_k)	P_k	(x_{k+1}, y_{k+1})	P_{k+1}
0	(0, 5)	-4	(1, 5)	-1
1	(1, 5)	-1	(2, 5)	4
2	(2, 5)	4	(3, 4)	3
3	(3, 4)	3	(4, 3)	

Points are (1,5) (2,5) (3,4) (4,3)

Applying 8-point symmetry in (1,5)

$(x, y) = (1, 5)$
$(x, -y) = (1, -5)$
$(-x, y) = (-1, 5)$
$(-x, -y) = (-1, -5)$
$(y, x) = (5, 1)$
$(-y, x) = (-5, 1)$
$(-y, -x) = (-5, -1)$
$(y, -x) = (5, -1)$

Now,

Translate with centre (2, 3)

$(x, y) = (3, 8)$
$(x, -y) = (3, -2)$
$(-x, y) = (1, 8)$
$(-x, -y) = (1, -2)$
$(y, x) = (7, 4)$
$(-y, x) = (-3, 4)$
$(-y, -x) = (-3, 2)$
$(y, -x) = (7, 2)$

Midpoint Ellipse Algorithm

Eqn of ellipse

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

For center at origin,

$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} = 1$$

$$\text{or, } x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2 = 0 \quad \text{—— (i)}$$

$$f(\text{ellipse}) = x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2 = 0 \quad \text{—— (ii)}$$

For region-1, slope $|m| < 1$

So, sampling is done in x-axis i.e. $x_{k+1} = x_k + 1$ and based on decision parameter next y increment is y_k or $y_k - 1$.

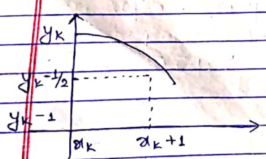
For region-2, slope $|m| > 1$

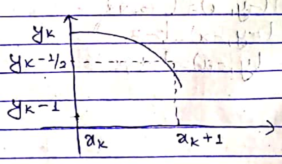
So, sampling is done in y-axis i.e, $y_{k+1} = y_k - 1$ and $x_{k+1} = x_k$ or x_k+1 based on decision parameter

For region R_1

Using midpoint $y_k - 1/2$

$$P_{1k} = f_{ellipse}(x_k+1,\ y_k - 1/2)$$





$P_{1k} < 0$ (point lies inside)

$y_{k+1} = y_k$

$P_{1k} > 0$ (outside)

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

$$P_{1k} = (x_k+1)^2 r_y^2 + (y_k - 1/2)^2 r_x^2 - r_y^2 r_x^2 \quad \text{—— (iii)}$$

For incremental recursive decision parameter.

$$P_{1k+1} = f_{ellipse}(x_{k+1}+1,\ y_{k+1} - 1/2)$$

$$= f_{ellipse}(x_k + 1 + 1,\ y_{k+1} - 1/2)$$

$$P_{1k+1} = (x_k+1+1)^2 r_y^2 + (y_{k+1} - 1/2) r_x^2 - r_y^2 r_x^2 \quad \text{—— (iv)}$$

$$P_{1k+1} - P_{1k} = r_y^2 (x_k+1+1)^2 + (y_{k+1} - 1/2)^2 r_x^2 - \cancel{r_x^2 r_y^2} - [r_y^2 (x_k+1)^2 + (y_k - 1/2) r_x^2 - \cancel{r_x^2 r_y^2}]$$

$$= r_y^2 [\cancel{(x_k+1)^2} + 2(x_k+1) + 1 - \cancel{(x_k+1)^2}] + r_x^2 [(y_{k+1} - 1/2)^2 - (y_k - 1/2)^2] \quad \text{—— (v)}$$

For $P_{1k} < 0$; $y_{k+1} = y_k$

$$P_{1k+1} - P_{1k} = 2r_y^2 (x_k+1) + r_y^2$$

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

For $P_{1k} > 0$; $y_{k+1} = y_k - 1$

$$P_{1k+1} - P_{1k} = 2r_y^2 (x_k+1) + r_y^2 + r_x^2 [(y_k - 1 - 1/2)^2 - (y_k - 1/2)^2]$$

$$= 2r_y^2 (x_k+1) + r_y^2 + r_x^2 [\cancel{(y_k - 1/2)^2} - 2(y_k - 1/2) + 1 - \cancel{(y_k - 1/2)^2}]$$

$$= 2r_y^2 (x_k+1) + r_y^2 + r_x^2 (-2y_k + 2)$$

$$= 2r_y^2 (x_k+1) + r_y^2 - 2r_x^2 (y_k - 1)$$

$$P_{1k+1} = P_{1k} + 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1} \quad \text{—— (vi)}$$

For initial decision parameter,
P_{10} ; initial points are $(0, r_y)$

$P_{10}(1, r_y - 1/2)$

$P_{10} = r_y^2 + r_x^2(r_y - 1/2)^2 - r_y^2 r_x^2$

$= r_y^2 + r_x^2[r_y^2 - 2r_y \cdot 1/2 + 1/4] - r_y^2 r_x^2$

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

$= r_y^2 - r_y r_x^2 + \frac{1}{4} r_x^2$ —— (vii)

Ending condition for region R_1.

R_1 lasts till $m = -1$

i.e. $\frac{dy}{dx} = -1$

Diff eqn (i) w.r.t x

$2x r_y^2 + r_x^2 2y \frac{dy}{dx} = 0$

or, $\frac{dy}{dx} = -\frac{2x r_y^2}{2y r_x^2}$

or, $\frac{dy}{dx} = -1 = -\frac{2x r_y^2}{2y r_x^2}$

or, $2r_y^2 x = 2r_x^2 y$

i.e. Region lasts till

$2r_x^2 y \geq 2r_y^2 x$

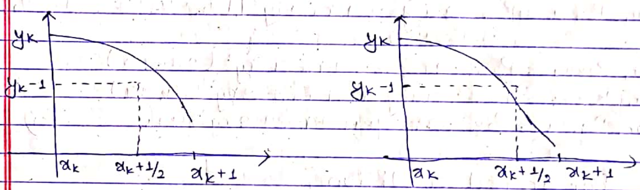
Consider region R_1 ends at (x_0, y_0)

$|m| > 1$ so $|\Delta x| < |\Delta y|$

$\therefore y_{k+1} = y_k - 1$ i.e. sampling occurs in y-axis.

Now,

x_{k+1} is either x_k or $x_k + 1$



For $P_{2k} < 0$ (inside)
$x_{k+1} = x_k + 1$

For $P_{2k} \geq 0$ (outside)
$x_{k+1} = x_k$

$P_{2k} = f_{ellipse}(x_k + 1/2, y_k - 1)$

$P_{2k} = r_y^2(x_k + 1/2)^2 + r_x^2(y_k - 1)^2 - r_y^2 r_x^2$ —— (viii)

So,

$P_{2k+1} = r_y^2(x_{k+1} + 1/2)^2 + r_x^2(y_{k+1} - 1)^2 - r_y^2 r_x^2$

$= r_y^2(x_{k+1} + 1/2)^2 + r_x^2(y_k - 1 - 1)^2 - r_y^2 r_x^2$

$P_{2k+1} - P_{2k} = r_y^2[(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] + r_x^2[(y_k - 1 - 1)^2 - (y_k - 1)^2] - \cancel{r_x^2 r_y^2} + \cancel{r_x^2 r_y^2}$

or, $P_{2k+1} - P_{2k} = r_y^2[(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] + r_x^2[\cancel{(y_k - 1)^2} - 2(y_k - 1) + 1 - \cancel{(y_k - 1)^2}]$

or, $P_{2k+1} - P_{2k} = r_y^2[(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] + r_x^2[-2(y_k - 1) + 1]$

or, $P_{2k+1} = P_{2k} + r_y^2[(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] - 2r_x^2(y_k - 1) + r_x^2$

i.e. $P_{2k+1} = P_{2k} + r_y^2[(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] - 2r_x^2 y_{k+1} + r_x^2$ —— (ix)

PAGE :
DATE : / /

For $P_{2k} > 0$ $\quad x_{k+1} = x_k$

$\therefore P_{2k+1} = P_{2k} + r_x^2 - 2r_x^2 y_{k+1}$ —— (x)

For $P_{2k} < 0$ $\quad x_{k+1} = x_k + 1$

$P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2 + r_y^2 \left[(x_k + 1/2 + 1)^2 - (x_k + 1/2)^2 \right]$

or, $P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2 + r_y^2 \left[\cancel{(x_k + 1/2)^2} + 2(x_k + 1/2).1 + 1 - \cancel{(x_k + 1/2)^2} \right]$

or, $P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2 + r_y^2 (2x_k + 2)$

or, $P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2 + 2r_y^2 (x_k + 1)$

or, $P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2 + 2r_y^2 x_{k+1}$ —— (xi)

For initial condition,
Let us assume region-2 starts at (x_0, y_0)

$\therefore P_{20} = f_{ellipse}(x_0 + 1/2, y_0 - 1)$

ie, $P_{20} = r_y^2 (x_0 + 1/2)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$ —— (xii)

Q. Digitize an ellipse with eqn

$$\left(\frac{x+3}{10}\right)^2 + \left(\frac{y+4}{8}\right)^2 = 0$$

Comparing with standard eqn

$$\left(\frac{x-x_c}{r_x}\right)^2 + \left(\frac{y-y_c}{r_y}\right)^2 = 0$$

⇒ Soln:

$r_x = 10$, $r_y = 8$

$x_c = -3$ $\quad y_c = -4$

For region-1,
Initial points $(x_0, y_0) = (0, r_y) = (0, 8)$

$P_0 = r_y^2 - r_y^2 r_x^2 + \frac{1}{4} r_x^2$

PAGE :
DATE : / /

$P_0 = (8)^2 - (8)^2 \times (10)^2 + \frac{1}{4} \times 10^2$

$= 64 - 800 + 25$

$= -711$

k	P_{1k}	x_{k+1}, y_{k+1}	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-711	(1, 8)	128	1600
1	-519	(2, 8)	256	1600
2	-199	(3, 8)	384	1600
3	-249	(4, 7)	512	1400
4	-575	(5, 7)	640	1400
5	129	(6, 6)	768	1200
6	-239	(7, 6)	896	1200
7	721	(8, 5)	1024	1000

For region-2,
starting points are $(8, 5) = (x_0, y_0)$

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

$P_{20} = r_y^2 (x_0 + 1/2)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$

$= (8)^2 \times (8 + 1/2)^2 + (10)^2 \times (5-1)^2 - (10)^2 \times (8)^2$

$= -176$

k	P_k	x_{k+1}, y_{k+1}	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-176	(9, 4)	1152	800
1	276	(9, 3)	1152	600
2	~~328~~ -224	(10, 2)	1280	400
3	7956	(10, 1)	1280	200
4	~~1936~~ 656	(10, 0)	1280	0

PAGE :
DATE : / /

Now, using four point symmetry we draw ellipse at center ~~orgin~~ origin.

$(-3,-4) = (x_c, y_c)$

Choosing any point $(1,8)$

$(x,y) = (1,8)$

$(x,-y) = (1,-8)$

$(-x,-y) = (-1,-8)$

$(-x,y) = (-1,8)$

Now shifting as per center $(-3,-4)$

$(-3+1,\ 8-4) = (-2,4)$

$(-3+1,\ -4-8) = (-2,-12)$

$(-1-3,\ -8-4) = (-4,-12)$

$(-1-3,\ 8-4) = (-4,4)$

PAGE :
DATE : / /

CHAPTER: 03

2-D TRANSFORMATIONS

There are five types of 2D transformations grouped into two categories.

1. Rigid Body Transformation
   (i) Translation
   (ii) Reflection
   (iii) Rotation

2. Non-Rigid Body Transformation
   (i) Scaling
   (ii) Shearing

Translation

$x' = x + t_x$

$y' = y + t_y$

where, x', y' are transformed points.
& x, y are original points.

In matrix form,

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

In homogeneous form,

$$\underbrace{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}}_{P'} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{T} \cdot \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{P} = \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix}$$

Scanned with CamScanner

Scaling

$x' = S_x \cdot x$

$y' = S_y \cdot y$

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

where S_x and S_y are scaling factor.

In homogeneous form

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

Note* For any transformation

$P' = T \cdot P$

where P is actual points in the form

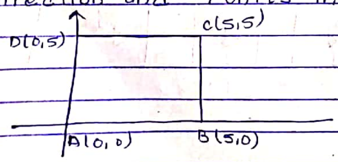
$$P = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

T is transformation matrix in homogeneous form.

and P' is transformed matrix in form: P

$$P' = \begin{bmatrix} x_1' & x_2' & \cdots & x_n' \\ y_1' & y_2' & \cdots & y_n' \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Translate the following square 3 units in x-direction and 4 units in negative y-direction.



$t_x = 3$

$t_y = -4$

$P' = T \cdot P$

$$P' = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 5 & 5 & 0 \\ 0 & 0 & 5 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\therefore P' = \begin{bmatrix} 3 & 8 & 8 & 3 \\ -4 & -4 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Rotation about origin

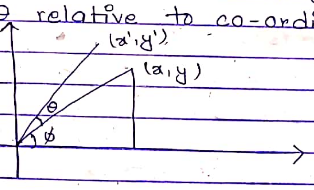
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

anticlockwise direction θ is +ve

clockwise direction θ is −ve

Derivation

Rotation of a point from (x,y) to (x',y') through an angle θ relative to co-ordinate origin.



Original angular displacement of point from x-axis is φ.

$x = r\cos\phi$

$y = r\sin\phi$

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

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

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

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

PAGE :
DATE : / /

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

Shearing

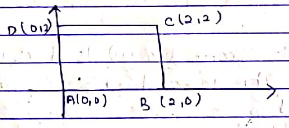
x-axis shear

$x' = x + y * sh_x$
$y' = y$ $\longrightarrow \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

y-axis shear

$x' = x$
$y' = y + x * sh_y$ $\longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Q.



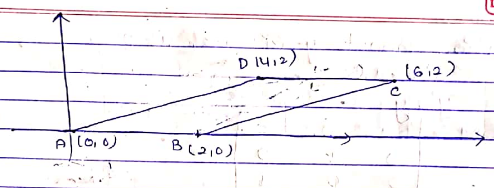
For shearing about x-axis with shearing factor 2 units i.e, $sh_x = 2$.

~~$x' = x + y * sh_x$~~ $P' = T \cdot P$

or, $P' = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

i.e, $P' = \begin{bmatrix} 0 & 2 & 6 & 4 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

PAGE :
DATE : / /

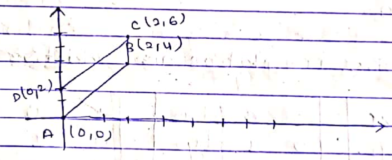


For shearing about y-axis with shearing factor 2 units i.e, $sh_y = 2$

$P' = T \cdot P$

or, $P' = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

$\therefore P' = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 4 & 6 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$



Reflection

① Reflection about x-axis i.e $y = 0$

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

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

Scanned with CamScanner

PAGE :
DATE : / /

ction about y-axis i.e, $x = 0$

$x' = -x$

$y' = y$

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

nposite Transformation

ose transformation that involve two or more sic transformations are known as composite ansformation.

g: Rotation about a fixed point, Reflection bout an arbitrary line.

Properties of some homogenous composite transformation.

Two successive translatios are additive.

$$\begin{bmatrix} 1 & 0 & tx_1 \\ 0 & 1 & ty_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & tx_2 \\ 0 & 1 & ty_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx_1+tx_2 \\ 0 & 1 & ty_1+ty_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Two succesive rotation are additive.

$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1+\cos\theta_2 & -(\sin\theta_1+\sin\theta_2) & 0 \\ \sin\theta_1+\sin\theta_2 & \cos\theta_1+\cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

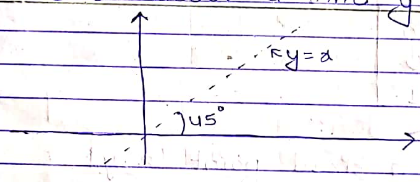
PAGE :
DATE : / /

3. Two successive scaling are multiplicative

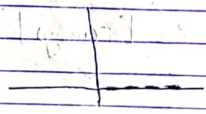
$$\begin{bmatrix} sx_1 & 0 & 0 \\ 0 & sy_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} sx_2 & 0 & 0 \\ 0 & sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} sx_1 \cdot sx_2 & 0 & 0 \\ 0 & sy_1 \cdot sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note* For convention $P' = T \cdot P$ composite transformation is written in reverse order.

Reflection about a line $y = x$.

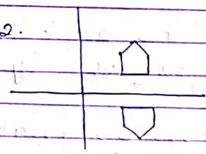


1.



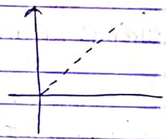
Clockwise rotation about 45°

2.



Reflection about x-axis

3.

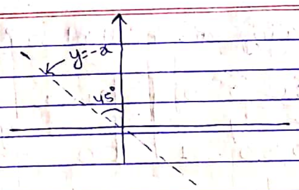


Anticlockwise rotation by 45°

$T = \cancel{R(\theta)} \cdot \cancel{Rx \cdot R(-\theta)}$ $R^{-1}(\theta)\, Rx\, R(\theta)$

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

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



$T = \cancel{R(\theta) \cdot R_y \cdot R(+\theta)}\ R^{-1}(\theta)\, R_y \cdot R_\theta$

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

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

Rotation about a fixed point (x, y)

Step 1: Translate the point to origin i.e. $[T(-x, -y)]$
$t_x = -x \quad t_y = -y$

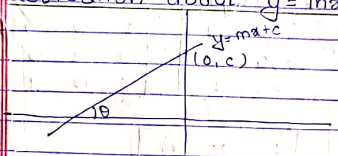
Step 2: Perform required rotation. R_θ

Step 3: Reverse translate to original position.
$[T'(x, y)]$
i.e. $t_x = x \quad t_y = y$

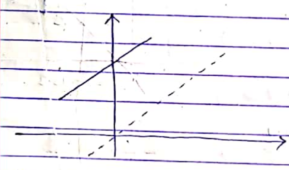
$T = T^{-1}(x, y)\ R_\theta \cdot T(x, y)$

→ Similar in scaling about a fixed point only on step 2 perform required scaling.

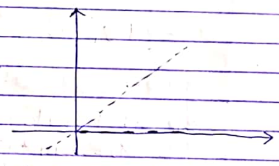
Reflection about $y = mx + c$



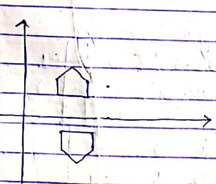
(i) First translate line so that it passes through origin.
(ii) Rotate line onto one of co-ordinate axis (usually x-axis)
(iii) Reflect object
(iv) Finally restore line to original position with inverse rotation and translation transformation.



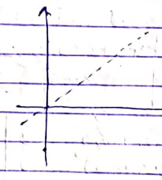
Step 1: translate line $t_x = 0$
$t_y = -c$



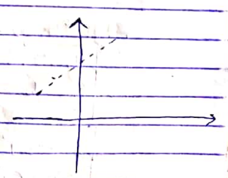
Step 2: Rotate to origin
$\theta = \tan^{-1}(m)$



Step 3: Perform ~~rotation~~ reflection about x-axis



Step 4: Inverse rotation $R(-\theta)$



Step 5: Inverse translation
$t_x = 0 \quad t_y = c$

Scanned with CamScanner

PAGE:
DATE: / /

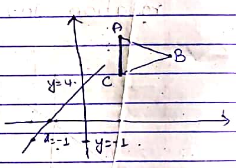
$T = T^{-1}(x,y) \cdot R^{-1}(\theta) \cdot R_x \cdot R_\theta \cdot T(x,y)$

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

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

Q. Reflect a triangle ABC with vertex A(2,10) B(4,9) and C(2,7) about a line $y = 5x + 4$.

Step 1: translate $t_x = 0$
$t_y = -c$.
(0, −4)



Step 2: Rotate to origin
$\theta = \tan^{-1}(m)$
or, $\theta = \tan^{-1}(5)$
$\therefore \theta = 78.69°$

$T = T^{-1}(x,y) \cdot R^{-1}(\theta) \cdot R_x \cdot R_\theta \cdot T(x,y)$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 78.69° & -\sin 78.69° & 0 \\ \sin 78.69° & \cos 78.69° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos 78.69° & \sin 78.69° & 0 \\ -\sin 78.69° & \cos 78.69° & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{bmatrix}$$

PAGE:
DATE: / /

$$\begin{bmatrix} -0.9923 & 0.38846 & -1.538 \\ 0.38846 & -0.9923 & 0.308 \\ 0 & 0 & 1 \end{bmatrix}$$

$P' = T \cdot P$

$$= \begin{bmatrix} -0.923 & 0.3840 & -1.538 \\ 0.3846 & 0.923 & 0.308 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 4 & 2 \\ 10 & 9 & 7 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.456 & -1.774 & -0.696 \\ 10.307 & 10.153 & 7.538 \\ 1 & 1 & 1 \end{bmatrix}$$

Q. Reflect a quadilateral A(3,1) B(2,4) C(6,3), D(7,5) about a line $3x - 9y = 12$.
or, $3x - 12 = 9y$
or, $y = \frac{1}{3}x - \frac{4}{3}$

$\theta = \tan^{-1}(m)$
or, $\theta = \tan^{-1}(1/3)$
$\therefore \theta = 18.43°$

$T = T^{-1}(x,y) \cdot R^{-1}(\theta) \cdot R_x \cdot R_\theta \cdot T(x,y)$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -4/3 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos 18.43 & -\sin 18.43 & 0 \\ \sin 18.43 & \cos 18.43 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos 18.43 & \sin 18.43 & 0 \\ -\sin 18.43 & \cos 18.43 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & +4/3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.8 & 0.6 & 0.8 \\ 0.6 & -0.8 & -2.4 \\ 0 & 0 & 1 \end{bmatrix}$$

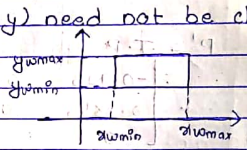
2-D Clipping

Point Clipping

- ~~Assume the~~ Any point (x, y) need not be clipped if it satisfies

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

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



Line Clipping

2 types

(i) Cohen Sutherland

(ii) Liang Barsky

1001	1000	1010
0001	0000	0010
0101	0100	0110

Region Code

Algorithm for Cohen Sutherland

1. Assign region code for each end point.
2. If both end points have a region 0000 → trivally accept these lines.
3. Else perform logical AND operations for both region codes.
   (i) If result is not 0000 → trivally reject lines.
   (ii) Else (result = 0000, need clipping)
   - Choose any non-zero end point of the line that is outside the window.
   - Find intersection point at window boundary (based on a region code).
   - Replace end point with intersection and uptade region code.
   - Repeat step 2 and step 3 until we find a clipped line either trivially accepted or rejected.
4. Repeat steps 1 to 3 for other lines.

Q. Line P_1P_2 with $P_1(0, 120)$ and $P_2(130, 5)$

=> Soln:

Step 1:

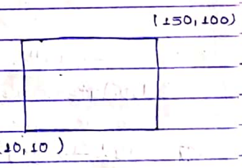
$P_1 = 1001$

$P_2 = 0100$

1001	1000	1010
0001	0000	0010
0101	0100	0110

Step 2:

Both 0000 → NO



Step 3:

AND operation

(a) $P_1 \rightarrow 1001$

$P_2 \rightarrow 0100$

$0000 \rightarrow$ need clipping.

Step 3.2.1

A. Choose $P_1(1001)$

Intersection with left boundary

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5 - 120}{130 - 0} = -0.8846$

$y = y_1 + m(x - x_1)$ where $x = 10$

$y = 111.15 \simeq 111$

Now,

$P_1' = (10, 111)$

$P_1' = 1000$ (top) [boundary → 0]

[update region code]

Now,

(b) $P_1' = 1000$

$P_2 = \underline{0100}$

0000 (AND)

Choose

$P_1' (1000)$ → top

i.e. intersection with top boundary

$m = -0.8846$

$x = x_1 + \dfrac{y - y_1}{m}$ where $y = 100$;

$= 22.44 \simeq 22$

$\therefore P_1'' = (22, 100)$

Updated region code $P_1'' = 0000$

(c) $P_1'' \rightarrow 0000$

$P_2 \rightarrow \underline{0100}$

0000 (AND)

Choose $P_2 (0100)$ → bottom

Intersection with bottom boundary.

$m = -0.8846$

$x = x_1 + \dfrac{y - y_1}{m} = 124.35 \simeq 124$. [$y = 10$]

$\therefore P_2' (124, 100)$

Updated region code.

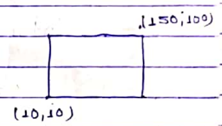
$P_2' = 0000$

Since both P_1'' & P_2' have region code 0000 so, acept the line.

So, acept the line

$P_1'' (22, 100)$ & $P_2 (124, 10)$ is the req^d clipped line.

Q. Line $P_3 (50, 80)$ $P_4 (160, 120)$



⇒ Soln:

Step 1:

$P_1 = 0000$

$P_2 = 1010$

Step 2:

1001	1000	1010
0001	0000	0010
0101	0100	0110

Step 3:

AND operation

$P_1 = 0000$

$P_2 = \underline{1010}$

0000 → need clipping

Step 3.2.1.

A. choose $P_2 (1010)$

Intersection with right boundary.

$m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{120 - 80}{160 - 50} = 0.3636$

$y = y_1 + m(x - x_1)$ where $x = 150$

$y = 80 + 0.3636(150 - \cancel{80}\,50)$

$= \cancel{75.45 \simeq 75}$ $\cancel{105.45 \simeq 105}$ $116.36 \simeq 116$

Now,

$P_2' = (\cancel{95, 10, 75})$ $(\overset{150}{\cancel{10}}, \overset{116}{\cancel{105}})$

$P_2' = 1000$ (top)

Now,

(b) $P_1 = 0000$

$P_2' = 1000$

$0000 \rightarrow$ AND

Choose $P_2'(1000)$
↓ top

Intersection with top boundary

$m = 0.3636$

$x = x_1 + \frac{y - y_1}{m}$ where, $y = 100$

$= 50 + \frac{100-80}{0.3636}$

$= 105.00 \simeq 105$

$P_2'' = (105, 100)$

$P_2'' = 0000$

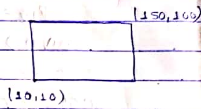
(c) Since both P_1 and P_2'' have region code 0000.
So, accept the lines

$P_1(50,80)$ and $P_2' = (105,100)$ is the reqd clipped line.

Q. $P_5(-50,40)$ & $P_6(30,120)$

Q. Clip lines $P_7(80,200)$, $P_8(250,70)$, $P_9(50,70)$, $P_{10}(50,-40)$, $P_{11}(70,80)$ and $P_{12}(200,80)$ using Cohen Sutherland line clipping algorithm. For a window with bottom left corner (10,10) and top right corner (150,100).

⇒ Soln:



$P_7(80,200)$

$P_8(250,70)$

Region Code for:

$P_7 = 1000$

$P_8 = 0010$

AND ⇒ 0000 → need clipping.

1001	1000	1010
0001	0000	0010
0101	0100	0110

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{70-200}{250-80} = \cancel{-4.29}\ -0.76$

Intersection with top boundary, $y = 100$

$x = x_1 + \frac{y - y_1}{m}$

$= 80 + \frac{100-200}{-0.76}$

$= 211.57 \simeq 212.$

Updated point $(212, 100)$

Updated region $(0010) = P_7'$

$P_7' = 0010$

$P_8 = 0010$

0010 (AND) → no need clipping.

Since the AND operation of P_7' & P_8 is not zero. The line is trivally rejected.

PAGE:
DATE: / /

$P_9(50, 70)$

$P_{10}(50, -40)$

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-40-70}{50-50} = -\infty$

Region code for P_9 & P_{10}.

$P_9 = 0000$

$P_{10} = 0100$

AND ⇒ 0000 → need clipping.

Choosing P_{10}

We find bottom intersect. $y = y_{wmin} = 10$

$x_1 = 50$, $y_1 = -40$

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

or, $\frac{10+40}{m(-\infty)} = x - x_1$

or, $0 = x - 50$

$\therefore x = 50$

Updated point $(50, 10)$

Updated region code = 0000 = P_{10}'

Since both region P_9 and P_{10}' has region code 0000.

The line is trivally accepted with points.

$P_9(50, 70)$

$P_{10}(50, 10)$.

PAGE:
DATE: / /

$P_{11}(70, 80)$

$P_{12}(200, 80)$

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{80-80}{200-70} = 0.$

Region code for P_{11} & P_{12}.

$P_{11} = 0000$

$P_{12} = 0010$

AND = 0000 → need clipping

Intersection with right boundary, $x = 150$

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

$= 80 + 0 \times (150 - 70)$

$= 80$

Updated point = $(150, 80)$

Updated region = 0000 = P_{12}'

Since both region P_{11} & P_{12}' has 0000 region code. The accepted line with point $P_{11}(70, 8)$ and P_{12} $(150, 80)$.

Liang Barsky Line Clipping Algorithm

- Based on analysis of parametric eqn of line.

$$\left.\begin{aligned} x &= x_1 + u\Delta x \\ y &= y_1 + u\Delta y \end{aligned}\right\} \quad 0 \le u \le 1$$

$\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$

Clipping window is represented by

$x_{wmin} \le x_1 + u\Delta x \le x_{wmax}$

$y_{wmin} \le y_1 + u\Delta y \le y_{wmax}$

Each inequality can be expressed as

$u p_k \le q_k$; $K = 1, 2, 3, 4$.

where p_k and q_k are defined as:

$p_1 = -\Delta x$	$q_1 = x_1 - x_{wmin}$
$p_2 = \Delta x$	$q_2 = x_{wmax} - x_1$
$p_3 = -\Delta y$	$q_3 = y_1 - y_{wmin}$
$p_4 = \Delta y$	$q_4 = y_{wmax} - y_1$

Clipped line will be

$$\left.\begin{array}{l} x_1' = x_1 + u_1 \Delta x \\ y_1' = y_1 + u_1 \Delta y \end{array}\right\} u_1 \geq 0$$

$$\left.\begin{array}{l} x_2' = x_1 + u_2 \Delta x \\ y_2' = y_1 + u_2 \Delta y \end{array}\right\} u_2 \leq 1$$

Reject the line if:

(i) $u_1 > u_2$

(ii) $p_k = 0$ and $q_k < 0$

Algorithm for Liang Barsky

1. Initialize value $u_1 = 0$, $u_2 = 1$
2. For $k = 1, 2, 3, 4$
   - 2.1) Calculate p_k and q_k
   - 2.2) Calculate $r_k = q_k / p_k$
   - 2.3) If $(p_k < 0)$, candidate for u_1
   - 2.4) If $(p_k > 0)$, candidate for u_2
   - 2.5) If $p_k = 0$ & $q_k < 0$ reject line goto 5.
3. If $(u_1 > u_2$, reject line, goto 5)

5. Find clipped line.

$x_1' = x_1 + u_1 \Delta x$ $x_2' = x_1 + u_2 \Delta x$

$y_1' = y_1 + u_1 \Delta y$ $y_2' = y_1 + u_2 \Delta y$

5. Repeat step 1 to step 5 for other lines.

4. Find u_1 & u_2 where u_1 = maxvalue (0 & candiate of u_1); u_2 = minvalue (1 & candidates of u_2)

Numerical

Q. $x_{wmin} = 0$ $x_{wmax} = 100$

$y_{wmin} = 0$ $y_{wmax} = 50$

and line we want to clip connects $P_1(10,10)$ & $P_2(110,40$

⇒ Soln:

Construct table

k	p_k	q_k	$u = r_k = (q_k/p_k)$
$k \to 1$ $(p_k < 0)$	$-\Delta x$ $= -(110-10)$ $= -100$	$x_1 - x_{wmin}$ $= 10-0$ $= 10$	$u_1 = \frac{10}{-100} = \frac{-1}{10}$
$k \to 2$ $(p_k > 0)$	Δx $= 110-10$ $= 100$	$x_{wmax} - x_1$ $= 100-10$ $= 90$	$u_2 = \frac{90}{100} = \frac{9}{10}$
$k \to 3$ $(p_k < 0)$	$-\Delta y$ $= -(40-10)$ $= -30$	$y_1 - y_{wmin}$ $= 10-0$ $= 10$	$u_1 = \frac{10}{-30} = \frac{-1}{3}$
$k \to 4$ $(p_k > 0)$	Δy $= 40-10$ $= 30$	$y_{wmax} - y_1$ $= 50-10$ $= 40$	$u_2 = \frac{40}{30} = \frac{4}{3}$

If $u_1 > u_2$, then reject line (completely outside the window)

Here, $u_1 = 0$ or $-1/10$ or $-1/3$

$= 0$ $(u_1 \geq 0) \to$ max value

$u_2 = 1$ or $9/10$ or $4/3$

$= 9/10$ $(u_2 \leq 1) \longrightarrow$ min value

∴ Clipped line will be

$x_1' = x_1 + u_1 \Delta x = 10 + 0 \times 100 = 10$

$y_1' = y_1 + u_1 \Delta x = 10 + 0 \times 30 = 10$

$x_2' = x_1 + u_2 \Delta x = 10 + 9/10 \times 100 = 100$

$y_2' = y_1 + u_2 \Delta x = 10 + 9/10 \times 30 = 37$

$P_1'(10,10)$ & $P_2'(100,37)$

PAGE :
DATE : / /

$\cdot 9 \cdot 23 = 10$

Q. $x_{wmin} = 10$ $\quad x_{wmax} = 150$
$y_{wmin} = 10$ $\quad y_{wmax} = 100$
and the line we want to clip connects $P_1(0, 120)$ & $P_2(130, 5)$

⇒ Solⁿ:

Construct table

K	P_K	q_K	$u = r_K = (q_K/p_K)$
$K \to 1$ ($P_K<0$)	$-\Delta x$ $= -(130-0)$ $= -130$	$x_1 - x_{wmin}$ $= 0-10$ $= -10$	$u_1 = \frac{-10}{-130} = \frac{1}{13}$
$K \to 2$ ($P_K>0$)	Δx $= 130-0$ $= 130$	$x_{wmax} - x_1$ $= 150-0$ $= 150$	$u_2 = \frac{150}{130} = \frac{15}{13}$
$K \to 3$ ($P_K>0$)	$-\Delta y$ $= -(5-120)$ $= +115$	$y_1 - y_{wmin}$ $= 120-10$ $= 110$	$u_2 = \frac{110}{115}$
$K \to 4$ ($P_K<0$)	Δy $= (5-120)$ $= -115$	$y_{wmax} - y_1$ $= 100-120$ $= -20$	$u_1 = \frac{-20}{-115} = \frac{20}{115}$

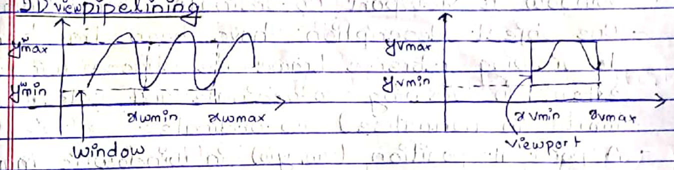
If $u_1 > u_2$, then reject line.
Here, $u_1 = 0$ or $\frac{1}{13}$ or $+\frac{20}{115}$
$= \frac{20}{115} = 0.17$ $(u_1 \geq 0) \to$ max value

$u_2 = 1$ or $15/13$ or $110/115$
$= 110/115 = 0.95$ $(u_2 \leq 1) \to$ min value

∴ Clipped line will be:
$x_1' = x_1 + u_1 \Delta x = 0 + 0.17 \times 130 = 22.1$
$y_1' = y_1 + u_1 \Delta y = 120 + 0.17 \times -115 = \cancel{139.55}\ 100.45$
$x_2' = x_1 + u_2 \Delta x = 0 + 0.95 \times 130 = 123.5$
$y_2' = y_1 + u_2 \Delta y = 120 + 0.95 \times -115 = 10.75$
$P_1' = (22, 100)$ & $P_2' = (124, 11)$

PAGE :
DATE : / /

2D viewpipelining

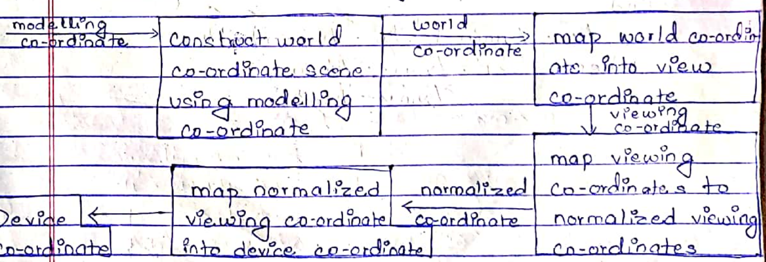


Window:
A rectangular area in world co-ordinate that defines what is to be displayed.

Viewport: A rectangular area in device co-ordinate that defines what is to be displayed.

By changing position of viewport, we can view objects at different positions on the display area of output device. By changing size of the view port, we can change size and proportions of displayed objects.

Two dimensional pipeling



CG III

PAGE : DATE : / /

Window to Viewport Co-ordinate Transformations

- One object description have been transferred to viewing reference frame, we choose window extent in viewing co-ordinate and select viewport limits in normalised co-ordinates.
- A point at position (x_w, y_w) in window is mapped into (x_v, y_v) in associated viewport.
- In order to maintain some relative placement in viewport as in window.

$$\frac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}} = \frac{X_w - X_{wmin}}{X_{wmax} - X_{wmin}}$$

$$\frac{Y_v - Y_{vmin}}{Y_{vmax} - Y_{vmin}} = \frac{Y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}}$$

or, $X_v - X_{vmin} = \left(\frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}\right) \times (X_w - X_{wmin})$

or, $X_v = X_{vmin} + (X_w - X_{wmin}) \cdot S_x$

where, $S_x = \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}$

And

$Y_v = Y_{vmin} + (Y_w - Y_{wmin}) \cdot S_y$

where, $S_y = \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}}$

CG III

PAGE : DATE : / /

Window to viewport transformation

Step 1: Translate the window along with the object such that it's left lower corner is now at origin.

$$\begin{bmatrix} 1 & 0 & -x_{wmin} \\ 0 & 1 & -y_{wmin} \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2: Scale the size of window to match the size of viewport.

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

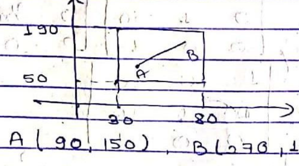
Step 3: Translate the window from origin to the position of viewport.

$$\begin{bmatrix} 1 & 0 & x_{vmin} \\ 0 & 1 & y_{vmin} \\ 0 & 0 & 1 \end{bmatrix}$$

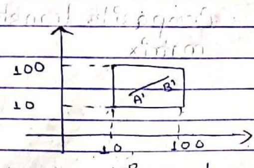
Composite transformation matrix

$$\begin{bmatrix} 1 & 0 & x_{vmin} \\ 0 & 1 & y_{vmin} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_{wmin} \\ 0 & 1 & -y_{wmin} \\ 0 & 0 & 1 \end{bmatrix}$$

Q.



A (90, 150), B(270, 170)



Viewport.

PAGE :
DATE : / /

Here,

$(X_{wmin}, Y_{wmin}) = (30, 50)$

$(X_{wmax}, Y_{wmax}) = (300, 190)$

$(X_{vmin}, Y_{vmin}) = (10, 10)$

$(X_{vmax}, Y_{vmax}) = (100, 100)$

$$S_x = \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}} = \frac{100-10}{300-30} = \frac{9\cancel{0}}{27\cancel{0}} = \frac{3}{9} = \frac{1}{3}$$

$$S_y = \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}} = \frac{100-10}{190-50} = \frac{9\cancel{0}}{14\cancel{0}} = \frac{9}{140}$$

$$\begin{bmatrix} 1 & 0 & -x_{wmin} \\ 0 & 1 & -y_{wmin} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -30 \\ 0 & 1 & -50 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cancel{9/27}\ 1/3 & 0 & 0 \\ 0 & 9/140 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & x_{vmin} \\ 0 & 1 & y_{vmin} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

Now,

Composite transformation matrix $= \begin{bmatrix} 1 & 0 & -30 \\ 0 & 1 & -50 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 9/140 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$

$$= \begin{bmatrix} 0.3333 & 0 & \begin{matrix} 0 \\ \cancel{26.66} \end{matrix} \\ 0 & 0.6428 & \begin{matrix} 6.7857 \\ \cancel{43.57} \end{matrix} \\ 0 & 0 & 1 \end{bmatrix}$$

PAGE :
DATE : / /

$P' = T \cdot P$

$$= \begin{bmatrix} 0.333 & 0 & \begin{matrix} 0 \\ \cancel{26.66} \end{matrix} \\ 0 & 0.6428 & \begin{matrix} 6.7857 \\ \cancel{43.57} \end{matrix} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 90 & 270 \\ 150 & 170 \\ 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{matrix} 30 \\ \cancel{3.31} \end{matrix} & \begin{matrix} 90 \\ \cancel{63.25} \end{matrix} \\ \begin{matrix} 16.428 \\ \cancel{52.85} \end{matrix} & \begin{matrix} 17.714 \\ \cancel{65.70} \end{matrix} \\ 1 & 1 \end{bmatrix}$$

PAGE :
DATE : / /

CHAPTER:04
3-D TRANSFORMATION

We follow notation

$$P' = T \cdot P$$

where $P' = \begin{bmatrix} x_1' & x_2' & \cdots & x_n' \\ y_1' & y_2' & \cdots & y_n' \\ z_1' & z_2' & \cdots & z_n' \\ 1 & 1 & \cdots & 1 \end{bmatrix}$

$$P = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

and T is transformation matrix.

Same as 2D there are 5 transformations

1. Translation

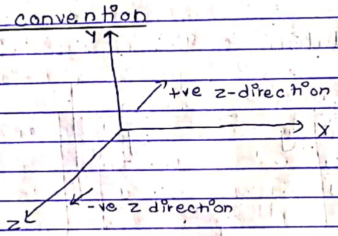
$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Scaling

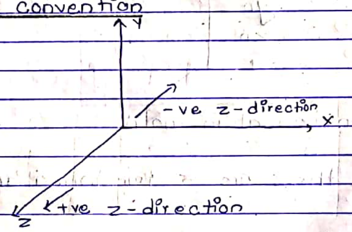
$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note :- In 3D there are two conventions.

① LHS convention



② RHS convention



3. Shearing

Assume a = shear constant along x-axis

b = shear constant along y-axis

c = shear constant along z-axis

→ z-axis shear (across xy-plane)

$x' = x + az$

$y' = y + bz$

$z' = z$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$P' = SHz \cdot P$

→ x-axis shear (across yz-plane)

$x' = x$

$y' = y + bx$

$z' = z + cx$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ b & 1 & 0 & 0 \\ c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$P' = SHx \cdot P$

→ y-axis shear (across xz-plane)

$y' = y$

$x = x + ay$

$z = z + cy$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Reflection of points relative to xy plane

$P' = Rf_{xy\text{-plane}}\, P$

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

PAGE:
DATE: / /

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

Relative to xz plane

$P' = Rf_{zx}$ plane $\cdot P$

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

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

Relative to yz plane

$P' = Rf_{yz}$ plane $\cdot P$

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

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

Three - dimensional Composite transformation.

- successive translation is additive.
- successive rotation is additive.
- successive scaling is multiplicative.

Rotation

- Rotation about z-axis (on xy-plane)

$$\left.\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned}\right\} \cdots\cdots (i)$$

PAGE:
DATE: / /

$P' = R_z(\theta) \cdot P$

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

Rotation about x-axis (yz-plane)

In eqⁿ (i) replace x by y, y by z and z by x.

$$\left.\begin{aligned} y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \\ x' &= x \end{aligned}\right\} \text{———} (ii)$$

$P' = R_x(\theta) \cdot P$

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

Rotation about y-axis (zx plane)

In eqⁿ (ii), replace x by y, y by z & z by x

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

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

$y' = y$

$P' = R_y(\theta) \cdot P$

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

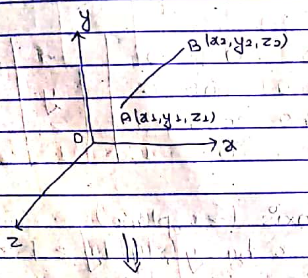
PAGE :
DATE : / /

Rotation about an axis parallel to any one standard axis.

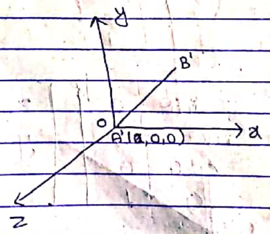
- Translate the axis to standard axis
- Perform desired rotation.
- Inverse translation.

$T_\theta = T^{-1} \cdot R_\theta \cdot T$

Rotation about an axis not parallel to any standard axis.



Translate point A to origin



$B' = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$

PAGE :
DATE : / /

For step 2.1

$u' \cdot u_z = \left(0, \frac{b}{\sqrt{b^2+c^2}}, \frac{c}{\sqrt{b^2+c^2}}\right) \cdot (0,0,1)$

$= \frac{c}{\sqrt{b^2+c^2}}$

$|u'| \cdot |u_z| = \sqrt{\left(\frac{b}{\sqrt{b^2+c^2}}\right)^2 + \left(\frac{c}{\sqrt{b^2+c^2}}\right)^2} \times 1$

$= 1$

$\cos\alpha = \frac{u' \cdot u_z}{|u_1||u_2|}$

$\boxed{\cos\alpha = \frac{c}{\sqrt{b^2+c^2}} = \frac{c}{d}}$

$\sin\alpha = \frac{u' \otimes u_z}{|u'||u_z|} \hat{n}$ ——— (i)

$u' \otimes u_z = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 0 & b/d & c/d \\ 0 & 0 & 1 \end{vmatrix}$

$= \frac{b\hat{i}}{d}$

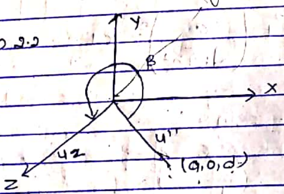
From (i) $\sin\alpha = \frac{b\hat{i}/d}{1} \times \hat{i}$

$\boxed{\sin\alpha = \frac{b}{d}}$

$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$R^{-1}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 2.2



$$u'' = \frac{a}{\sqrt{a^2+d^2}}\vec{i} + 0\vec{j} + \frac{d}{\sqrt{a^2+d^2}}\vec{k}$$

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

$$\cos\beta = \frac{u'' \odot u_z}{|u''||u_z|}$$

$$u'' \odot u_z = \left(\frac{a}{l}, 0, \frac{d}{l}\right) \cdot (0, 0, 1)$$

$$= \frac{d}{l}$$

$$|u''| \cdot |u_z| = \sqrt{\frac{a^2}{l^2} + \frac{d^2}{l^2}} \cdot 1$$

$$= \sqrt{\frac{a^2+d^2}{l^2}}$$

$$= \frac{\sqrt{a^2+d^2}}{l} = \frac{\cancel{l}}{\cancel{l}} = 1$$

$$\boxed{\cos\beta = \frac{d}{l}}$$

$$\sin\beta = \frac{u'' \otimes u_z}{|u''|\,|u_z|} \cdot \hat{n} \quad \text{——— (ii)}$$

$$u'' \otimes u_z = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a/l & 0 & d/l \\ 0 & 0 & 1 \end{vmatrix}$$

$$= -\frac{a}{l}\vec{j}$$

From (ii) $\sin\beta = \dfrac{\left(-\frac{a}{l}\vec{j}\right)}{1} \cdot \vec{j}$

$$\boxed{\sin\beta = -\frac{a}{l}}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d/l & 0 & -a/l & 0 \\ 0 & 1 & 0 & 0 \\ a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

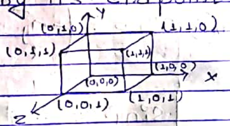
$$R^{-1}_y(\beta) = \begin{bmatrix} d/l & 0 & a/l & 0 \\ 0 & 1 & 0 & 0 \\ -a/l & 0 & d/l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PAGE :
DATE : / /

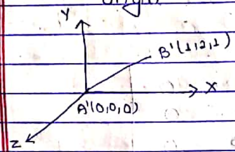
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R(\theta) = T^{-1} R_x^{-1}(\alpha)\, R_y^{-1}(\beta)\, R_z(\theta)\, R_y(\beta)\, R_x(\alpha)\, T$$

Q. Find the new co-ordinates of a unit cube 90°- rotated about an axis defined by its endpoint A(2,1,0) and B(3,3,1)

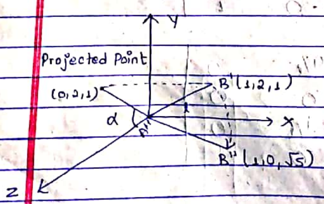


Step 1: Translate point A to origin



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

Step 2: Rotate axis A'B' about the x-axis by and angle α, until it lies on the xz plane.



$$\sin\alpha = \frac{2}{\sqrt{2^2+1^2}} = \frac{2}{\sqrt{5}} = \frac{2\sqrt{5}}{5}$$

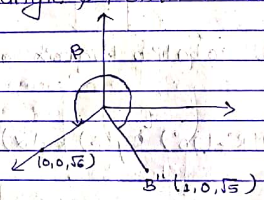
$$\cos\alpha = \frac{1}{\sqrt{5}} = \frac{\sqrt{5}}{5} = \frac{c}{d}$$

$$l = \sqrt{1^2+2^2+1^2} = \sqrt{6}$$

PAGE :
DATE : / /

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{5}/5 & -2\sqrt{5}/5 & 0 \\ 0 & 2\sqrt{5}/5 & \sqrt{5}/5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 3: Rotate axis A'B'' about the y-axis by and angle ϕ, until it coincides with z-axis



$$\sin\beta = -\frac{a}{l} = -\frac{1}{\sqrt{6}} = -\frac{\sqrt{6}}{6}$$

$$\cos\beta = \frac{d}{l} = \frac{\sqrt{5}}{\sqrt{6}} = \frac{\sqrt{30}}{6}$$

$$R_y(\beta) = \begin{bmatrix} \sqrt{30}/6 & 0 & -\sqrt{6}/6 & 0 \\ 0 & 1 & 0 & 0 \\ \sqrt{6}/6 & 0 & \sqrt{30}/6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1}_y(\beta) = \begin{bmatrix} \sqrt{30}/6 & 0 & \sqrt{6}/6 & 0 \\ 0 & 1 & 0 & 0 \\ -\sqrt{6}/6 & 0 & \sqrt{30}/6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{5}/5 & 2\sqrt{5}/5 & 0 \\ 0 & -2\sqrt{5}/5 & \sqrt{5}/5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 4: Rotate the cube 90° about z-axis

$$R_z(90^\circ) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally the concatenated rotation matrix about arbitrary axis becomes.

$$R(\theta) = T^{-1} R_x^{-1}(\alpha)\, R_y^{-1}(\beta)\, R_z(90^\circ)\, R_y(\beta)\, R_x(\alpha)\, T$$

3D View Pipelining

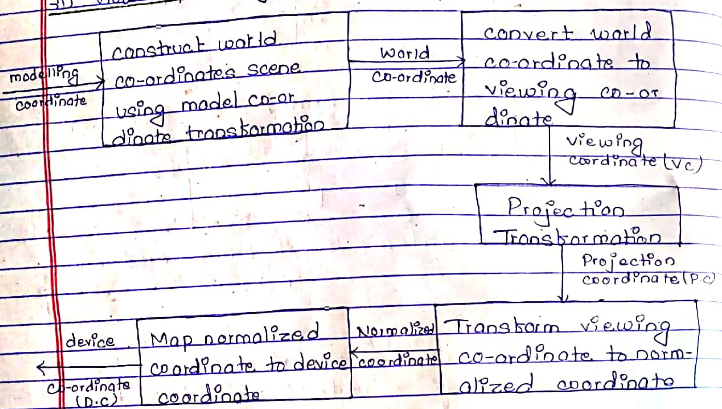


Fig. 3D View Pipelining

Once the scene has been modelled, WC are converted to VC which is used in graphics packages as a reference for specifying observer viewing position and position of projection plane.

Projection operations are performed to convert VC to coordinate positions on projection plane which are normalized and then mapped to output devices.

Normalized coordinates define which part will be visible and where it will appear on the display surface.

Transformation from world to viewing coordinate

1. Translate viewing coordinate to world co-ordinate origin.
2. Apply rotation to align x_v, y_v & z_v axes with world x_w, y_w & z_w axes.
3. The rotation can be either by rotating viewing coordinate axes along x_w, y_w, z_w axes or the composite rotation matrix can be obtained by setting up u, v, n vectors and forming a complete rotation matrix.

Given reference points P_0 and look at point P_{ref} and view up vector $V = (V_x, V_y, V_z)$

$$N = P_0 - P_{ref} = (N_x, N_y, N_z)$$

$$V = (V_x, V_y, V_z)$$

$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

Again, unit vector along x_v axis

$$u = \frac{V \times N}{|V \times N|} = \cancel{(v_x, v_y, v_z)}\ (u_x\ u_y\ u_z)$$

where, $V \times N = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ V_x & V_y & V_z \\ N_x & N_y & N_z \end{vmatrix}$

and $\vec{v} = n \times u = (v_x\ v_y\ v_z)$

where, $n \times u = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ n_x & n_y & n_z \\ u_x & u_y & u_z \end{vmatrix}$

The composite rotation matrix is then,

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, final transformation matrix is

$V_c = R \cdot T$

Q. Find viewing transformation matrix for which,
Pref (look at position) = (0, 0, 0)
P_0 (eye position) = (1, 1, 1)
& view up vector, V = (0, 1, 0)

⇒ Soln:
We have,

$N = P_0 - P_{ref} = (1, 1, 1)$

$$n = \frac{N}{|N|} = \frac{(1, 1, 1)}{\sqrt{1^2+1^2+1^2}} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

n_x, n_y, n_z

Given,

$V = (0, 1, 0)$

$$u = \frac{V \times N}{|V \times N|}$$

$$V \times N = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}$$

$V \times N = \vec{i}(1-0) - \vec{j}(0-0) + \vec{k}(-1)$
$= \vec{i} - \vec{k}$

$|V \times N| = \sqrt{2}$

$$u = \frac{1}{\sqrt{2}}\vec{i} - \frac{1}{\sqrt{2}}\vec{k} = \left(\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}\right)$$

Now,

$$v = n \times u = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \end{vmatrix}$$

$$= -\frac{1}{\sqrt{6}}\vec{i} + \frac{2}{\sqrt{6}}\vec{j} + \frac{1}{\sqrt{6}}\vec{k}$$

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

$$R = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ -1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now,

$$M_{wc,vc} = R \cdot T = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ -1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PAGE :
DATE : / /

Projection: Process of converting n dimension object into n-1 or less dimensions.

Taxonomy of Projection

- Projection
  - Parallel
    - Orthographic
      - Top (y) (xz plane)
      - Side (z) (xy plane)
      - Front (z) (yz plane)
      - Axonometric (multiple faces, doesnot lie in any plane)
    - Oblique
      - Cavilear — viewplane, Projection line
      - Cabinet — Projection line
  - Perspective
    - 1pt
    - 2pt
    - 3pt

Parallel Projection: Projection lines are parallel to eachother.

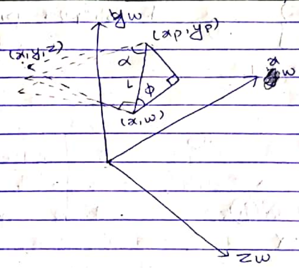
- Relative size of object is maintained.
- Unrealistic

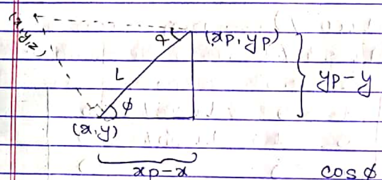
Orthographic: Projection lines are perpendicular to the projection plane view plane

Oblique: Projection lines are not perpendicular to view plane.

PAGE :
DATE : / /

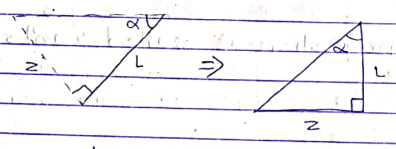
Composite matrix for oblique projection





$xp = x + \cos\phi\, L$ $\qquad \cos\phi = \frac{xp-x}{L}$

$yp = y + \sin\phi\, L$ $\qquad \sin\phi = \frac{yp-y}{L}$



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

The figure shown is oblique projected projection of a point (x, y, z) to position (xp, yp) on viewplane.

Scanned with CamScanner

PAGE :
DATE : / /

From fig.   L- distance betn orthographic & oblique projected point

$$\left.\begin{aligned} x_p &= x + L\cos\phi \\ y_p &= y + L\sin\phi \end{aligned}\right\} \text{——— (i)}$$

α depends on z & L

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

$$\text{or, } L = \frac{z}{\tan\alpha}$$

$$\text{or, } L = L_1 z \text{ ——— (ii)}$$

where, $L_1 = \cot\alpha$

From (i) & (ii),

$$x_p = x + z(L_1\cos\phi)$$
$$y_p = y + z(L_1\sin\phi)$$

In matrix,

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

Common value for $\phi = 63.3^\circ$ or 45°

$\tan\alpha = 1$ or 2

If $\tan\alpha = 1$, projection is called cavilear projection

PAGE :
DATE : / /

Perspective Projection

For a perspective position, object positions are transformed to view plane along lines that converge to a point called projection reference point (PRP) or center of projection (COP). The projected view of an object is determined by calculating the intersection of projection lines with a view plane.

A perspective projection produces realistic views but doesnot preserve relative proportions. Projections of distant objects are smaller than the projections of objects of same size that are closer to projection plane.

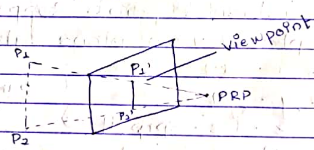
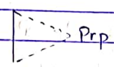


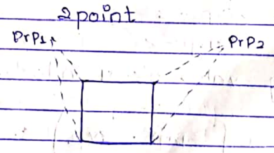
Fig. Perspective projection of a line of a viewplane

Types

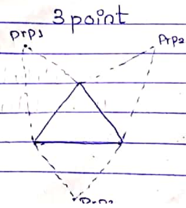
1 point



2 point



3 point



PAGE :
DATE : / /

<u>Composite transformation</u>

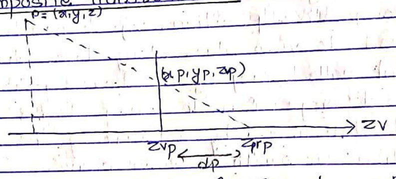


Fig. Perspective projection of a point p with co-ordinates (x, y, z) to position (xp, zp, zvp) on view plane

To obtain perspective ~~position~~ projection we transform points along projection lines that meet at projection refrence point. Suppose zprp is PRP and we place view plane at zvp.

$$\left.\begin{aligned} x' &= x - xu \\ y' &= y - yu \\ z' &= z - (z - z_{prp})u \end{aligned}\right\} \text{parametric form}$$

where (x', y', z') represents any point on projection line and u takes value from 0 to 1.
When $u = 0$, we are at position $P(x, y, z)$ when $u = 1$ we are at zprp $(0, 0, z_{prp})$ on viewplane.

$$z' = z_{vp}$$

And we can solve z' eqn for parameter is,

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

Substituting value of u into eqns of x' and y', we obtain

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = x\left(\frac{d_p}{z_{prp} - z}\right) \Rightarrow x \cdot \frac{1}{h} = x_p$$

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

or, $x_p = \dfrac{x(z_{prp} - z) - x(z_{vp} - z)}{z_{prp} - z}$

or, $x_p = \dfrac{x(z_{prp} - z_{vp})}{z_{prp} - z}$

PAGE :
DATE : / /

and $y_p = y\left(\dfrac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = y\left(\dfrac{d_p}{z_{prp} - z}\right)$

where,
$d_p = z_{prp} - z_{vp}$ is distance of view plane from projection refrence point.

Using three dimentional homogeneous coordinate representation, we can write perspective projection transformation in matrix form as:

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

In this representation, homogeneous factor is,

$$h = \frac{z_{prp} - z}{d_p} = \frac{z_{prp}}{d_p} - \frac{z}{d_p} = -\left(\frac{1}{d_p}\right)z + \left(\frac{z_{prp}}{d_p}\right)$$

and projection coordinates on view plane are calculated from homogeneous coordinate as:

$$x_p = \frac{x_h}{h}, \quad y_p = \frac{y_h}{h}$$

where,
Original z-coordinate value would be obtained in projection co-ordinates for visible surface and other depth processing.

PAGE :
DATE : / /

CHAPTER: 06
SURFACE MODELLING

3D-Representation

- Graphics scene can contain many kinds of objects like trees, clouds, rocks, marbles etc.
- To produce realistic display of scene, we need to use representations that accurately models objects characteristics.
- Representation scheme for solid objects are often divided into two broad categories:

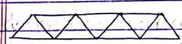
① **Space partitioning Representation**
- Used to describe interior properties of partitioning the spatial region containing the object into a set of small, non-overlapping contiguous solids (usually cubes). Eg: Octree

② **Boundary Representations**
- Describes 3D-objects as a set of surfaces that separate the object interior from the environment.
- Most commonly used representation for 3D-object.
- Speeds up surface rendering.
- greater the number of small surfaces, better the approximation.

Eg: Polygon surfaces, spline surfaces etc.

PAGE :
DATE : / /

Polygon Surface
- 3D graphics object is a set of surface polygons that enclose object interior.
- A polygon mesh is a collection of edges vertices and polygons connected such that each edge is shared by atmost two polygons.



Triangle strip
(Eg of Polygon Mesh)

Quadilateral Mesh

Data Representation for Polygon Surface

- We specify polygon ~~sor~~ surface with a set of vertex coordinates and associated attribute parameters
- The information for each polygon are stored in polygon tables
- Polygon Table can be organized into two groups:

Geometric Tables contain vertex co-ordinates and parameters to identify spatial orientation of polygon surfaces.

Attribute Tables consist of parameters that specify transparency, surface texture, color etc.

PAGE:
DATE: / /

Polygon Data Table

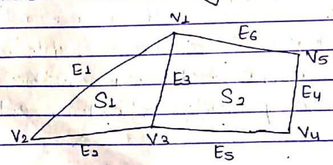
- Geometric Tables
  - Vertex Table
  - Edge Table
  - Polygon Surface Table
- Attribute Tables

Vertex Table: stores the co-ordinates of each vertex in the object.

Edge Table: contains the pointer back to the vertex table to identify the vertices ~~also~~ for each polygon edge.

Polygon Surface Table: contains the pointer back to the edge table to identify edges for each polygon.

Example:
Let's consider two adjacent polygon surfaces formed with six edges and five vertices.



PAGE:
DATE: / /

	Vertex Table	Edge Table	Polygon Surface Table
V_1:	x_1, y_1, z_1	E_1: V_1, V_2	S_1: E_1, E_2, E_3
V_2:	x_2, y_2, z_2	E_2: V_2, V_3	S_2: E_3, E_4, E_5, E_6
V_3:	x_3, y_3, z_3	E_3: V_3, V_1	
V_4:	x_4, y_4, z_4	E_4: V_3, V_4	
V_5:	x_5, y_5, z_5	E_5: V_4, V_5	
		E_6: V_5, V_1	

Alternate Representation
For quick identification of redundancy, pointers to polygon tables are added to edge table so as to notice common edges quickly.

Edge Table
E_1: V_1, V_2, S_1
E_2: V_2, V_3, S_2
E_3: V_3, V_1, S_1, S_2
E_4: V_3, V_4, S_2
E_5: V_4, V_5, S_2
E_6: V_5, V_1, S_2

Guidelines to generate error free tables:
- Every vertices listed as end points for at least two edges.

- Every edge is a part of atleast one polygon.
- Every polygon is closed.
- Every polygon has atleast one shared edge.
- If edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to polygon.

Spatial Orientation of Polygon surface

This information is obtained from the vertex coordinate values and the equation that describes the polygon planes

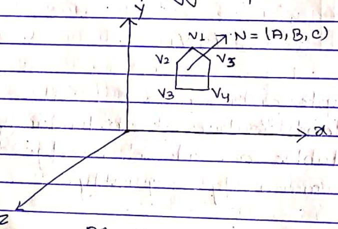


Fig. Vector N, Normal to the Plane

Surface Normal

- A vector normal to the surface of the plane.
- Describes the orientation of the polygon surface

The eqn of the plane surface can be expressed in the form

$$Ax + By + Cz + D = 0$$

where, (x, y, z) is any point on plane and coefficient A, B, C, D are constant describing spatial properties of the plane.

Let 3 successive polygon vertices be $V_1(x_1, y_1, z_1)$ $V_2(x_2, y_2, z_2)$ $V_3(x_3, y_3, z_3)$ then

$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$
$$\Rightarrow \frac{A}{D}x_k + \frac{B}{D}x_k + \frac{C}{D}z_k = -1 \quad [k = 1, 2, 3]$$

Applying Cramer's rule

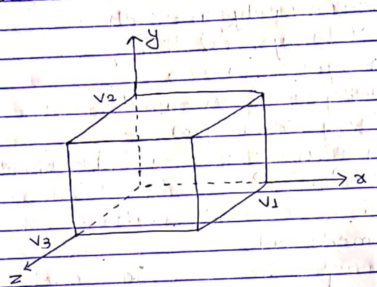
$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \qquad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad D = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Expanding determinants

$$A = y_1(z_1 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$
$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$
$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$
$$D = -x_1(y_2 z_3 - y_3 z_2) - x_2(y_3 z_1 - y_1 z_3) - x_3(y_1 z_2 - y_2 z_1)$$

Since we are usually dealing with polygon surface that encloses the object interior, we need to distinguish between the two sides of the surface. The side of the plane that faces the object interior is called "inside face" and the visible outerside is called "outer face".



The element of the surface normal (for shaded surface) can be obtained using a vector cross-product calculation.

Let us take three vector position $\vec{V_1}$, $\vec{V_2}$ and $\vec{V_3}$ taken in counter clockwise order when viewed from outside the surface in a right handed cartesian system. Forming two vector one from $\vec{V_1}$ to $\vec{V_2}$ and another from $\vec{V_1}$ to $\vec{V_3}$. We calculate $\vec{N}$ as:

$$\vec{N} = (\vec{V_2} - \vec{V_1}) \times (\vec{V_3} - \vec{V_1})$$

This generates value for plane parameter A, B and C. we can obtain value of D by substituting these value and co-ordinates of one vertex.

The plane equation can be expressed in vector form using normal $\vec{N}$ and position $\vec{P}$ of any point plane as:

$$\vec{N} \cdot \vec{P} = -D \left[Ax_p + By_p + Cz_p = \vec{N} \cdot \vec{P} \right]$$

For any point (x, y, z) not on a plane with parameters A, B, C, D we have

$$Ax + By + Cz + D \neq 0$$

- If $Ax + By + Cz + D < 0$, (x, y, z) is inside surface
- If $Ax + By + Cz + D > 0$, (x, y, z) is outside surface

CHAPTER: 05
CURVE MODELLING

Spline

In drafting terminology, a spline is a flexible strip used to produce a smooth curve designated set of points. Several small weights are distributed along the length of the strip to hold it in position on the drafting table as curve is drawn.

The term spline curve originally referred to a curve drawn in this manner.

We can mathematically describe such a curve with a piecewise cubic polynomial function whose first and second derivatives are continuous across various curve sections.

In computer graphics, the term spline curve refers to any composite curve formed with polynomial sections satisfying specified continuity conditions at boundary of pieces.

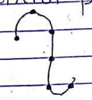
Typical CAD applications for splines include the design of automobile bodies, aircraft and spacecraft surfaces and ship hulls.

PAGE :
DATE : / /

Interpolation and approximation splines:

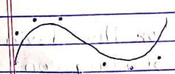
- A spline curve is specified by giving a set of co-ordinate positions called control points, which indicates general shape of curve.
- These control points are then fitted with piecewise continuous parametric polynomial functions in one of two ways:

a. When polynomial sections are fitted so that curve passes through each control point, resulting curve is said to interpolate set of control points.



Fig; A set of six control points interpolated with piecewise continuous polynomial sections.

b. When polynomials are fitted to general control point path ~~withat with~~ necessarily passing through any control point, resulting curve is said to approximate set of control points.



Fig; A set of six control points approximated with piecewise continuous polynomial section.

PAGE :
DATE : / /

- Interpolation curves are commonly used to digitize drawings or specify animation paths. Approximation curves are primarily used as design tools to structure object surfaces.
- A spline curve is defined, modified and manipulated with operations on control points.
- IOE The convex polygon boundary that encloses a set of control points is called convex hull.
- One way to envision the shape of a convex hull is to imagine a rubber band streched around the positions of control points so that each control point is either on perimeters of hull or inside it.

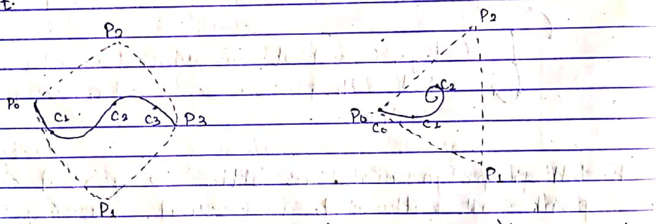


Fig. convex hull shapes (dashed lines) for two set of control points.

- Convex hulls provides a measure for the deviation of a curve or surface region bounding control points. Some splines are bounded by convex hull, thus ensuring that polynomials smoothly follow control points without erratic oscillation.

PAGE :
DATE : / /

- A polyline connecting sequence of control points for an approximation spline is usually displayed to remind a designer of control points. Ordering set of connected line segments is often referred to as control graph curve.

- Other names for series of straight line sections connecting control points in order specified are control and characteristic polygon.

Spline can be described in following ways:
- Blending function
- Characterization matrix
- boundary condition

Some common spline curves are:
(i) Piecewise cubic spline
(ii) Hermite spline (Interpolation spline)
(iii) B-spline
(iv) Bezier curve (Approximation Spline)

Parametric Cubic Curve

A parametric cubic curve is defined as

$$P(t) = \sum_{i=0}^{3} a_i t^i \quad \text{—— (i)} \qquad 0 \le t \le 1$$

where $P(t)$ is a point on curve.
Expanding eqn (i)

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

The eqn is separated into 3 components of $P(t)$

- Interpolation curves are commonly used to digitize drawings or specify animation paths. Approximation curves are primarily used as design tools to structure object surfaces.
- A spline curve is defined, modified and manipulated with operations on central points.
- IOE - The convex polygon boundary that encloses a set of control points is called convex hull.
- One way to envision the shape of a convex hull is to imagine a rubber band streched around the positions of control points so that each control point is either on perimeters of hull or inside it.

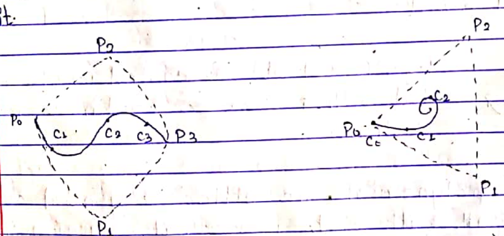


Fig. convex hull shapes (dashed lines) for two set of control points

- Convex hulls provides a measure for the deviation of a curve or surface region bounding control points. Some splines are bounded by convex hull, thus ensuring that polynomials smoothly follow control points without erratic oscillation.

- A polyline connecting sequence of control points for an approximation spline is usually displayed to remind a designer of control points. Ordering set of connected line segments is often referred to as control graph curve.

- Other names for series of straight line sections connecting control points in order specified are control and characteristic polygon.

Spline can be described in following ways:
- Blending function
- Characterization matrix
- boundary condition

Some common spline curves are:
(i) Piecewise cubic spline (Interpolation spline)
(ii) Hermite spline
(iii) B-spline (Approximation spline)
(iv) Bezier curve

Parametric Cubic Curve
A parametric cubic curve is defined as
$$P(t) = \sum_{i=0}^{3} a_i t^i \quad \text{---} \quad (i) \qquad 0 \le t \le 1$$

Where $P(t)$ is a point on curve
Expanding eqn (i)
$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$
The eqn is separated into 3 components of $P(t)$

PAGE :
DATE : / /

$x(t) = a_{3x}t^3 + a_{2x}t^2 + a_{1x}t + a_{0x}$
$y(t) = a_{3y}t^3 + a_{2y}t^2 + a_{1y}t + a_{0y}$
$z(t) = a_{3z}t^3 + a_{2z}t^2 + a_{1z}t + a_{0z}$

Continuity Condition
- 2 types

IOE (a) Parametric Continuity Condition
- To ensure a smooth transition from one section of a piecewise parametric curve to next. Each section of spline is described with a set of parametric curve to next. Each section of a spline is described with a set of parameters co-ordinate functions of form.

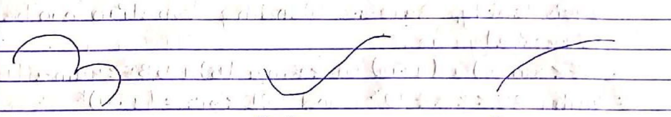
$X = x(u)$, $Y = y(u)$, $Z = z(u)$
$U_1 \leq U \leq U_2$ —— (i)

(1) Zero Order Parametric Continuity
- described as c^0 continuity means simply that curve meet i.e. the values of X, Y and Z evaluated at U_2 for first curve section are equal respectively, to values of x, y and z evaluated at U_1 for next curve section.

(2) First Order Parametric Continuity
- referred to as c^1 ~~coefficien~~ continuity means that first parametric derivations (tangent lines) of co-ordinate function in eqn (i) for two successive cosine sections are equal at their joining point.

PAGE :
DATE : / /

(3) Second Order Parametric Continuity
- referred to as c^2 continuity means that both first and second parametric derivations of two curve sections are same at intersection.



(a) zero order (b) first order (c) second order

Fig. Piecewise construction of a curve by joining two curve segment using different order continuity.

Beizer Curve
Unlike piecewise spline curve, a Beizer Spline Curve can be fitted to any number of control points. Instead of finding out the tangent vectors at any of the control points, a set of characteristic polynomial approximating functions called as Beizer blending function is used.

Definition: Given a set of $(n+1)$ control points $P_0, P_1, P_2, \ldots P_n$ that will fit to those points as mathematically defined by
$$P(u) = \sum_{k=0}^{n} P_k \, BEZ_{k,n}(u) \quad ; \quad 0 \leq U \leq 1 \text{ —— (i)}$$

The Beizer Blending function $BEZ_{k,n}(u)$ are the Bernstein Polynomial.

PAGE:
DATE: / /

$$BEZ_{k,n}(u) = C(n,k)\, u^k (1-u)^{n-k}$$

where, $C(n,k)$ are binomial coefficients

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

Equivalently, Beizer blending function can be defined recursively as.

$$BEZ_{k,n}(u) = (1-u) BEZ_{k,n-1}(u) + u BEZ_{k-1,n-1}(u),\ n>k\geq 1$$

with, $BEZ_{k,k} = u^k$ and $BEZ_{0,k} = (1-u)^k$

Eqn (i) represents the set of three parametric equation for individual curve co-ordinates.

$$x(u) = \sum_{k=0}^{n} x_k\, BEZ_{k,n}(u)$$

$$y(u) = \sum_{k=0}^{n} y_k\, BEZ_{k,n}(u)$$

$$z(u) = \sum_{k=0}^{n} z_k\, BEZ_{k,n}(u)$$

As a rule, Beizer curve is a polynomial of degree one less than the number of control points used 3 points generate parabola, 4-cubic curve and so forth

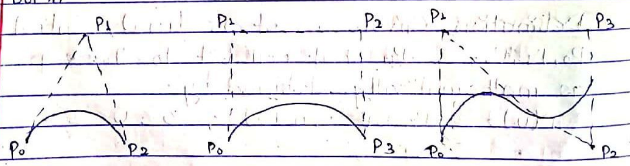


Fig. Beizer Curve

PAGE:
DATE: / /

CHAPTER: 07
VISIBLE SURFACE DETERMINATION

Q. Determine whether two surfaces of object with normals $2\vec{i} - 3\vec{j} + 4\vec{k}$ and $\vec{i} + \vec{j} - 2\vec{k}$ respectively viewed from a given direction $\vec{i} - \vec{j} + \vec{k}$ are backface or frontface.

⇒ Soln:

$\vec{N_1} = 2\vec{i} - 3\vec{j} + 4\vec{k}$

$\vec{N_2} = \vec{i} + \vec{j} - 2\vec{k}$

$\vec{V} = \vec{i} - \vec{j} + \vec{k}$

Now,

$\vec{N_1} \cdot \vec{V} = (2\vec{i} - 3\vec{j} + 4\vec{k}) \cdot (\vec{i} - \vec{j} + \vec{k})$

$= 2 + 3 + 4$

$\vec{N_1} \cdot \vec{V} = 9 > 0$

$\vec{N_2} \cdot \vec{V} = (\vec{i} + \vec{j} - 2\vec{k}) \cdot (\vec{i} - \vec{j} + \vec{k})$

$= 1 - 1 - 2$

$\vec{N_2} \cdot \vec{V} = -2 < 0$

Since, $\vec{N_1} \cdot \vec{V} = 9 > 0$ (backface)

$\vec{N_2} \cdot \vec{V} = -2 < 0$ (front face)

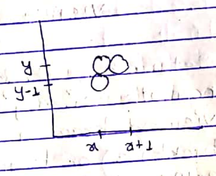
Drawback of Back Face Detection Method

1. Doesnot detect multiple objects.
2. Doesnot detect concave polyhedron
3. Doesnot detect spline surfaces.

Depth Calculation:

$$Ax + By + Cz + D = 0$$

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



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

$$z' = \frac{-A(x+1) - By - D}{C}$$

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

$$z' = z - \frac{A}{C}$$

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

$$z' = \frac{-Ax - B(y-1) - D}{C}$$

$$= \frac{-Ax - By - D + B}{C}$$

$$z' = z + \frac{B}{C}$$

CHAPTER: 06 (CONTINUED)

Properties of Beizier Curve

1. It always passes through the first and last control points i.e, the boundary conditions at two ends of curve are:

$$\left.\begin{aligned} P(0) &= P_0 \\ P(1) &= P_n \end{aligned}\right\} \text{-----} (i)$$

2. Values of parametric first derivatives of a bezier curve at end points can be calculated from control point co-ordinate as

$$P'(0) = -nP_0 + nP_1$$
$$P'(1) = -nP_{n-1} + nP_n$$

3. Similarly, parametric 2nd derivatives of bezier curve at end points are calculated as

$$P''(0) = n(n-1)[(P_2 - P_1) - (P_1 - P_0)]$$
$$P''(1) = n(n-1)[(P_{n-2} - P_{n-1}) - (P_{n-1} - P_n)]$$

4. The degree of the polynomial defining the curve segment is one less than the number of defining polygon point.

5. The curve is always contained within the convex hull of the control points.

6. The sum of all Bezier Blending functions is always 1 and are always positive i.e

PAGE:
DATE: / /

$$\sum_{k=0}^{n} BEZ_{k,n}(0) = 1$$

(7) The curve can be translated and rotated by applying transformation on the control points.

Q. Find the eqn of Bezier curve whose control paths are $P_0(2,6)$, $P_1(6,8)$ & $P_2(9,12)$. Also find co-ordinate of point at $u = 0.8$

⇒ Soln:

$P_0(2,6) = (x_0, y_0)$
$P_1(6,8) = (x_1, y_1)$
$P_2(9,12) = (x_2, y_2)$

The parametric eqn for Bezier curve of order 2 is:

$$\sum_{k=0}^{2} P_k \, BEZ_{k,n}(u)$$

$$= P_0(1-u)^2 + 2P_1 u(1-u) + P_2 u^2 \quad \text{—— (i) for } 0 \le u \le 1$$

For individual curve co-ordinates,

$$x = x_0(1-u)^2 + 2ux_1(1-u) + x_2 u^2 \quad \text{—— (ii)}$$
$$y = y_0(1-u)^2 + 2uy_1(1-u) + y_2 u^2 \quad \text{—— (iii)}$$
$$z = 0$$

Eqn (i) is the generic equation for the Bezier curve with specified control points.

PAGE:
DATE: / /

Eqn (ii) and (iii) are for individual curve co-ordinates

For $u = 0.8$,

$x = 7.76 \cong 8$
$y = 10.48 \cong 10$
$z = 0$

∴ The co-ordinate at $u = 0.8$ is $(8, 10)$

Note*

$$\boxed{\begin{aligned} BEZ_{k,n}(u) &= C(n,k)\, u^k (1-u)^{n-k} \\ C(n,k) &= \frac{n!}{k!\,(n-k)!} \end{aligned}}$$

Q. Construct a Bezier curve of order 3 with polygon vertices $A(1,1)$, $B(2,3)$, $C(4,3)$ and $D(6,4)$

⇒ Soln:

$P_0(1,1) = (x_0, y_0)$
$P_4(6,4) = (x_4, y_4)$

The parametric eqn for Bezier curve of order 3

$$\sum_{k=0}^{3} P_k \, BEZ_{k,n}(u)$$

$$= P_0(1-u)^3 + 3P_1 u(1-u)^2 + 3P_2 u^2(1-u) + P_3 u^3 \quad \text{—— (i)}$$

for $0 \le u \le 1$.

For individual curve co-ordinates,

$$x = x_0(1-u)^3 + 3ux_1(1-u)^2 + 3u^2 x_2(1-u) + x_3 u^3$$
$$y = y_0(1-u)^3 + 3uy_1(1-u)^2 + 3u^2 y_2(1-u) + y_3 u^3$$
$$z = 0$$

PAGE :
DATE : / /

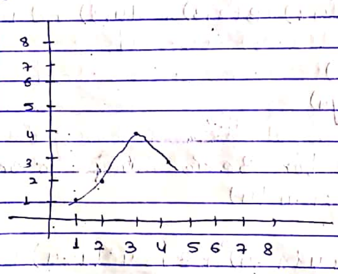
Eqn (i) is the generic eqn for the Bezier curve with specified control points.

Eqn (ii) and (iii) are for individual curve co-ordinates

For $u = 0.25$
$x = 1.92 \cong 2$
$y = 2.17 \cong 2$

For $u = 0.5$
$x = 3.125 \cong 3$
$y = 2.87 \cong 4$

For $u = 0.75$
$x = 4.51 \cong 4$
$y = 3.39 \cong 3$



PAGE :
DATE : / /

CHAPTER:09
OPEN GL

- ✓ – Describe open GL
- ✓ – What is called buck function. Explain any fewof them.
- – Draw a line, rectangle, circle, ellipse using open GL commands.
- ✓ – Explain about GLUTE - open GL application.
- – Lightening condition in open GL.