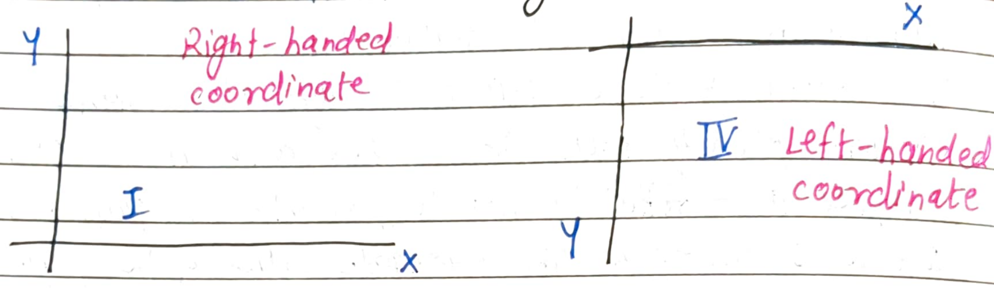
CG

Unit 5 – 2 Dimensional Viewing.

The Coordinate System

1] If a counter clockwise 90° rotation about the origin aligns with the positive x-axis and the positive y-axis then the coordinate system is called right-handed system otherwise it is left-handed system.

2] 

World Coordinate System (WCS)

1] The world coordinate system is the right-handed coordinate system.

2] The object which is in the object space is described in WCS.

3] This model represents the object in physical units of length.

4] Theoretically, WCS is infinite in extent.

Device coordinate System (DCS)

1] The coordinate system that corresponds to the device or workstation where the picture of the object is to be displayed.

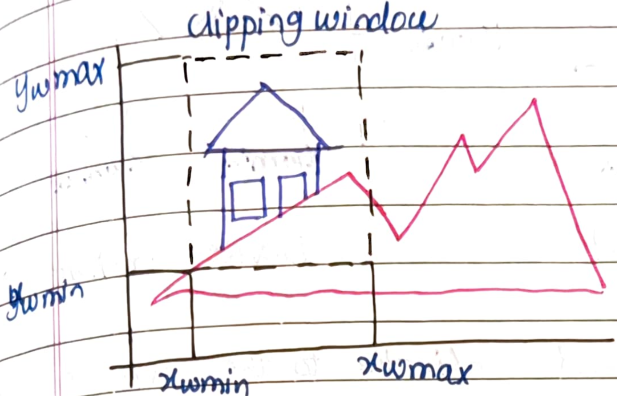
2] ~~call~~ It can be left-handed or a right-handed coordinate system depending upon the workstation.

Normalized Device Coordinate System (NDCS)

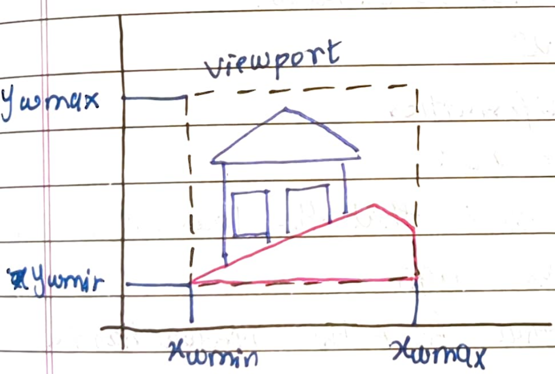
It is a right-handed coordinate system in which the display area of the virtual display device corresponds to the unit Square.

Concept of window and viewport

Consider a house described in WCS.



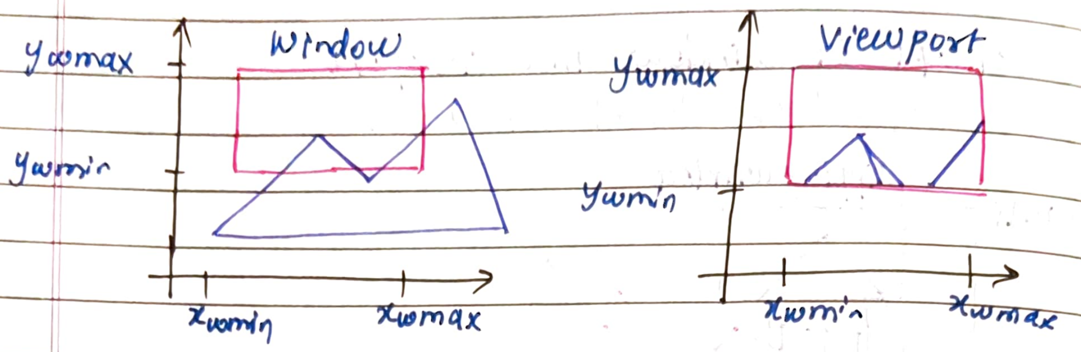
1] A world coordinate area selected for display is called a window.
2] The window can be rectangular or of any orientation.
3] The window can be directly mapped onto the display area or onto a subregion of display device.
4] The window defines what is to be displayed.



1] An area on the display device to which a window is mapped is called a viewport.
2] It defines where it is to be displayed.

1] If a window is changed, a different part of the object is displayed at the same position of the screen.
2] If a viewport is changed, the same portion of the object model is displayed at different positions of the screen.
3] Viewing Transformation: The mapping of a part of the world coordinate scene to device coordinate scene.

Window to viewport Tranformation Algorithm



World-coordinates | Device Coordinates.

Algorithm

The mapping of the world coordinates to the device coordinates is called viewing transformation. It is the transformation from window to viewport. The viewing transformation is formed by:

1] The Normalized transformation that maps the world coordinates to the normalized device coordinates.

2] The Workstation transformation that maps the normalized world coordinates to device coordinates.

3] ~~The following~~ $V = NW$
viewing transformation

3] Steps for transformation are:

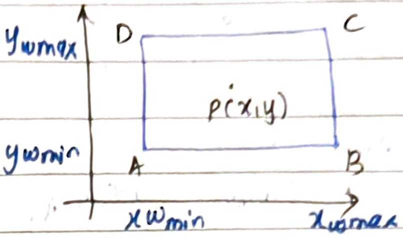
(i) Translate the object ~~to~~ so that the lower left corner of the object is on the origin.

(ii) Scale the object so that ~~it is~~ the window has the dimensions of the viewport.

(iii) Translate the object so that the scaled window area is ~~at the~~ positioned to the viewport.

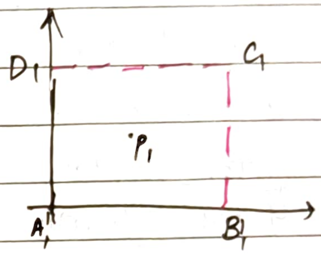
Steps

I] Initial position of the window.



II] Translate (xw_{min}, yw_{min}) to the origin. Transformation matrix is:

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

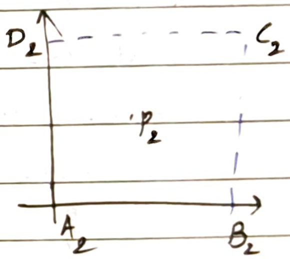


III] Scale wrt origin

Scaling matrix, $S_x = \dfrac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$

$$S_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

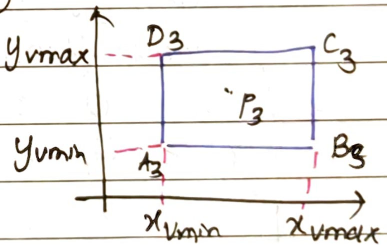
$= \dfrac{\text{viewport y extent}}{\text{window y extent.}}$



IV] Translate to (xv_{min}, yv_{min})

Translation matrix:

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ xv_{min} & yv_{min} & 1 \end{bmatrix}$$

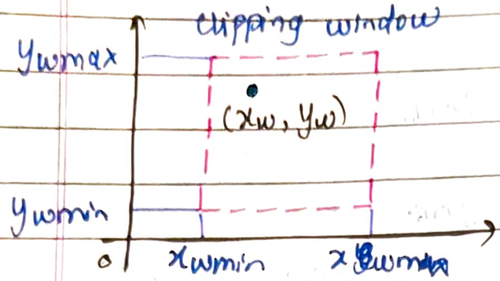


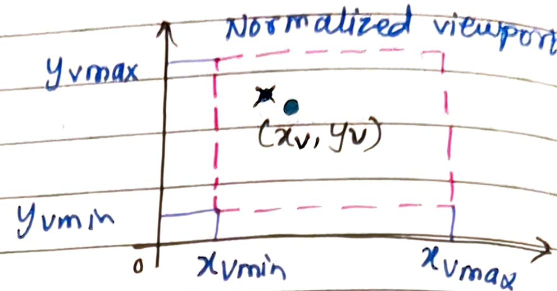
The composite transformation matrix would be:

$$V = T_1 S T_2$$

& $P' = V \cdot P$.

$P(x,y)$ is transformed to $P'(x', y')$.





Maintain the relative size and position between the clipping window & viewport.

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

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

1) Perform window to viewport for the point (20,15). Assume that x_{wmin}, y_{wmin} is (0,0), x_{wmax}, y_{wmax} is (100,100). x_{vmin}, y_{vmin} is (5,5) and x_{vmax}, y_{vmax} is (20,20). Find the value of x and y in viewport.

$$S_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} = \frac{20-5}{100-0} = \frac{15}{100} = \frac{3}{20}$$

$$S_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} = \frac{20-5}{100-5} = \frac{3}{20}$$

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3/20 & 0 & 0 \\ 0 & 3/20 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\frac{x_v - 5}{20 - 5} = \frac{20 - 0}{100 - 0}$$

$$x_v - 5 = \frac{\cancel{20}}{\cancel{100}_{5}} \times \cancel{15}^{3}$$

$$\therefore x_v = 8$$

Viewing Pipeline

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

$$\frac{y_v - 5}{20 - 5} = \frac{\cancel{8.}\,15 - 0}{100 - 0}$$

$$y_v - 5 = \frac{3}{\cancel{20}\,4} \times \cancel{15}\,3$$

$$y_v = \frac{9}{4} + 5 = \frac{29}{4} = 7.25$$

Viewing Pipeline

MC → [construct world-coordinate scene using the modelling coordinate transformation] → WC → [Convert world coordinates to ~~normal~~ veiwing coordinates] → VC → [Map viewing coordinates to normal viewing coordinates using window to viewport specification ~~transformation~~] → NVC → [Map normal viewing coordinates to the device coordinates] → DC

1] The view pipeline is the a series of transformations which are passed by geometric data to end up as image data being displayed on a device.

2] MC: ~~when~~ The coordinates in which indivial objects (models) are created are called modelling coordinate.

3] WC: When several objects are assembled into one ~~se~~ scene, they are world coordinates.

4] VC: After transformation into the coordinate system of the camera (viewer), they become viewing coordinates.

5] NVC: Their projection onto a common plane (window) gives independent normalized coordinates.

6] DC: Finally, after mapping those NVC to a specific device, we get device coordinates.

Clipping Operations

1] Point Clipping

Any point is inside the window if the following inequalities are satisfied. Let the point be $P(x, y, z)$:

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

2] Cohen-Sutherland Line clipping Algorithm

1001	1000	1010
0001	0000 Window	0010
0101	0100	0110

$\boxed{TBRL}$

Step 1: Take the input of the line endpoints (x_1, y_1) and (x_2, y_2) and window coordinates (x_{wmin}, y_{wmin}), (x_{wmax}, y_{wmax}).

Step 2: Determine the four-bit region code for the endpoints of the line (refer diagram), it can be anything depending upon which of the nine regions of the plane it lies on.

Step 3:

(i) If the region code for both the end points is (0, 0, 0, 0) that means it lies inside the window, display it.

(ii) If the LOGICAL AND of the endpoints codes is not

(0,0,0,0) the segment is not visible discard it.

(iii) If the LOGICAL AND of the endpoints codes is (0,0,0,0) the line segment is a clipping candidate.

Step 4: Determine the intersecting boundary:

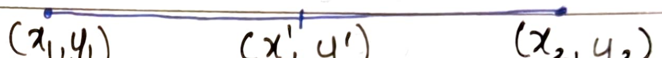
If (T=1) intersect with $y = y_{wmax}$

If (B=1) intersect with $y = y_{wmin}$

If (L=1) intersect with $x = x_{wmin}$

If (R=1) intersect with $x = x_{wmax}$.

Step 5: The eqn. of the line passing through (x_1, y_1) and (x_2, y_2)



$$\frac{x' - x_1}{x_2 - x_1} = \frac{y' - y_1}{y_2 - y_1}$$

$$x' - x_1 = \frac{x_2 - x_1}{y_2 - y_1}(y' - y_1)$$

$$x' = x_1 + \frac{1}{m}(y' - y_1).$$

[where $y' = y_{wmin}$ or y_{wmax}]

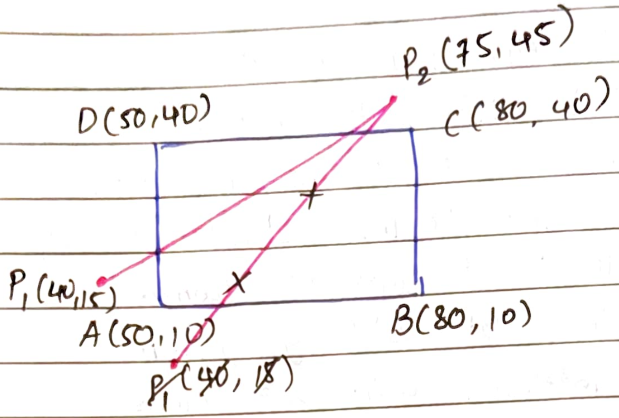
$$y' = y_1 + m(x' - x_1)$$

[where $x' = x_{wmin}$ or x_{wmax}].

Step 6: Go to step 2.

NUMERICALS

1) Use CSA Algorithm to clip a line $P_1(40, 15)$ and $P_2(75, 45)$ passing through the window having coordinates $A(50, 10)$, $B(80, 10)$, $C(80, 40)$, $D(50, 40)$.



$P_1 : 0010$ $\qquad$ $P_2 : 1000$

Logical AND

$$\begin{array}{r} 0010 \\ 1000 \\ \hline 0000 \end{array}$$

P_1

$L = 1$

$\therefore x = x_{wmin} = 50$

$\cancel{x' = x_1 + \frac{1}{m}(y' - y_w)}$

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

$$= 15 + \left(\frac{45-15}{75-40}\right)(50-40)$$

$$= 15 + \frac{\cancel{30}^{6}}{\cancel{35}_{7}} \times 10$$

$= 23.5714$ $\qquad$ $I_1 = (50, 23.5714)$

P_2

$T = 1$

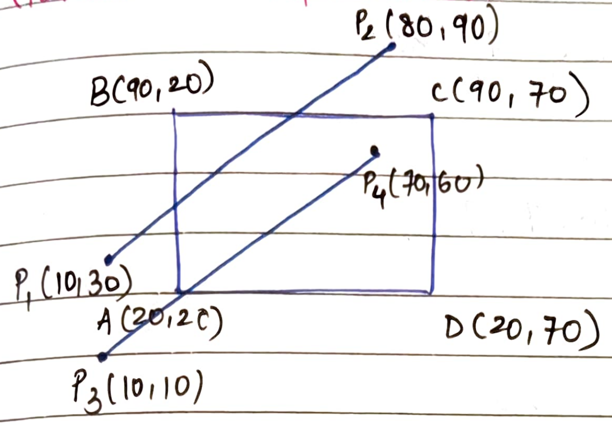
$y = y_{wmax} = 40.$

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

$$= 40 + \frac{35}{30}(40-15)$$

$= 69.167$ $\qquad$ $I_2 = (69.167, 40)$

2] Let ABCD be the rectangular window with A(20, 20), B(90, 20), C(90, 70), D(20, 70). Find the region code for the end points and use CSA to clip $P_1 P_2$ where P_1 (10, 30) and P_2 (80, 90) and $P_3 P_4$ where P_3 (10, 10) and P_4 (70, 60).



P_1 : 0010 $\quad$ P_2 : 1000

P_3 : 0110 $\quad$ P_4 : 0000

Logical AND

$P_1 P_2$ $\quad$ $P_3 P_4$

0010 $\quad$ 0110

1000 $\quad$ 0000

0000 $\quad$ 0000

P_1

$L = 1$

$x = x_{wmin} = 20$

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

$= 30 + \left(\frac{\cancel{8}90-30}{80-10}\right)(20-10)$

$= 30 + \frac{60}{7\cancel{0}} \times 1\cancel{0}$

$= 38.5714$

$I_1 = (20, 38.5714)$

P_2

$T = 1$

$y = y_{wmax} = 70$

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

$= 10 + \frac{7\cancel{0}}{6\cancel{0}} \times (70 - 30)$

$= 56.67$

$I_2 = (56.67, 70)$.

P_3

$B = 1$

$\therefore y = y_{wmin} = 20$

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

$= 10 + \left(\frac{70-10}{60-10}\right)(20 - 10)$

$= 10 + \frac{6\cancel{0}}{5\cancel{0}} \times 1\cancel{0}$

$= 22$

$L = 1$

$x = x_{wmin} = 20$

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

$= 10 + \frac{50}{60}(20 - 10)$

$= 18.33$

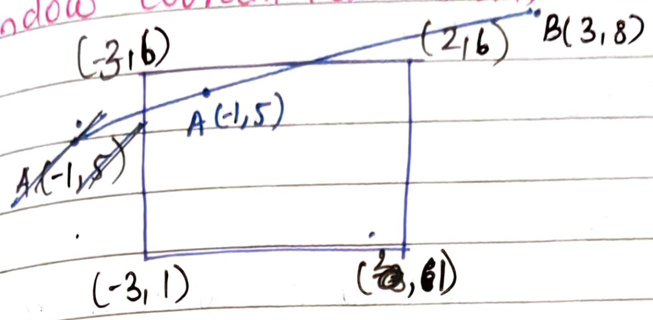
$I_3 = (22, \cancel{18.33}\ 20)$.

P_4

Code : 0000

∴ The ~~line~~ point lies inside the window.

3] Clip a line A(-1,5) and B(3,8) using CSA with window coordinates (-3,1) and (2,6)



A : 0000 B : 1001

Logical AND

0000
1001
0000

P_1

$L = 1$

$x = x_{wmin} = -3$

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

$= -1 + \left(\frac{6-5}{2+1}\right)(-3+1)$

$= -1 + \frac{1}{3} \times (-2)$

$= -1.67$

$I_1\ (-3, -1.67)$

$P_1 = 0000$

∴ Lies inside the window

$I_1 = (-1,5)$

P_2

$T = 1$

$y = y_{wmax} = 6$

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

$= -1 + \left(\frac{3+1}{8-5}\right)(6-5)$

$= -1 + \frac{4}{3} = 0.333$

$k = 1$

$x = x_{wmax} = 2$

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

$= 5 + \frac{3}{4}(2+1)$

$= 7.25$

7.25 can't be y-coordinate since it would lie outside the window

$\therefore I_2 = (0.33, 6)$

Liang Barsky Line Clipping Algorithm

Step 1: Input coordinates of the end points of lines and the window coordinates: x_{wmin}, y_{wmin}, x_{wmax}, y_{wmax}.

Step 2: Calculate p_k and q_k for $k = 1, 2, 3, 4$

$p_1 = -\Delta x \qquad q_1 = x_1 - x_{wmin} \quad (L)$

$p_2 = \Delta x \qquad q_2 = x_{wmax} - x_1 \quad (R)$

$p_3 = -\Delta y \qquad q_3 = y_1 - y_{wmin} \quad (B)$

$p_4 = \Delta y \qquad q_4 = y_{wmax} - y_1 \quad (T)$

Step 3: (i) If $p_k = 0$ { then the line is parallel to the kth boundary.

(ii) If $q_k < 0$ {

Then the line is outside the boundary

Discard the line segment

STOP

}

(iii) If $q_k \geq 0$ then the line is inside the parallel boundary. }

Step 4: Calculate $r_k = \frac{q_k}{p_k}$ for all $k = 1, 2, 3, 4$

Step 5: Determine U_1 for all $P_k < 0$ from the set containing $\{r_k, 0\}$.

select r_k for all $P_k < 0$

then $U_1 = \{r_k, 0\}_{max}$

Determine U_2 for all $P_k > 0$ from the set $\{r_k, 1\}$.

select r_k for all $P_k > 0$

Then $U_2 = \{r_k, 1\}_{min}$

Step 6: If $U_1 > U_2$ Then

{

The line is completely outside the boundary

discard the line segment

STOP

}

Step 7: Calculate end points of clipped line

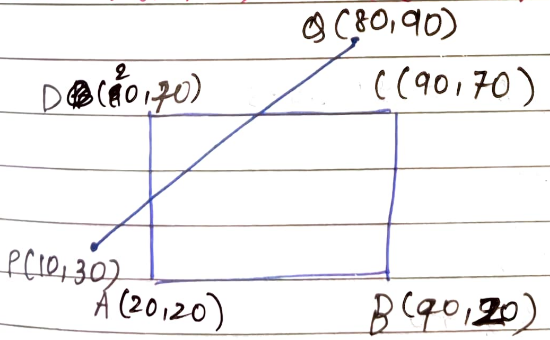
$x' = x_1 + u_1 \Delta x$ $\quad$ $x'' = x_1 + u_2 \Delta x$

$y' = y_1 + u_1 \Delta y$ $\quad$ $y'' = y_1 + u_2 \Delta y$.

Step 8: Display $I_1 I_2$. STOP

NUMERICALS

1) The window size is given A(20,20), B(90,20), C(90,70) and D(20,70). The line coordinates are P(10,30) and Q(80,90). Find the intersection points.



PAGE No

DATE / /

~~Step 1:~~

Step 1: $x_{wmin} = 20$; $y_{wmin} = 20$

$x_{wmax} = 90$; $y_{wmax} = 70$

$\Delta x = 80 - 10 = 70$

$\Delta y = 90 - 30 = 60$

Step 2:

$p_1 = -70$	$q_1 = 10 - 20 = -10$
$p_2 = 70$	$q_2 = 90 - 10 = 80$
$p_3 = -60$	$q_3 = 30 - 20 = 10$
$p_4 = 60$	$q_4 = 70 - 30 = 40$

Step 3:

$r_k = \frac{q_k}{p_k}$

$p_k < 0$	$p_k > 0$
p_1 p_3	p_2 p_4

$$u_1 = \left\{ \frac{-10}{-70}, \frac{10}{-60}, 0 \right\}_{max}$$

$$= \frac{1}{7}$$

$$u_2 = \left\{ \frac{80}{70}, \frac{40}{60}, 1 \right\}_{min}$$

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

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

$$= 10 + \frac{1}{7} \times 70 = 20$$

$$y' = 30 + \frac{2}{7} \times 60 = \cancel{50}\ 38.57$$

$$I_1 = (20, \cancel{50}\ 38.57)$$

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

$$= 10 + \frac{2}{3} \times 70$$

$= 56.67$

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

$$= 30 + \frac{2}{3} \times 60 = \cancel{90}\ 70.$$

$$I_2 = (56.67, \cancel{90})$$

Applications of Clipping Operation

1] Identifying the visible surfaces in 3D views.

2] Extracting the ~~defined~~ part of a defined scene for viewing.

3] Creating objects using solid-modelling procedures.

4] Antialiasing line segments or line boundaries.

5] Displaying multiwindow environment.

6] Depending on the application, the clip window can be a polygon or even a curved ~~surface~~ boundary.

7] Drawing and painting options that allow a ~~window to be~~ part of a picture to be moved for copying, ~~pasting~~ moving, erasing, or duplicating.

Advantages of CSA Clipping

1] Easily extended to 3 dimension by adding 2 bits to the outcode for z-axis.

2] Calculations can be then reduced to the intersection of a line with plane.

3] Algorithm is most most efficient when most segments can be trivially accepted or trivially rejected.