Viewing Transformation

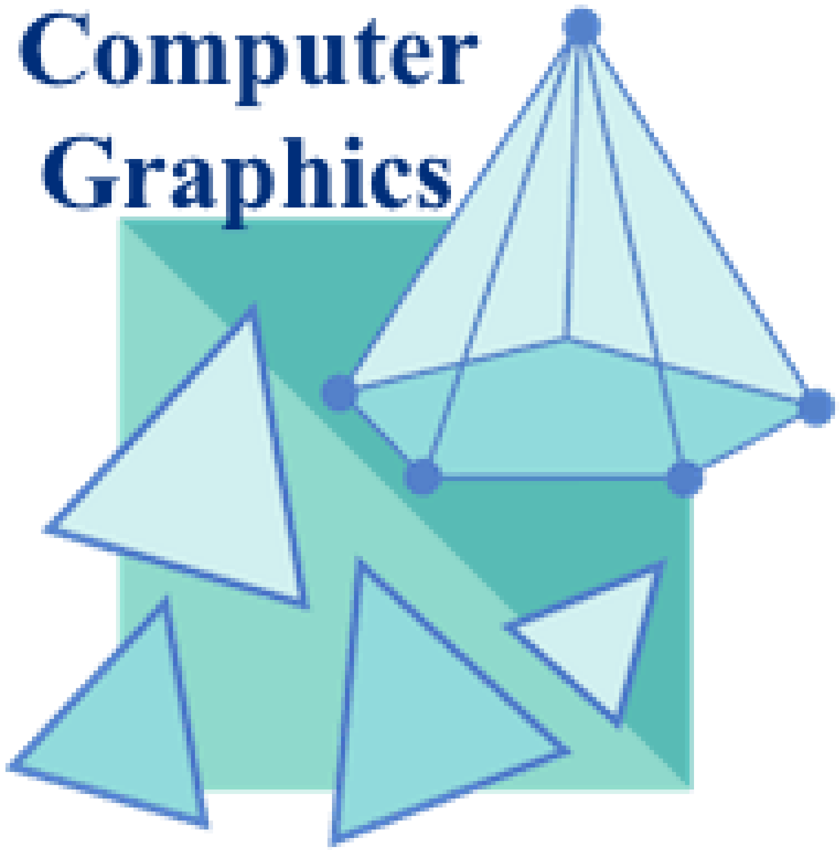
Course Title: Computer Graphics

Course Code: CSE – 413

Instructor: Khandaker Jannatul Ritu, Lecturer, Dept. of CSE(BAIUST)

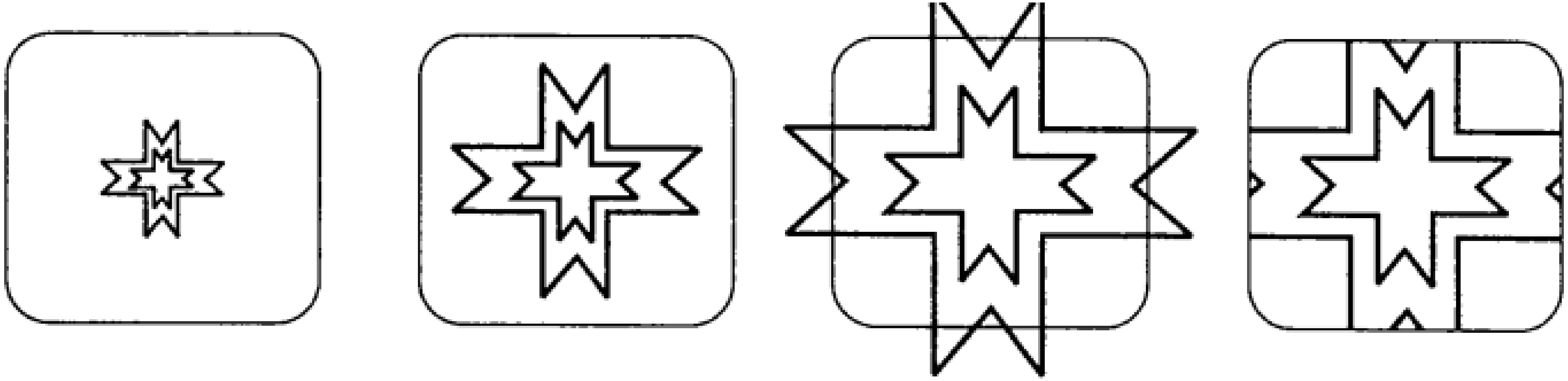
Outlines

- Introduction to Viewing Transformation
- Window port and Viewport in Computer Graphics
- Window to Viewport Transformation / Viewing transformation / windowing transformation
- Example problems
- Viewing transformation process
- Normalization transformation
- Workstation transformation
- 2D Viewing-Transformation Pipeline
- Example problems
- Difference Between Window Port And Viewport
- **Computer Graphics Zooming**
- **Computer Graphics Panning**
- **Inking**
- **Scissoring**



❑ Introduction to Viewing Transformation:

- ✓ Typically, a graphics package allows us to specify which part of a defined picture is to be displayed and where that part is to be displayed on the display device. Furthermore, the package also provides the use of the scaling, translation and rotation techniques to generate a variety of different views of a single picture.
- ✓ We can generate different view of a picture by applying the appropriate scaling and translation. While doing this, we have to identify the visible part of the picture for inclusion in the display image.
- ✓ This selection process is not straight forward. Certain lines may lie partly inside the visible portion of the picture and partly outside. These lines cannot be omitted entirely from the display image because the image would become inaccurate.
- ✓ The process of selecting and viewing the picture with different views is called **windowing**, and a process which divides each element of the picture into its visible and invisible portions, allowing the invisible portion to be discarded is called **clipping**.



Window port and Viewport in Computer Graphics

Capturing images from the real world and displaying them on the screen is an astonishing process, only if we do not know the underlying process. Here, we will be studying how the images are captured. This process is held by the Window port and Viewport in Computer Graphics.

Window Port

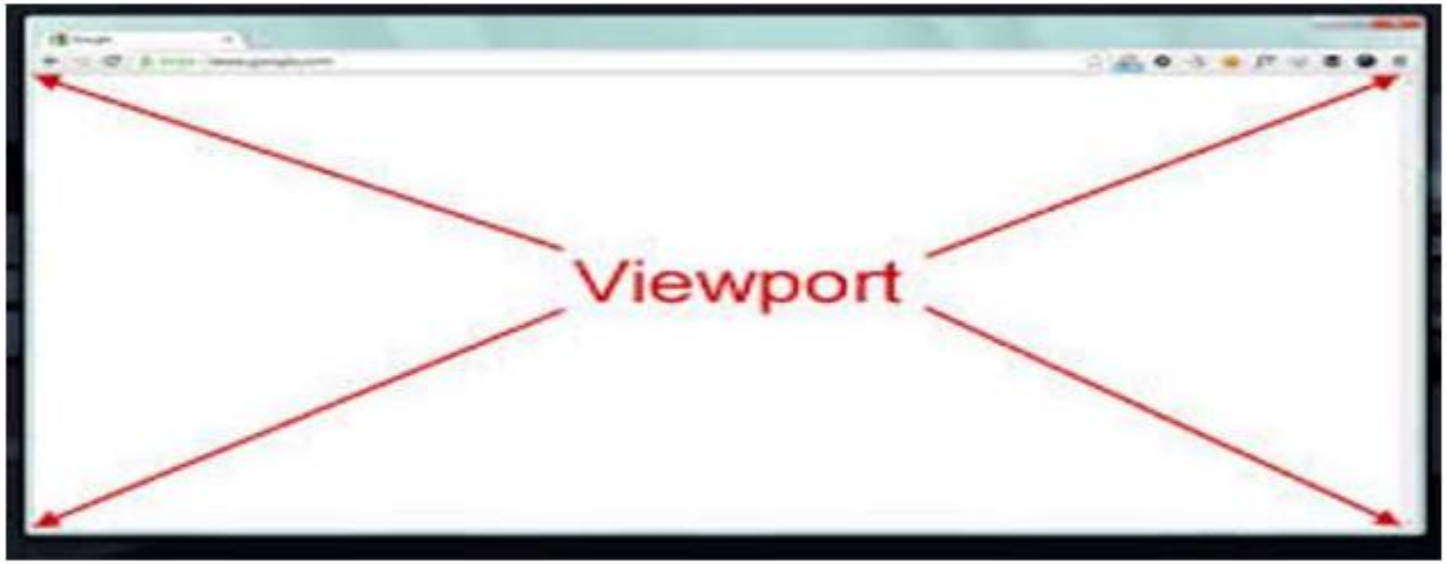
The window port can be confused with the computer window but it isn't the same. The window port is the area chosen from the real world for display. This window port decides what portion of the real world should be captured and be displayed on the screen.

So, the Window defines what is to be viewed. The widow port can thus be defined as,

- ✓ *"A world-coordinate area selected for display is called a window. A window defines a rectangular area in the world coordinates."*
- ✓ *"The Capability to show some part of an object in a window is known as windowing."*
- ✓ *"The rectangular area describes in the world coordinate system is called the window."*

<u>Viewport</u>

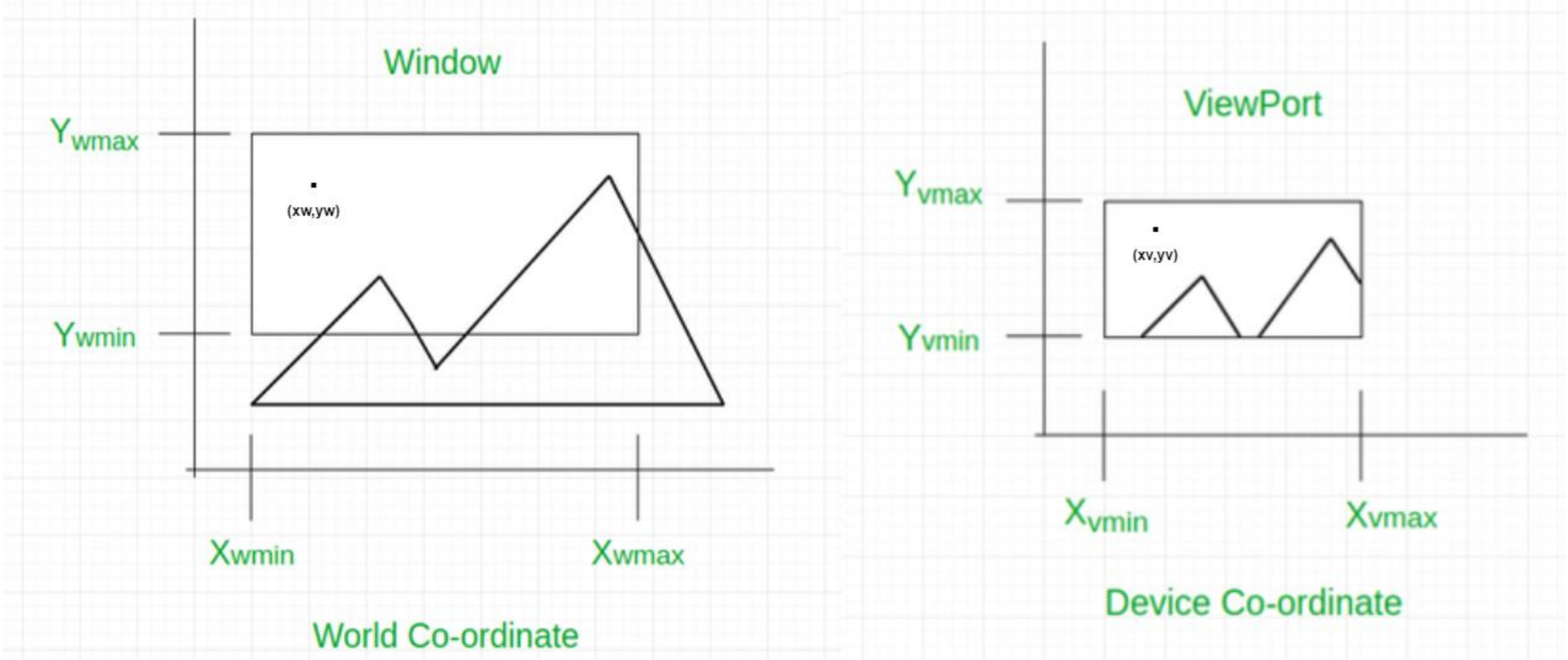
Now, the Viewport is the area on a display device to which a window is mapped. ***"The viewport can be defined as an area on the screen which is used to display the object".*** So, Viewport defines where is to be viewed. Thus, the viewport is nothing else but our device's screen. The viewport can thus be defined as follows:
"A viewport is a polygon viewing region in computer graphics. The viewport is an area expressed in rendering-device-specific coordinates, e.g. pixels for screen coordinates, in which the objects of interest are going to be rendered."



So, to display the image on the computer screen, we must map our window port to the viewport. The capture ratio of the window might not always be similar to or easily adjustable to the viewport. Thus, some necessary transformations adjustments like clipping and cropping are performed on the window.

❑ Window to Viewport Transformation /Viewing transformation/windowing transformation

Window to Viewport Transformation is the process of transforming 2D world-coordinate objects to device coordinates. Objects inside the world or clipping window are mapped to the viewport which is the area on the screen where world coordinates are mapped to be displayed.



General Terms:

•**World coordinate** – It is the Cartesian coordinate w.r.t which we define the diagram, like X_{wmin}, X_{wmax}, Y_{wmin}, Y_{wmax}

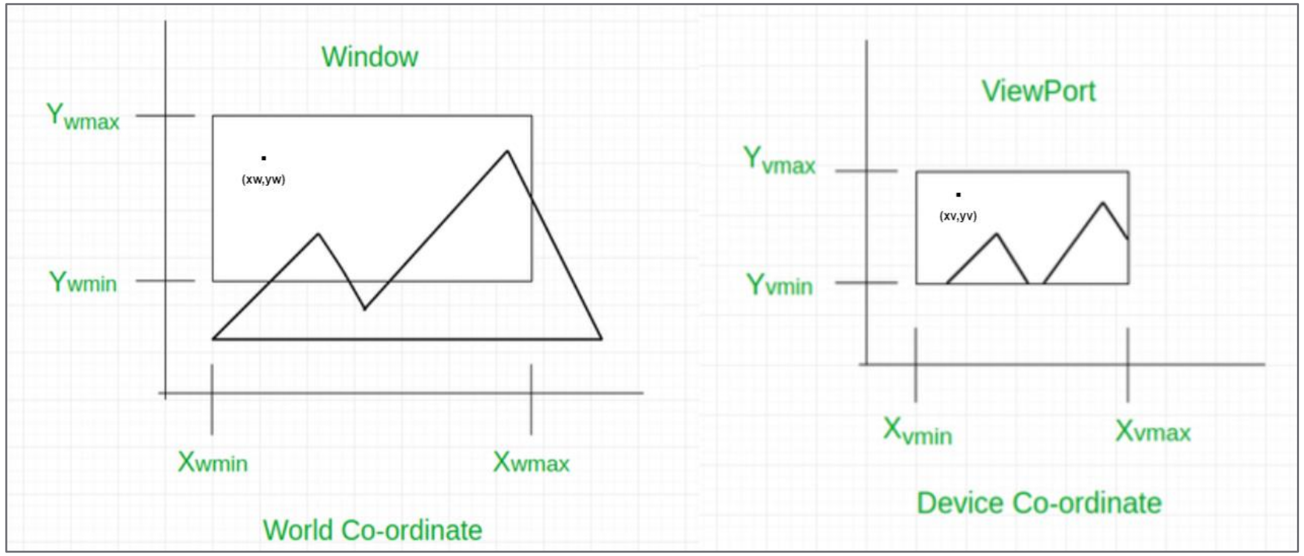
•**Device Coordinate** –It is the screen coordinate where the objects are to be displayed, like X_{vmin}, X_{vmax}, Y_{vmin}, Y_{vmax}

•**Window** –It is the area on the world coordinate selected for display.

•**ViewPort** –It is the area on the device coordinate where graphics is to be displayed.

Mathematical Calculation of Window to Viewport:

If (x_w, y_w) is a point on Window port. Then (x_v, y_v) is the Corresponding point on Viewport. What is the value of this viewport points (x_v, y_v) ?



<u>**Solution:**</u> We have to calculate the point $\mathbf{(x_v, y_v)}$

Normalized Point on Window $\left(\dfrac{X_w - X_{wmin}}{X_{wmax} - X_{wmin}}, \dfrac{Y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}} \right)$

Normalized Point on Viewport $\left(\dfrac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}}, \dfrac{Y_v - Y_{vmin}}{Y_{vmax} - Y_{vmin}} \right)$

Now the relative position of the object in Window and Viewport are same.

<u>***For x coordinate,***</u>

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

<u>***For y coordinate,***</u>

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

❖Considering scaling factor

calculating for x coordinate:

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

$$\Rightarrow (X_v - X_{vmin})\ (X_{wmax} - X_{wmin}) = (X_w - X_{wmin})\ (X_{vmax} - X_{vmin})$$

$$\Rightarrow X_v - X_{vmin} = (X_w - X_{wmin}) * \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}$$

$$\Rightarrow X_v = X_{vmin} + (X_w - X_{wmin})\ S_x$$

Where s_x is the scaling factor of x coordinate

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

calculating for y coordinate:

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

$$\Rightarrow (Y_v - Y_{vmin})\ (Y_{wmax} - Y_{wmin}) = (Y_w - Y_{wmin})\ (Y_{vmax} - Y_{vmin})$$

$$\Rightarrow Y_v - Y_{vmin} = (Y_w - Y_{wmin}) * \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}}$$

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

Where s_y is the scaling factor of y coordinate

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

Considering translation factor

calculating for x coordinate:

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

$$\Rightarrow (X_v - X_{vmin})(X_{wmax} - X_{wmin}) = (X_w - X_{wmin})(X_{vmax} - X_{vmin})$$

$$\Rightarrow X_v - X_{vmin} = \frac{(x_w - x_{wmin})(x_{vmax} - x_{vmin})}{(x_{wmax} - x_{wmin})}$$

$$\Rightarrow X_v = X_{vmin} + t_x$$

Where t_x is the translation factor of x coordinate

$$t_x = \frac{(x_w - x_{wmin})(x_{vmax} - x_{vmin})}{(x_{wmax} - x_{wmin})}$$

$$t_x = \frac{xw_{max}xv_{min} - xw_{min}xv_{max}}{xw_{max} - xw_{min}}$$

calculating for y coordinate:

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

$$\Rightarrow (Y_v - Y_{vmin})(Y_{wmax} - Y_{wmin}) = (Y_w - Y_{wmin})(Y_{vmax} - Y_{vmin})$$

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

$$\Rightarrow Y_v = Y_{vmin} + t_y$$

Where t_y is the translation factor of y coordinate

$$t_y = \frac{(Y_w - Y_{wmin})(Y_{vmax} - Y_{vmin})}{(Y_{wmax} - Y_{wmin})}$$

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

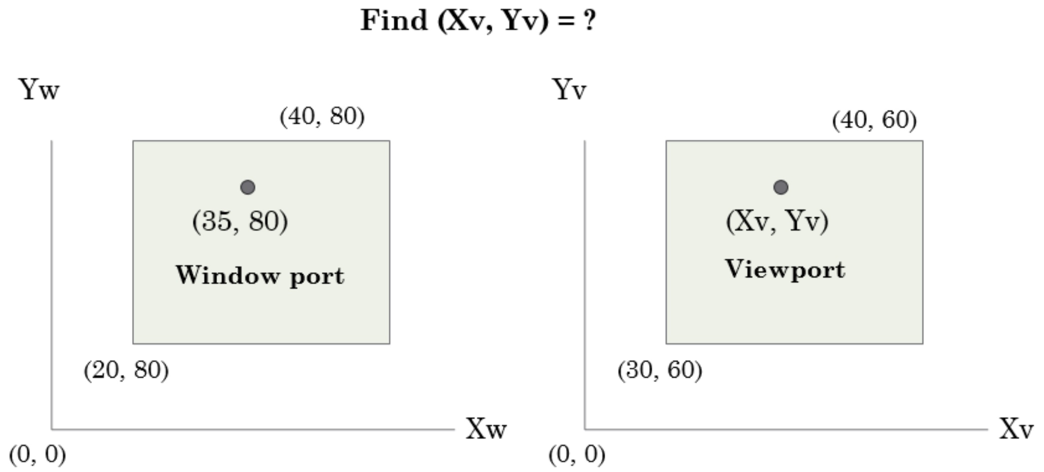
<u>Example-01</u>: If there is a coordinate (x_w, y_w) in the window port what will the corresponding coordinate value(x_v, y_v) in the viewport with respect to (a) scaling factor and (b) translation factor? Find the equation. [see previous equation.]

<u>Example-02:</u> If there is a coordinate point (30, 80) on the window port where the coordinate of lower left corner is (20, 80) and the upper right corner is (40, 80), then will be the corresponding coordinate point(x_v, y_v) on viewport where the coordinate of lower left corner is (30, 60) and the upper right corner is (40, 60),

<u>Solution:</u>

Let us assume,

- for window, $\mathbf{X_{wmin} = 20, X_{wmax} = 80, Y_{wmin} = 40, Y_{wmax} = 80}$.
- for viewport, $\mathbf{X_{vmin} = 30, X_{vmax} = 60, Y_{vmin} = 40, Y_{vmax} = 60}$.



- Now a point (X_w, Y_w) be (30, 80) on the window. We have to calculate that point on the viewport i.e **(X_v, Y_v)**.
- First of all, calculate the scaling factor of x coordinate S_x and the scaling factor of y coordinate S_y using the above-mentioned formula.

$$Sx = (60 - 30) / (80 - 20) = 30 / 60$$
$$Sy = (60 - 40) / (80 - 40) = 20 / 40$$

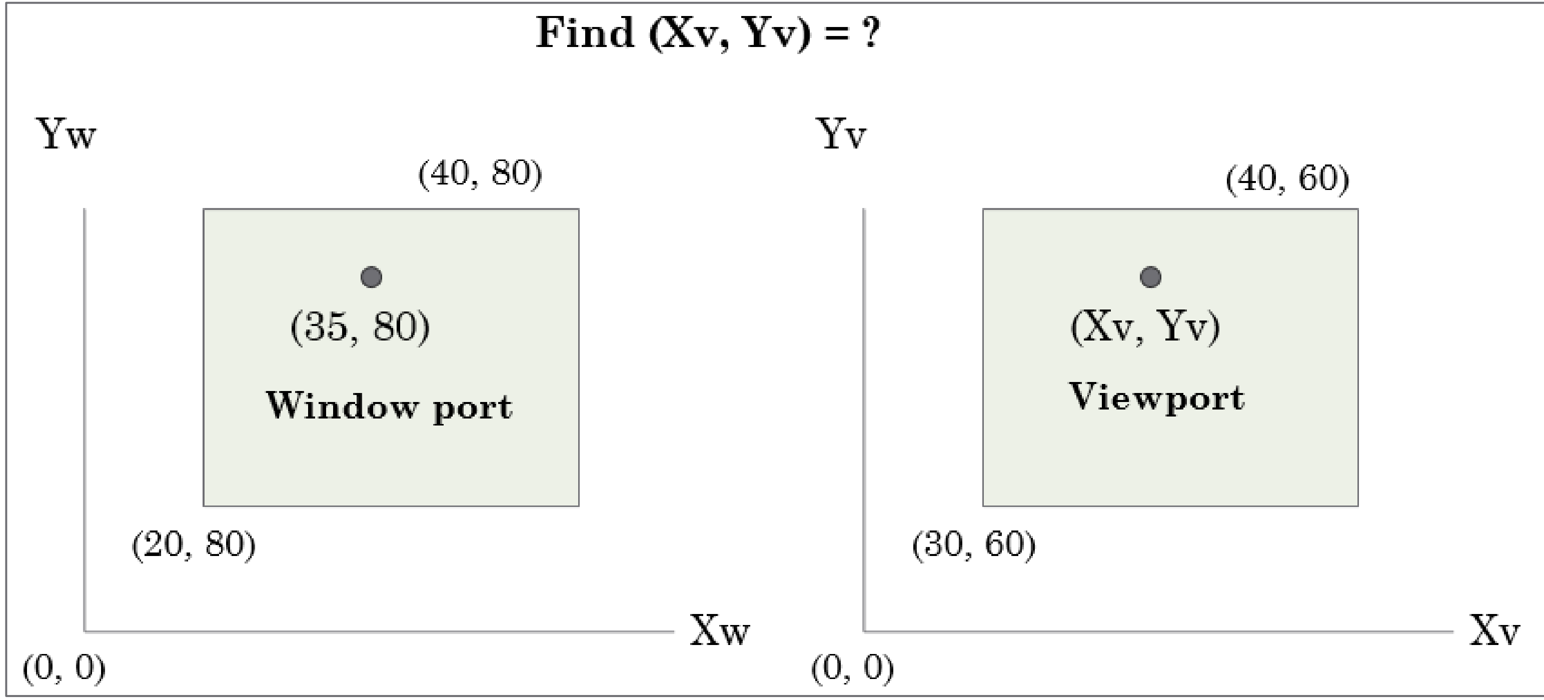
So, now calculate the point on the viewport (Xv, Yv).

$$Xv = 30 + (30 - 20) * (30 / 60) = 35$$
$$Yv = 40 + (80 - 40) * (20 / 40) = 60$$

So, the point on window (Xw, Yw) = (30, 80) will be (Xv, Yv) = (35, 60) on viewport.

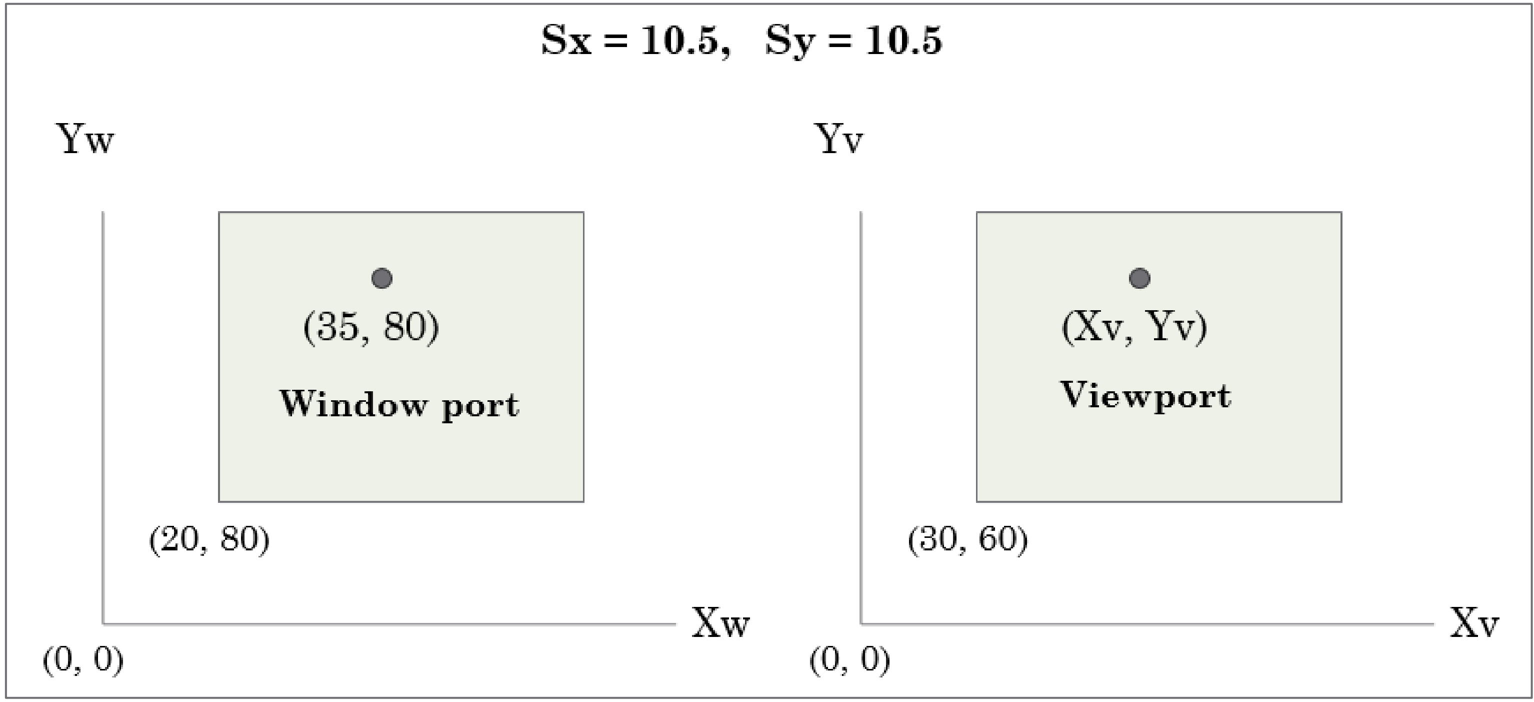
Example-03:

i) Find (Xv, Yv) = ?
ii) Find (Sx, Sy) = ?
iii) Find (tx, ty) = ?



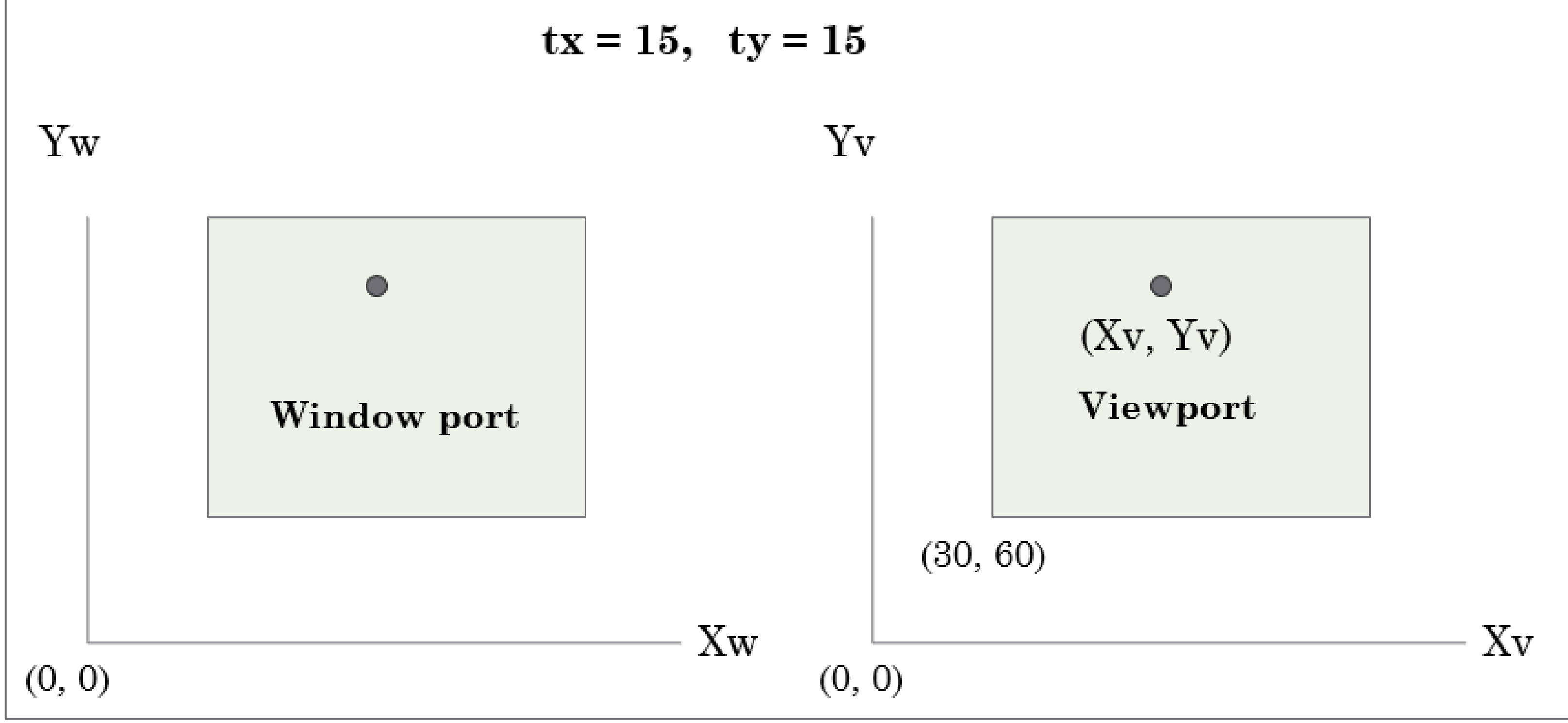
Example-04:

i) Find (Xv, Yv) = ?
ii) Find (tx, ty) = ?



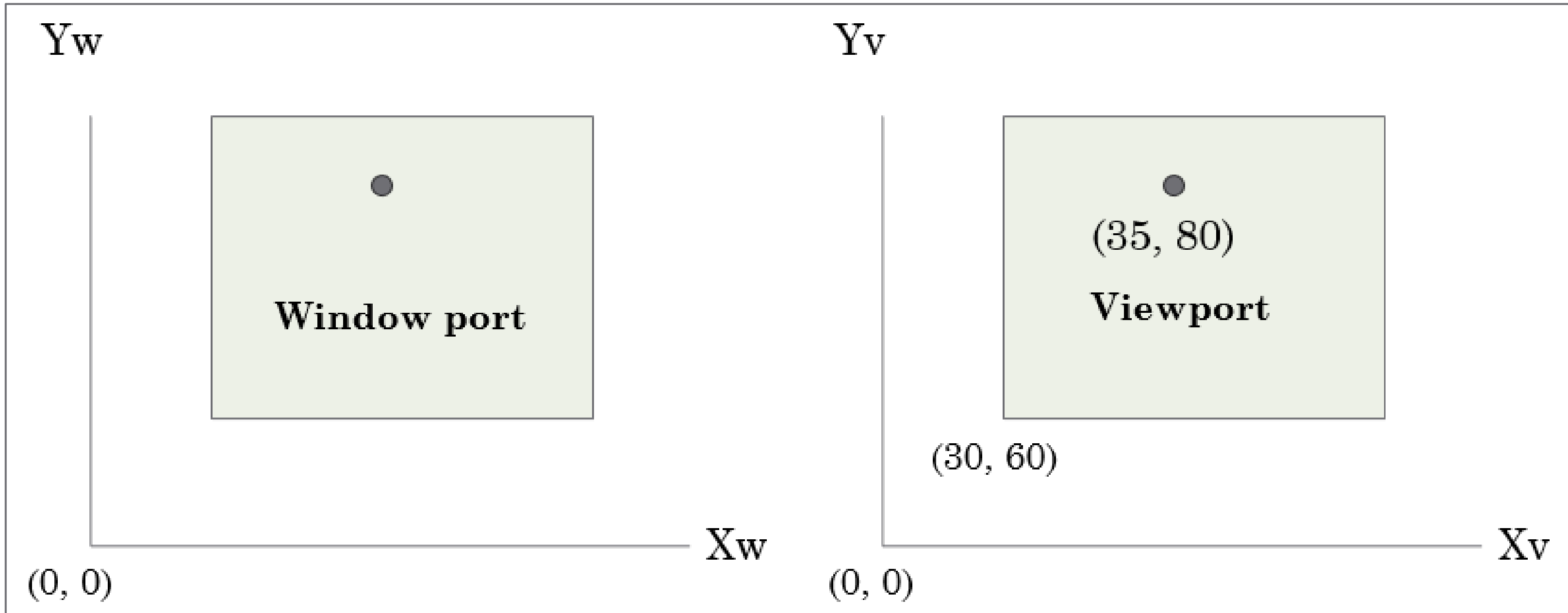
Example-05:

i) Find (Xv, Yv) = ?



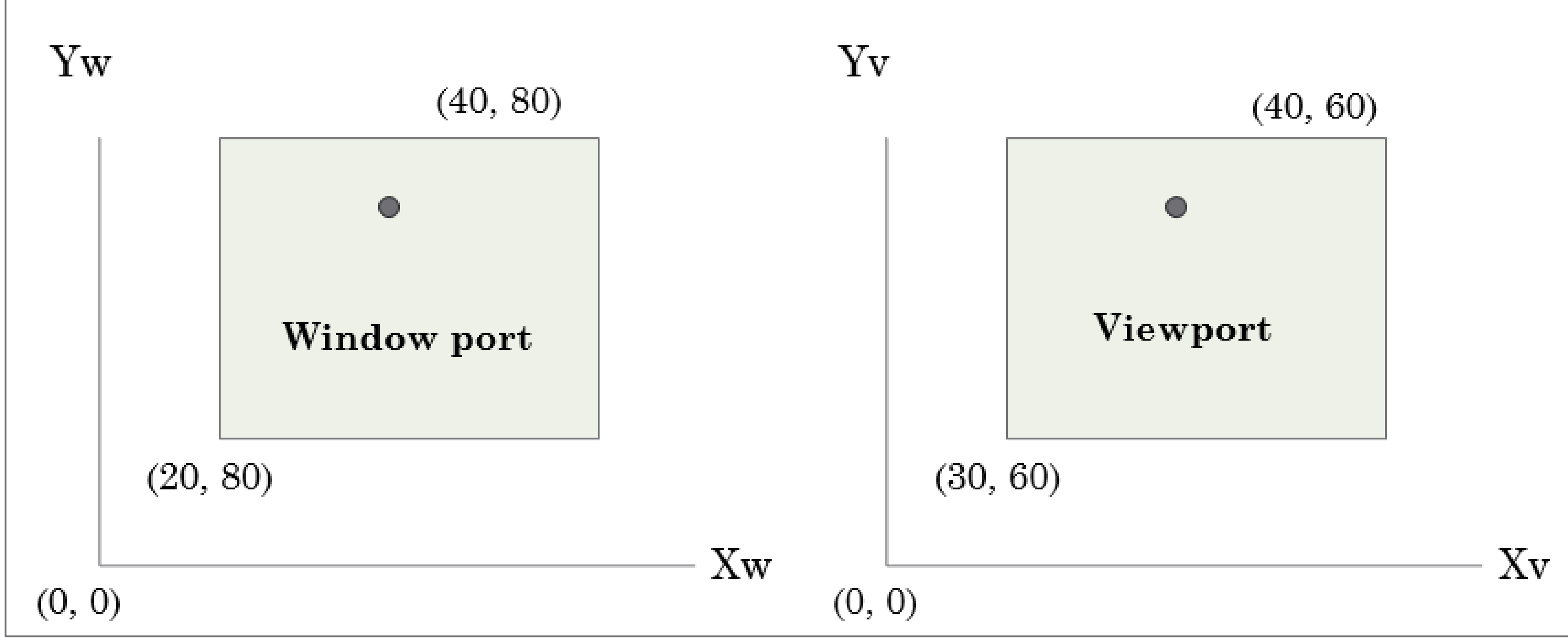
Example-06:

i) Find (tx, tx) = ?



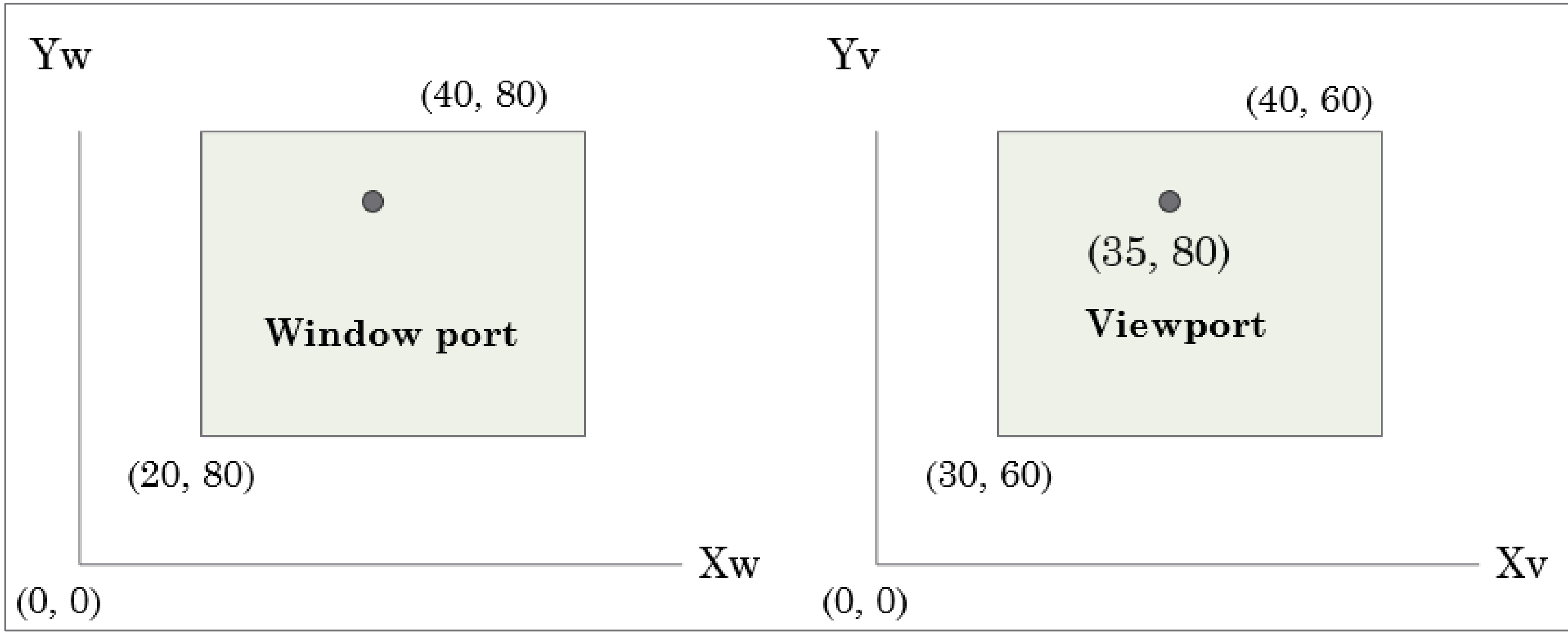
Example-07:

i) Find (Sx, Sy) = ?



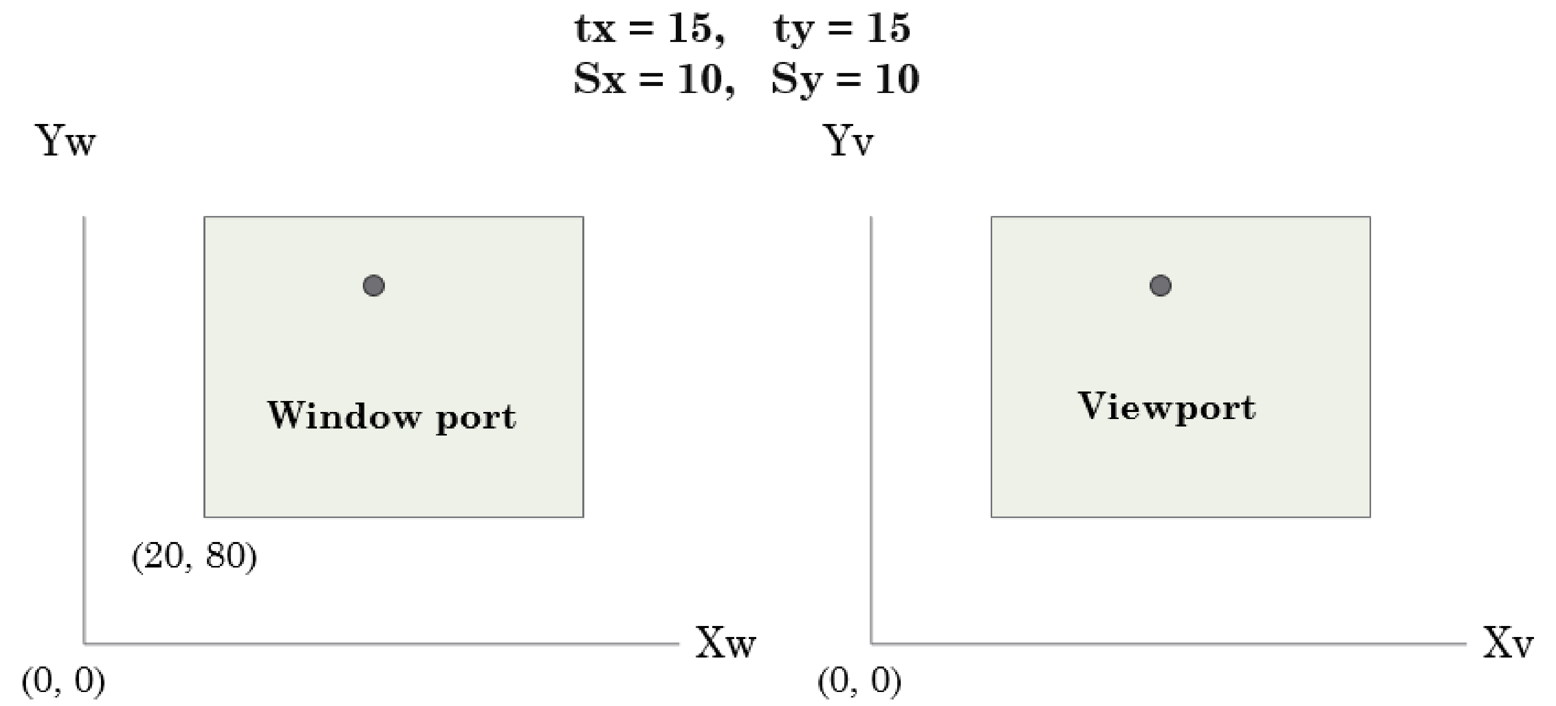
Example-08:

i) Find (Xw, Yw) = ?
ii) Find (Sx, Sy) = ?
iii) Find (tx, ty) = ?



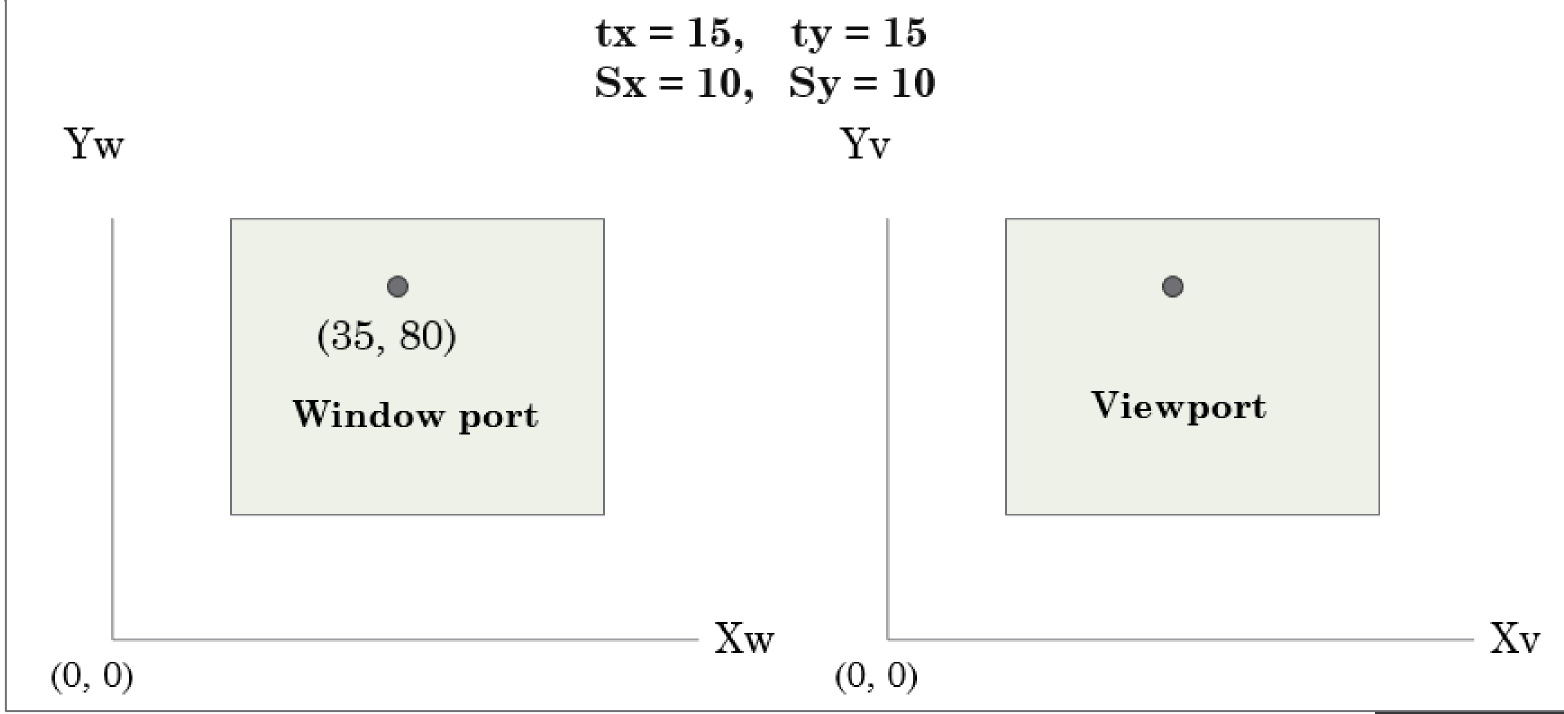
Example-09:

i) Find (Xw, Yw) = ?



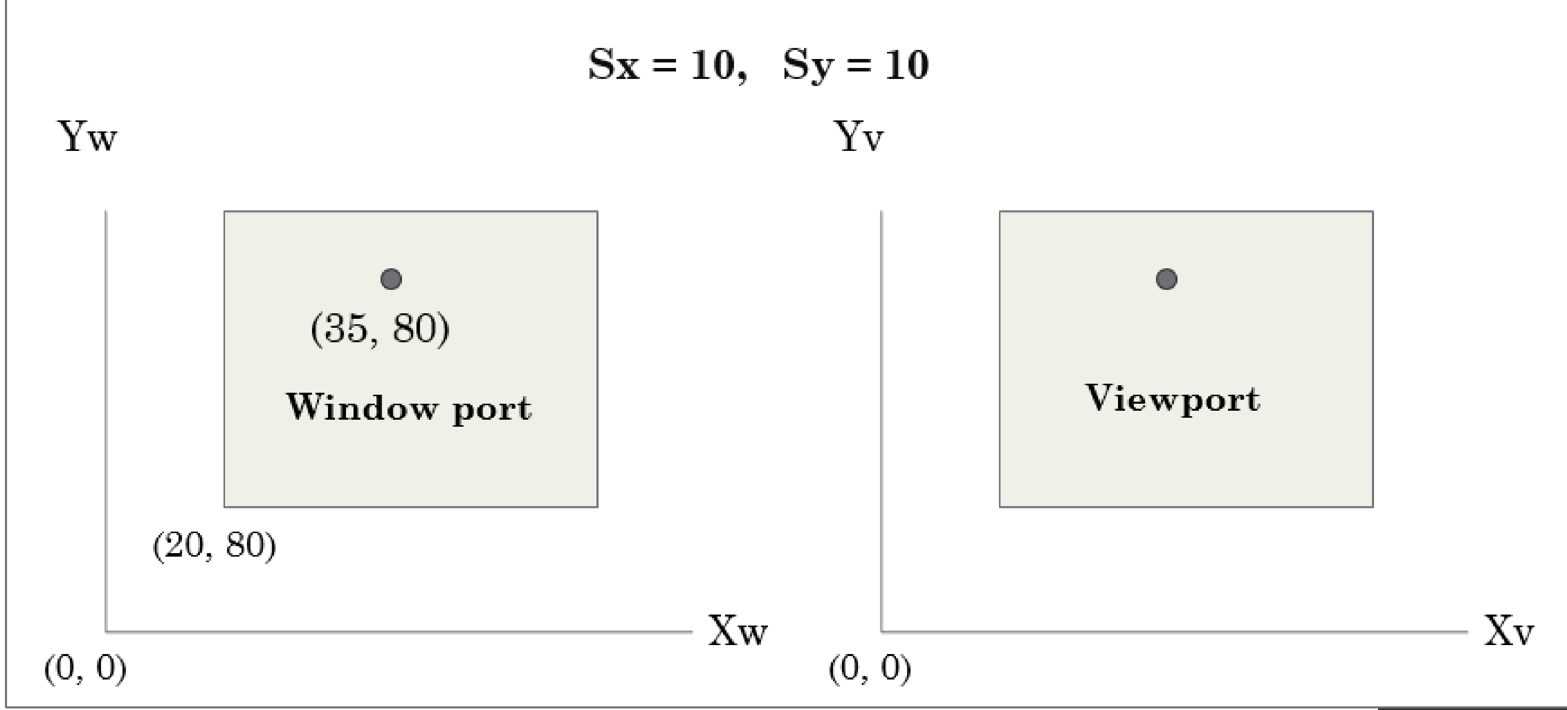
Example-10:

i) Find (Xwmin, Ywmin) = ?



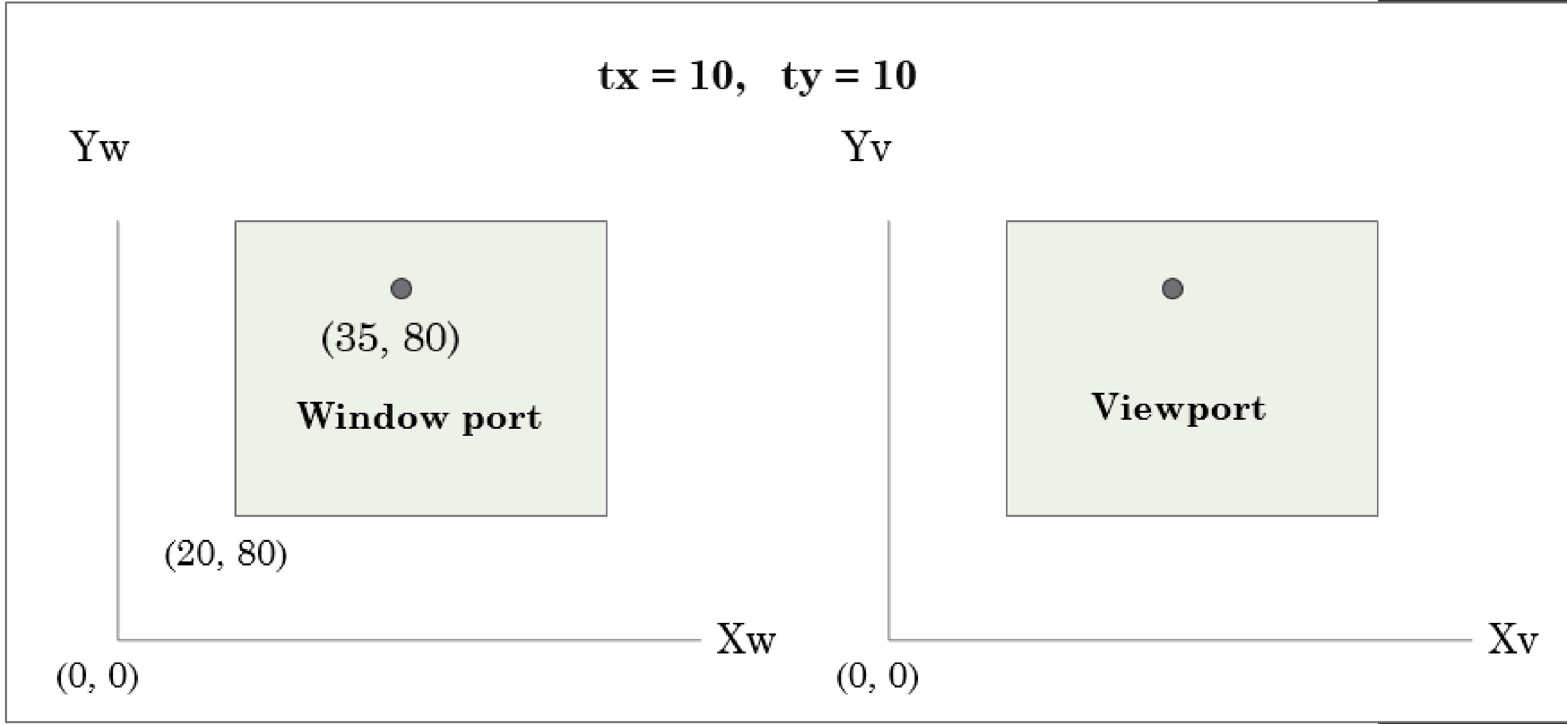
Example-11:

i) Find (tx, ty) = ?



Example-12:

i) Find (tx, ty) = ?



❑ Viewing transformation

- We know that the picture is stored in the computer memory using any convenient cartesian coordinate system, referred to as **world coordinate system (WCS)**. However, when picture is displayed on the display device it is measured in **physical device coordinate system (PDCS)** corresponding to the display device.
- Therefore, displaying an image of a picture involves mapping the coordinates of the points and lines that form the picture into the appropriate physical device coordinate where the image is to be displayed.
- This mapping of coordinates is achieved with the use of coordinate transformation known as **viewing transformation**.
- The viewing transformation which maps picture coordinates in the WCS to display coordinates in PDCS is performed by the following transformations :
  - **Normalization transformation (N) and**
  - **Workstation transformation (W)**

❑ Normalization transformation

- ✓ We know that, different display devices may have different screen sizes as measured in pixels. Size of the screen in pixels increases as resolution of the screen increases.
- ✓ When picture is defined in the pixel values then it is displayed large in size on the low resolution screen while small in size on the high resolution screen.
- ✓ To avoid this and to make our programs to be device independent, we have to define the picture coordinates in some units other than pixels and use the interpreter to convert these coordinates to appropriate pixel values for the particular display device. The device independent units are called the **normalized device coordinates**.
- ✓ In these units, the screen measures 1 unit wide and 1 unit length as shown in the Fig. 5.3. The lower left corner of the screen is the origin, and the upper-right corner is the point (1, 1). The point(0.5, 0.5) is the center of the screen no matter what the physical dimensions or resolution of the actual display device may be.

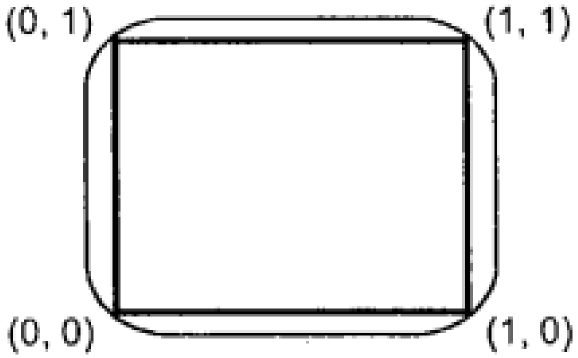


Fig. 5.3: Entire normalized device screen coordinates.

The interpreter uses a simple linear formula to convert the normalized device coordinates to the actual device coordinates.

$X = X_n \times X_w$ .... (5.1)

$Y = Y_n \times Y_H$ .... (5.2)

where

- X : Actual device x coordinate
- Y : Actual device y coordinate
- X_n : Normalized x coordinate
- Y_n : Normalized y coordinate
- X_w : Width of actual screen in pixels
- Y_H : Height of actual screen in pixels

The transformation which maps the world coordinate to normalized device coordinate is called **normalization transformation**. It involves scaling of x and y, thus it is also referred to as **scaling transformation**.

Normalized transformation

$$N = T \,.\, S \,.\, T^{-1}$$

$$T = \begin{vmatrix} 1 & 0 & Xvmin \\ 0 & 1 & Yvmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$S = \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$T^{-1} = \begin{vmatrix} 1 & 0 & -Xwmin \\ 0 & 1 & -Ywmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$N = \begin{vmatrix} 1 & 0 & Xvmin \\ 0 & 1 & Yvmin \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & -Xwmin \\ 0 & 1 & -Ywmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$N = \begin{vmatrix} Sx & 0 & Xvmin \\ 0 & Sy & Yvmin \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & -Xwmin \\ 0 & 1 & -Ywmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$N = \begin{vmatrix} Sx & 0 & -Xwmin*Sx + Xvmin \\ 0 & Sy & -Ywmin*Sy + Xvmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$N = \begin{vmatrix} Sx & 0 & Xvmin - Xwmin*Sx \\ 0 & Sy & Yvmin - Ywmin*Sy \\ 0 & 0 & 1 \end{vmatrix}$$

where,

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

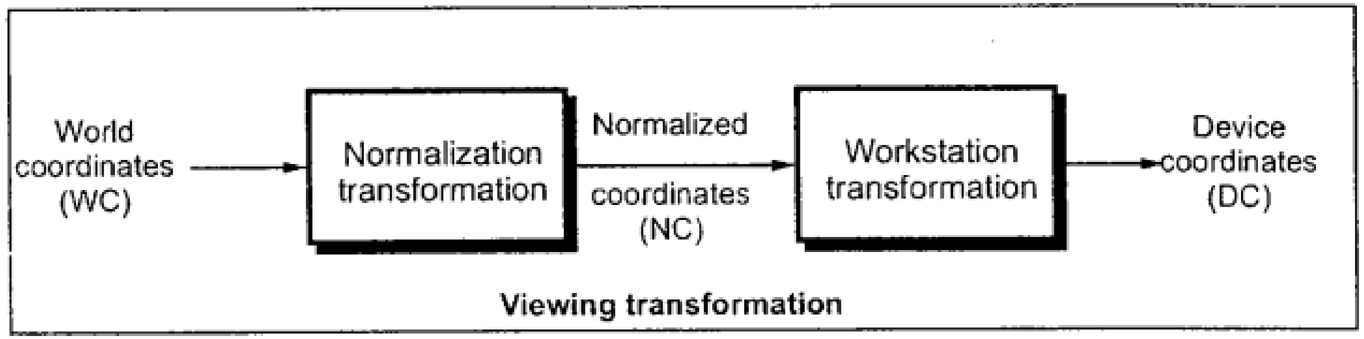
$$S_y = \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}}$$

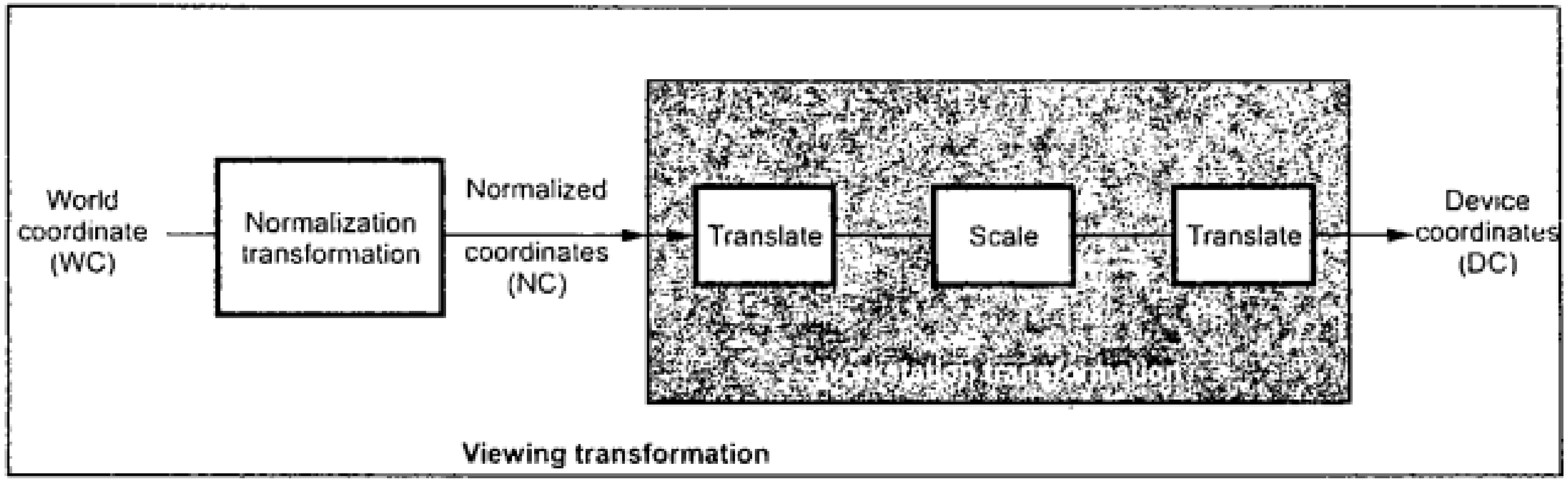
❑ Workstation transformation

- The transformation which maps the normalized device coordinates to physical device coordinates is called **workstation transformation**.

- The viewing transformation is the combination of normalization transformation and workstation transformations as shown in the Fig. 5.4. It is given as

$$V = N.W \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(5.3)}$$





❑ 2D Viewing-Transformation Pipeline

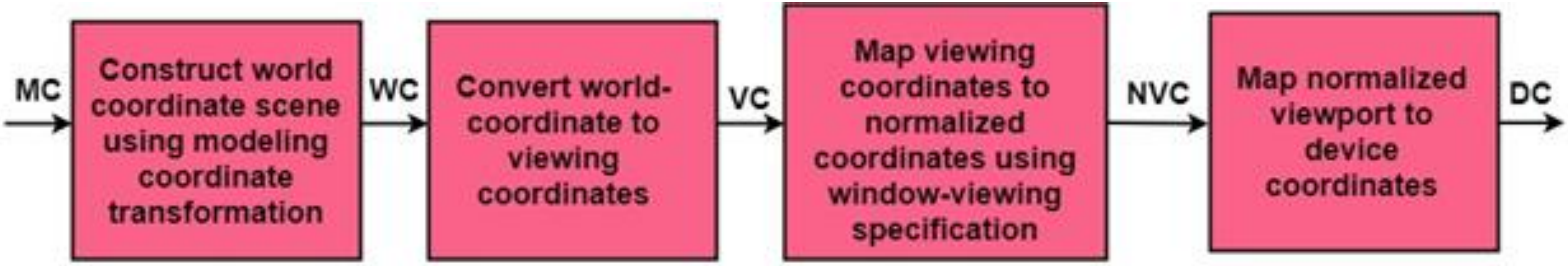


Fig: The two-dimensional viewing-transformation.

- ✓ First, we construct the scene in world coordinate using the output primitives and attributes.
- ✓ To obtain a particular orientation, we can set up a 2-D viewing coordinate system in the window coordinate plane and define a window in viewing coordinates system.
- ✓ Once the viewing frame is established, are then transform description in world coordinates to viewing coordinates.
- ✓ Then, we define viewport in normalized coordinates (range from 0 to 1) and map the viewing coordinates description of the scene to normalized coordinates.
- ✓ At the final step, all parts of the picture that (i.e., outside the viewport are dipped, and the contents are transferred to device coordinates).

- We know that **world coordinate system (WCS)** is infinite in extent and the device display area is finite. Therefore, to perform a viewing transformation we select a finite world coordinate area for display called a **window**. An area on a device to which a window is mapped is called a **viewport**. The window defines what is to be viewed; the viewport defines where it is to be displayed, as shown in the Fig. 5.5.

- The window defined in world coordinates is first transformed into the normalized device coordinates. The normalized window is then transformed into the viewport coordinate. This window to viewport coordinate transformation is known as **workstation transformation**. It is achieved by performing following steps :

1. The object together with its window is translated until the lower left corner of the window is at the origin.
2. Object and window are scaled until the window has the dimensions of the viewport.
3. Translate the viewport to its correct position on the screen.

Therefore, the workstation transformation is given as:

$$W = T \cdot S \cdot T^{-1}$$

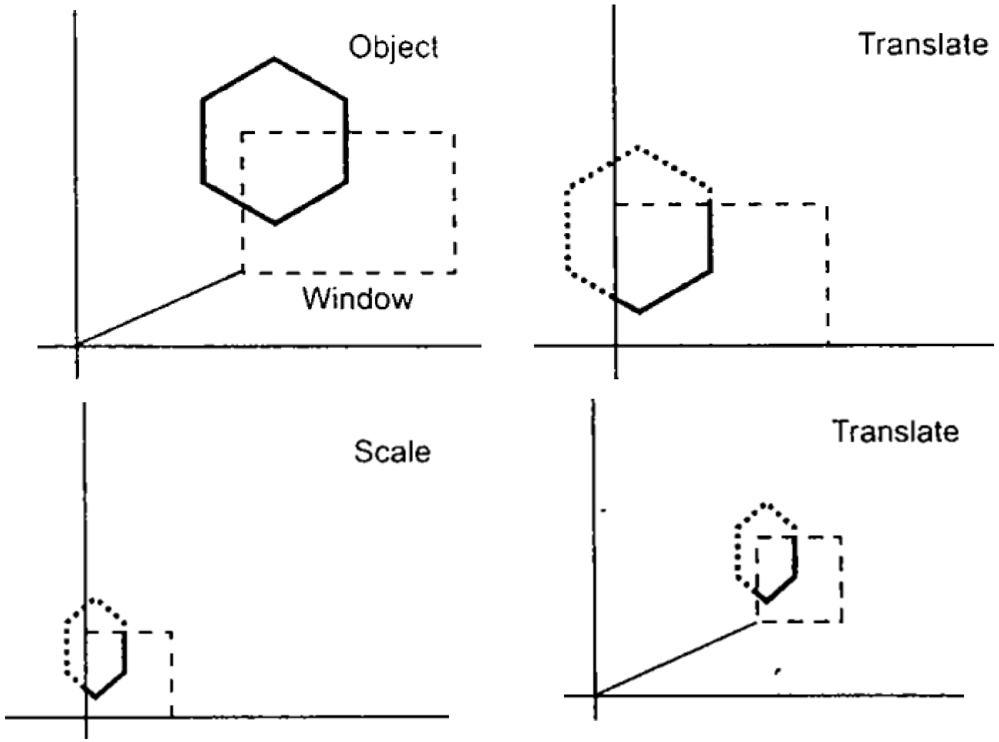


Fig:- Step in workstation transformation

The transformation matrices for individual transformation are as given below :

$$T = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Xw_{min} & -Yw_{min} & 1 \end{vmatrix}$$

$$S = \begin{vmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad \text{where} \quad S_x = \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}$$

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

$$T^{-1} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Xv_{min} & Yv_{min} & 1 \end{vmatrix}$$

The overall transformation matrix for W is given as

$$W = T \cdot S \cdot T^{-1}$$

$$= \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Xw_{min} & -Yw_{min} & 1 \end{vmatrix} \begin{vmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Xv_{min} & Yv_{min} & 1 \end{vmatrix}$$

$$= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ x_{vmin} - x_{w\,min} \cdot S_x & y_{v\,min} - y_{w\,min} \cdot S_y & 1 \end{bmatrix}$$

The Fig. 5.7 shows the complete viewing transformation.

Formula to solve workstation transformation problems:

$$W = T \,.\, S \,.\, T^{-1}$$

$$T = \begin{vmatrix} 1 & 0 & -Xwmin \\ 0 & 1 & -Ywmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$W = \begin{vmatrix} 1 & 0 & -Xwmin \\ 0 & 1 & -Ywmin \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & Xvmin \\ 0 & 1 & Yvmin \\ 0 & 0 & 1 \end{vmatrix}$$

$$S = \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$W = \begin{vmatrix} Sx & 0 & Xvmin - Xwmin * Sx \\ 0 & Sy & Yvmin - Ywmin * Sy \\ 0 & 0 & 1 \end{vmatrix}$$

$$T^{-1} = \begin{vmatrix} 1 & 0 & Xvmin \\ 0 & 1 & Yvmin \\ 0 & 0 & 1 \end{vmatrix}$$

Example-01: What are the steps of workstation transformation? Explain with example and find the equation for workstation transformation.

Solution: see previous topic.

Example-02: Find normalization transformation that maps a window whose lower-left corner is at (1,1) and upper right corner is at (3,5) onto: a) Viewport with lower-left corner (0,0) and upper right corner (1,1) b) Viewport with lower left corner (0,0) and upper right corner (1/2,1/2).

Solution: Given that,

wxmin=1, wxmax=3,

wymin=1, wymax=5

a)

vxmin=0, vxmax=1

vymin=0, vymax=1

Sx = (vxmax– vxmin) / (wxmax– wxmin)

Sy = (vymax– vymin) / (wymax– wymin)

While, Putting the values:

Sx = (1 – 0) / (3 – 1) = 1/2,

Sy = (1 – 0) / (5 – 1) = ¼

The normalized form:

$$N = \begin{bmatrix} 1 & 0 & vx_{min} \\ 0 & 1 & vy_{min} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -wx_{min} \\ 0 & 1 & -wy_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

b)

vxmin=0, vxmax=1/2 ,
vymin=0, vymax=1/2

Sx = (vxmax– vxmin) / (wxmax– wxmin),
Sy = (vymax– vymin) / (wymax– wymin)

While, Putting the values:

Sx = (1/2-0)/3-1 = (½)/2 = 1/4
Sy= (1/2-0)/5-1 = (1/2)/4 = 1/8

$$N=\begin{bmatrix} 1 & 0 & vx_{min} \\ 0 & 1 & vy_{min} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -wx_{min} \\ 0 & 1 & -wy_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

$$N=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/8 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$N=\begin{bmatrix} 1/4 & 0 & -1/4 \\ 0 & 1/8 & -1/8 \\ 0 & 0 & 1 \end{bmatrix}$$

Example-03: Find normalization transformation that maps a window whose lower-left corner is at (1,1) and upper right corner is at (3,5) onto: a) Viewport that is the entire normalized device screen.

Solution:

(a) The window parameters are $wx_{\min} = 1$, $wx_{\max} = 3$, $wy_{\min} = 1$, and $wy_{\max} = 5$. The viewport parameters are $vx_{\min} = 0$, $vx_{\max} = 1$, $vy_{\min} = 0$, and $vy_{\max} = 1$. Then $s_x = \frac{1}{2}$, $s_y = \frac{1}{4}$, and

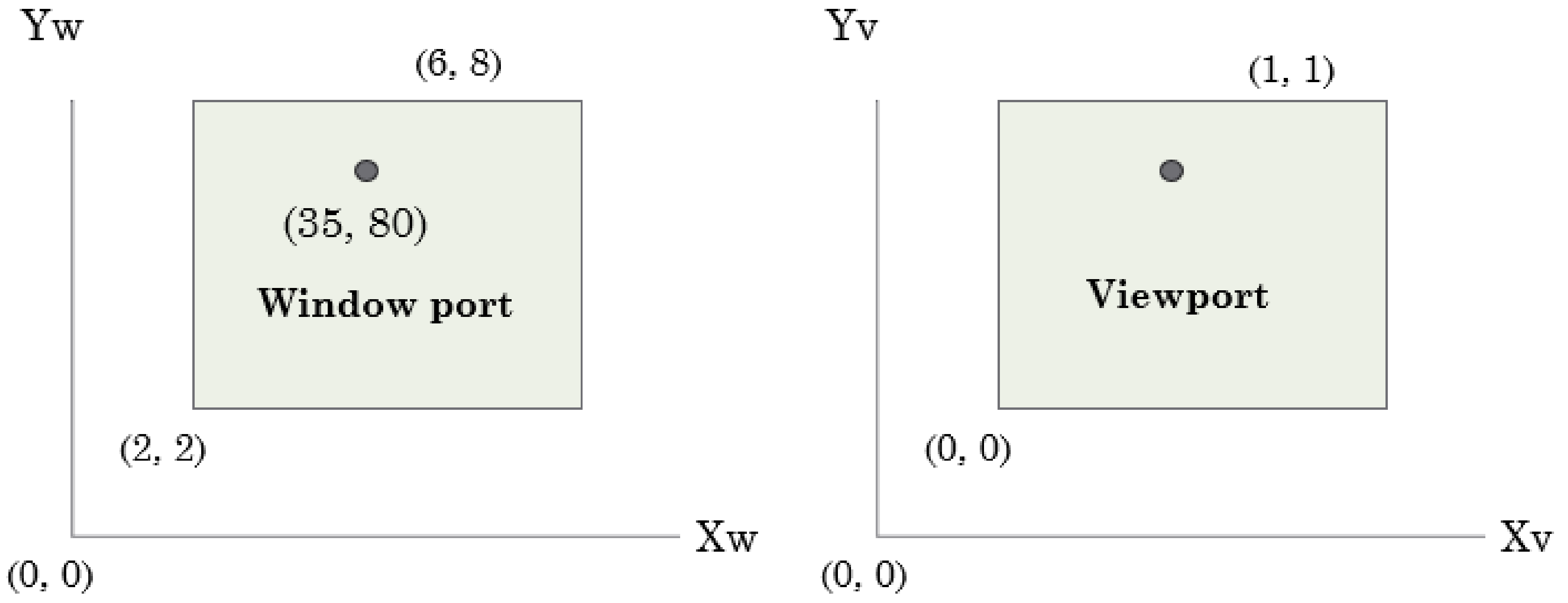
$$N = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix}$$

Example-04:

i) Find (Sx, Sy) = ?

ii) Find N = ?

Solution: Solve By YourSelf !



Example-05: Find normalization transformation that maps a window whose lower-left corner is at (2,2) and upper right corner is at (3,4) onto:

a) Viewport that is the entire normalized device screen.

b) Viewport with lower left corner (0,0) and upper right corner (3/2,3/2).

Solution: Solve By YourSelf !

Example-06:

Find the complete viewing transformation that maps a window in world coordinates with x extent 1 to 10 and y extent 1 to 10 onto a viewport with x extent $\frac{1}{4}$ to $\frac{3}{4}$ and y extent 0 to $\frac{1}{2}$ in normalized device space, and then maps a workstation window with x extent $\frac{1}{4}$ to $\frac{1}{2}$ and y extent $\frac{1}{4}$ to $\frac{1}{2}$ in the normalized device space into a workstation viewport with x extent 1 to 10 and y extent 1 to 10 on the physical display device.

SOLUTION

From Prob. 5.1, the parameters for the normalization transformation are $wx_{min} = 1$, $wx_{max} = 10$, $wy_{min} = 1$, $wy_{max} = 10$, and $vx_{min} = \frac{1}{4}$, $vx_{max} = \frac{3}{4}$, $vy_{min} = 0$, and $vy_{max} = \frac{1}{2}$. Then

$$s_x = \frac{1/2}{9} = \frac{1}{18} \qquad s_y = \frac{1/2}{9} = \frac{1}{18}$$

and

$$N = \begin{pmatrix} \frac{1}{18} & 0 & \frac{7}{36} \\ 0 & \frac{1}{18} & -\frac{1}{18} \\ 0 & 0 & 1 \end{pmatrix}$$

The parameters for the workstation transformation are $wx_{\min} = \frac{1}{4}$, $wx_{\max} = \frac{1}{2}$, $wy_{\min} = \frac{1}{4}$, $wy_{\max} = \frac{1}{2}$, and $vx_{\min} = 1$, $vx_{\max} = 10$, $vy_{\min} = 1$, and $vy_{\max} = 10$. Then

$$s_x = \frac{9}{1/4} = 36 \qquad s_y = \frac{9}{1/4} = 36$$

and

$$W = \begin{pmatrix} 36 & 0 & -8 \\ 0 & 36 & -8 \\ 0 & 0 & 1 \end{pmatrix}$$

The complete viewing transformation V is

$$V = W \cdot N = \begin{pmatrix} 36 & 0 & -8 \\ 0 & 36 & -8 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{18} & 0 & \frac{7}{36} \\ 0 & \frac{1}{18} & -\frac{1}{18} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -10 \\ 0 & 0 & 1 \end{pmatrix}$$

Difference Between Window Port And Viewport

Window port	Viewport
Window port is the coordinate area specially selected for the display.	An area on display device to which a window is mapped.
Region created according to the world coordinates	Region created according to device coordinates
It is a region selected from the real world. It is a graphically control thing and composed of visual areas along with some of its program controlled with help of window decoration.	It is the region in computer graphics which is a polygon viewing region.
A window port can be defined with the help of a GWINDOW statement.	A view port can be defined with the help of a GPORT command.
You can define the window to be larger than, the same as or smaller than the actual range of data values, depending on whether you want to show all of the data or only part of the data.	The rectangular portion of the interface window that defines where the image will defines where the image will actually appear.
Window defines what is to be viewed.	Viewport defines where the window to be displayed.

❑ Computer Graphics Zooming

- **Zooming** is a transformation often provided with an imaginary software. The transformation effectively scales down or blows up a pixel map or a portion of it with the instructions from the user. Such scaling is commonly implemented at the pixel level rather than at the coordinates level. A video display or an image is necessarily a pixel map, i.e., a collection of pixels which are the smallest addressable elements of a picture. The process of zooming replicates pixels along successive scan lines.

- **Example:** for a zoom factor of two
  - Each pixel value is used four times twice on each of the two successive scan lines.
  - Figure shows the effect of zooming by a factor of 2.

Fig:Effect of zooming by a factor of two.

- Such integration of pixels sometimes involves replication using a set of ordered patterns, commonly known as Dithering.
- The two most common dither types are:
  - Ordered dither.
  - Random dither.
- There are widely used, especially when the grey levels (share of brightness) are synthetically generated.

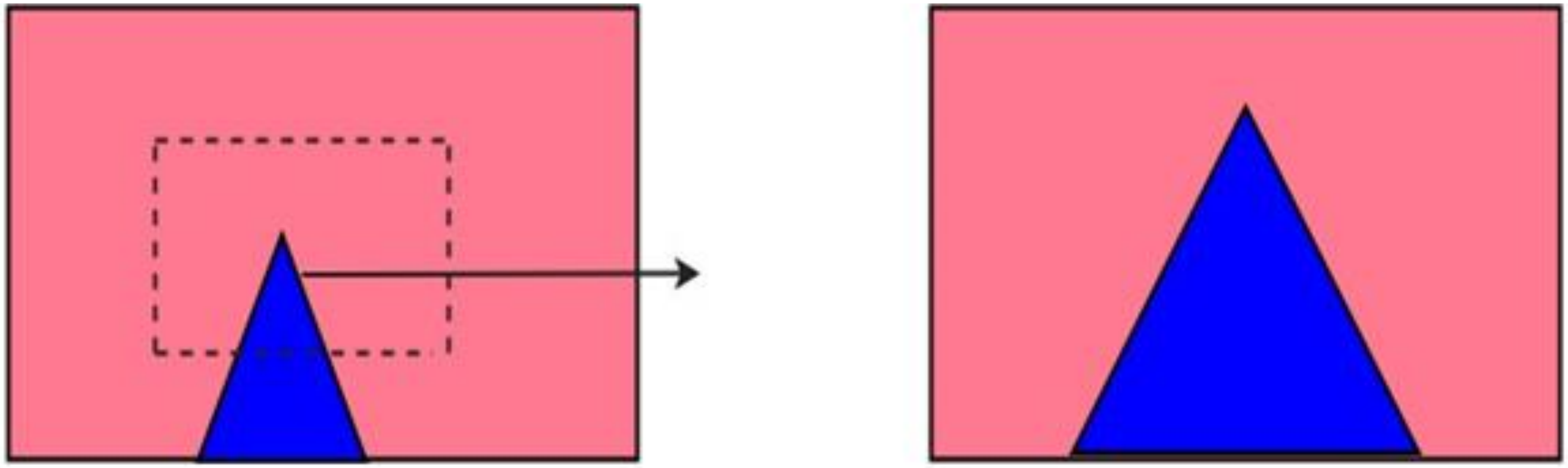


Fig:The zooming of the selected portion of an image (In the case of panning,the blown up part is shifted to the center of the screen)

❑ Computer Graphics Panning

- The process of panning acts as a qualifier to the **zooming transformation**. This step moves the scaled up portion of the image to the center of the screen and depending on the scale factor, fill up the entire screen.

- **Advantage:** Effective increase in zoom area in all four direction even if the selected image portion (for zooming) is close to the screen boundary.

❑ Inking:

If we sample the position of a graphical input device at regular intervals and display a dot at each sampled position, a trial will be displayed of the movement of the device. This technique which closely simulates the effect of drawing on paper is called Inking.

For many years the primary use of inking has been in conjunction with online character-recognition programs.

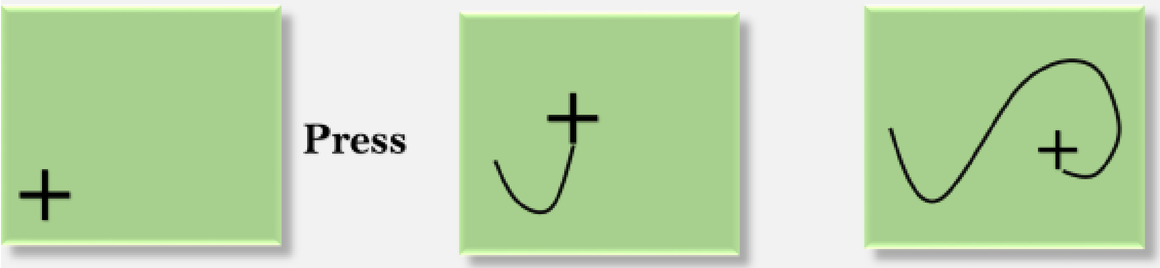


Fig: Freehand inking

❑ Scissoring:

In computer graphics, the deleting of any parts of an image which falls outside of a window that has been sized and laid the original vision ever. It is also called the **clipping**.