# Two Dimensional Viewing

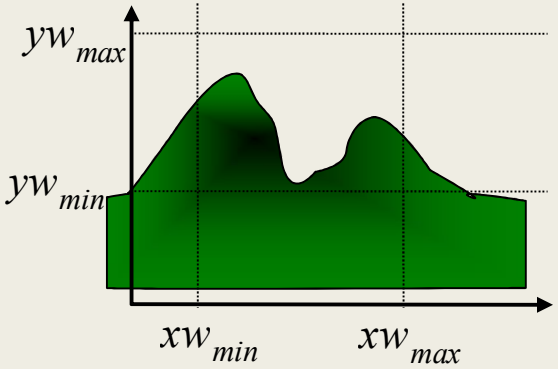
# The Viewing Pipeline

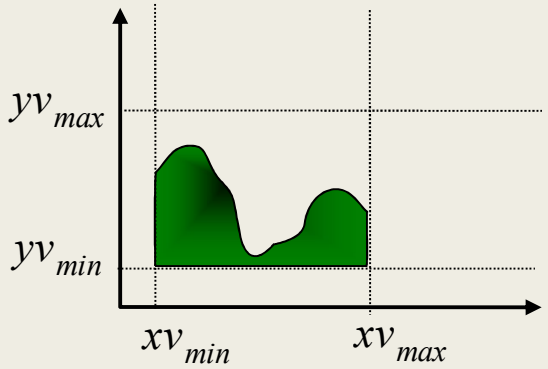
- Graphics package allows the user to specify which part of a defined picture to be displayed where that part is to displayed on the display device
- **Window**
  - *A world-coordinate area selected for display.*

    *defines what is to be viewed*
- **Viewport**
  - *An area on a display device to which a window is mapped.*

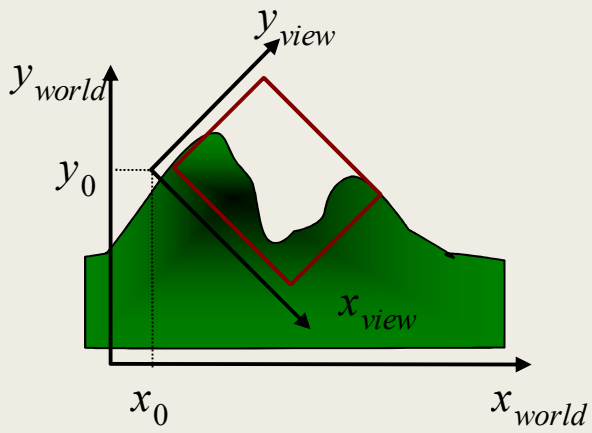
    *defines where it is to be displayed*
- **Viewing transformation**
  - *The mapping of a part of a world-coordinate scene to device coordinates.*
  - *The two dimensional viewing transformation is referred as windowing transformations.*

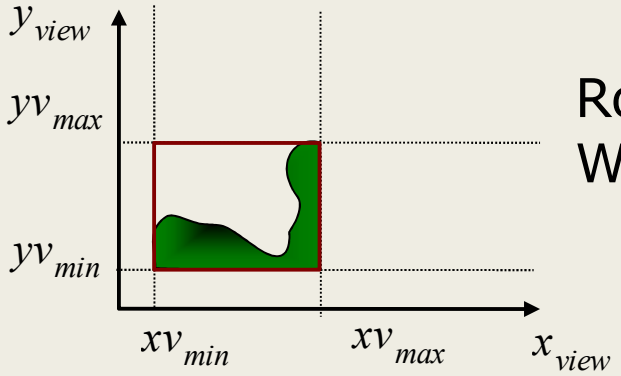
# Two-Dimensional Viewing




Rectangular Window




Rotated Window

# Two-Dimensional Viewing

- Rectangular window of different orientation viewing transformations follows the following steps
- Construct the scene in world coordinates using the output primitives and attributes.
- Obtain the particular orientation for the window, set up a two dimensional viewing coordinate system in the world coordinate plane and define a window in the viewing coordinate system.
- Once the reference frame is established transform the descriptions in world coordinates to viewing coordinates.
- We then define a viewport in normalized coordinates and map the viewing coordinate of the scene to normalized coordinates

# The Viewing Pipeline

- All parts of the picture that lie outside the viewport are clipped .
- Contents  of the viewport are transferred to device coordinates.

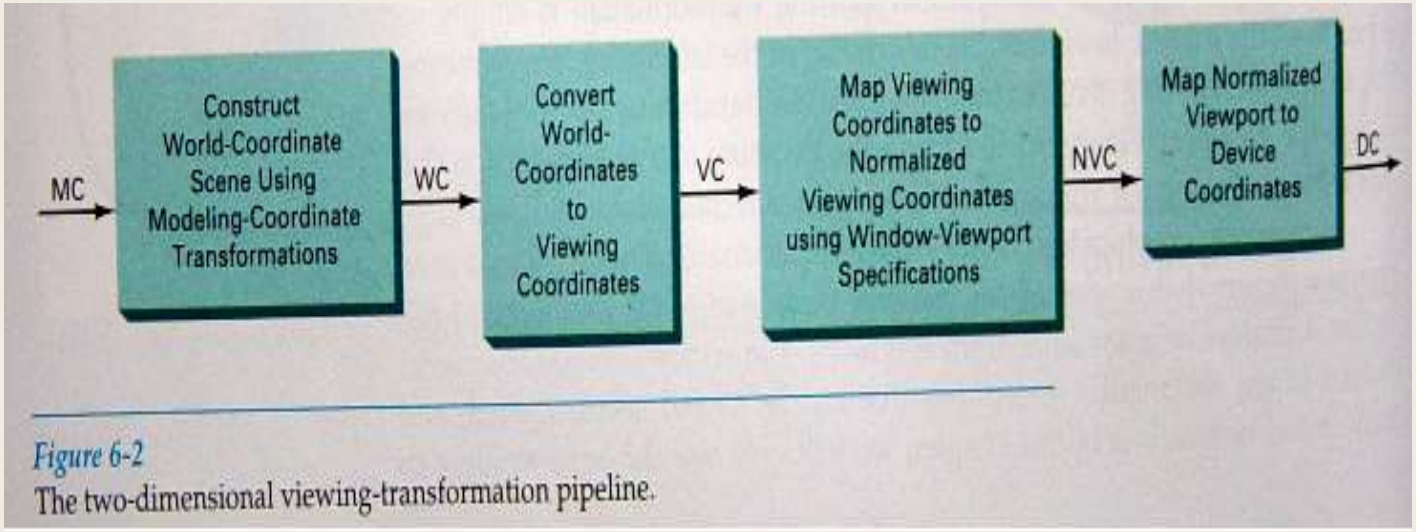


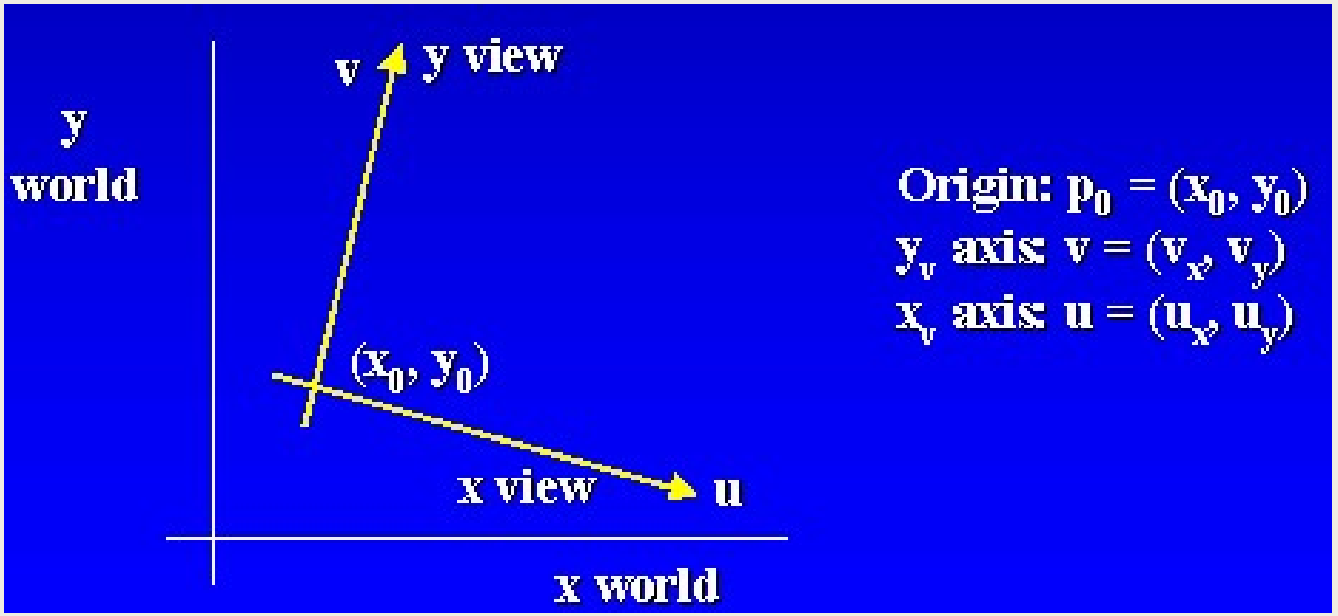

*Figure 6-2*
The two-dimensional viewing-transformation pipeline.

# Two-Dimensional Viewing

- By changing the position of the viewport we can view objects at different positions on the display area of the output device.
- **Zooming effects**
  - *Successively mapping different-sized windows on a fixed-sized viewports.*
- **Panning effects**
  - *Moving a fixed-sized window across the various objects in a scene.*
- **Device independent**
  - *Viewports are typically defined within the unit square. (normalized coordinates)*
  - *This provides means for separating the viewing and other transformations from specific output device requirements.*

# Viewing Coordinate Reference Frame

- The reference frame for specifying the world-coordinate window.
  - *Viewing-coordinate origin: P0 = (x0, y0)*
  - *Establish the orientation or rotation of this reference frame.*
  - *Specify View up vector V: Define the viewing $y_v$ direction*

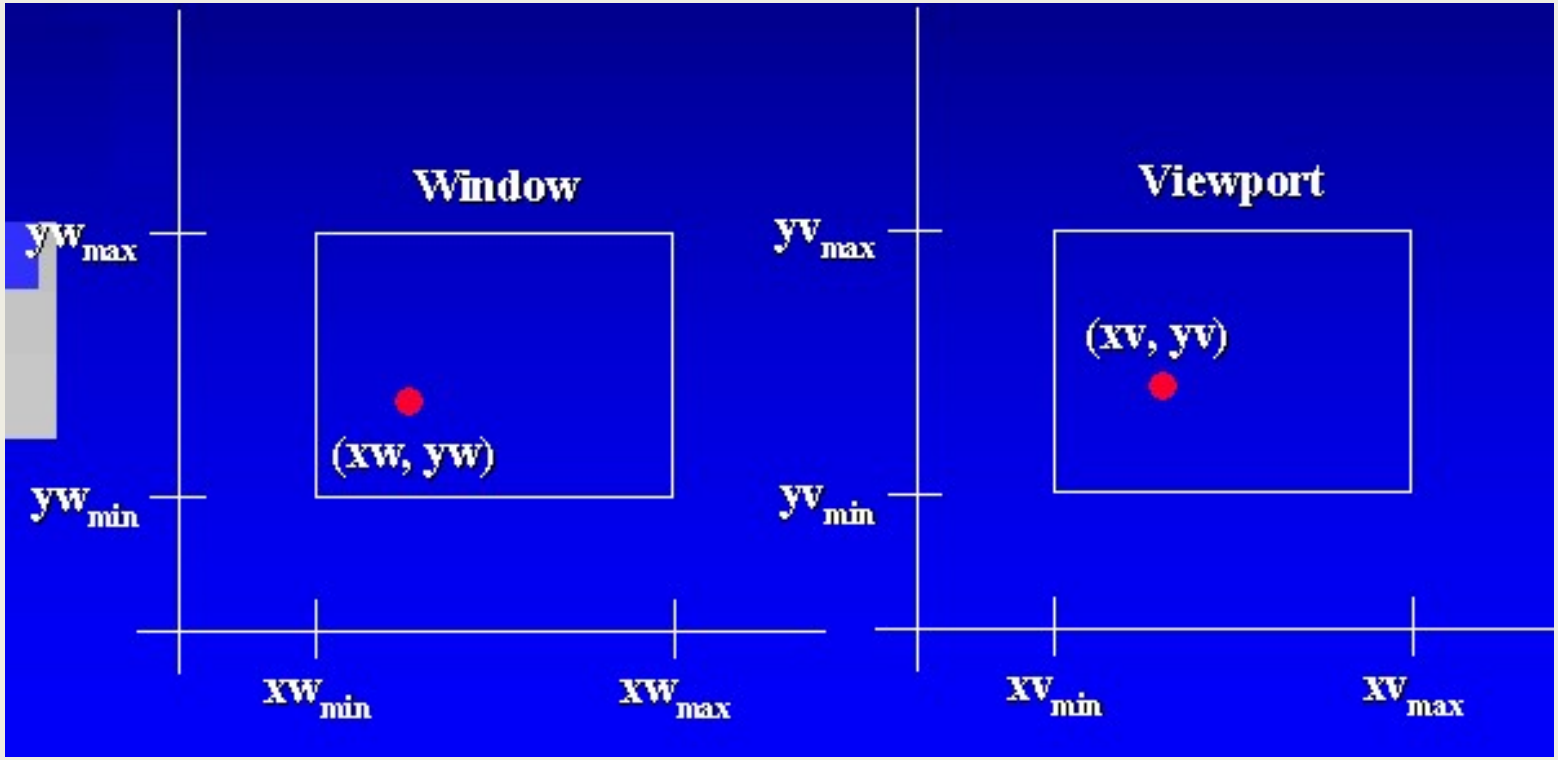


# Viewing Coordinate Reference Frame

- Given V calculate the components of uint vectors $v_x$,$v_y$ and $u_x$,$u_y$

  for the viewing $y_v$ and $X_v$ .
- These unit vectors aligns the viewing $x_v$,$y_v$ axes with world axes $x_w$,$y_w$
- The composite two dimensional transformation to convert world coordinates to viewing coordinates is
- Translate the viewing origin to the world origin
- Rotate to align the two coordinate reference frames.
  - $M_{wc,vc}=R.T$

# Window-to-Viewport Coordinate Transformation



# Window-to-Viewport Coordinate Transformation

$$\frac{xv - xv_{\min}}{xv_{\max} - xv_{\min}} = \frac{xw - xw_{\min}}{xw_{\max} - xw_{\min}} \qquad \frac{yv - yv_{\min}}{yv_{\max} - yv_{\min}} = \frac{yw - yw_{\min}}{yw_{\max} - yw_{\min}}$$

$$xv = xv_{\min} + (xw - xw_{\min})sx$$

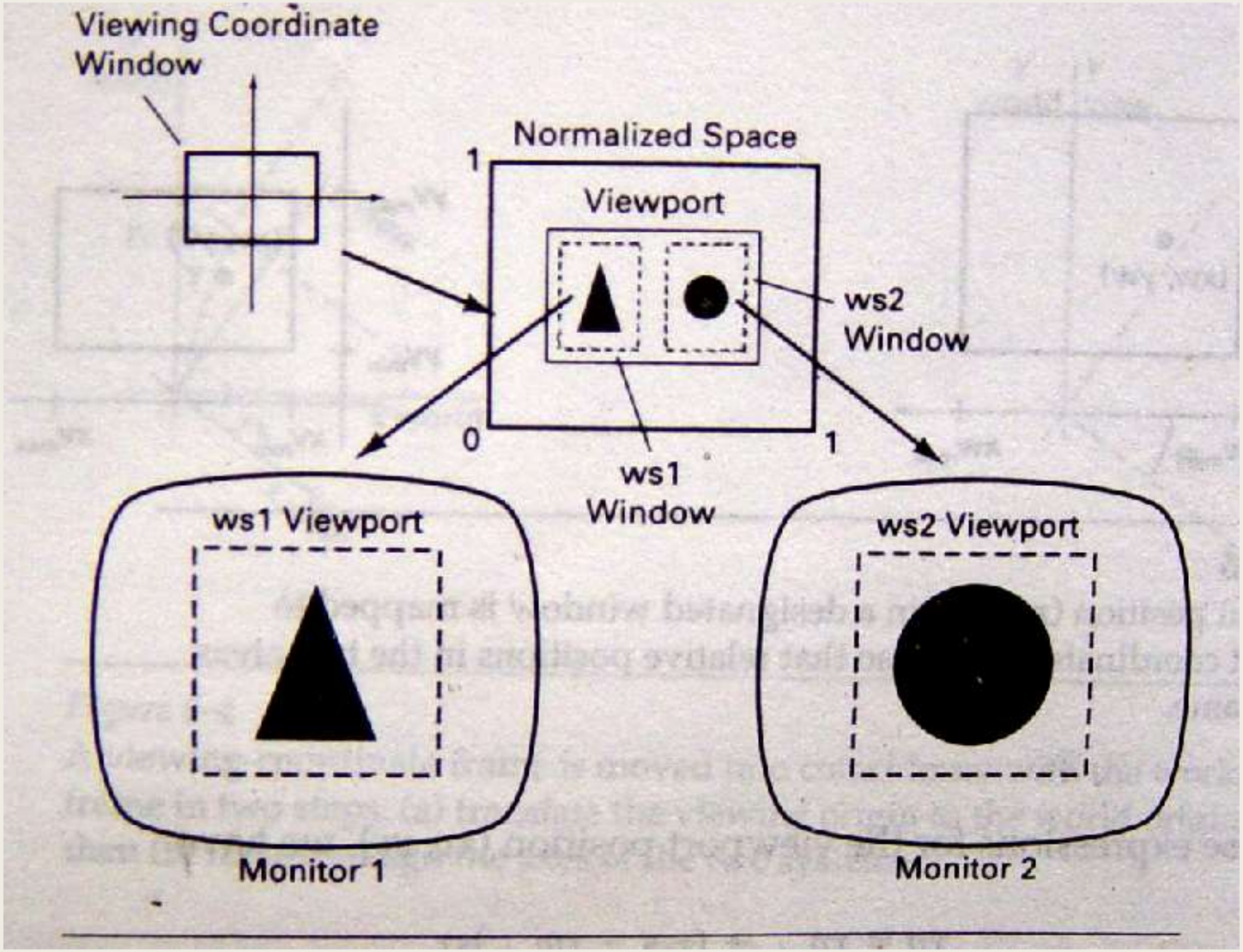
$$yv = yv_{\min} + (yw - yw_{\min})sy$$

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

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

A point at poisition ($x_w$,$y_w$) is mapped into position ($x_v$,$y_v$)

# Workstation transformtion



# Workstation Transformtion

- From normalized coordinates , object description  are mapped to various display devices.
- Any no of output devices can be used and window to viewport transformation can be performed for each open output device.
- This mapping called the **workstation transformation.**
- Window area in normalized space and viewport area in the coordinates of the display device.
- Controls the positioning of parts of a scene on individual output devices.

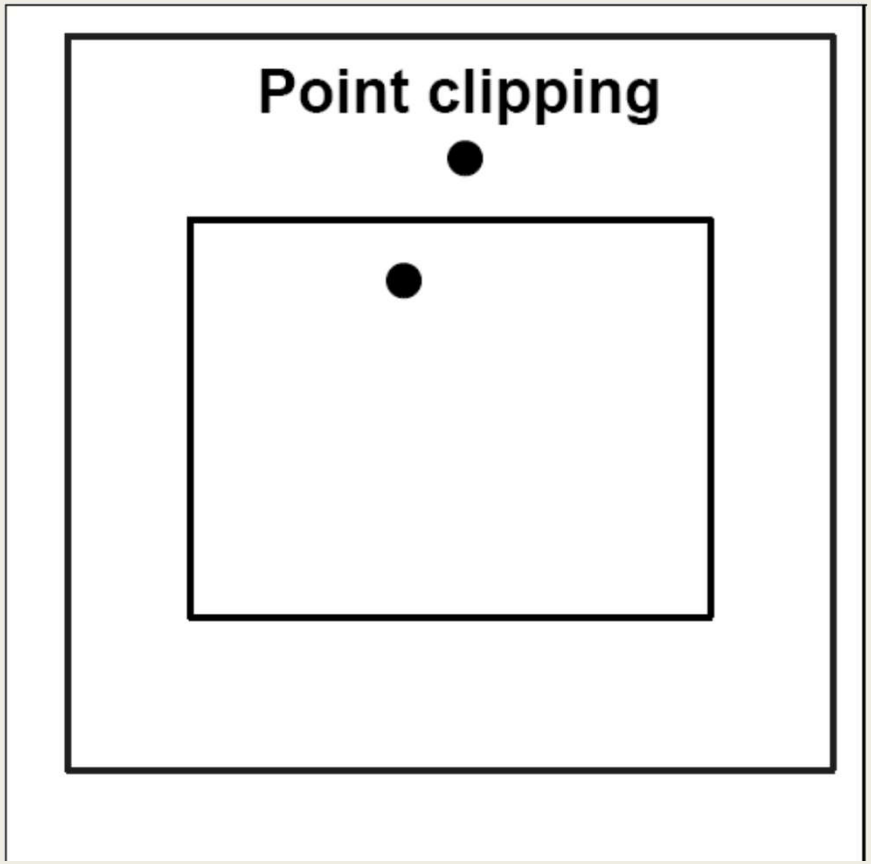
# Clipping Operations

- Clipping
  - *Identify those portions of a picture that are either inside or outside of a specified region of space.*
- Clip window
  - *The region against which an object is to be clipped.*
  - *The shape of clip window*
- Applications of clipping :extracting part of the defined scene for viewing , identifying visible surfaces in 3d views etc.
- World-coordinate clipping: Clipping algorithm can be applied to the world coordinate , so the contents of the window are mapped to device coordinates.

# Clipping Operations

- Viewport clipping
  - *It can reduce calculations by allowing concatenation of viewing and geometric transformation matrices.*
- Types of clipping
  - *Point clipping*
  - *Line clipping*
  - *Area (Polygon) clipping*
  - *Curve clipping*
  - *Text clipping*
- Point clipping (Rectangular clip window)

# Point clipping



# Point clipping

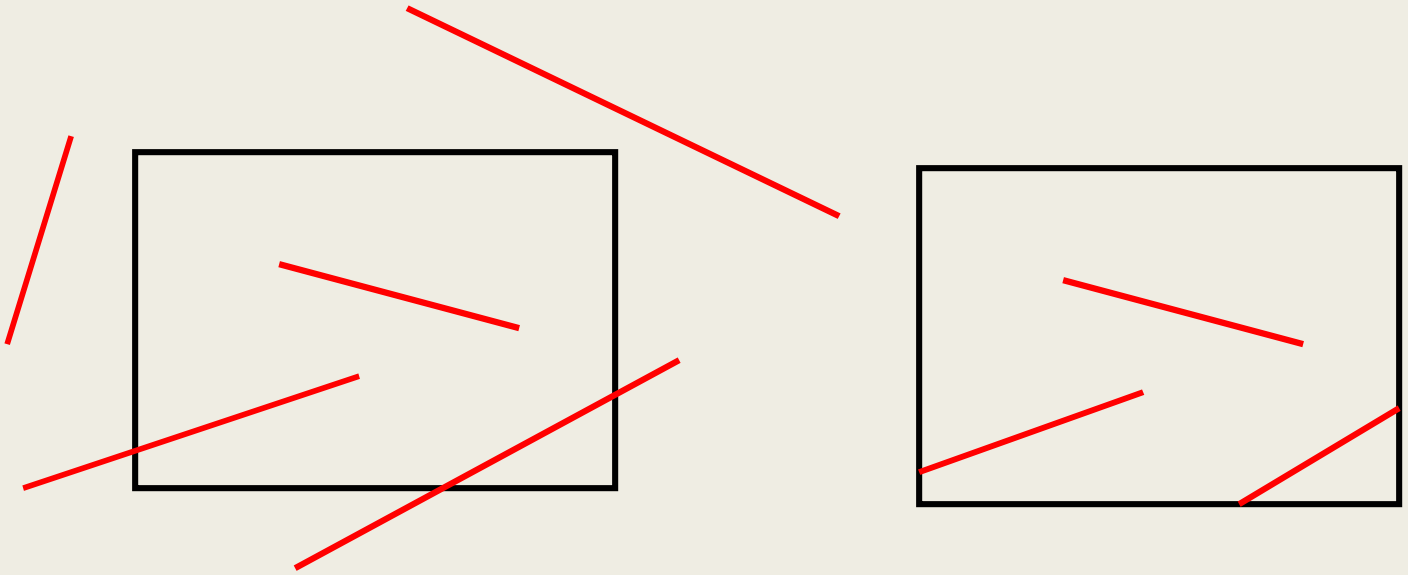
- Is point (x,y) inside the clip window?
- Considering the clip window is a rectangular window.
- A pint is inside if it satisfies the following inequalities are satisfied
  - $XW_{win} <= x <= XW_{max}$
  - $YW_{win} <= x <= YW_{max}$

where the edges of the clip window can be either world coordinate window boundaries or viewport boundaries

- If any of the inequalities are not satisfied the point is clipped.

# Line Clipping

- **Possible relationships between line positions and a standard rectangular clipping region**



Before clipping

after clipping

# Line Clipping

- Possible relationships
  - *Completely inside the clipping window*
  - *Completely outside the window*
  - *Partially inside the window*
- Parametric representation of a line segment with endpts (x1,y1) and (x2,y2)

  $x = x_1 + u(x_2 - x_1)$

  $y = y_1 + u(y_2 - y_1)$  $0 <= u <= 1$
- The value of u for an intersection with a rectangle boundary edge
  - *Outside the range 0 to 1 ,line does not enter the interior of window*
  - *Within the range from 0 to 1,the line segment crosses the clipping area.*