

# # COMPUTER GRAPHICS..!!!

## #. CLIPPING

# # Overview #

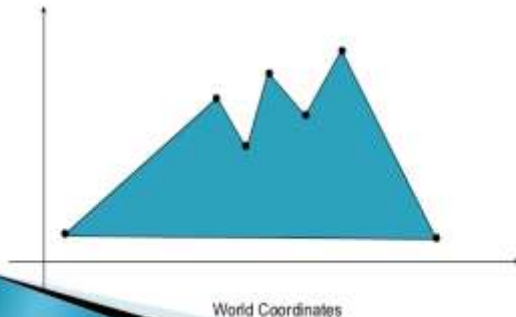
- Introduction.
- Point Clipping.
- End Point Codes.
- Algorithm :
  1. Cohen-Sutherland Algorithm.
  2. Mid-Point Subdivision Algorithm.
  3. Sutherland-Hodgeman Algorithm.
- Text Clipping.

# # Clipping

- Clipping is a process of extracting a portion of a data base or identifying elements of a scene or picture inside or outside a specified region, called the clipping region.
- E.g.,

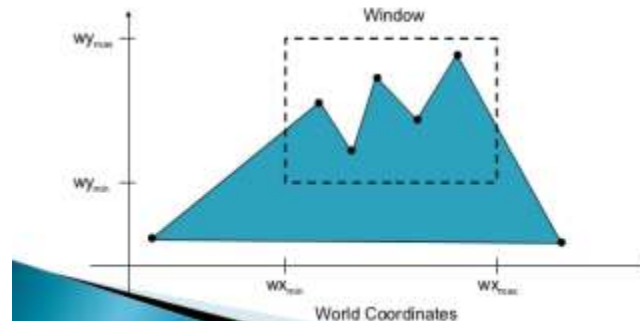
## Windowing I

A scene is made up of a collection of objects specified in world coordinates



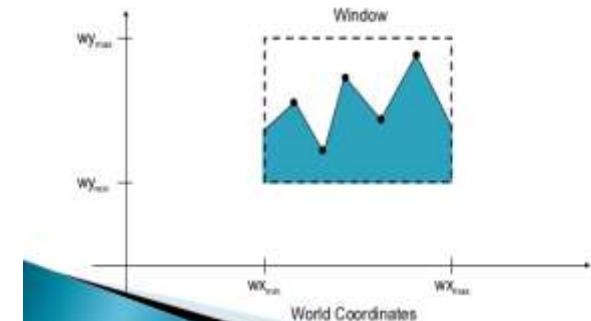
## Windowing II

When we display a scene only those objects within a particular window are displayed



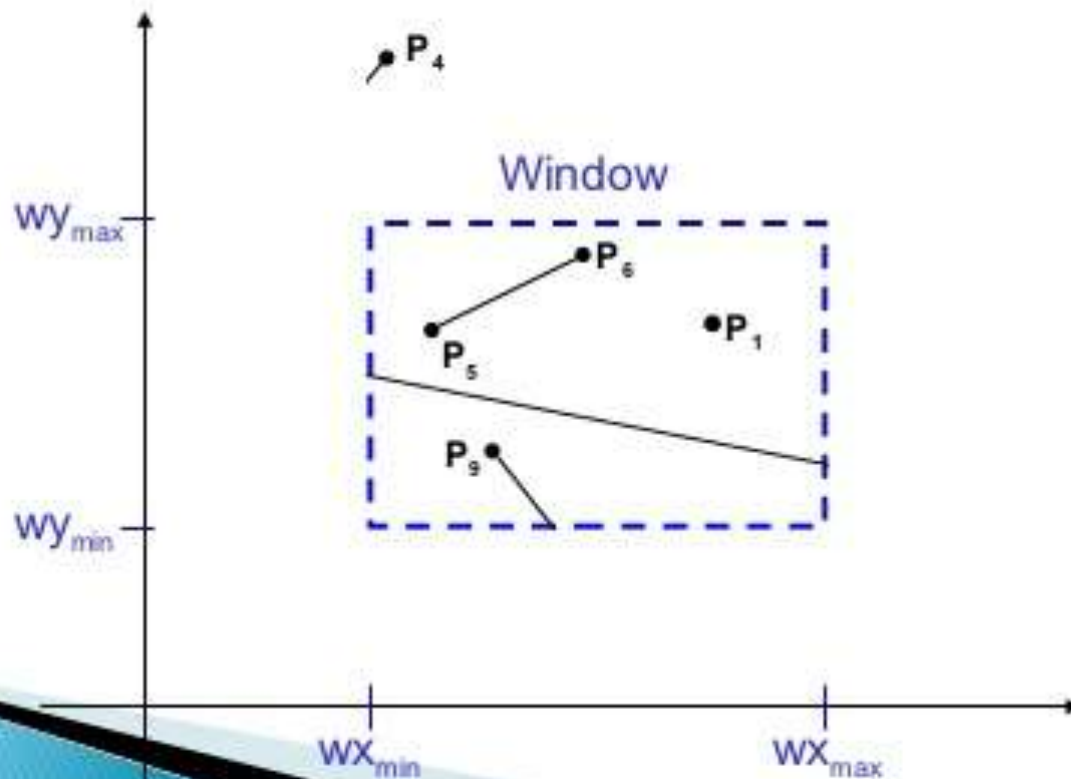
## Windowing III

Because drawing things to a display takes time we *clip* everything outside the window



# Clipping

For the image below consider which lines and points should be kept and which ones should be clipped

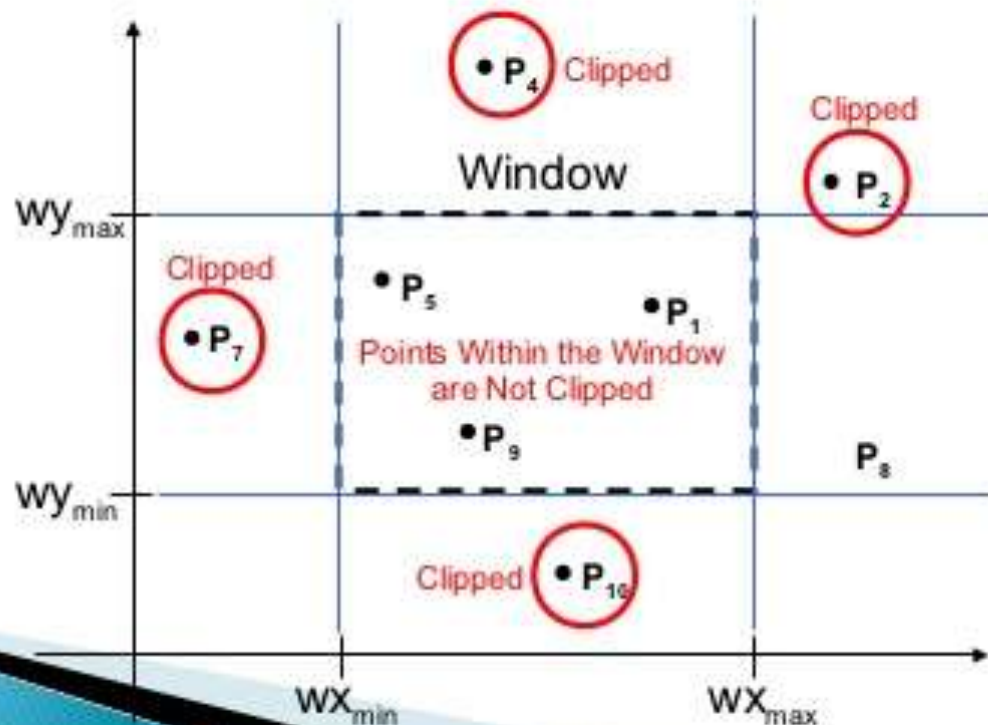


# Point Clipping

Easy - a point  $(x,y)$  is not clipped if:

$$wx_{min} \leq x \leq wx_{max} \text{ AND } wy_{min} \leq y \leq wy_{max}$$

otherwise it is clipped



# # End Point Codes :

- This technique uses a 4 bit (digit) code to indicate which of nine regions contains the end point of a line.
- This test is formalized by Dan Cohen & Ivan Sutherland.
- The rightmost bit is the first bit.
- The bits are set to 1, based on the following scheme:
  - 1<sup>st</sup>-bit set                      -if the end pt is to the left of the window.
  - 2<sup>nd</sup>-bit set                      -if the end pt is to the right of the window.
  - 3<sup>rd</sup>-bit set                      -if the end pt is below the window.
  - 4<sup>th</sup>-bit set                      -if the end pt is above the window.

# Cohen-Sutherland: World Division

World space is divided into regions based on the window boundaries

- Each region has a unique four bit region code
- Region codes indicate the position of the regions with respect to the window

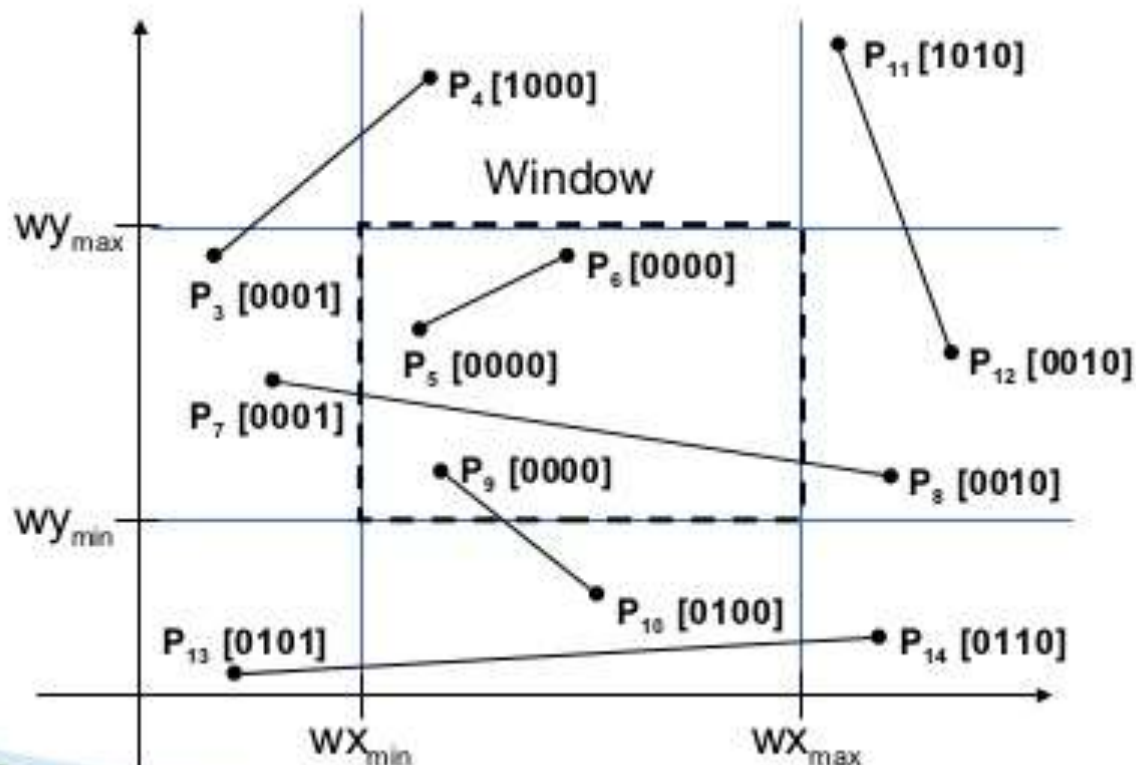
3	2	1	0
above	below	right	left

Region Code Legend

1001	1000	1010
0001	0000 Window	0010
0101	0100	0110

# Cohen-Sutherland: Labelling


Every end-point is labelled with the appropriate region code



# Cohen-Sutherland: Other Lines

Lines that cannot be identified as completely inside or outside the window may or may not cross the window interior


These lines are processed as follows:

- Compare an end-point outside the window to a boundary (choose any order in which to consider boundaries e.g. left, right, bottom, top) and determine how much can be discarded
  - If the remainder of the line is entirely inside or outside the window, retain it or clip it respectively
- 

## Cohen-Sutherland: Other Lines (cont...)

- Otherwise, compare the remainder of the line against the other window boundaries
- Continue until the line is either discarded or a segment inside the window is found

We can use the region codes to determine which window boundaries should be considered for intersection

- To check if a line crosses a particular boundary we compare the appropriate bits in the region codes of its end-points
  - If one of these is a 1 and the other is a 0 then the line crosses the boundary
- 

# Calculating Line Intersections

Intersection points with the window boundaries are calculated using the line-equation parameters

- Consider a line with the end-points  $(x_1, y_1)$  and  $(x_2, y_2)$
- The y-coordinate of an intersection with a vertical window boundary can be calculated using:

$$y = y_1 + m (x_{\text{boundary}} - x_1)$$

where  $x_{\text{boundary}}$  can be set to either  $wx_{\min}$  or  $wx_{\max}$

## Calculating Line Intersections (cont...)

- The x-coordinate of an intersection with a horizontal window boundary can be calculated using:

$$x = x_1 + (y_{\text{boundary}} - y_1) / m$$

where  $y_{\text{boundary}}$  can be set to either  $wy_{\text{min}}$  or  $wy_{\text{max}}$

- $m$  is the slope of the line in question and can be calculated as  $m = (y_2 - y_1) / (x_2 - x_1)$

# Mid Point Subdivision Algorithm

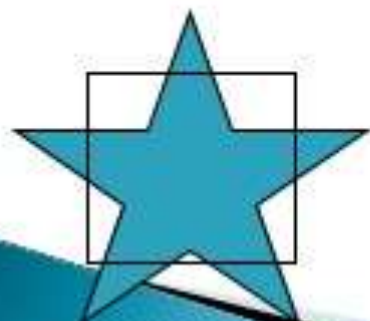
- The Cohen-Sutherland algorithm requires the calculation of the intersection of the line with the window edge.
- The direct calculation is avoided by performing a binary search for the intersection by always dividing the line at its midpoint.

# Sutherland-Hodgman Area Clipping Algorithm

A technique for clipping areas developed by Sutherland & Hodgman

Put simply the polygon is clipped by comparing it against each boundary in turn

Sutherland turns up again. This time with Gary Hodgman with whom he worked at the first ever graphics company Evans & Sutherland



Original Area



Clip Left



Clip Right




Clip Top



Clip Bottom

# Sutherland–Hodgman Area Clipping Algorithm (cont...)

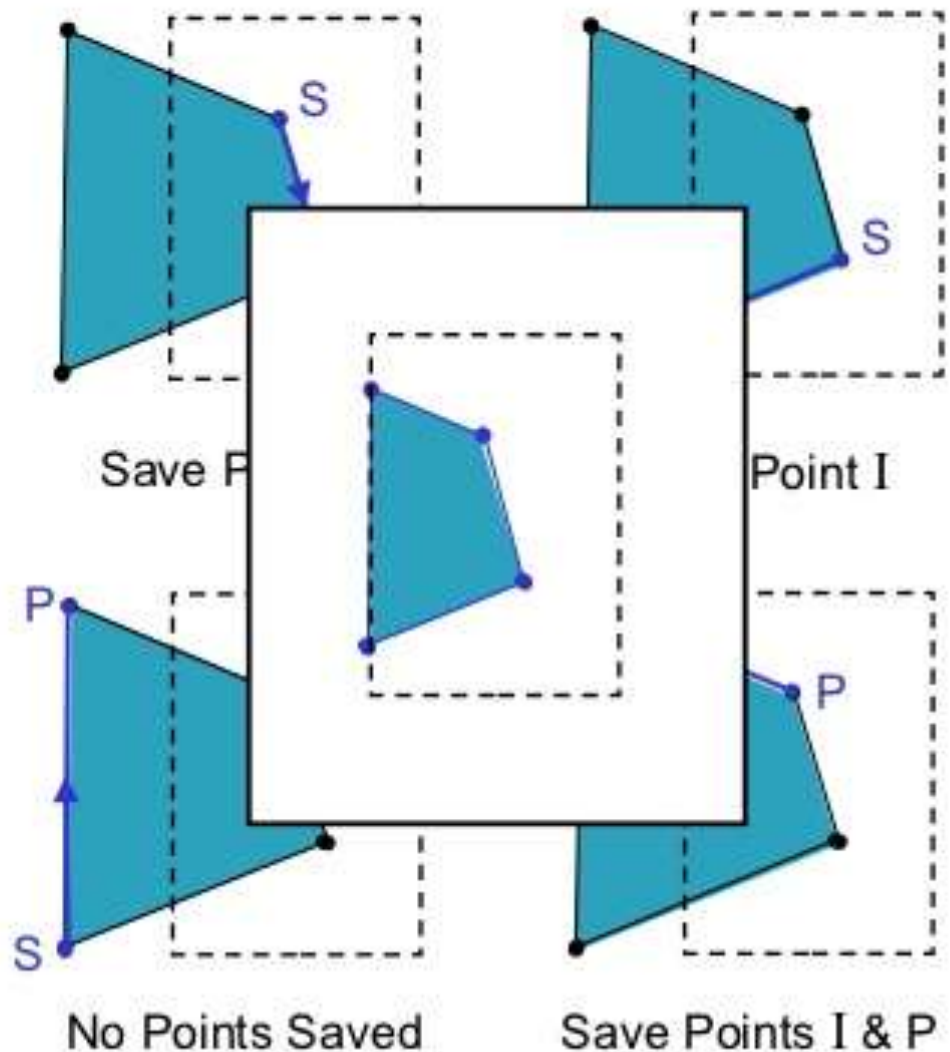
To clip an area against an individual boundary:

- Consider each vertex in turn against the boundary
  - Vertices inside the boundary are saved for clipping against the next boundary
  - Vertices outside the boundary are clipped
  - If we proceed from a point inside the boundary to one outside, the intersection of the line with the boundary is saved
  - If we cross from the outside to the inside intersection point and the vertex are saved
- 

# Sutherland-Hodgman Example

Each example shows  
the point being  
processed (P) and the  
previous point (S)

Saved points define  
area clipped to the  
boundary in question



# Text Clipping

In general, methods depend on how characters are represented

However, three strategies can be followed:

- ▶ **all-or-none string clipping**
  - use a bounding rectangle for the string
- ▶ **all-or-none character clipping**
  - use a bounding rectangle for the character
- ▶ **individual character clipping or component clipping**
  - like line/curve clipping (outlined char's)
  - compare individual pixels (bit-mapped)

