Window to Viewport

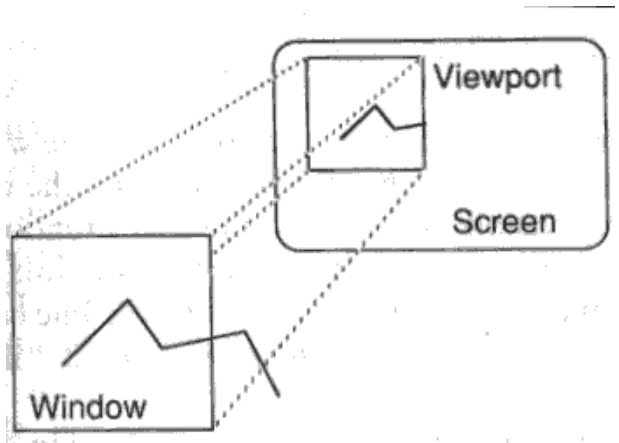
Raster methods

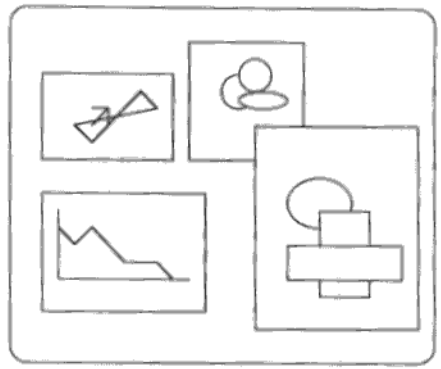
Vector methods

- line clipping
- polygon clipping

Window to Viewport

- Display picture on output device



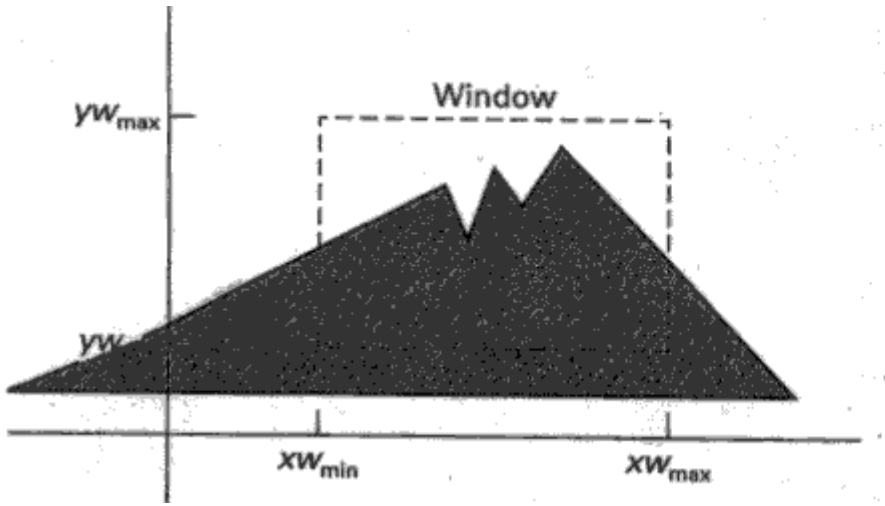


Screen showing many viewports

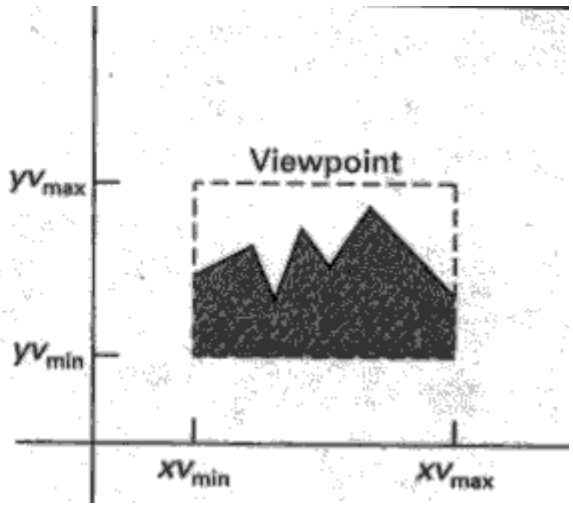
Window - area selected for display (Note not a window as in Microsoft windows) ‘what is to be viewed’

Viewport - area on display device to which window is mapped ‘where it is to be displayed’

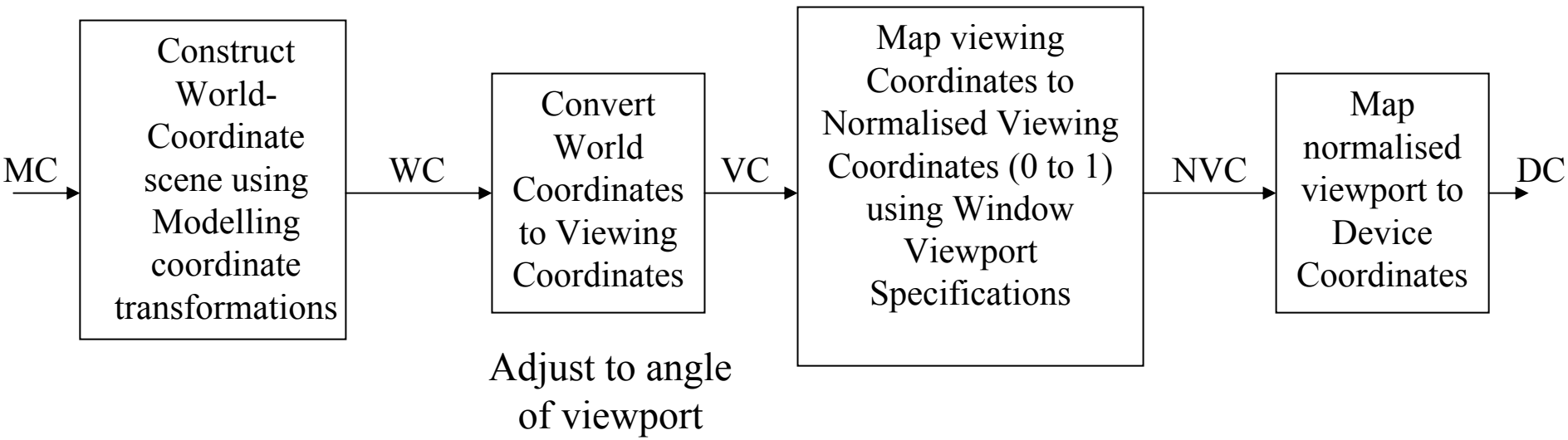
Raster Method



World Coordinates



Device Coordinates



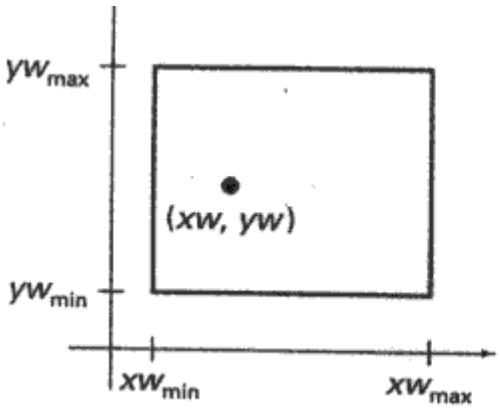
Normalising

- Why normalise the coordinates?
  - Makes the graphics package device-independent
  - the normalised coordinates are simply mapped to the display area for the particular device in use at that time
  - different viewing devices can be used by providing the appropriate device drivers

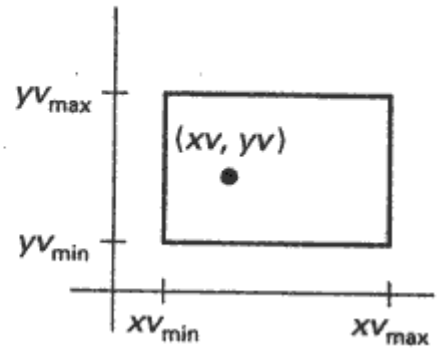
Zoom and Pan

- Zoom
  - successively mapping different sized windows on a fixed size viewport
- Pan
  - moving a fixed size viewport across various objects in a scene

Window to Viewpoint Mapping



Window coordinates



Viewport coordinates

To maintain relative placement in the viewport

$$\frac{xv - xv_{\min}}{xv_{\max} - xv_{\min}} = \frac{xw - xw_{\min}}{xw_{\max} - xw_{\min}}$$

(Similar for y coords)

The scale factor must be

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

Clipping

- Clipping applications
  - extracting part of a scene for viewing
  - identifying visible surfaces in 3D views
  - displaying a multi-window environment
  - selecting part of a picture to be edited say erased or copied.
- Clipping Algorithms covered in this module
  - Point, line, area (polygon), text

Point Clipping

A point will be displayed if

$$xw_{\min} \leq x \leq xw_{\max}$$

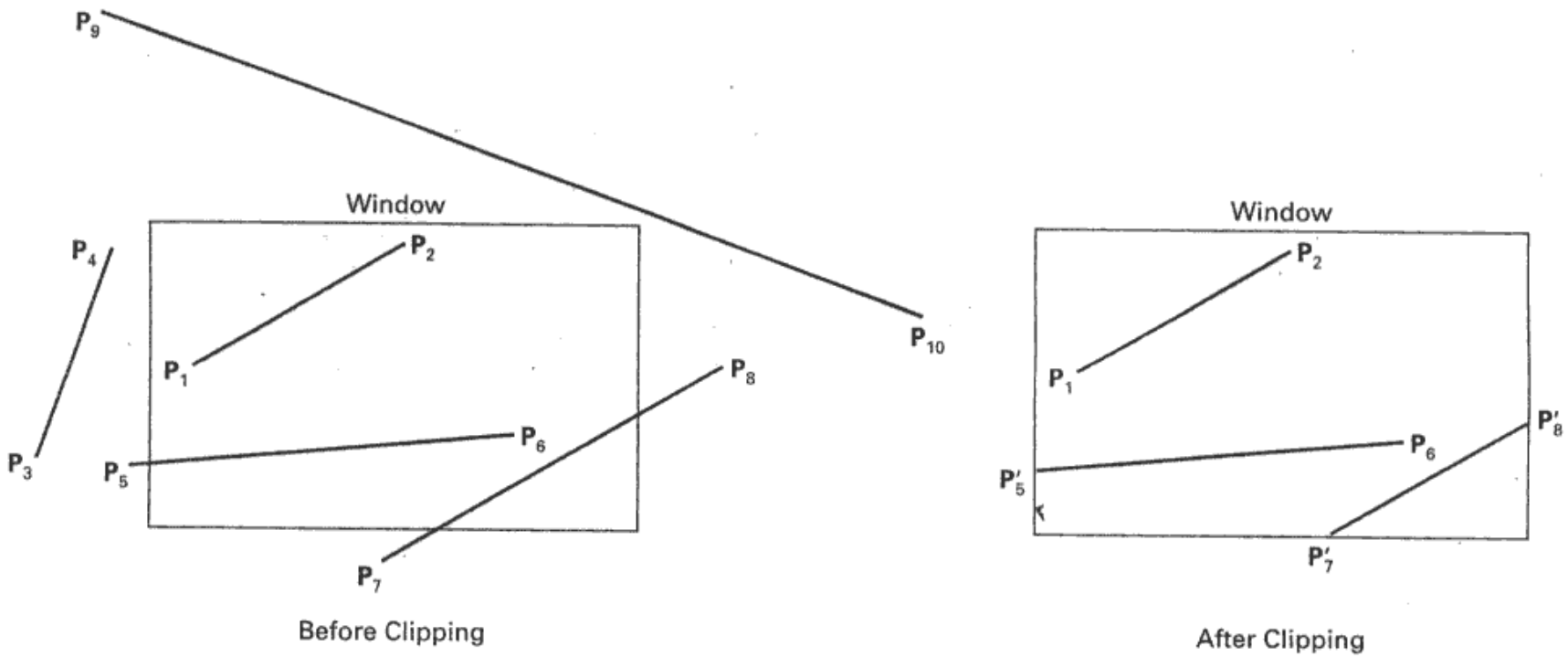
and

$$yw_{\min} \leq y \leq yw_{\max}$$

- Few applications
  - diagram showing particles
  - hatching for soft soil

Line Clipping

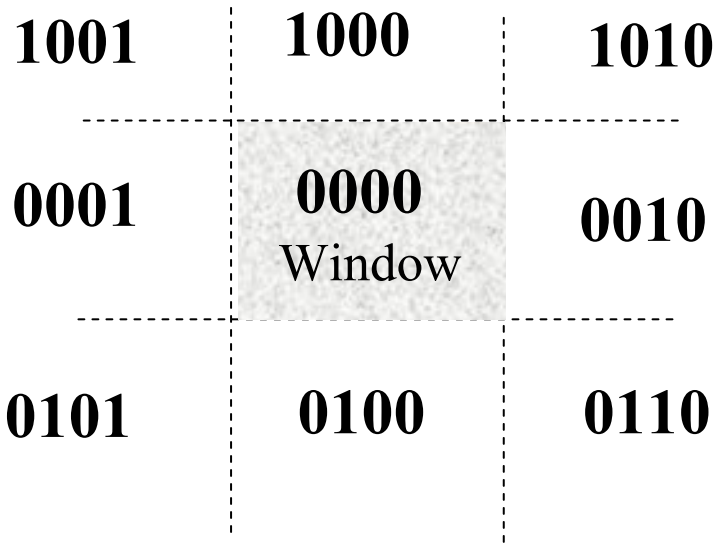
- Need an algorithm to work out
  - whether lines are in, out or across the window
  - if they cross the window, need the intersection point



Cohen-Sutherland Line Clipping

- Step 1 - Region codes used to establish if line is in, out or across window

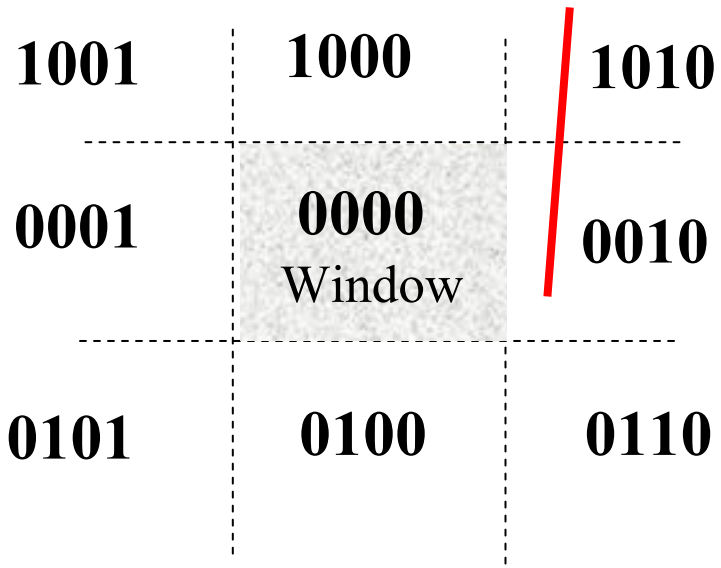
Assign line ends a binary code - on their position relative to the window.



Binary (Base 2)				
above	below	right	left	Position of End points
1	**0**	**0**	**0**	Above
0	**0**	**1**	**0**	Right
0	**1**	**0**	**0**	Below
0	**0**	**0**	**1**	Left
1	**0**	**1**	**0**	Above and right
0	**1**	**1**	**0**	Below and right
1	**0**	**1**	**0**	Above and right
0	**1**	**0**	**1**	Below and left
0	**0**	**0**	**0**	In window

Cohen-Sutherland 2

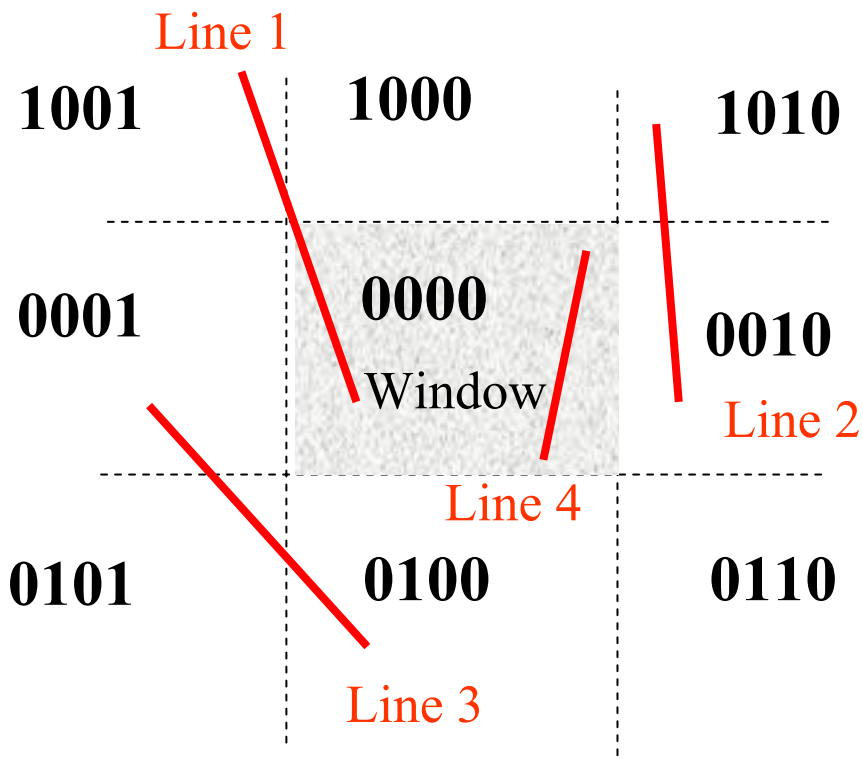
- Step 2 – If both lines have code 0000 - accept
- Step 3 – If AND of codes is 0000 – check for intersection
- Step 4 – Else - reject



AND

1010
0010
0010 so reject

Practice AND



Line 1

1001
0000 AND
0000 check for intersection

Line 2

1010
0010 AND
0010 reject

Line 3

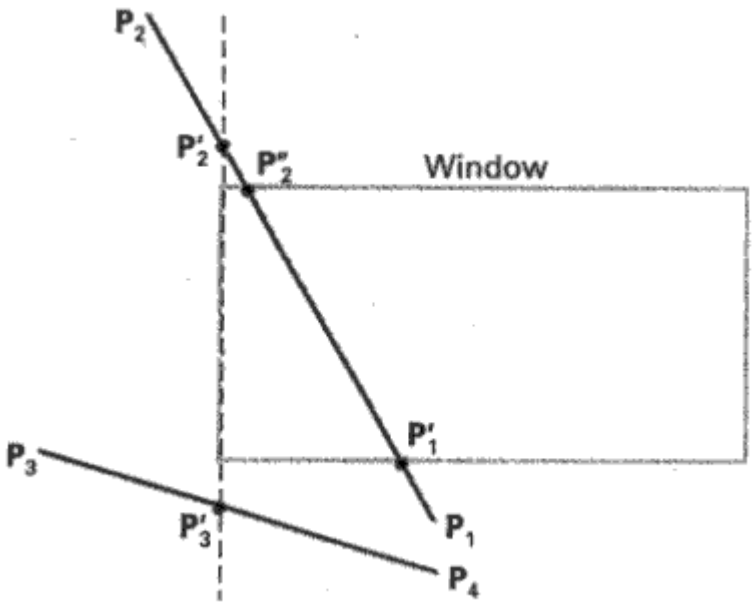
0001
0100 AND
0000 check for intersection

Line 4

0000
0000
Both 0000 so accept

Boundary intersections

- Checking for intersections



Consider the line P_1 to P_2

P_1 is below the boundary. Find intersection point P_1'. This point is on the boundary so save it.

P_2 is to the left of the boundary. Find intersection point P_2' this is outside the boundary so find point P_2''. This point is on the boundary so save it.

Save the portion of the line from P_1 to P_2''. Repeat the procedure for P_3 to P_4, reject this line

Finding boundary intersections

- Y coordinate of intersection with vertical boundary is

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

Where x value is set to xw_{min} or xw_{max}. x_1 and y_1 are the end point coordinates and m=gradient

- x coordinate of intersection with horizontal boundary is

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

Where y value is set to yw_{min} or $yw_{max.}$ x_1 and y_1 are the end point coordinates and m=gradient

Worked Example

A clipping window has opposite corners at 10,10 and 100,60. Show how the Cohen-Sutherland algorithm would clip a line from 20,5 to 80,20.

Answer

First sketch the problem

Worked Example cont

Codes are 0100 and 0000.

0100
AND 0000
0000 so part of this line may be inside the window.

Need to clip the line against the lower boundary.

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

So $x = 20 + \frac{10-5}{0.25}$

So $x = 40$

The line would be clipped below 40,10

Find the gradient

$$\frac{20-5}{80-20} = \frac{15}{60} = 0.25$$

Line Clipping Problems

1 A clipping window has opposite corners at 0,0 and 20,10. Show how the Cohen-Sutherland algorithm would clip a line from 10,5 to 25, 20.(Ans 15,10)

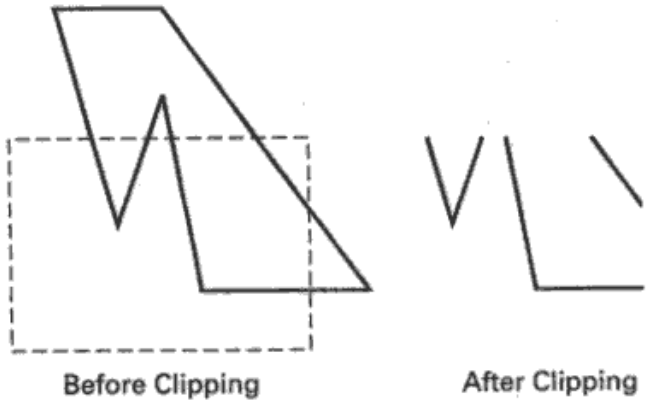
2 A clipping window has opposite corners at 10,10 and 80,50. Show how the Cohen-Sutherland algorithm would clip a line from 20,20 to 100,40.(Ans 80,35)

Polygon Clipping

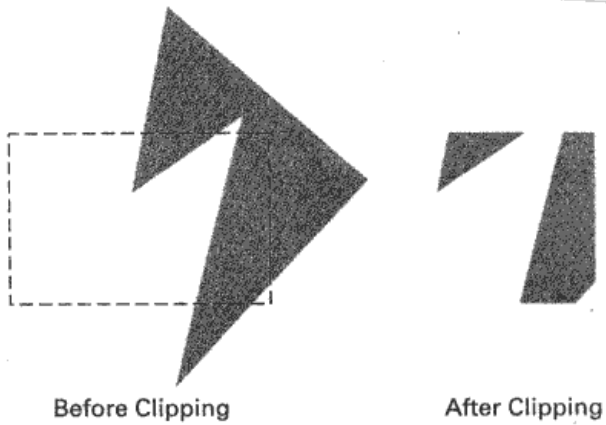
- Need to modify the line algorithm to clip polygons.

Incorrectly clipped

Polygon clipped using line clipping Techniques.

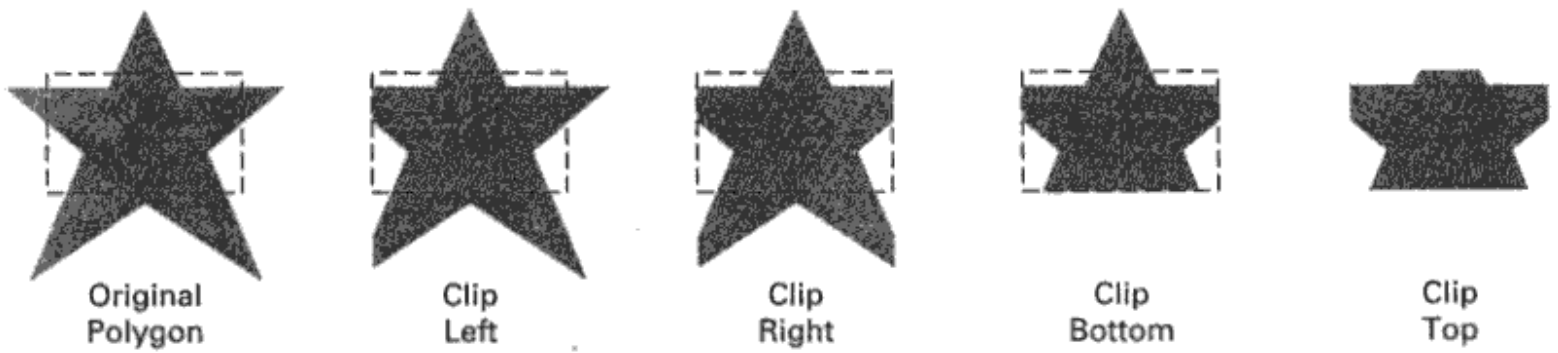


Correctly clipped



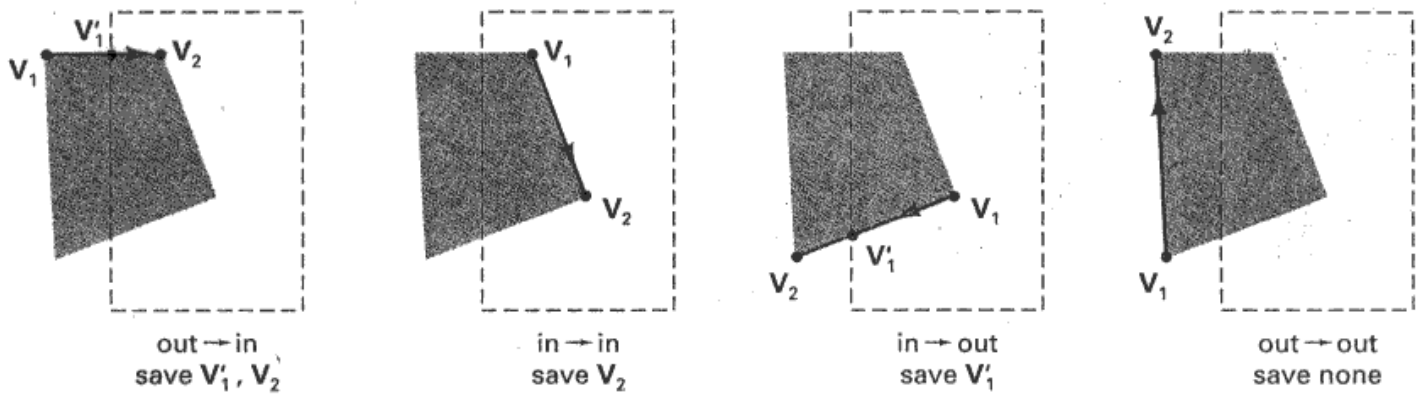
Polygon Clipping

- Sutherland-Hodgeman Polygon Clipping
  - Process the polygon boundary as a whole against each window edge
  - start by clipping the polygon against the left boundary to produce a new clipped left edge for the polygon, then right, bottom and finally top
  - last step go round the sides in turn next slide



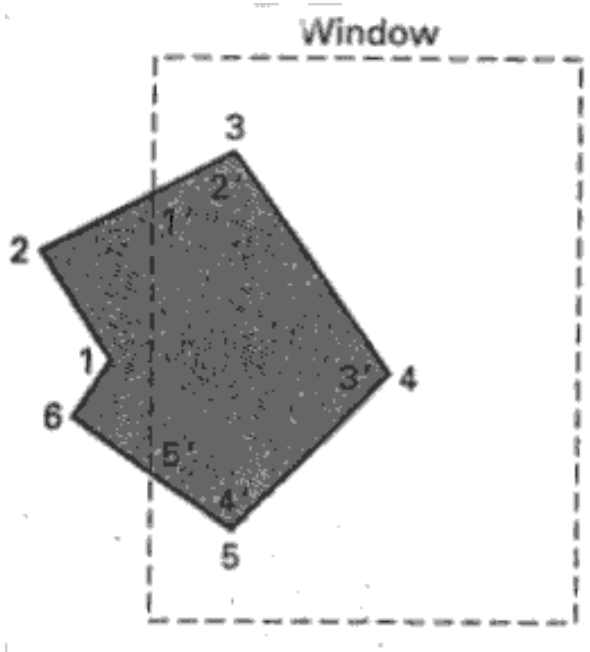
Clipping Procedure

- Go round the sides in turn
  - if line goes out to in, keep intersection and and inside point
  - if line is entirely inside, keep destination point
  - if line goes from in to out, keep intersection
  - if line is entirely outside, do not keep any points



Clipping Example

1 to 2	out to out	keep no points
2 to 3	out to in	keep 1’ and 2’
3 to 4	……..	
4 to 5		
5 to 6		
6 to 1		



Final clipped, filled shape has vertices 1’,3,4,5,5’

Polygon Clipping Problem

Question 1

Three lines are drawn

Line 1 from 110,20 to 100,60

line2 from 100,60 to 70,40

line 3 from 70,40 to 110,20

Show how these lines would be clipped by a window having opposite corners at 20,10 and 90,80. Use Cohen-Sutherland.

Suppose these three lines formed a filled polygon. Show how the Sutherland-Hodgeman procedure would handle this polygon.

Answer

Line 2 clipped at 90,53.3. Line 3 clipped at 30,90.

Filled polygon vertices at (90,53.3) (70,40) and (30,90)