Chapter-3: Remaining of Clipping

Clipping

The process of discarding those parts of a picture which are outside of a specified region or window is called clipping. The procedure using which we can identify whether the portions of the graphics object is within or outside a specified region or space is called clipping

Algorithm.

The region or space which is used to see the object is called window and the region on which the object is shown is called view port. Clipping is necessary to remove those portions of the object which are not necessary for further operations. It excludes unwanted graphics from the screen. So, there are three cases:

1. The object may be completely outside the viewing area defined by the window port.

2. The object may be seen partially in the window port.

3. The object may be seen completely in the window port.

For case 1 & 2, clipping operation is necessary but not for case 3.

1. Point Clipping

Assuming that the clip window is a rectangle in standard position, we save a point P=(x, y) for display if the following inequalities are satisfied:

$$xwmin \leq x \leq xwmax$$

$$ywmin \leq y \leq ywmax$$

Where the edges of the clip window ($xwmin$, $xwmax$, $ywmin$, $ywmax$) can be either the world coordinate window boundaries or viewport boundaries. If any one of these four inequalities is not satisfied, the point is clipped.

Algorithm:

1. Get the minimum and maximum coordinates of the viewing plane.

2. Get the coordinates for a point.

3. Check whether given input lies between minimum and maximum coordinates of viewing plane.

4. If yes display the point which lies inside the region otherwise discard it.

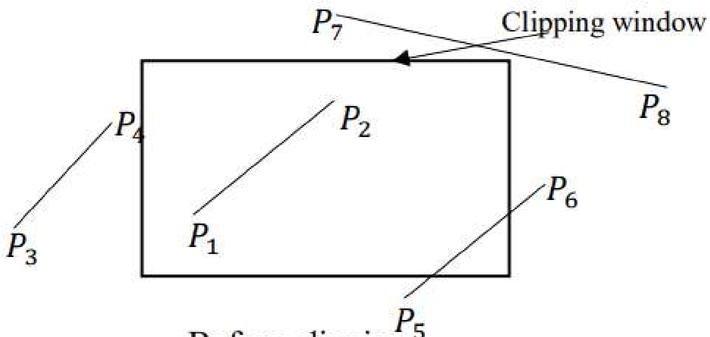
2. Line Clipping

In line clipping, a line or part of line is clipped if it is outside the window port. There are three possibilities for the line:

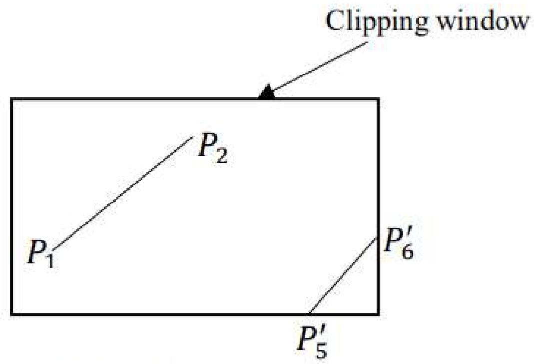
a. Line can be completely inside the window (This line should be accepted).

b. Line can be completely outside of the window (This line will be completely removed from the region).

c. Line can be partially inside the window (We will find intersection point and draw only that portion of line that is inside region).



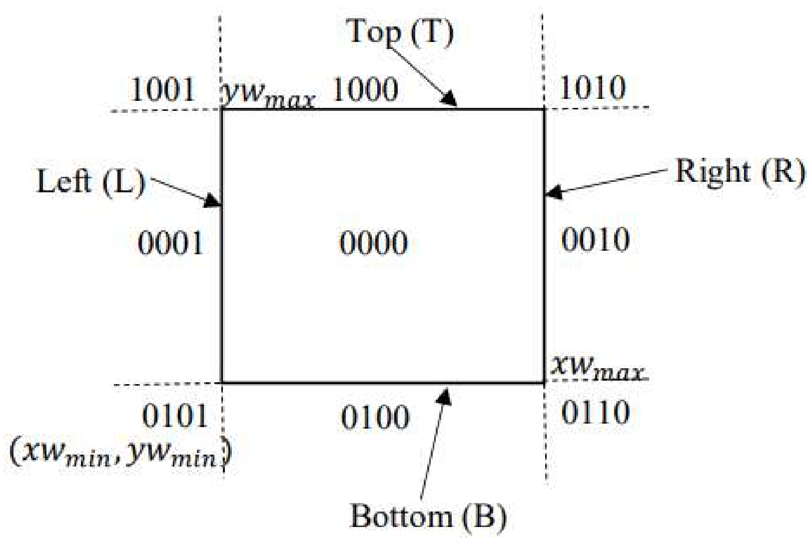
Before clipping



After clipping

Cohen-Sutherland Line Clipping Algorithm

1. Assign the four digit binary value called region code to each end point of a line.

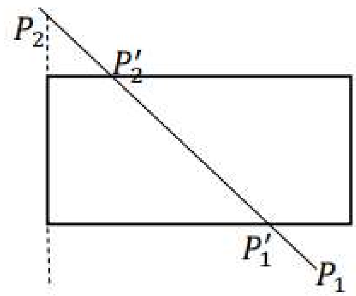


- A region code is represented as TBRL with 0000 inside clipping window.
- To calculate region code, perform following steps:
    a) Calculate the difference between endpoint coordinates and clipping boundary i.e. $x - xw_{min}, xw_{max} - x, y - yw_{min}$ & $yw_{max} - y$.
    b) Use '1' when resultant sign is $-ve$ otherwise use '0'.

2. Determine which lines are completely inside the clipping window & which lines are completely outside.

- Perform OR operation on line endpoint region code, if we get 0000, the line is completely inside clipping window & save these points.
- Perform AND operation on line endpoint region code, if we get value not equal to 0000, the line is completely outside & discard these points.

3. If condition 2 fails, the line crosses the clipping window boundary & find the point of intersection.



- Windows edges are processed in left, right, top and bottom. Here, the region code for point P_1 is 0100 & P_2 is 1001.
- To decide which boundary edges the line crosses, check for the bit position in line endpoint.
- Line crosses these window boundary edges for which bit position value are opposite.

Here,

	T	B	R	L
P1:	0	1	0	0
P2:	1	0	0	1

Since, the value at T, B & L are opposite, the line P1P2 crosses the clipping window at top, bottom & left edge.

- Now find the point of intersection with the clipping window edge.
- For calculation of intersection point, first find the slope,

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

For bottom edge intersection point; $y = yw_{min}$

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

For top edge intersection point; $y = yw_{max}$

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

For left edge intersection point; $x = xw_{min}$

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

For right edge intersection point; $x = xw_{max}$

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