These two functions perform the workstation transformation by selecting a workstation window-viewport pair. If a workstation viewport is not specified, the normalized reference frame is mapped onto the largest square area possible on an output device. The transformation maps the origin of the normalized space to origin of device coordinates and retains the aspect ratio.

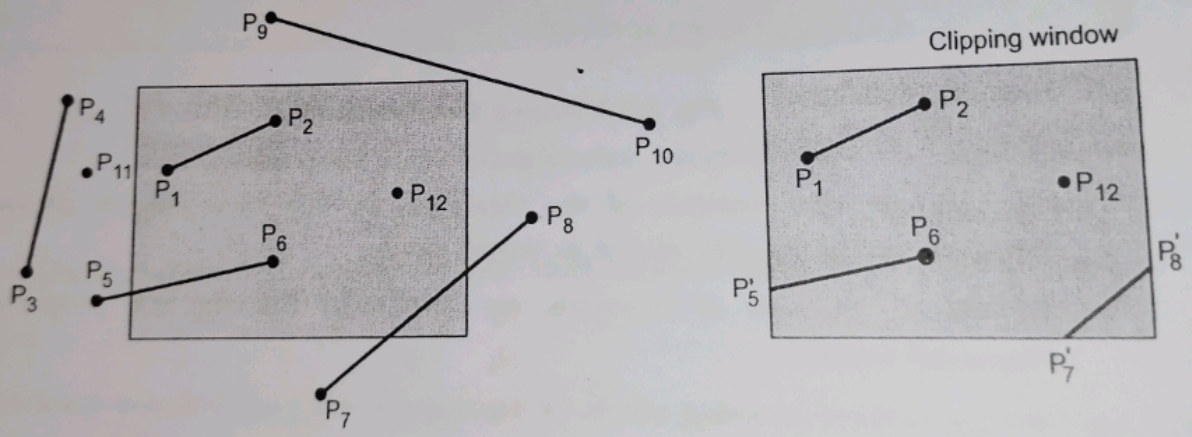
Review Question

1. *Write a short note on viewing functions.*

AU : May-13

5.7 2 D Clipping

- The procedure that identifies the portions of a picture that are either inside or outside of a specified region of space is referred to as clipping.
- The region against which an object is to be clipped is called a **clip window** or **clipping window.** It usually is in a rectangular shape, as shown in the Fig. 5.7.1.
- The clipping algorithm determines which points, lines or portions of lines lie within the clipping window. These points, lines or portions of lines are retained for display. All others are discarded.



(a) Before clipping (b) After clipping

Fig. 5.7.1

5.7.1 Point Clipping

- The points are said to be interior to the clipping window if

  $x_{wmin} \leq x \leq x_{wmax}$ and

  $y_{wmin} \leq y \leq y_{wmax}$

The equal sign indicates that points on the window boundary are included within the window.

5.7.2 Line Clipping

- The lines are said to be interior to the clipping window and hence visible if both end points are interior to the window, e.g. line P_1P_2 in Fig. 5.7.1.
- If both end points of a line are exterior to the window, the line is not necessarily completely exterior to the window, e.g. line P_7P_8 in Fig. 5.7.1.
- If both end points of a line are completely to the right of, completely to the left of, completely above, or completely below the window, then the line is completely exterior to the window and hence invisible. For example, line P_3P_4 in Fig. 5.7.1.
- The lines which across one or more clipping boundaries require calculation of multiple intersection points to decide the visible portion of them.
- To minimize the intersection calculations and to increase the efficiency of the clipping algorithm, initially, completely visible and invisible lines are identified and then the intersection points are calculated for remaining lines.
- There are many line clipping algorithms. Let us discuss a few of them.

Review Questions

1. *What is point clipping and line clipping ?*
2. *How will you clip a point ?*

AU : May-13, Marks 2

5.8 Cohen-Sutherland Line Subdivision Clipping Algorithm

AU : May-12, Dec.-12

- This is one of the oldest and most popular line clipping algorithm developed by Dan Cohen and Ivan Sutherland.
- To speed up the processing this algorithm performs initial tests that reduce the number of intersections that must be calculated.
- This algorithm uses a four digit (bit) code to indicate which of nine regions contain the end point of line.
- The four bit codes are called **region codes** or **outcodes**. These codes identify the location of the point relative to the boundaries of the clipping rectangle as shown in the Fig. 5.8.1.

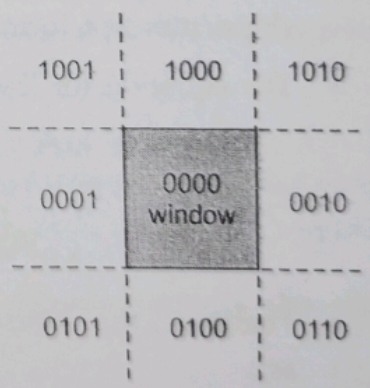


Fig. 5.8.1 Four-bit codes for nine regions

- Each bit position in the region code is used to indicate one of the four relative co-ordinate positions of the point with respect to the clipping window : To the

left, right, top or bottom. The rightmost bit is the first bit and the bits are set to 1 based on the following scheme :

- Set Bit 1 - If the end point is to the **left** of the window
- Set Bit 2 - If the end point is to the **right** of the window
- Set Bit 3 - If the end point is **below** the window
- Set Bit 4 - If the end point is **above** the window

Otherwise, the bit is set to zero.

- Once we have established region codes for all the line endpoints, we can determine which lines are completely inside the clipping window and which are clearly outside.
- Any lines that are completely inside the window boundaries have a region code of 0000 for both endpoints and we trivially accept these lines.
- Any lines that have a 1 in the same bit position in the region codes for each endpoint are completely outside the clipping rectangle and we trivially reject these lines.
- A method used to test lines for total clipping is equivalent to the logical **AND** operator.
- If the result of the logical AND operation with two end point codes is not 0000, the line is completely outside the clipping region.
- The lines that cannot be identified as completely inside or completely outside a clipping window by these tests are checked for intersection with the window boundaries.

Example 5.8.1 *Consider the clipping window and the lines shown in Fig. 5.8.2. Find the region codes for each end point and identify whether the line is completely visible, partially visible or completely invisible.*

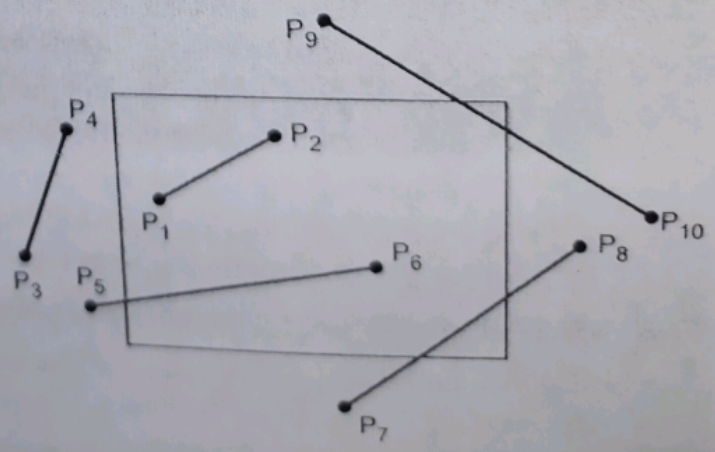


Fig. 5.8.2

Solution : The Fig. 5.8.3 shows the clipping window and lines with region codes. These codes are tabulated and end point codes are logically ANDed to identify the visibility of the line in Table 5.8.1.

Sutherland and Cohen subdivision line clipping algorithm :

1. Read two end points of the line say $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$.
2. Read two corners (left-top and right-bottom) of the window, say (W_{x1}, W_{y1}) and (W_{x2}, W_{y2}).
3. Assign the region codes for two endpoints P_1 and P_2 using following steps :

   Initialize code with bits 0000

   Set Bit 1 - if($x < W_{x1}$)

   Set Bit 2 - if ($x > W_{x2}$)

   Set Bit 3 - if ($y < W_{y2}$)

   Set Bit 4 - if ($y > W_{y1}$)

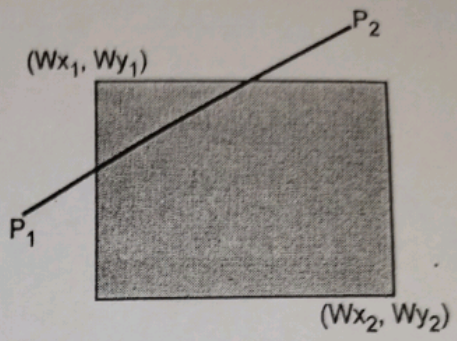


Fig. 5.8.5

4. Check for visibility of line P_1P_2
   a) If region codes for both endpoints P_1 and P_2 are zero then the line is completely visible. Hence draw the line and go to step 9.
   b) If region codes for endpoints are not zero and the logical ANDing of them is also non-zero then the line is completely invisible, so reject the line and go to step 9.
   c) If region codes for two endpoints do not satisfy the conditions in (4a) and (4b) the line is partially visible.
5. Determine the intersecting edge of the clipping window by inspecting the region codes of two endpoints.
   a) If region codes for both the end points are non-zero, find intersection points P_1' and P_2' with boundary edges of clipping window with respect to point P_1 and point P_2, respectively
   b) If region code for any one end point is non-zero then find intersection point P_1' or P_2' with the boundary edge of the clipping window with respect to it.
6. Divide the line segments considering intersection points.
7. Reject the line segment if any one end point of it appears outsides the clipping window.
8. Draw the remaining line segments.
9. Stop.