Clipping points and Lines

By

Prof. Vaibhav P. Vasani

Assistant Professor

Department of Computer Engineering

K. J. Somaiya College of Engineering

Somaiya Vidyavihar University

Point Clipping

- Point clipping is essentially the evaluation of the following inequalities:

$$x_{min} \le x \le x_{max} \quad \text{and} \quad y_{min} \le y \le y_{max}$$

- where Xmin, Xmax, Ymin and Ymax define the clipping window. A point (x, y) is considered inside the window when the
inequalities all evaluate to true.

Clipping Lines

- In an OpenGL environment each object is automatically clipped to the world window using a particular algorithm

Clipping a Line

- A classic line-clipping algorithm ,the Cohen-Sutherland clipper

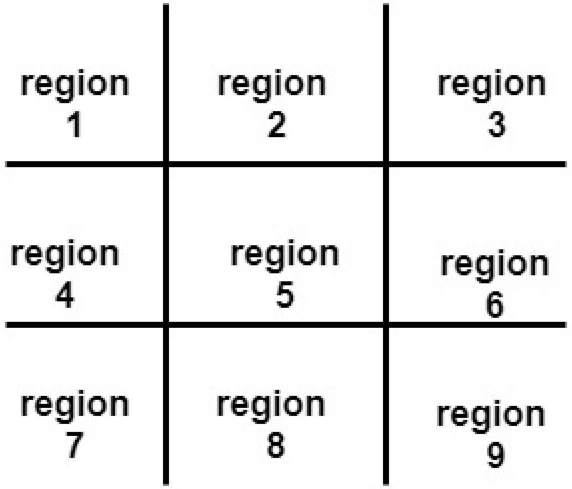
Cohen Sutherland Line Clipping Algorithm

- In the algorithm, first of all, it is detected whether line lies inside the screen or it is outside the screen. All lines come under any one of the following categories:
    - Visible
    - Not Visible
    - Clipping Case
- **1. Visible:** If a line lies within the window, i.e., both endpoints of the line lies within the window. A line is visible and will be displayed as it is.

- **Not Visible:** If a line lies outside the window it will be invisible and rejected. Such lines will not display. If any one of the following inequalities is satisfied, then the line is considered invisible.
- Example:
  - Let A (x_1,y_2) and B (x_2,y_2) are endpoints of line.
  - x_{min}, x_{max} are coordinates of the window.
  - y_{min}, y_{max} are also coordinates of the window.

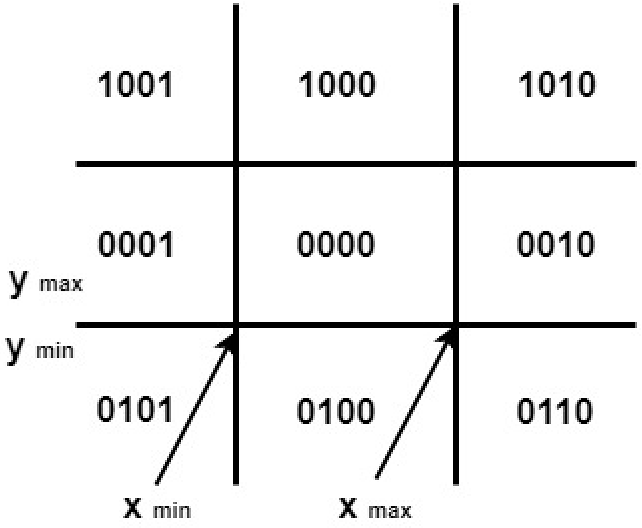
$x_1 > x_{max}$
$x_2 > x_{max}$
$y_1 > y_{max}$
$y_2 > y_{max}$
$x_1 < x_{min}$
$x_2 < x_{min}$
$y_1 < y_{min}$
$y_2 < y_{min}$

- **3. Clipping Case:**
  - If the line is neither visible case nor invisible case. It is considered to be clipped case.
  - The category of a line is found based on nine regions given below. All nine regions are assigned codes.
  - Each code is of 4 bits. If both endpoints of the line have end bits zero, then the line is considered to be visible.



region 1	region 2	region 3
region 4	region 5	region 6
region 7	region 8	region 9

9 region

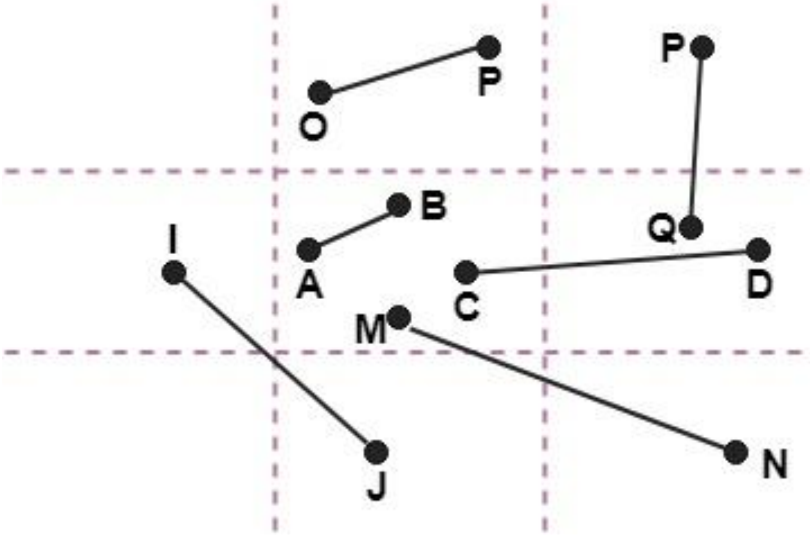


1001	1000	1010
0001	0000	0010
0101	0100	0110

bits assigned to 9 regions

- The center area is having the code, 0000, i.e., region 5 is considered a rectangle window.

- Line AB is the visible case
- Line OP is an invisible case
- Line PQ is an invisible line
- Line IJ are clipping candidates
- Line MN are clipping candidate
- Line CD are clipping candidate



Advantage of Cohen Sutherland Line Clipping

- It calculates end-points very quickly and rejects and accepts lines quickly.
- It can clip pictures much large than screen size.

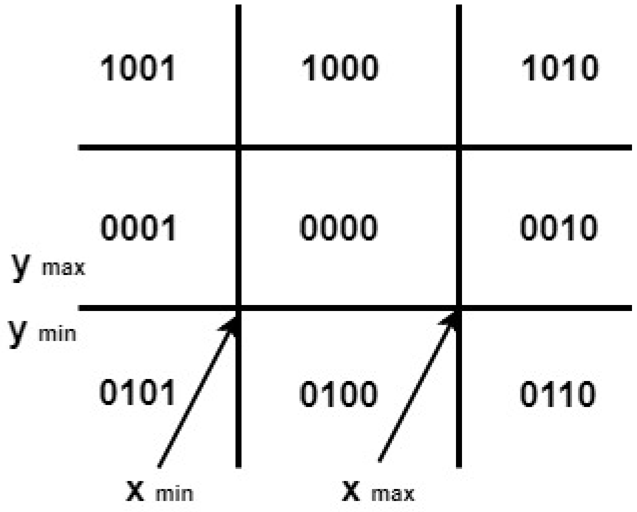
Algorithm of Cohen Sutherland Line Clipping

Step1: Calculate positions of both endpoints of the line

Step2: Perform OR operation on both of these end-points

Step3: If the OR operation gives 0000

```
      Then
              line is considered to be visible
      else
         Perform AND operation on both endpoint
     If And ≠ 0000
         then the line is invisible
       else
     And=0000
   Line is considered the clipped case.
```



1001	1000	1010
0001	0000	0010
0101	0100	0110

Algorithm of Cohen Sutherland Line Clipping

Step4: If a line is clipped case, find an intersection with boundaries of the window

$m=(y_2-y_1)(x_2-x_1)$

(a) If bit 1 is "1" line intersects with left boundary of rectangle window

$y_3=y_1+m(x-X_1)$

where $X = X_{wmin}$

where X_{wmin} is the minimum value of X co-ordinate of window

(b) If bit 2 is "1" line intersect with right boundary

$y_3=y_1+m(X-X_1)$

where $X = X_{wmax}$

where X more is maximum value of X co-ordinate of the windo

(c) If bit 3 is "1" line intersects with bottom boundary

$X_3=X_1+(y-y_1)/m$

where $y = y_{wmin}$

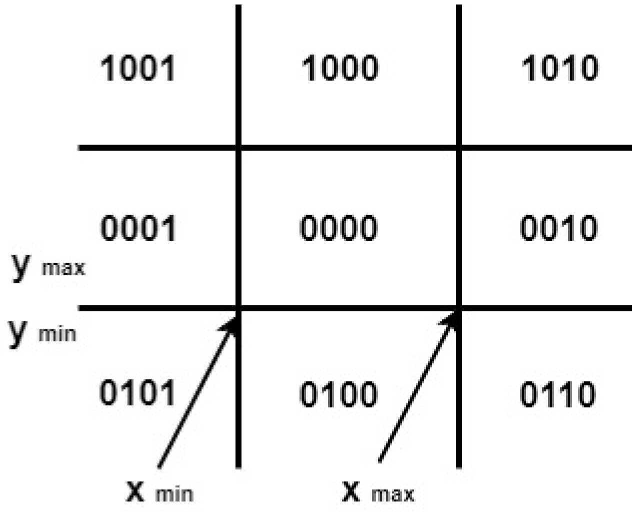
y_{wmin} is the minimum value of Y co-ordinate of the window

(d) If bit 4 is "1" line intersects with the top boundary

$X_{3=X}1+(y-y_1)/m$

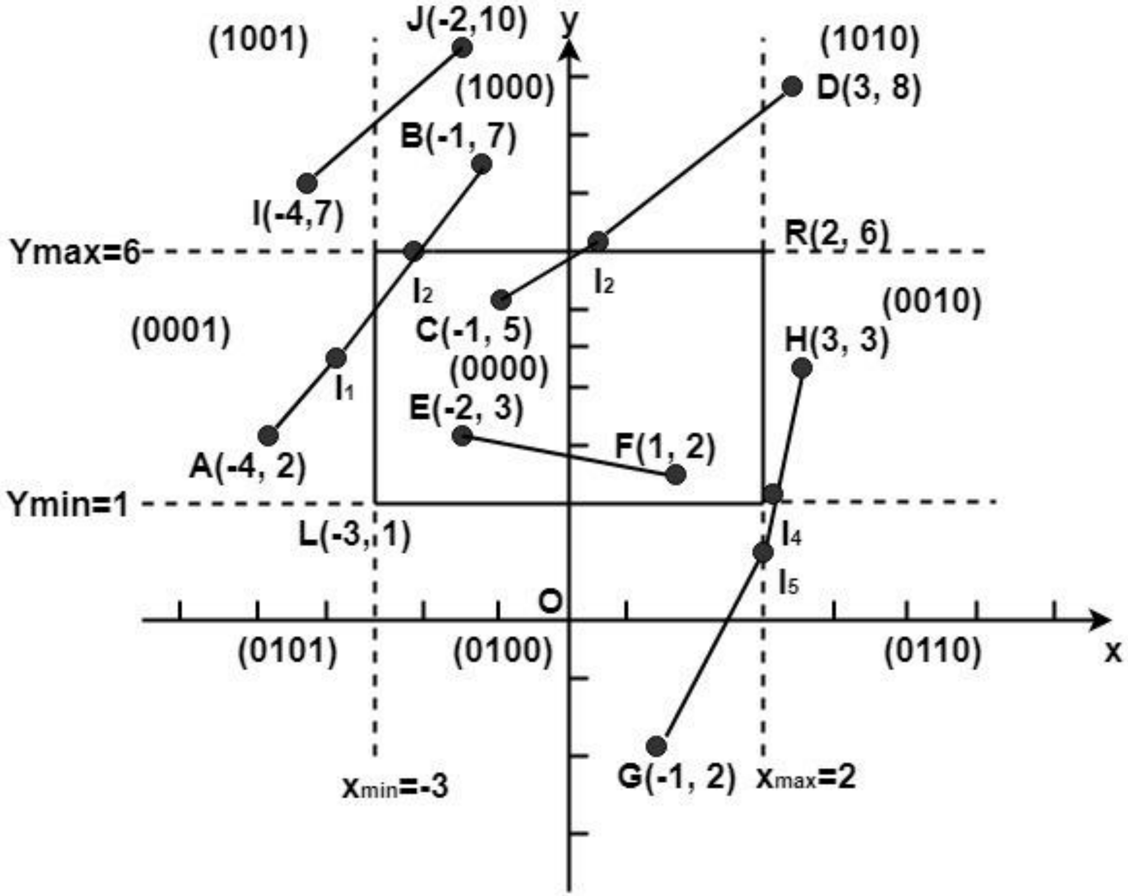
where $y = y_{wmax}$

y_{wmax} is the maximum value of Y co-ordinate of the window



Example of Cohen-Sutherland Line Clipping Algorithm

- Let R be the rectangular window whose lower left-hand corner is at L (-3, 1) and upper right-hand corner is at R (2, 6). Find the region codes for the endpoints in fig:



- The region code for point (x, y) is set according to the scheme
  Bit 1 = sign $(y-y_{max})$=sign $(y-6)$ Bit 3 = sign $(x-x_{max})$= sign $(x-2)$
  Bit 2 = sign $(y_{min}-y)$=sign$(1-y)$ Bit 4 = sign $(x_{min}-x)$=sign$(-3-x)$
- Here

$$\left\{ \begin{array}{lll} \text{sign (a) =} & 1 & \text{if a is positive} \\ & 0 & \text{otherwise} \end{array} \right\}$$

-

A (-4, 2)→ 0001	F (1, 2)→ 0000
B (-1, 7) → 1000	G (1, -2) →0100
C (-1, 5)→ 0000	H (3, 3) → 0100
D (3, 8) → 1010	I (-4, 7) → 1001
E (-2, 3) → 0000	J (-2, 10) → 1000

- **Category1 (visible):** EF since the region code for both endpoints is 0000.
- **Category2 (not visible):** IJ since (1001) AND (1000) =1000 (which is not 0000).
- **Category 3 (candidate for clipping):** AB since (0001) AND (1000) = 0000, CD since (0000) AND (1010) =0000, and GH. since (0100) AND (0010) =0000.
- The candidates for clipping are AB, CD, and GH.

- In clipping AB, the code for A is 0001. To push the 1 to 0, we clip against the boundary line x_{min}=-3. The resulting intersection point is I_1 (-3,3). We clip (do not display) AI_1 and I_1 B. The code for $I_1$is 1001. The clipping category for I_1 B is 3 since (0000) AND (1000) is (0000). Now B is outside the window (i.e., its code is 1000), so we push the 1 to a 0 by clipping against the line y_{max}=6. The resulting intersection is l_2 (-1,6). Thus I_2 B is clipped. The code for I_2 is 0000. The remaining segment I_1 I_2 is displayed since both endpoints lie in the window (i.e., their codes are 0000).
- For clipping CD, we start with D since it is outside the window. Its code is 1010. We push the first 1 to a 0 by clipping against the line y_{max}=6. The resulting intersection I_3 is (,6),and its code is 0000. Thus I_3 D is clipped and the remaining segment CI_3 has both endpoints coded 0000 and so it is displayed.
- For clipping GH, we can start with either G or H since both are outside the window. The code for G is 0100, and we push the 1 to a 0 by clipping against the line y_{min}=1.The resulting intersection point is I_4 (2,1) and its code is 0010. We clip GI_4 and work on I_4 H. Segment I_4 H is not displaying since (0010) AND (0010) =0010.

Thank you