

Title: Polygon Line Clipping

Problem Statement: Write a C++/Java program to implement line clipping algorithm for given window. Draw a line using the mouse interfacing to draw polygon.

Learning objective:

To learn and implement the Cohen-Sutherland line clipping algorithm.

Learning outcome:

- 1) We should be able to implement the Cohen-Sutherland line clipping algorithm.
- 2) We should be able to understand the line drawing and concept of line clipping.

Theory:

Digital Differential Analyzer (DDA):

In any 2-D plane if we connect two points (x_1, y_1) and (x_2, y_2) we get line segment. But in computer graphics we can directly join two coordinates point for that we need to calculate intermediate coordinates.

Algorithm:

Integer-integer function

sign: return 1, 0, -1 for argument.

Step 1: Read end points (x_1, y_1) and (x_2, y_2) .

Step 2: approx the length of line.

if ($\text{abs}(x_2 - x_1) \geq \text{abs}(y_2 - y_1)$)
 $\text{length} = \text{abs}(x_2 - x_1);$

else

$\text{length} = \text{abs}(y_2 - y_1);$

Step 3: Select raster unit

$A_x = (x_2 - x_1) / \text{length}$

$A_y = (y_2 - y_1) / \text{length}$

Step 4: Round the values.

$x = x_1 + 0.5 * \text{sign}(A_x)$

$y = y_1 + 0.5 * \text{sign}(A_y)$

Step 5: Plot the pixel $i = 1$

while ($i \leq \text{length}$)

{ setpixel (Integer(x), Integer(y));

$x = x + A_x$

$y = y + A_y$

$i = i + 1;$

}

* Visibility of line:

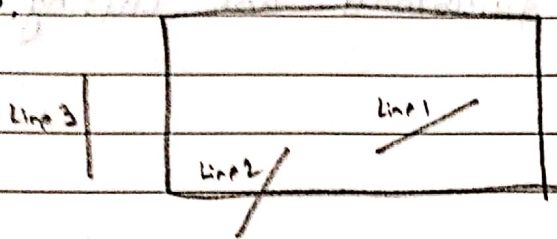
1) Completely visible:

When both endpoints of line are lies completely inside or at point of window intersection then line called as completely visible.

2) Partially visible:

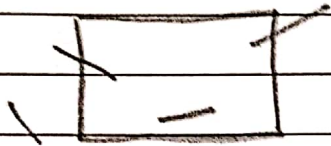
When one of the endpoint of line lies inside the window and other lies the window and other lies line partially visible.

- 3) Completely visible:-
When both end points lies outside the window then line is completely invisible.

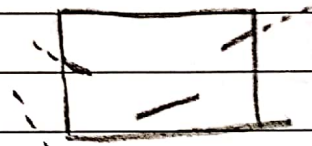


Line 1 is completely visible
Line 2 is partially visible
Line 3 is completely invisible.

* Visibility Algorithms:-
Before Algorithm



After Algorithm



* End point codes:-

Cohen and Sutherland technique uses 4 bit code to indicate nine regions which contains end point.

↑	1 0 0 1	1 0 0 0	1 0 1 0
top	0 0 0 1	window (0 0 0 0)	0 0 1 0
↓	0 1 0 1	0 1 0 0	0 1 1 0
bottom			
	← left		→ right

* 1) Cohen - Sutherland algorithm end point code to accept or reject line segment.

2) If the line segment is not trivially accepted or rejected then search for end points which is outside the window and segment from this point to the intersection point can always be rejected.

Conclusion:-

Thus from this experiment we have learned and implemented the Cohen-Sutherland line clipping algorithm.

01 01

00 01

1001

01 00

(0000) window

1000

01 10

0010

1010

0110

0111