IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411152

# Cohen Sutherland Line Clipping Algorithm

**Mrs. Pournima Abhishek Kamble[1], Mrs. Sujata Shankar Gawade[2]**

Lecturer, Computer Technology, BVIT, Navi Mumbai, India[1]

Lecturer, Computer Technology, BVIT, Navi Mumbai, India[2]

**Abstract**: Deciding visible and invisible portion of the line and discarding invisible line segments from line is known as Line Clipping.

**Keywords:** Region Codes, Outcodes.

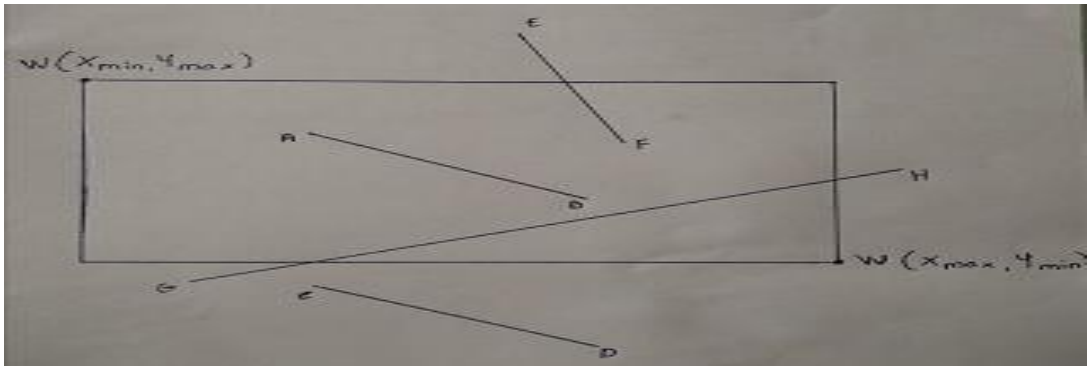
## I. INTRODUCTION

The process of selecting and viewing the picture with different views is called windowing and a process which divides each element of the picture into its visible and invisible portions, allowing the invisible portion to be discarded is called clipping.
There are four algorithms which can be used for line clipping.

1. Cohen Sutherland Line Clipping Algorithm.
2. Cyrus-Beck Line Clipping Algorithm.
3. Liang Barsky Line Clipping Algorithm.
4. Mid-Point Subdivision Line Clipping Algorithm.

## II. LINE CLIPPING

If both end points are interior to the window the line is said to be completely in-side of the window and hence visible.
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 outside to the window and hence invisible.
If both end point of a line are exterior to the window, the line is not necessarily completely outside of the window. Some part of this line is visible and some is invisible. In this case intersection points are calculated and visible portion is decided. To calculate these visible portions Line Clipping algorithms are used.




## III. COHEN SUTHERLAND LINE CLIPPING WORKING

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 reduces 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.

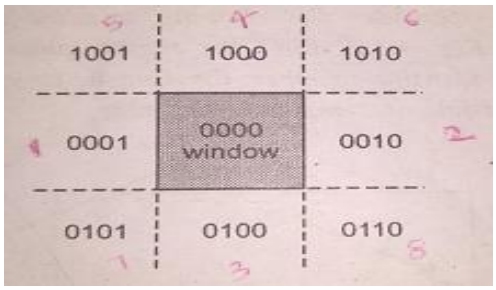
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411152

These codes identify the location of the point relative to the boundaries of the clipping rectangle as shown in the figure.




Each bit position in the region code is used to indicate one of the four relative coordinate 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 to the below of the window.
Set Bit 4 – If the end point is to the above of the window.
Otherwise, the bit is set to zero.

Once we have established region codes for, all the line endpoint, 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.

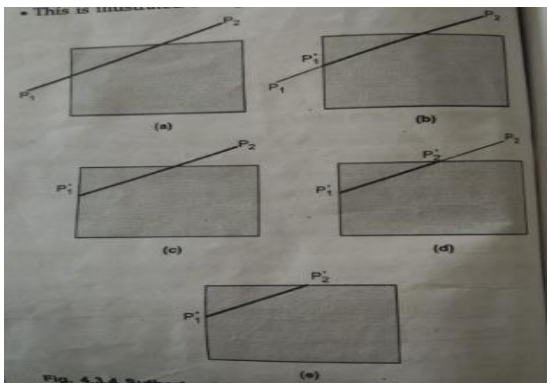
A method used to test lines for total clipping is equivalent to the logical AND operator.

If the result of the logic AND operation with two end point codes is not 0000, then that 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.

The Cohen-Sutherland algorithm begins the clipping process for a partially visible line by comparing an outside endpoint to a clipping boundary to determine how much of the line can be discarded.

Then the remaining part of the line is checked against the other boundaries and the process is continued until either the line is totally discarded, or a section is found inside the window.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411152

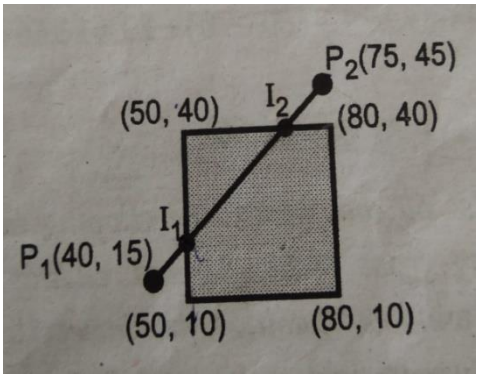
Line p1p2 is a partially visible and p1 is outside the window.
Starting with point p1, the intersection point is found and we get two line segments p1-p1' and p1'-p2.
We know that, for p1-p1' one end point i.e. p1 is outside the window and thus the line segment p1-p1' is discarded.
The line is now reduced to the section p1'-p2.
Since p2 is outside the clip window, it is checked against the boundaries and intersection point p2' is found.
Again the line segment is divided into two segments giving p1'-p2' and p2'-p2.
We know that, for p2'-p2 one end point i.e. p2 is outside the window and thus the line segment p2'-p2 is discarded.
The remaining line segment p1'-p2' is completely inside the clipping window and hence made visible.

## IV. COHEN SUTHERLAND LINE CLIPPING ALGORITHM

Step 1: Read two end points of the line say p1(x1,y1) and p2(x2,y2).
Step 2: Read two corners (left-top and right-bottom) of the window as (xL,yT) and (xR,yB).
Step 3: Assign the region codes for two end points p1 an dp2 using following steps:
Initialize code with bits 0000
Set Bit 1 - if(x<xL)
Set Bit 2 - if(x>xR)
Set Bit 3 - if(y<yB)
Set Bit 4 - if(y>yT)
Step 4: Check for visibility of line p1p2
a) If region code for both endpoints p1 and p2 are 0000 then the line is completely visible. Hence draw the line and go to step 9.
b) If region codes for endpoints are not 0000 and the logical ANDing of them is also not 0000 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.
Step 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 p1' and p2' with boundary edges of clipping window with respect to point p1 and point p2 respectively.
b) If region code for any one end point is non-zero then find intersection point p1' or p2' with the boundary edge of the clipping window with respect to it.
Step 6: Divide the line segments considering intersection points.
Step 7: Reject the line segment if any one end point of it appears outsides the clipping window.
Step 8: Draw the remaining line segments.
Step 9: Stop.

## V. EXAMPLE

Use the Cohen-Sutherland outcode algorithm to clip two lines p1(40,15)-p2(75,45) and p3(70,20)-p4(100,10) against a window A(50,10), B(80,10), c(80,40) and D(50,40).




For Line 1: xL=50, xR=80, yT=40, yB=10
x1=40, y1=15, x2=75, y2=45
m=(y2-y1)/(x2-x1)
=30/35
=6/7

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.1411152

I1(xL,y)=(50,y)=y1+m(xL-x1)
=15+6/7(10)
=15+8.57
=23.57

**Thus I1 is (50,23.57)**

I2(x,yT)=(x,40)=x1+(1/m)(yT-y1)
=40+(7/6)(25)
=40+29.167
=69.167

**Thus I2 is (69.167,40)**

For Line 2: xL=50, xR=80, yT=40, yB=10
x1=70, y1=20, x2=100, y2=10
m=(y2-y1)/(x2-x1)
= (-10)/(30)
= -1/3
I2(xR,,y)=(80,y)=y1+m(xR-x1)
=20+ (-1/3) (10)
=20-3.33
=16.667

**Thus I2 is (80,16.667)**

## VI. CONCLUSION

Cohen Sutherland Line Clipping Algorithm is the oldest and most popular line clipping algorithm used for clipping the lines against a given viewing window.

## REFERENCES

[1]. Donald Hearn, Baker M. Pauline, “Computer Graphics”, Pearson Education, New Delhi, June 2012, ISBN: 817758765X.
[2]. Maurya Rajesh K., “Compute Graphics”, Wiley-India 2011, Delhi ISBN: 978-81-265-3100-4.
[3]. Dr. Chopra Rajiv, “Computer Graphics”, S. Chand 2016, New Delhi, ISBN: 978-93-856-7633-8.
[4]. Foley James, “Computer Graphics principles and practices”, Pearson Education, New Delhi 2014, ISBN: 978-0-321-39952-6.