International Journal of Advance Engineering and Research Development (IJAERD)
Special Issue, Volume 1,Issue 4, April 2014, e-ISSN: 2348 - 4470 , print-ISSN:2348-6406

Survey Of Line Clipping Algorithm In Computer Graphics

Pamal Parekh[1], Vatsal Shah[2], Jayna Donga[3]

[1]IT Department, BVM Eng. College, pamalparekh@yahoo.in
[2]Assistant Professor, BVM Eng. College, vatsal.shah@bvmengineering.ac.in
[3] Assistant Professor, MBICT College, jaynadonga@yahoo.com

Abstract—Line clipping algorithm is represents of clipping line segment in rectangular window. There are basically three algorithm namely cohen-sutherland line clipping algorithm, Liang-Barskey Line Clipping and Nicholl-Lee-Nicholl Line Clipping.

Keywords- Line clipping, Outcode, Algorithm, Intersection test, clipping window.

I. INTRODUCTION

Line clipping are possible relationships between line positions and a standard rectangular clipping region. It is consider several part, like we can test given line segment are inside or not in the clipping region. Line clipping is useful in geographic information system, VLSI circuits design, designing building architecture etc. the survey is organized as: Intersection test, three line clipping algorithms and conclusion.

II. INTERSECTION TEST

Let (x1,y1) and (x2,y2) denote two endpoints of a given line segment S.Then the line segment can be described parametric equations are,

$$x = x1 + u\,(x2 - x1)$$
$$y = y1 + u\,(y2 - y1)$$

Where, u varies over the interval $0 \leq u \leq 1$.

Now, if the boundary edge outside the region then line does not enter the interior of the window at that boundary. Else cross into the clipping area.

There are three algorithms.

1. Cohen-Sutherland Line Clipping
2. Liang- Barskey Line Clipping
3. Nicholl-Lee-Nicholl Line Clipping

III. COHEN-SUTHERLAND LINE CLIPPING

It is the oldest and most popular line clipping algorithm. Every line-end point have four binary code, it is called outcode, that identifies position of the line points of the clipping rectangle.[1,2]

0110	0010	0011
0100	0000	0001
1100	1000	1001

Figure 1. Outcode[2]

Code bits are set according to:

First bit is left: $x < xmin$
Second bit is right: $x > xmax$
Third bit is bottom: $y < ymin$
Fourth bit is Top: $y > ymax$
The sequence of code is "LRBT"

Algorithm :

step 1. Read two end points of the line.

step 2.Read boundary points of the clip window.

step 3.Calculate outcode for end point.

step 4.Check whether line is fully inside the clip window, then draw the line. And go to step 7(If outside of both end points are zero, then line is fully inside)

step 5.Calculate whether line is fully outside the clipping region. If so, go to step 7.(If outcodes gives non-zero, the line is fully outside)

step 6.If step 4 and 5 are not satisfied, line is partially inside.

6.1 Find end points which outside of the clip region.
6.2 Check the selected end points(from 6.1) is crossing or intersecting with which clipping boundry.Perform AND operation between outcode of end points and boundry. If line intersect the boundary result will be non-zero. Find intersection point using following formula:

$$x=(y-y_0/m)+x_0 \qquad y=m(x-x_0)+y_0$$

6.3 Replace the selected end points with the intersection points and find outcode of that points.
6.4 Continue till, it determine to reject or accept the line and go to step 4.

step 7.End

IV. LIANG- BARSKEY LINE CLIPPING

This algorithm named after You-Dong Liang and Brian A. Barsky.Liang-Barskey Line Clipping algorithm[2,4] that uses floating-point arithmatic.In general, the this algorithm is more efficient than the Cohen-Sutherland algorithm[1,2] because finds the appropriate end points with reduce calculations.

now, first find the intersection points of the straight line:

$$x = x1 + u\Delta x$$
$$y = y1 + u\Delta y$$

where $\Delta x = x2 - x1$ and $\Delta y = y2 - y1$ when $0 \le u \le 1$

A point clipping conditions are,

$$xmin \le x1 + u\Delta x \le xmax$$
$$ymin \le y1 + u\Delta y \le ymax$$

Which can be expressed as, $up_k \le q_k$, where $k=1,2,3,4$

p and q are defined as ,

$p1 = -\Delta x,$ $q1 = x1 - xmin$

$p2 = \Delta x,$ $q2 = xmax - x1$

$p3 = -\Delta y,$ $q3 = y1 - ymin$

$p4 = \Delta y,$ $q4 = ymax - y1$

algorithm:

step 1.A line parallel to a clipping window edge has pk=0 for that boundary.

step 2. If for that k, qk<0, the line is completely outside and can be eliminated.

step 3. When pk<0 the line proceeds outside to inside the clip window and when pk>0, the line proceeds inside to outside.

step 4. For nonzero pk , u=qk/pk gives the intersection point.

step 5. For each line, calculate u1 and u2. For u1, look at boundaries for which pk<0(outside -> in).Take u1 to be the largest among (0,qk/pk).For u2, look at boundaries for which pk>0 (inside ->out). Take u2 to be the minimum of (1, qk/pk). If u1>u2, the line is outside and therefore rejected.

V. NICHOLL-LEE-NICHOLL LINE CLIPPING

This algorithm avoids multiple clipping of an individual line segment. In this algorithm, the area around the clipping window is divided into a number of different areas, depending on the position of the initial point of the line to be clipped. For intial point mainly three regions are defined. Comparing the above both algorithm, Nicholl-Lee- Nicholl Line clipping algorithm [1,2] is fewer comparisons and divisions.

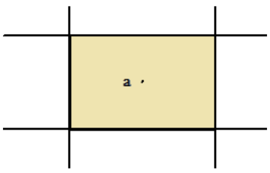

Figure 2(a)
Inside the region[2]

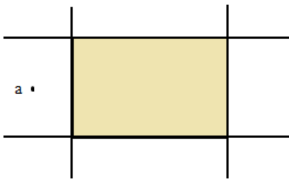

Figure 2(b)
Outside the region
(left side)[2]

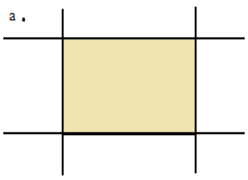

Figure 2(c)
Outside the region
(Top side)[2]

In this algorithm three cases are explained below:

Case 1:
Clipping point p1 is inside the clip rectangle, thus the cross side regions are L, B, R, T.

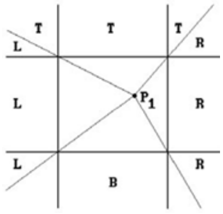

Figure 3(a)[2]

Case 2:
Clipping point p1 is outside edge region (left side),then the cross side regions are LB, LR, LT.

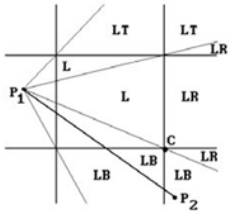

Figure 3(b)[2]

Case 3:
Clipping point p1 is outside corner edge region (top-left side),then the cross side regions are LB, TB, TR.

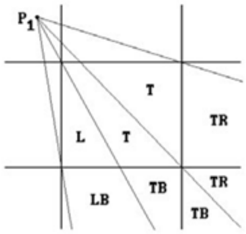

Figure 3(c)[2]

VI. CONCLUSION

The Line Clipping algorithm, removes the unwanted graphics from the region. To start with the Cohen-Sutherland algorithm it is clearly removes the outside the window. The Liang-barskey algorithm is parametric equations of the line segment. And last Nicholl-Lee-Nicholl algorithm is slightly changed of the Liang-barskey algorithm. Nicholl-Lee-Nicholl algorithm is efficient than the other Line clipping algorithm.

REFERENCES

[1] Bimal Kumar Ray, "A Line Segment Clipping Algorithm in 2D", International Journal of Computer Graphics, Vol. 3, No. 2, November, 2012

[2] D. Hearn and P. Baker, "Computer Graphics C version", 2nd ed., Prentice Hall, 2004.

[3] M. Cyrus and J. Beck, "Generalized two - and three-dimensional clipping", Computers & Graphics, vol. 3 ,1978, no. 1.

[4] Y. D. Liang and B. A. Barsky, "A new concept and method for line clipping", ACM Transactions on Graphics, 1984, vol. 3, no. 1.

