ENGINEERING & TECHNOLOGY

Khulna University Studies 3(2) : 509-512

AN ENHANCEMENT OF COHEN-SUTHERLAND ALGORITHM FOR LINE CLIPPING BY REDUCING NEEDLESS CLIPPING

M.A. Rahman*, S.K. Islam, S.M.R. Haque and M.K. Islam

Computer Science and Engineering Discipline, Khulna University, Khulna-9208, Bangladesh

KUS-01/56-311201

Manuscript received: December 31, 2001; Accepted: March 10, 2001

Abstract: Cohen-Sutherland Algorithm is a very well known algorithm in computer graphics for line clipping. In this algorithm the clipping region is divided into nine regions. In our proposed algorithm we have proposed an algorithm which divides the clipping region into five regions. This modified algorithm is faster than Cohen-Sutherland algorithm in many cases. We have presented our algorithm and given comparison between Cohen-Sutherland Algorithm with ours.

Keywords: Line clipping, clipping candidate, needless clipping, clipping window

Introduction

Line clipping is used in the cases where a portion of a picture is required to be shown through a rectangular window. Different algorithms have been devised for this purpose among which Cohen-Sutherland algorithm plays a significant role. Cohen-Sutherland algorithm (Xiang, 2001) divides the plane of the rectangular clipping region called window, into nine regions and gives specific 4 bit codes to each region. The two major drawbacks of C-S algorithm are needless clipping and selecting some lines as clipping candidates which are totally outside the clipping window. These two are explained in section 2. In our algorithm we could partially overcome the problem of needless clipping by dividing the plain into five sections. This modified algorithm has been presented in section 3. We gave the comparison between C-S and our algorithm in section 4.

Drawbacks of the Cohen-Sutherland algorithm

There are two major drawbacks of C-S algorithm, these are:

1. Needless clipping

C-S algorithm at first clips the line AB in figure 2.1 at point C then it clips at point F. AB could directly be clipped at point F. So BC is a needless clipping. Similarly AD is also a needless clipping. Our proposed algorithm avoids such needless clipping in many cases.

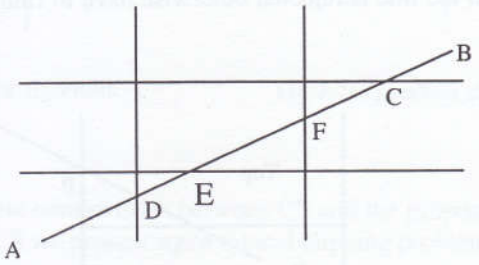

Fig.1. Needless clipping of C-S algorithm

2. Clipping Candidate

In some cases where the line segments are actually outside of the clipping window, C-S algorithm treats those lines as clipping candidates. In fig.2. although the line AB is outside the window, it is considered as a clipping candidate in C-S. This limitation has also been partially removed in our algorithm.

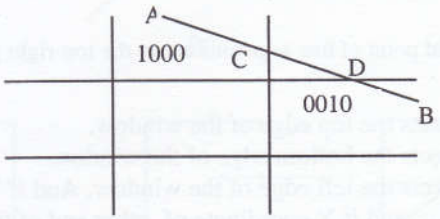

Fig. 2. AB line is considered as a clipping candidate

*Corresponding author: Tel: + 880-41-720171-6-3/213; Fax: + 880-41-731244; e-mail: ku@bdonline.com

DOI:https://doi.org/10.53808/KUS.2001.3.2.0156-se

Proposed Algorithm

We have divided the plane of the rectangular clipping region, called window, into five regions and given specific names and corresponding short integer code (as depicted in fig.3).

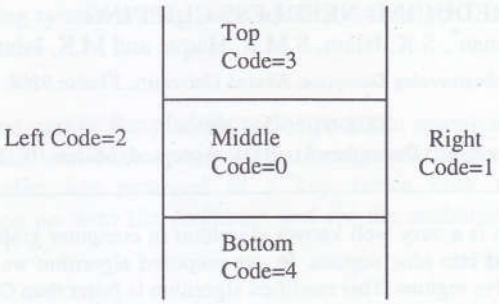

Fig.3. Codes for different regions

code=1, if $X>X_{max}$ (The end point is on the right)
code=2, if $X<X_{min}$ (The end point is on the left)
code=3, if $Y>Y_{max}$ (The end point is on the top)
code=4, if $Y<Y_{min}$ (The end point is on the bottom)
otherwise code=0.

X_{min}, X_{max}, Y_{min}, Y_{max}, are minimum and maximum X and Y co-ordinates respectively. The co-ordinates of the two end points of the line are assumed to be (X_1,Y_1) and (X_2,Y_2) in the following algorithm.
After assigning codes to both end of the line the following steps are followed:
1) If ($X_1<X_{min}$ and $X_2<X_{min}$) or ($X_1>X_{max}$ and $X_2>X_{max}$) or ($Y_1<Y_{min}$ and $Y_2<Y_{min}$) or ($Y_1>Y_{max}$ and $Y_2>Y_{max}$) then trivially rejected.
2) If both end code=0 then trivially accepted.
3) If code=1, then the line intersects the right edge of the window. And if Y coordinate of intersecting point P (shown in Fig.4) is greater than Y_{max}, and if Y coordinate of other end of that line is greater than Y_{max}, then the line is rejected otherwise we have to find intersect point which intersects the top edge. And if Y coordinate of intersecting point D (shown in Fig.4) is smaller than Y_{min}, and if Y coordinate of other end of that line is smaller than Y_{min}, then the line is rejected otherwise have to find intersect point which intersects the bottom edge.

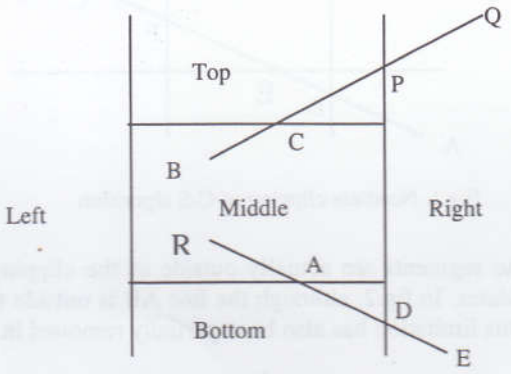

Fig.4. The end point of line segments are in the top-right and bottom-right corner

4) If code=3, then the line intersects the top edge of the window.
5) If code=4, then the line intersects the bottom edge of the window.
6) If code=2, then the line intersects the left edge of the window. And if Y coordinate of intersecting point P (shown in Fig.5) greater than Y_{max}, and if Y coordinate of other end of that line is greater than Y_{max}, then the line is rejected otherwise have to find intersect point which intersects the top edge. And if Y coordinate of intersecting point D (shown in Fig.5) smaller than Y_{min}, and if Y coordinate of other end of that line is smaller than Y_{max}, then the line is rejected otherwise have to find intersect point which intersects the bottom edge.

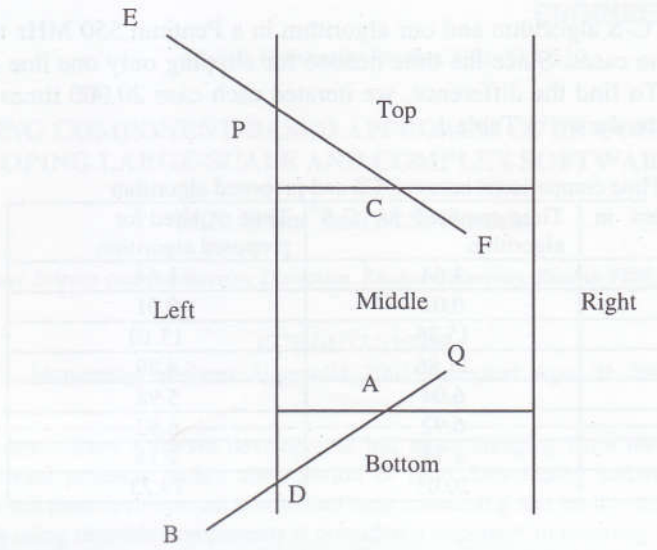

Fig.5. The end point of line segments are in the top-left and bottom-left corner

Advantages of our proposed Algorithm over C-S algorithm

1. In case of external intersections, the C-S algorithm performs needless clipping for all cases. In fig.6, CB and AD are needless clippings. But our Algorithm doesn't have to perform needless clipping in such cases (Fig.7).

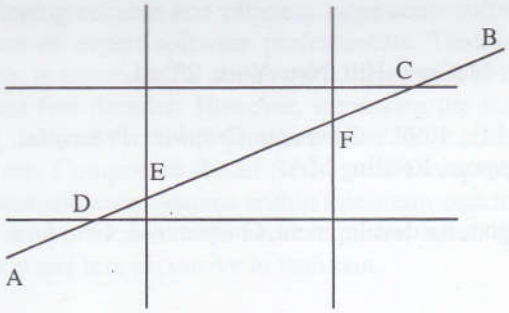

Fig.6. Needless clipping of the C-S algorithm

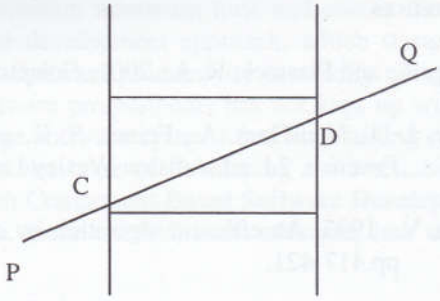

Fig.7. No needless clipping in the proposed algorithm

Time comparisons

In this section we have shown time comparisons between CS and the proposed algorithm. We selected some typical cases to work with. In fig. 8 we present some typical clipping problem.

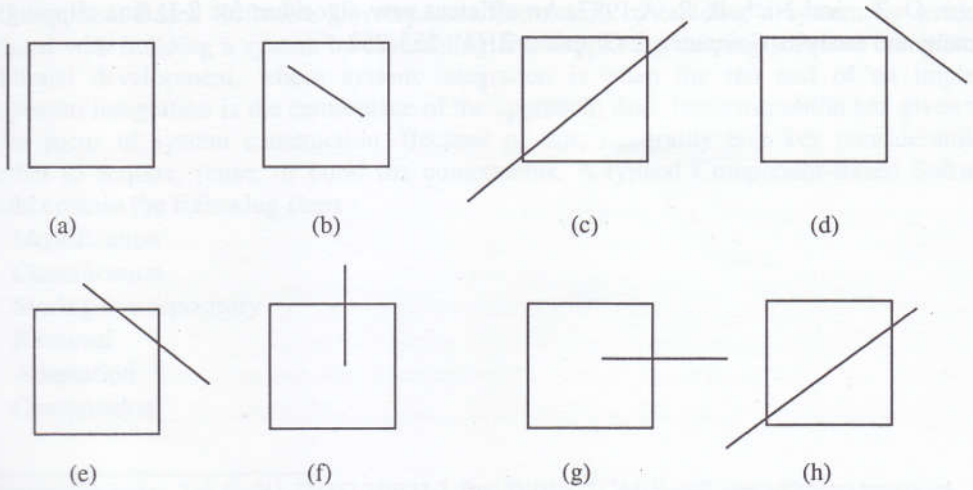

Fig.8. Some typical cases of line clipping

We tested all the cases by C-S algorithm and our algorithm in a Pentium 550 MHz machine. We recorded the execution time in all the cases. Since the time needed for clipping only one line computer showed 0.00 seconds for all the cases. To find the difference, we iterated each case 20,000 times and then recorded the time. The required times are shown in Table 1.

Table-1: Time comparisons between CS and proposed algorithm

Shape index in Fig.	Time required for C-S algorithm	Time required for proposed algorithm
a	1.64	1.64
b	0.01	0.01
c	15.76	15.10
d	11.86	4.89
e	6.04	5.98
f	6.92	6.92
g	2.14	2.14
h	20.05	19.23

Conclusion

We have presented a modified version of the Cohen-Sutherland algorithm in order to overcome its few major drawbacks. Our modified algorithm is more efficient than the Cohen-Sutherland in terms of execution time. The concept of our algorithm is easy to understand as well as easy to implement. Still the proposed algorithms suffers from some draw backs e.g., in some cases it also performs needless clipping. We are working to improve our algorithm so that needless clipping can be avoided as much as possible.

References

Xiang, Z., and Plastock, R. A., 2001. *Computer Graphics*, McGraw-Hill, New York, 2nd ed.

Foley, J. D., Van Dam, A., Feiner, S. K. and Hughes, J.F., 1999. *Computer Graphics Principles and Practice*, 2d. ed, Adision Wesley Longman Singapore, Reading MA.

Skala, V., 1993. An efficient algorithm by convex polygon, its development *Computer & Graphics* 17(4), pp.417-421.

Haque, M. M., Khan T.I., Tarafdar,A.H., 2001. Study and Development of a line clipping algorithm, B.Sc. Thesis, CSE Discipline, Khulna University, Khulna, Bangladesh.

Sharma, N.C. and Manohar, S.,1992. Two efficient algorithms based on simple geometric observations, its development *Computer & Graphics* 16(1), 51-54.

Andreev, R., and Sofianska, E.,1991. its development *Computer & Graphics* 15(4), pp.519-526.

Nicholl, T. M., Lee, D. T. and Nicholl, R. A.1987. An efficient new algorithm for 2-D line clipping : its development and analysis *Computer & Graphics* 21(4), 253-262.