ISSN: 2321-7782 (Online)
Volume 1, Issue 4, September 2013
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

# Polygon Clipping – An Approach to Simplified Clipping Techniques

**Prabhjyot Kaur Haryal**
Department of Computer Science and Application
St. Aloysius College (Autonomous)
Jabalpur – India

***Abstract: Clipping procedures are generally classified as either a Line Clipper or a Polygon Clipper. A Line clipper determines which lines are wholly within the window boundaries and which lines are to be totally or partially clipped. This is known as clipping of lines by polygons. Here the Concave Polygons will be clipped against a Convex Window.***

***Keywords: Clipping, Line and Polygon Clipper, Concave Polygons Clipping, Degeneracy.***

## I. Introduction

Following up on the clipping of lines by polygons the following solutions exists:

1. Clipping of each edge of the polygon by the clipping window and then joining the points to get the resultant clipped polygon.
2. This method works only with simple convex polygons; it does not however generate correct polygons for more complex cases like Concave Polygons.
3. It also becomes difficult rearranging the resulting polygon from disjoint clipped lines especially when there may be more than one clipped polygons generated.

One such algorithm is Sutherland Hodgeman Polygon Clipping in which it abandons thinking of a polygon as a set of line segments and clipping individual edges. The algorithm works iteratively by considering a clipping boundary and clips the entire polygon against this boundary. For a polygon to be clipped against a rectangle there would be 4 clipping boundaries.

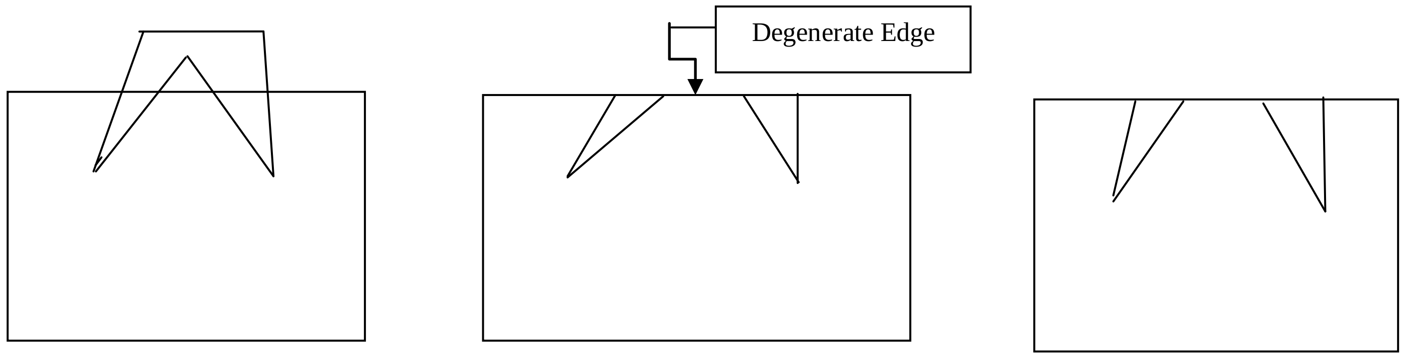
Thus each clip boundary is used to clip the polygon edges in turn and give a new set of points. This procedure is repeated for each clipping boundary and the final set of points representing the clipped polygon.

***Pipelining the Algorithm:***

The SH Algorithm requires at each step to store an array of order O (n) that contains the vertices of the intermediate polygon generated. To make the algorithm use less memory, a modification is suggested. When a polygon edge is clipped, it is not stored but is immediately forwarded to be clipped by the next window boundary. This pipelining method eliminates the need for intermediate storage of the entire polygon. Only the last generated point needs to be stored. The new modified algorithm will have an additional parameter 'range' which indicates the total number of boundaries.

***Degenerate Edges:***

While the SH Algorithm works very well with simple polygons. In the case of complex polygons it produces erroneous results. The algorithm tries to generate a fresh polygon by clipping with each boundary of the clipping window, but does not take into account the fact that the polygon might be clipped into more than one smaller polygon. Therefore the SH algorithm will have to be modified to take care of the Degenerated Edges between these smaller polygons.




Original Polygon with Clipping Window Clipped Polygon Actual Clipped Polygon

***Modified SH Algorithm***

This section explains a modification to the basic SH Algorithm that will not generate degenerate edges, instead it will output a list of polygons corresponding to the output of the basic SH Algorithm. The modified SH Algorithm takes as input the list of vertices output by the basic SH Algorithm and outputs a list of polygons.

The Modified Algorithm (MSH) takes as input the list of vertices output by the Basic Algorithm (BSH). It is known as the Output List. Another list called the Orientation List is constructed according to the following specifications.

**Step-1:** Sort those points in the output list that lie on the clipping window boundary in the same direction in which the BSH Algorithm traversed the polygon to generate the output list. The list of sorted points is called the Orientation List.

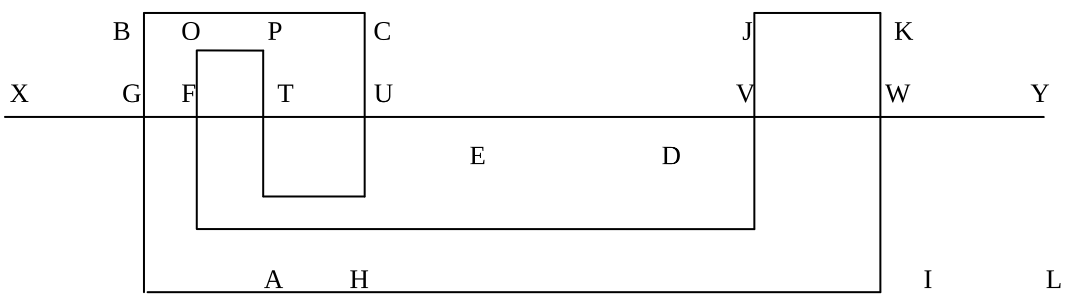
**Step-2:** Start traversing the output list one vertex at a time with the following modification. While going from a certain node A in the output list to its successor B, follow the same node A in the Orientation List. If node A is present in the Orientation List and node B is not its successor, do the following.

**Step-3:** If node C is node A's successor in the Orientation List, rearrange the pointer of node A in the output list so that its successor is also node C.

**Step-4:** Repeatedly perform a similar operation on the next node in the Orientation List till node B is reached on the Orientation List.

**Step-5:** Continue this process till the entire length of the output list is traversed.

At the end of the traversal of the output list, it will contain a number of lists, each corresponding to a separate polygon thus eliminating Degenerate Edges. The concept can also be explained with the help of following Example.




Orientation List- G → F→T→U→V→W

Initially Output List- A→G→U→D→E→T→F→H→I→V→W→L

After excluding Degenerated Edges i.e. G→F

Final Output List- A→G→F→H→I→V→W→L

## II. CONCLUSION

The above algorithm is first described in the simple case of clipping a polygon against just one edge of the window and then extended to include clipping over the entire window. If the output polygon contains a degenerate edge that goes across a corner of the clipping window, pre-process the Output and the Orientation Lists.

## References

1. Computer Graphics C Version Donald Hearn, M. Pauline Baker 2nd Edition.
2. A Parallel Polygon Clipping Algorithm, Chandrasekhar Narayanaswami.
3. An Extension of Polygon Clipping to Resolve Degenerate Cases, Dae Hyun Kim, Myoung-Jun Kim
4. An Efficient Software for Polygon Clipping, Liu Yong-kui , Gao Yun , Huang You-qun , Liu Yk , Gao Y , Huang Yq , An Journal Software.
5. An Analysis and Algorithm for Polygon Clipping, Communications of the ACM, Volume 26 Issue 11, Nov. 1983 Pages 868-877 , New York, NY, USA

## AUTHOR PROFILE

**Ms. Prabhjyot Kaur Haryal** received B.Sc (CS-Economics) degree from Rani Durgavati Vishwavidyalaya, Jabalpur in 2006 and MCA Degree from Rajeev Gandhi Proudyogiki Vishwavidyalaya, Bhopal in 2009, respectively. She is working as Assistant Professor, Department of Computer Science and Applications at St. Aloysius' College (Autonomous) Jabalpur, M.P. Since January 2010 till present.