POLYGON CLIPPING

Polygons can be clipped by modifying the line-clipping algorithm. **A** polygon boundary processed with a line clipper may be displayed as a series of unconnected line segments, depending on the orientation of the polygon to the clipping window. The portion needs to be displayed is a bounded area after clipping. Polygon clipping algorithm generates one or more closed areas that are then scan converted for the appropriate area fill. The output of a polygon clipper should be a sequence of vertices that defines the clipped polygon boundaries.

The Sutherland-Hodgeman Polygon-Clipping Algorithm

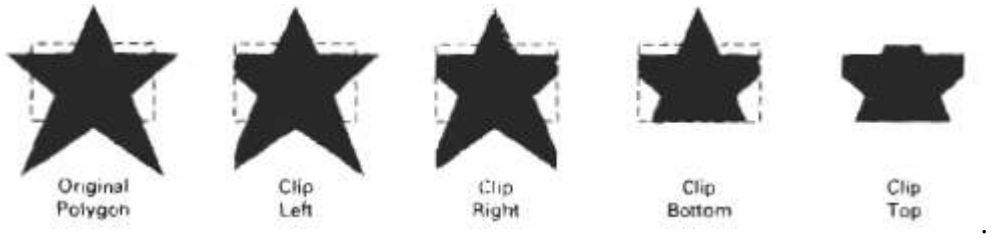
A polygon can be clipped by processing the polygon boundary as a whole against each (clip) window edge. This could be accomplished by processing all polygon vertices against each clip rectangle boundary (left, right, bottom and top) in turn.

Beginning with the initial set of polygon vertices, first clip the polygon against the left rectangle boundary to produce a new sequence of vertices.

The new set of vertices could then be successively passed to a right boundary clipper, a bottom boundary clipper, and a top boundary clipper.

At each step, a new sequence of output vertices is generated and passed to the next window boundary clipper

The final set of vertices represents the vertices of the final clipped polygon



Four possible cases when processing vertices in sequence around perimeter of a polygon:

As each pair of adjacent polvgon vertices is passed to a window boundary clipper, we make the following tests:

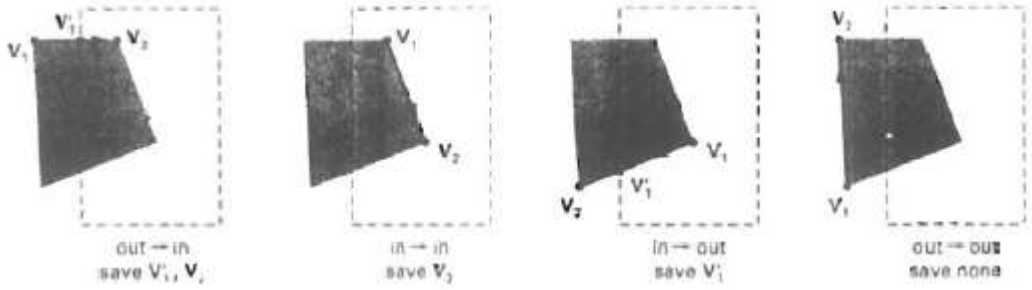
Case 1: If the first vertex is outside the window boundary and the second vertex is inside, both the intersection point of the polygon edge with the window boundary and the second vertex are added to the output vertex list.

Case 2: If both input vertices are inside the window boundary, only the second vertex is added to the output vertex list.

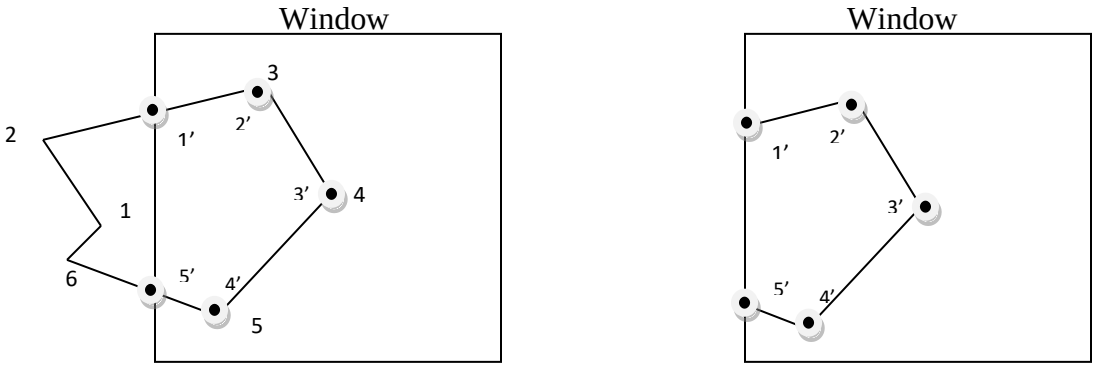
Case 3: If the first vertex is inside the window boundary and the second vertex is outside, only the edge intersection with the window boundary is added to the output vertex list.

Case 4: If both input vertices are outside the window boundary, nothing is added to the output list.

Once all vertices have been processed for one clip window boundary, the output list of vertices is clipped against the next window boundary.



Example



Before Clipping

After Clipping

i. Process area in figure against left window boundary .

ii. a. Vertices 1,2 are outside of boundary

b. Moving along to vertex 3 which is inside calculate the intersection . Save both intersection point and vertex 3
c. Vertices 4 and 5 are determined to be inside save them both
d. Vertex 6 is outside so find intersection point so save the intersection point.

Required setting up storage for an output list vertices as a polygon is clipped against each window boundary

The final set of vertices of the clipped polygon are 1’ 2’ 3’ 4’ 5’