CHAPTER 4 – ROLE OF VERTEX CLASSIFICATION

4.1 USE IN POLYGON FILLING

The vertices of a polygon are classified based on the action taken by the filling algorithms in determining the transformations of the edges bounding the region being filled.

A START vertex is one where a region begins. A STOP vertex is one where a region ends. Vertices may also be labeled as CONTINUE, MERGE, or SPLIT. A CONTINUE vertex occurs where a bounding edge is replaced by another. A MERGE vertex occurs where two regions can be joined into one region. A SPLIT vertex occurs when a region must be split into two regions.

Since there may be multiple regions being filled at a time even if only one polygon is present, the vertices in the polygon need to be ordered. The vertices are ordered from an arbitrary starting vertex by traversing the polygon counterclockwise so that the interior of the polygon is on the left as it is traversed. Therefore each vertex has a successor and a predecessor and links are created to the adjacent vertices. CONTINUE vertices, which cause an edge transition without changing the number of regions being filled, are classified as CONTINUE_PRED or CONTINUE_SUCC to indicate which vertex is to be used in defining the new edge. Edges are thus defined as being the line between a vertex and its successor or predecessor where the choice of successor or predecessor is determined by the “left-right” relationship of the edge to the region being filled.

The algorithm fills any simple polygon, including those with internal holes. The vertices that define the hole are classified by examining them in a clockwise manner. Therefore, the region
being filled is on the same side of an edge that is defined by a vertex and its successor (or predecessor) as if the vertex were on the outside of the polygon. MERGE and SPLIT vertices that are on the hole boundary indicate the same action as if they were on the outside of the polygon.

Figure 4.1 Polygon with each type of vertex  
Arrows indicate direction of traversal as vertices are classified.

An additional complication arises if the polygon is being filled in a vertical direction and contains horizontal edges. For example, if a polygon has a horizontal “bottom” edge, two vertices determine the beginning of a region. Similar situations exist for STOP, CONTINUE, MERGE, and SPLIT vertices which bound a polygon or a hole.

Therefore, vertices are also classified as SUCC_HORIZON (indicating that the successor vertex is connected by a horizontal edge), PRED_HORIZON (indicating that the predecessor vertex is connected by a horizontal edge), or NO_HORIZON (indicating that there is no horizontal edge
incident to the vertex). It is assumed that the polygon does not contain 3 adjacent colinear vertices.

For the algorithms to be general and robust they must handle any type of polygon. In addition to horizontal edges and holes, a difficult polygon might include one (or several) MERGE or SPLIT vertices with the same y value. Some examples of difficult regions are illustrated below.

![Figure 4.2 Polygon with hole and horizontal edges](image)

STT: START
STP: STOP
CP: CONTINUE PREDECESSOR
CS: CONTINUE SUCCESSOR
SPL: SPLIT
MRG: MERGE

Figure 4.2 Polygon with hole and horizontal edges

It is also possible for multiple regions to begin, end, or have transitions at the same y value. This situation is trivially supported in the case of MERGE vertices where two regions end and are replaced by a new one that is bound by two of the edges present in the two regions. Also, in the
case of SPLIT vertices one region ends and two begin which are bound by the two original edges and two new edges. Of course, it is also possible that an edge transition might occur at the same location.

Multiple START or STOP vertices with the same y value indicate independent regions. These regions could be filled in parallel.

4.2 MERGE/SPLIT ALGORITHM

In the Merge/Split algorithm regions are filled beginning with a START vertex until a STOP, MERGE, or SPLIT vertex is encountered.

A MERGE vertex indicates that the current region is to merged with another. This second region may already be filled, may be in the process of being filled by another process in a parallel implementation, or may be "waiting". In any case, one of the regions will be finished and need to wait on the other region to reach the same MERGE vertex. Thus, sufficient information must be stored and associated with the MERGE vertex. It is also possible that several MERGE vertices will occur at the same vertical location so that many regions will need to be combined.

A SPLIT vertex creates two regions to be filled. These regions can be viewed as independent. In practice it is more efficient to continue one vertex as if a CONTINUE vertex had been encountered and to store the second region for filling later or as another process. It is also possible that several SPLIT vertices occur at the same vertical location so that a region will need to be split into many regions. Such events can be handled easily if the stopping points (vertices)
are sorted horizontally as well as vertically. A region is split into two regions and one of these regions is again split into two regions before any filling is actually done.

It is important to note that the terminology “stopping point” refers to any vertex when it is encountered and causes an action to be taken. STOP vertices are those that are determined to indicate the end of a region.

4.3 ACTIVE EDGE LIST ALGORITHM

A modification of the Merge/Split algorithm does not treat the regions being filled independently at the scan line level. In this algorithm, all regions active for a given scan line are filled at the same time by the same process.

The AEL algorithm fills a polygon in much the same manner as the standard scanline method. A list of bounding edges, named active edge list, is maintained. However, the bucket-sorted edge table of scanline is not used. While the scanline algorithm examines the data structures for each new scanline, the AEL algorithm modifies the active edge list for each vertex encountered as in a plane sweep algorithm.

The two algorithms might be filling regions of a polygon at a given time as illustrated below.
Figure 4.3 Regions being filled independently