11.1 CONTRIBUTIONS

The following new approaches to polygon filling have been developed:

- A point based approach to polygon filling that is driven by vertices rather than edges has been presented. The concept of classifying vertices as an aid to processing has been extended. Two algorithms have resulted.

- One of the algorithms, Merge/Split, is conceptually different from other polygon filling algorithms. The other algorithm, AEL, is similar in many respects to the standard scanline approach.

- The algorithms, especially Merge/Split, have natural parallel implementations. Parallel Merge/Split, has been implemented in both a distributed memory and a shared memory model.

- The concept of polygon filling has been extended to generalized polygons that allow edges to be any type of curve that can be rasterized in single step increments.
The algorithms produce correct results for all polygons tested including polygons with many segments, polygons with many transitions at the same scan line, polygons with interior holes, polygons with curved edges, and polygons with various combinations of these features.

The algorithms are more efficient than scanline with an average time ratio of 0.281 for a wide variety of polygons. The Merge/Split algorithm is slightly more efficient than AEL for most polygons.

The shared memory parallel implementation of Merge/Split performs very well. The distributed memory implementation is less efficient than a sequential algorithm due to the amount of communication relative to arithmetic performed.

11.2 FUTURE WORK

Additional parallel implementations are possible. A SIMD implementation would be interesting as a group of processors could each be filling a polygon segment at the same time. Several issues would need to be considered including how to start new segments and deal with any change caused by a vertex at the end of an edge. To be practical such an implementation would need to spend the great majority of its time filling segments rather than processing vertices that alter segments.

A thread based implementation on a Symmetric Multiple Processor would be very similar to the shared memory implementation but would have the advantage of much lower context switch times.
Another area for exploration would be filling multiple polygons at the same time on a parallel system so that there would not be a fixed relationship between a polygon and a processor.

More careful analysis of the parallel algorithms could result in motivation to consider a hardware based implementation.