Surface Simplification Using Quadratic Error Metrics Summary

Trenton Plager

Introduction & Related Work (3 sent.)

The introduction of the paper reveals that the topic being discussed is simplifying 3D models so that complex models can be used in applications that desire less complexity and greater speed since complexity is directly related to computational cost. The authors of the paper indicate that they have created an algorithm that can simplify 3D models efficiently while still maintaining quality and while still being fairly generalized. To introduce their work, they begin by discussing 3 other classes of algorithms that are used for this purpose: vertex decimation, vertex clustering, and iterative edge contradiction. None of these 3 classes are without fault though, at least in terms of the applications the authors have in mind.

Methods (Sections 3 – 6) **(5 sent.)**

Regarding the actual algorithm, it begins with a method derived from the earlier mentioned "Iterative Edge Contraction." This method instead contracts vertex pairs, moving 2 vertices that are near each other each to a new location, assigning all connections to one of the 2 vertices, and deleting the other vertex. As a result, the author's algorithm succeeds where others fail in that it can connect previously disconnected sections of a model. To select the pair, the authors developed a method of approximating error using Quadrics. Using matrices and matrix math, they determine the cost of contracting each potential pair of vertices. However, to do this, they must also choose a heuristic that can be used to arrive at the general error. Some of their methods are similar to other methods used by similar algorithms.

Results (5 sent.)

The results of this algorithm seem promising. The authors have been able to use it to drastically decrease the number of vertices in a model, while still maintaining quality and the distinctive features of the model. The algorithm has also proven to be extremely efficient, with a series of 5 cow models of varying complexities being returned in only about a second. In the paper, the authors also demonstrate the effect of their algorithm on larger models. With these models, the process understandably takes more time, but still returns largely positive results. Finally, the authors show examples of their algorithm's results compared to the results of other algorithms by showing the results of each on a model of a human foot.

Discussion (5 sent.)

The main discussion in this section of the paper states that this algorithm provides features and quality that have never before been seen in previous algorithms. Specifically, the authors refer to their algorithm's efficiency, quality, and generality. They go on to cite the specific difference between their own algorithm and other related algorithms. Finally, the authors do write some about the improvements that could be made on their algorithm. The first that they cite is that they could use a more sophisticated vertex pair selection system. They also note that they have not addressed the properties of a models' surfaces such as color or roughness. They do also cite a

few weaknesses of their algorithm such as the method they use to estimate error when choosing vertex pairs, and that it is difficult for them to remove faces they no longer need because of vertex and edge restructuring.

Conclusion (3 sent.)

Overall, it seems the authors have done a tremendous job at improving a process that is vital for the continued advancement of graphics programming. Their algorithm using vertex pair contractions combines incredible efficiency with incredible quality and incredible generality. They do this by measuring error to select vertex pairs to contract. Then they store data within vertices to assist in determining the final shape of the model. The high generality of their algorithm also allows them to join previous disconnected surfaces, an ability that is not found in many other similar algorithms.