Progressive linear triangle surface mesh implementation. Pitfalls and challenges.

Bartosz Chaber[1]

Institute of Theory of Electrical Engineering, Measurement and Information Systems, Faculty of Electrical Engineering, Warsaw University of Technology, chaberb@iem.pw.edu.pl

This paper describes conclusions from author' implementation of progressive format for surface meshes obtained as the result of FEM simulations. It's main goal is to point out challenges and issues which author ran into during development and to present way of handling them.

1 Introduction

The need of progressive mesh scheme originated from an idea of web service-based ordered computations system described in [6]. Progressive format introduced by Hoppe in [2, 3] was found very suitable in the Internet transmission of large data sets. In result of its flexibility, many derivatives was developed [5, 8]. Despite its straightforward basis, there are some challenges that must to be overtaken by developer facing implementation of progressive mesh format.

Progressive meshes bases on two complementary operations: *ecol* and *vsplit*. They are responsible for simplification and refinement of processed model. Edge collapse (*ecol*) operation contracts edge between two vertices v_a and v_b, replacing them by one vertex in the middle of the edge. Inverse of *ecol* is called Vertex Split (*vsplit*). With help of some additional info about a topology before *ecol* it can split given vertex v_c and replace it with an edge between two new vertices.

It turned out that strategy of mesh processing is crucial aspect in the whole process. As also pointed in [7], cost function definition is another essential element during simplification as it directly reflects in results of mesh simplification. Finally, during processing mesh is vulnerable for degeneration caused by edge collapsing.

1.1 Edge collapse and vertex split strategy

As it was mentioned earlier, progressive mesh scheme define two operations performed on mesh structure. However, it does not define explicitly how those modifications are actually made. It's up to developer which geometry and topology modification strategy is used. During implementation of described application two edge collapsing strategies were considered: replacing both edge's vertices (Fig. 1) and replacing only one vertex (Fig. 2).

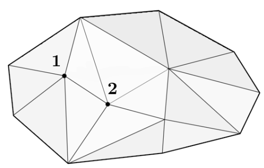

Fig. 1. First type of reconnection. Vertices 1 and 2 are replaced with new vertex 3.

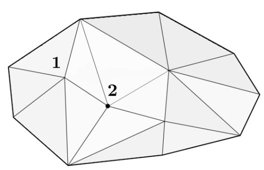

Fig. 2. Second reconnection type. Vertex 1 is moved to new position and vertex 2 is deleted.

Although first solution seems simpler and more natural, solution depicted in the Fig. 2 has much lower computational cost. It's worth mentioning that using method 2 less vertices are being reconnected, than using method 1 (note number of reconnection operations marked as an arrows in the Fig. 1 and the Fig. 2). Finally, author implemented second strategy, which made him to set additional rule: when collapsing the edge or splitting the lower index indicates vertex being translated, where higher one correspond to new vertex (during *vsplit*) or removed vertex (during *ecol*). This rule prevents from errors caused by accidental mixing indexes.

1.2 Cost function

In [2] Hoppe uses minimization of an energy function. Order of edges to collapse is determined in two phases. The first phase is choosing the best position for vertex replacing the edge, when the second phase is choosing best edge to collapse in the current step. Garland describes in [8] another approach based on quadratic error metric. On the other hand much simpler solutions also have been proposed. Liu et al. in [4] proposed metric based on volume change introduced by collapsed edge. They used some of adjacent faces to calculate volume of contracted part of the mesh. Author of this paper developed extended version of this metric. To calculate approximate volume difference all adjacent faces are being used. However calculating volume bounded by the surface consisting of those faces is nontrivial, as it requires to find centroid of this surface in order to calculate its volume. Idea behind using centroid to determine volume is depicted in the Fig. 3. To find approximated difference of volume caused by the edge collapse, one has to know two volumes: $\mathbf{V_0}$ (before *ecol*) and $\mathbf{V_1}$ (after *ecol*).

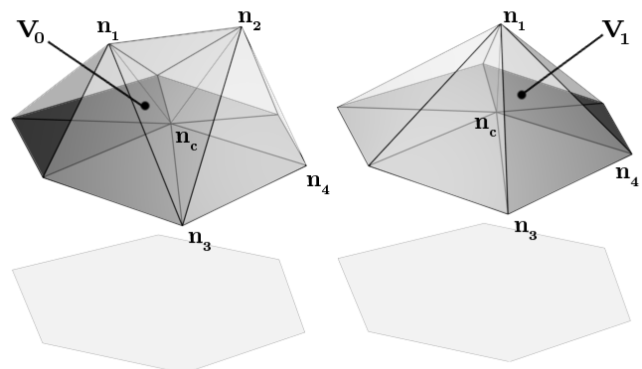

Fig. 3. Figure depicts how *ecol* $(\mathbf{n_1}, \mathbf{n_2})$ affects volume of presented mesh.

Finally, each edge of mesh is rated using:

$$r(e) = |\mathbf{V_0} - \mathbf{V_1}|. \tag{1}$$

It turned out that this cost function leads to very good results. Results of processing meshes using this cost function can be found in section 1.4.

1.3 Mesh degeneration

One of the biggest issues connected with processing meshes into progressive form was to preserve valid surface topology and geometry. Beside from rating edges using (1), each edge has to be tested if its collapsing doesn't degenerate mesh. If it does degenerate the mesh, this edge is blocked and cannot be collapsed. To determine if the mesh is valid, all edges are checked if they connect exactly two faces. If they connect more faces it means that mesh is degenerated and cannot be further processed. Example of the mesh which can be degenerated in result of *ecol* operation is presented in the Fig. 4.

However degeneration presented in the Fig. 4 can be easily prevented, there are some situations when more tests are required.

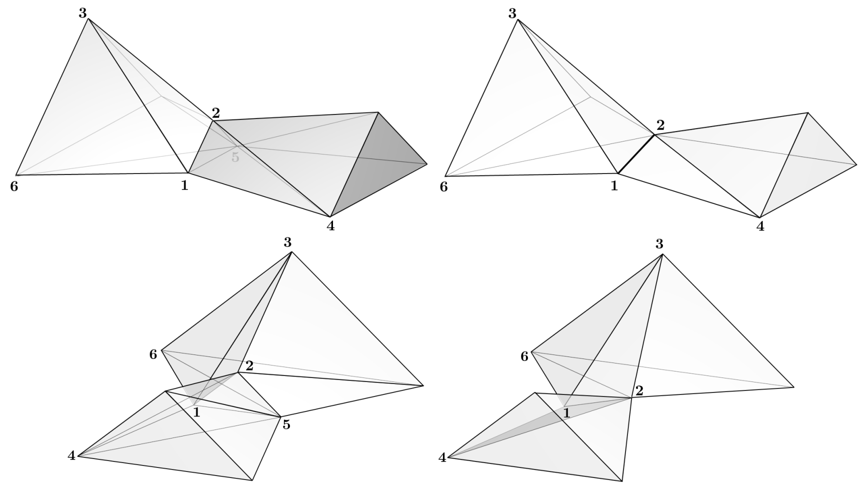

Fig. 4. Example of situation, when $ecol(2, 5)$ results in degenerated mesh (on the right side). After *ecol* the edge between 1 and 2 is shared by three faces: 1–2–3, 1–2–4 and 1–2–6.

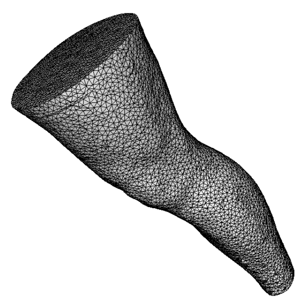

Fig. 5. `leg` model stored in non progressive format.

1.4 Results

During development of the processor of a standard meshes into their progressive form, author encountered many issues which had an impact on final application architecture. Fortunately all issues had been resolved. Result of processing input mesh (Fig. 5) into progressive format is presented in the Fig. 6. It can be seen that none of those meshes is invalid nor degenerated. What is more, according to chosen cost function, whole model seems to be simplified uniformly. However, plain areas are simplified more than the others, which is expected and desired effect.

Conclusions from author's implementation are as follows:

- before implementation of progressive mesh format one has to choose strategy of collapsing edges and modifying topology of mesh,

- cost function is essential element of progressive mesh processor and there exists many metrics (volume, quadratic error, edge length),
- mesh degeneration caused by edge collapse is serious problem, and preventing it is nontrivial task.

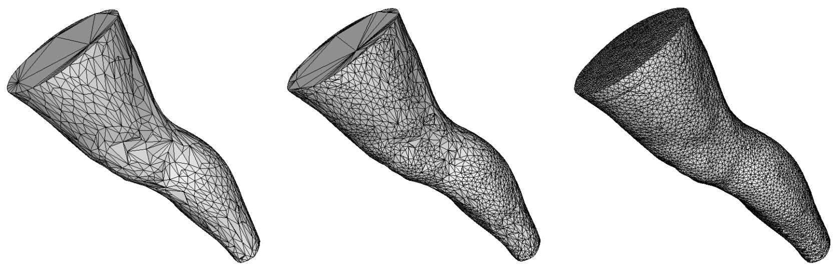

Fig. 6. `leg` model stored in progressive format. From left to right: base mesh after no refinements, mesh after 2000 *vsplit* operations applied, recovered original mesh.

References

1. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. (1993), Mesh optimization, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 19–26, New York, USA, ACM Press/Addison-Wesley Publishing Co.
2. Hoppe H. (1996), Progressive meshes, Proceedings of the 23th annual conference on Computer graphics and interactive techniques, pp. 99–108, New York, USA, ACM Press/Addison-Wesley Publishing Co.
3. Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. (1998), Efficient Implementation of Progressive Meshes, Computers & Graphics
4. Liu X., Bao H., Peng Q., Heng P., Wong T., Sun H. (2000), Progressive Geometry Compression for Meshes, Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, Washington, DC, USA, ACM Press/Addison-Wesley Publishing Co.
5. Chen Z., Bodenheimer B., Barnes J. F. (2003), Extending progressive meshes for use over unreliable networks, Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03) - Volume 03, Washington, DC, USA, ACM Press/Addison-Wesley Publishing Co.
6. Sawicki B., Szmurło R., Starzyński J., Wincenciack S. (2011), Service Oriented Modular System for Modeling of the Human Body Stimulation Progress In Electromagnetics Research Symposium Abstracts, Marrakesh, Morocco
7. Staadt O. G., Gross M. H., ETH Zurich, Avoiding Errors In Progressive Tetrahedralizations
8. Garland M. (1999) Quadric-based polygonal surface simplification, Phd Thesis, Carnegie Mellon University, Pittsburgh, PA, USA