

Mesh Simplification Using a Two-Sided Error Minimization

Elena Ovreiu^{1,2+}, Juan Gabriel Riveros², Sebastien Valette² and Rény Prost²

¹ University Politehnica of Bucharest, Romania

² Université de Lyon, CREATIS; CNRS UMR5220; Inserm U1044; INSA-Lyon; Université Lyon 1; France

Abstract. Reducing the complexity of a very large data set has become an important problem in the last years because of the rapid evolution of the acquisition techniques. In this paper we propose a mesh simplification algorithm based on two-sided error. The two-sided error metric permits us an accurately evaluation of the geometric deviation introduced by an edge simplification for the models with boundaries, islands.

Keywords: mesh simplification, two-sided error, quadratic error.

1. Introduction

Nowadays, meshes are presented in multiple and different areas such medical imaging, movie production, virtual reality, computer games. Due to the technological improvements from the recent years, a mesh could now have millions of elements. That means the reality could be reproduced more accurately with a complex mesh. The drawback of the complexity is the difficulty to manipulate this kind of meshes. For this reason, mesh simplification has become an extremely exploited topic in the last years.

The goal of mesh simplification is to reduce the complexity but keeping as possible as high fidelity of the original model.

Having this goal, a multitude of mesh simplification algorithms were developed during the time. For a detailed classification of those algorithms, we refer the reader to [4].

In the following we will describe some methods to compute the geometric deviation introduced by mesh simplification.

One of the most rapid simplification algorithms is the one proposed in [2]. The error introduced by one edge collapse is given by the sum of squared distances from the new vertex to its supporting planes. This error is called quadratic error metric (QEM). The drawback of this simplification method is given by the computed distance which is only an approximation. The algorithm presented in [5] is similar to QEM, but the error introduced by an edge collapse is considered the maximum of the distances from the resulted vertex to its supporting planes, and not the sum of them as in QEM.

In [6] the edge collapse is done accordingly to a complex error which represents the sum of four terms: the distance from the new vertex to the original model. This term penalizes contractions which do not preserve the sharp edges. The third term controls the accuracy of mesh's scalar attributes. The last term permits the optimization to get a desirable local minimum.

In [7] the simplification is realized according to the Hausdorff distance. A vertex is deleted from triangulation only if the introduced Hausdorff distance is smaller than a predefined Hausdorff distance.

⁺ Corresponding author. Tel.: +33 667923072.
E-mail address: Elena.ovreiu@gmail.com.

We propose an algorithm which simplifies a mesh accordingly to a two-sided error measure. Using this two-sided error we can measure more accurately the deviation introduced by one edge collapse.

2. Simplification Algorithm

Our simplification algorithm reduces the mesh complexity while retaining the mesh fidelity. The simplification is realized using an iteratively edge collapsing. The simplification algorithm follows the idea from [1].

The algorithm is outlined as follows:

1. On the original model we compute the error introduced by each possible edge collapse.
2. The edge with the minimum associated error is chosen to be collapsed.
3. We collapse the chosen edge to a single vertex $e = (v_1, v_2) \rightarrow \bar{v}$. The following operations are performed:
 - the position of the resulting vertex is computed: $(v_1, v_2) \rightarrow \bar{v}$
 - the vertex v_2 and degenerated faces are eliminated;
 - all faces connected to v_2 are connected to v_1 ;
 - the error is recomputed for all edges in the new simplified model.
4. Those steps are repeated until the stop condition is achieved.

There is a main problem: how to define the error generated by the edge to be collapsed in order to keep a high fidelity to the original. We detail this problem in the following subsections.

In order to get an accurate measure of the error introduced by an edge collapse, we introduce a two sided quadratic error. For each possible edge contraction, we call *model state* the possible mesh configuration. That means for each possible edge collapse we get a possible model state and compute the quadratic error between this model and the original one and the reverse quadratic error (between the original model and possible model state).

2.1. Direct error

We call *direct error* the quadratic error between the possible *model state* and the original model. We compute the distance from each face of the possible *model state* to the original model. The area weighted sum of the squared distances represents the error introduced by one edge collapse (eqn. 1).

$$d(\hat{M}, M) = \frac{1}{\sum_{c \in \hat{M}} w_c} \sum_{c \in \hat{M}} w_c d^2(c, M) \quad (1)$$

where \hat{M} is the approximated model and M the original one.

In order to have more accurately measurement of the distance between two meshes, we apply a repetitive one-to-four subdivision (1:4) for each face (Fig.2). The number of subdivisions is variable for each face and it is chosen as so to have a proportion between the number of faces of the simplified mesh and the original one.

Thus, $d(c, M)$ represents the distance from a cell (sub-triangle) of a subdivided triangle to the original mesh.

In practice, we compute the distance from a cell to the mesh as the arithmetic mean of the distances from the cell's vertices to the mesh:

$$d(c, M) = \frac{1}{3} \sum_{i=1}^3 d(v_i, M) \quad (2)$$

Where $d(v, M) = \min_{p \in M, v \in C} \|v - p\|$ is the minimum distance from the one vertex of subdivided cell to the closest face of M . $\|\cdot\|$ is the Euclidean vector length operator.

w_c is a scalar weight factor which in our method is the area of a subdivided cell.

Like in [2], we weight the quadratic error by area in order to make simplification more robust to irregular sampling.

To reduce the complexity of computing the minimum Euclidean distance, we use the Proximity Query Package (PQP) library [3].

For each step of simplification, the errors for all edges are recomputed.

For one edge collapse, only the faces around the respective edge will be modified, so we recompute only the distances for those faces, and the other distances are not modified.

2.2. Reverse error

We call the reverse error, the error between the original model and the simplified one. This error is similar to the direct one (equation 1), but here M and \hat{M} are interchanged.

For each possible edge contraction, for all faces, the distance from each face of the original model to the possible model state is modified. Computing all those distances is very expensive. Creating a copy of the possible model state in order to compute distances to this model is expensive as well. To avoid the computation of the distances for all faces in the original model, we compute only the distances for the faces in the original model which are affected by an edge collapse. For those faces in the simplified model, we look for the vertices in the original model for which the collapsed simulated faces are the closest. Afterwards, we compute the distances only for the faces in the original model which share those vertices. In this way we determine the faces on the original mesh whose distances are affected by one edge collapse.

To avoid recreating the simplified model for all possible edge collapses, we split the simplified model in more submodels, and we recreate only the submodel which contains the edge for which we compute the error.

2.3. Two-sided error

The error for one edge is the sum of the direct error and the reverse one. Even if the direct error and the reverse one compute the distances between the same two models (possible model state and original one), these distances are not equal because they are computed in different directions.

The error associated to each edge is the global symmetric error between simplified model and original one. We use the term global because the error is computed from all the faces of one model to the other one. This global symmetric error gives us better approximations of the introduced geometric deviation, than a one sided error.

3. Results

The approximations obtained with our simplification algorithm are compared with the ones obtained with QEM [2]. We evaluate the error of approximations using quadratic error metric.

In Fig.1 Pieta is simplified up to 450 vertices using our simplification algorithm and QEM. We can see that for flat surfaces, our simplification algorithm uses fewer faces for the approximation.

Fig. 2 represents the base for Pieta, which is a flat surface, represented with more triangles for the approximation done with QEM (b) and fewer by our algorithm (a). That means for the same budget of faces, our algorithm uses fewer faces for flat regions (as the base of Pieta) using the rest of them to approximate the curved regions, leading to a better approximation of the original model.

The quadratic error between approximations produced with our simplification algorithm is always lower than for approximations produced with QEM (Fig. 3). For instance, for an approximation of Pieta with 630 vertices, the quadratic error produced with our approximation algorithm is 0.088, while the one produced with QEM is 0.105. This difference is higher for a reduction factor higher than 50.

Fig. 4 represents an approximation for Octaflower model using our algorithm and QEM. We also compute the geometric deviations introduced by our simplification algorithm and by QEM using Metro tool

[8]. For Octoflower model simplified up to 109 vertices (Fig.4) using our algorithm, the Hausdorff distance measured by Metro is 0.745652 while for the model simplified by QEM (with the same number of vertices), the Hausdorff distance is 1.355617. Also, the mean error is 0.0101324 (from the simplified model to original one) and 0.00382 (from the original to the simplified) for the model obtained with our algorithm while for the model simplified with QEM the mean errors are 0.130543 and 0.002525, respectively. For a drastic simplification (Octoflower with 39 vertices) we get the Hausdorff distance 0.215756 with our algorithm and 0.422858 with QEM. The mean errors are 0.29203 and 0.001669, respectively 0.035 and 0.001407 for QEM. For a Pieta model with 130 vertices, the Hausdorff distance (measured by Metro tool) is 7.288526 for the model simplified by our algorithm while on model simplified by QEM the Hausdorff distance is 19.107645. The mean error is 0.981812 (from simplified model to the original) and 0.06451 (the backward) for our simplification and 1.12817 and 0.056734 for QEM.

4. Conclusions

In conclusion, we propose an iterative edge collapsing algorithm which produces high quality approximations. We get high quality results using a two-sided error to characterize the geometric deviation introduced by a possible edge collapse. As, for each possible edge collapse, we compute the error between the whole simplified mesh and the original one, and the reverse error, between the original mesh and simplified one, we are able to perform an accurate characterization of the geometric error introduced by an edge collapse.

Even if we obtain better results than QEM in terms of quality of approximations, our algorithm is several times slower than QEM.

The simplification algorithm proposed in this paper gives us better approximations and better running time than the algorithm proposed in [1]. We obtain better approximations because, after each edge collapse, we recompute the error for all edges in the simplified mesh, and not only for the edges modified after one collapse.

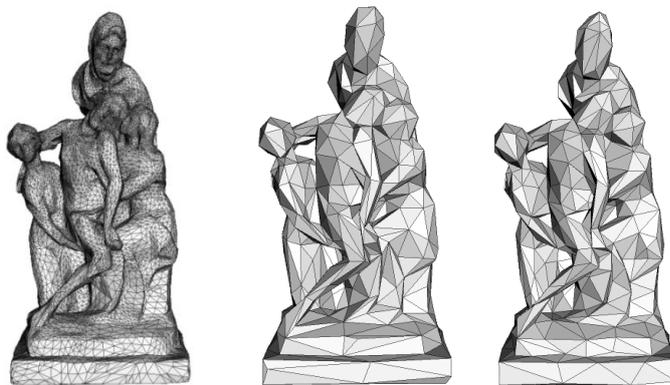


Fig. 1: Approximations of Pieta. From left to right: Original model with 13940 vertices (27904 triangles), simplified models with 450 vertices using our simplification algorithm and QEM.

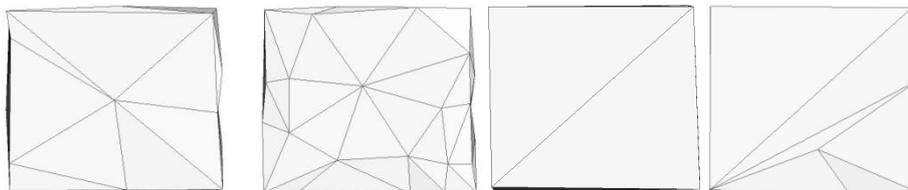


Fig. 2: Approximations for the base of Pieta model.

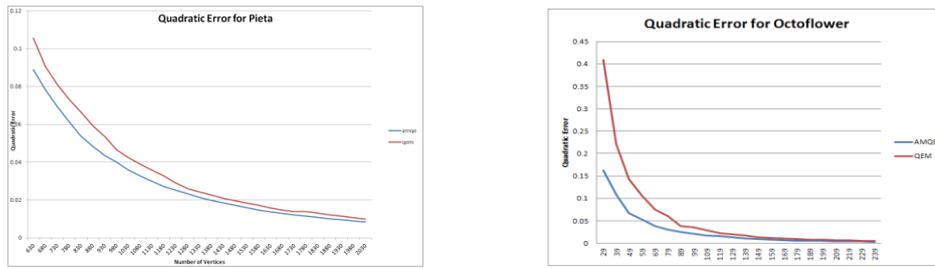


Fig. 3: Geometric error vs. number of vertices for Pieta and Octoflower. The quadratic error is computed between approximated mesh and original one for each step of simplification. Simplifications are made using QEM (red line) and our algorithm (blue line).

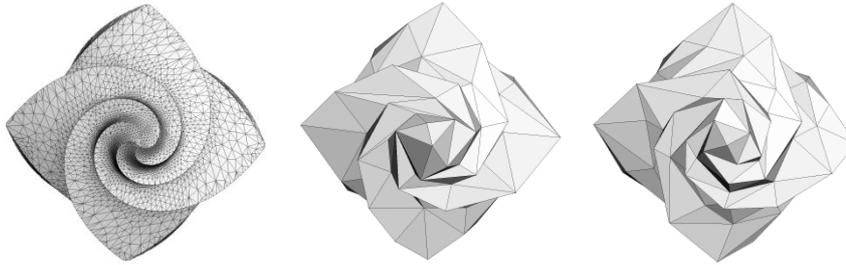


Fig. 4: Approximations for Octoflower model: From left to right: Original model with 1919 vertices (15834 triangles), simplified models with 109 vertices using our simplification algorithm and QEM.

5. References

- [1] M. Garland, P. Heckbert. *Surface Simplification Using Quadric Error*. In: Proc. of ACM SIGGRAPH 1997, pp. 209-216.
- [2] E. Larsen, S. Gorttschalk, S. Lin and D. Manocha. *Rectangular Swept Sphere Volumes*. In: Proc. of IEEE Int. Conference on Robotics and Automation 2000.
- [3] J. Talton. *A Short Survey of Mesh Simplification Algorithms*. Computers and Graphics, Elsevier, Volume 22, 2004.
- [4] R. Ronfard, J. Rossignac. *Full-range approximations of triangulated polyhedra*. In: Proc. of Eurographics Vol. 15, 1996.
- [5] H. Hoppe. *Progressive meshes*. In: SIGGRAPH 96 Conference Proceedings, pages 99–108, 1996.
- [6] R. Klein, G. Liebch, W. Strasser. *Mesh reduction with error control*. In: ACM Visualization 96, 1996.
- [7] P. Cignoni, C. Rocchini, R. Scopigno. *Metro: measuring error on simplified surfaces*. In: Computer Graphics Forum 17(2), 167-174, 1998