A Memory Effective Two-phase Approach for Large Scanned Surface Mesh Simplification

Yi-Ling Chen¹
yilin@cs.nthu.edu.tw

Xiang Zhang²
xiangzhang@gmail.com

ABSTRACT
We present a novel two-phase multi-attribute algorithm suitable for large surface mesh simplification. By employing a linear combination of error metrics to control the process, the proposed algorithm incorporates geometric error control and preserves other attributes of the original model such as the texture (vertex color) and surface normal. In the first phase, we utilize the volume-surface tree [1] (VS-Tree) for vertex clustering to achieve memory effectiveness and computational efficiency. In the second phase, an iterative edge contraction process is applied to obtain the final simplified model. We experiment the proposed algorithm to large mesh models and the results are compared with those from other state of art algorithms such as the octree clustering simplification[5] (OCS) and the original quadric error metric (QEM) based simplification (Q-Slim) ([2][3]).

KEYWORDS: Mesh simplification, vertex clustering, edge contraction, multi-attributes, linear combined cost function.

1 INTRODUCTION
Mesh simplification has a lot of applications in areas such as computer graphics, computer vision, and computer-aided design. The goal of mesh simplification is to obtain a model of reduced complexity that maintains a close approximation of the original model. As the recent advances in three-dimensional modelling technologies, the acquisition of extremely large scale models becomes possible. In addition, more and more scanned models are obtained with additional surface attributes such as vertex color. Realistic applications often require that the high resolution meshes to be simplified to achieve the best balance of data storage, online transportation, visualization, and geometric accuracy.

In this paper, we propose an approach for large scale surface model mesh simplification that achieves memory efficiency and accounts for multiple surface attributes. Considering that different types of surface attributes, e.g., the geometric shape and vertex color, are normally independent of each other, we define cost function as a linear combination of the geometric QEM and the normalized error metrics of other attributes. In addition, we apply a two-phase approach to take the advantages of both the vertex clustering and edge contraction. The values of the above cost function are directly passed from the vertex clustering phase to the edge contraction phase to control the process of mesh simplification and to preserve the quality of results. The values of the cost function are passed from the vertex clustering phase to the edge contraction phase to achieve the computation efficiency and the quality of results. We adopt the structure of VS-Tree[1] for the vertex clustering phase to effectively divide the simplification of a massive mesh into a set of problems of smaller scale.

2 ALGORITHM
In the proposed algorithm, a linear combination of normalized error metrics of multiple attributes if defined as the cost function to control the process of mesh simplification and to preserve the quality of results. The values of the cost function are passed from the vertex clustering phase to the edge contraction phase to achieve the computation efficiency and the quality of results. We adopt the structure of VS-Tree[1] for the vertex clustering phase to effectively divide the simplification of a massive mesh into a set of problems of smaller scale.

2.1 Linear Combination of Error Metrics
We need an inexpensive criterion that consistently reflects the homogeneity of the surface attributes for both clustering subdivision and edge contraction. Considering many attributes are independent of each other, e.g., the surface texture/color and vertex positions, we define the cost function over each cluster as a linear combination as follows:

$$E(C) = \sum_{i} w_i E_i(C)$$

where$$\sum_{i} w_i = 1$$

In the equations, $$m$$ is the number of attributes; $$E_i(C)$$ the normalized error term corresponding to one attribute; $$w_i$$ the corresponding weight coefficient. For example, in the case that the vertex coordinates, vertex color, and vertex normal are taken into consideration for the simplification cost, the equation can be written as:

$$E(C) = w_g E_{g}(C) + w_f E_{f}(C) + w_s E_{s}(C)$$

Where the geometric, color, and surface normal error metrics are:

$$E_{g}(C) = \sum_{i} Q_{g}, \forall f_i \in C$$
\[ E(C) = \sum_{j=1}^{n} \| e_j - c_i \| \quad \forall e_j \in C \]
\[ E(C) = \sum_{j=1}^{n} \| e_j - n_j \cdot n_i \| \quad \forall e_j \in C \]

Here we use the QEM to represent the error associated with the vertex coordinates only. Our experimental results show that this linear combination works well for large scale scanned models with a significant improvement on computation effectiveness and memory efficiency (in comparison with, e.g., [3]). Our formulation of the simplification cost function also allows flexibility to adjust the weights of different attributes to obtain results with preferred properties (see Figure 3).

2.2 Two-phase Approach

We adopt the Volume-Surface Tree developed by Boubekeur et al. [1] for the vertex clustering phase of our algorithm. A VS-Tree is a surface-based spatial partitioning technique that combines the octree and quadtree subdivision to describe a 3D surface adaptively. In our implementation, we incorporate the above linear combination as the cost function to control the subdivision and subdivide the VS-Tree in an iterative greedy manner, i.e., for each iteration step, the leaf node with largest cost value is chosen for further subdivision. The QEM and the linear combination of error metrics computed from the vertex clustering phase are passed to the next phase of iterative edge contraction.

In the second phase, we apply an iterative edge contraction procedure ([2][3]) to produce the final results. Instead of initializing from the intermediate geometry, we directly apply the values of cost function passed from the first phase, which means we continue to refer to the geometric and surface properties of the original model for the second phase of mesh simplification. This approach leads to final simplification results that are closer to the original model in comparison.

3 Experiments and Discussions

We test the proposed algorithm with publically available models [6] and compared the results from those of existing algorithms. Figure 1 shows the simplification result of a color scanned model (the manuscript). The comparison of different results is shown in Figure 2. A zoomed-in portion of the model shows that the proposed algorithm produces result that is visually (and also quantitatively) comparable to the results obtained using the extended Q-Slim[3], but requires only a sixth of memory usage and a third of computation time. Please refer the authors for more detailed and quantitative results.

Figure 3 shows the meshes of a portion of the simplified model obtained with different weights of attributes. Figure 4 shows the visualization of meshes simplified from a very large model. After reduced to 1/90 of the original size, the result from the proposed algorithm preserved a lot of geometric details. In this case the result from the original Q-Slim is not available because the memory requirement exceeds the capability of the computer used to produce the results (2GB RAM).

Figure 3. Different appearance can be achieved by adjusting the weights in the cost function with the same final model size.

Figure 4. Mesh simplification of large model (statuette).

4 Summary

To summarize, we present a novel 3D surface mesh simplification algorithm suitable for large scale scanned models. To achieve memory effectiveness and computation efficiency, we use a two-phase approach that takes the advantages of both vertex clustering and iterative edge contraction. By applying a linear combination of error metrics over multiple attributes as the cost function, the proposed algorithm takes multiple attributes into control. Experimental results have shown that the proposed algorithm is both effective and efficient in generating high-quality simplified meshes from large scale scanned models.

Acknowledgement: The authors thank for the help from Dr. Genc and other colleagues at the Siemens Corporate Research, Princeton, New Jersey.

References