JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 29, 509-525 (2013)

A Simple Yet Effective User Controllable Mesh Simplification

TAN-CHI HO AND JUNG-HONG CHUANG
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan

Mesh simplification has become a key ingredient for real-time graphics applications. However, practitioners have found that automatic simplification methods usually fail to produce satisfactory result when models of very low polygon count are desired. This is due to the fact that existing methods take no semantic or functional metric into account, and moreover, each error metric proposed previously has its own strength and weakness. In this paper, we propose a user-controllable mesh simplification framework that allows users to achieve a predictable resolution improvement in selected regions of a simplified mesh derived by using any error metric. The framework consists of two stages. The first stage employs weighting schemes that allow users to refine unsatisfactory regions to a user-expected resolution. The second stage is a local refinement aiming to provide a user-guided fine-tune to recover local sharp features. Two weighting schemes, namely uniform weighting and nonuniform weighting, are proposed. In uniform weighting scheme, a weight value is applied to all original mesh vertices in a selected region, resulting in a uniform improvement on vertex resolution in the region, while in the nonuniform weighting scheme varying weights are applied to vertices in a selected region and obtain a nonuniform resolution improvement in the region. Different from the previous user-assisted simplification methods that reorder the collapsing sequence indirectly by weighting the collapsing cost of edges, the two proposed weighting schemes directly reorder the edge collapsing sequence, ensuring a predictable resolution improvement in the selected regions. Moreover, the proposed weighting schemes are independent on error metric used for simplification and ensure the same resolution improvement when a weighting value is applied to simplified meshes with different resolutions.

Keywords: level of detail, mesh simplification, user-assisted simplification, geometric modeling, computer graphics

1. INTRODUCTION

Polygonal mesh is one of the most common model presentations in computer graphic applications. With the development of 3D scanning technologies and modeling tools, raw meshes can be composed of thousands to millions of polygons. Real-time rendering of such a large amount of data is always a challenge.

Many mesh simplification algorithms have been proposed to decrease the complexity of models while maintaining similarity with the original models. Among the previously proposed methods, progressive mesh approach employed a series of primitive collapsing, such as edge collapsing, in the increasing order of simplification cost [1-4]. The algorithms usually differ in how the simplification cost is measured. Each of these metrics has its own strength and weakness in preserving geometric and texture features. However, all of these metrics do not take semantic or functional features into account. As

---

Received April 14, 2011; accepted June 13, 2011.
Communicated by Jen-Hui Chuang.

a result, practitioners have found that existing metrics are not able to produce satisfactory result when the simplified mesh of very low-polygon count is expected.

To overcome such limitations, the concept of user-assisted or user-guided simplification becomes attractive. One way to this end is to perform refinement or simplification on the simplification hierarchy [5-7]. Such a setup is usually constrained by the vertex-split dependence problems. Another approach reorders the primitive collapsing by weighting the collapsing cost [8, 9]. Since the collapsing cost cannot be described by a simple function, the weights applied have no direct relation to the result of the refinement. In consequence, the weights are usually chosen in a trial and error basis. Moreover, the weights that are appropriate to a simplified mesh derived by an error metric may not be appropriate to the one derived by another error metric.

In this paper, we propose a user-controllable simplification framework that allows users to improve the quality of simplified meshes derived by using any existing error metric, such as QEM [2] or APS [3]. The framework consists of two stages. The first stage employs weighting schemes that allow users to refine unsatisfactory regions and achieve a user-expected resolution improvement. The second stage is a local refinement scheme that utilizes vertex splits performed on the vertex hierarchy [10], aiming to provide a user-guided fine-tune for recovering sharp features. Two weighting schemes are proposed, namely uniform weighting and nonuniform weighting. In uniform weighting scheme, a weight value is applied to all original mesh vertices in a selected region, resulting in a predictable uniform improvement on vertex resolution in the region. On the other hand, in the nonuniform weighting scheme varying weights are applied to vertices in a selected region and obtain a predictable nonuniform resolution improvement in the region. The proposed weighting schemes differ from the previous approaches [8, 9] in that the proposed weighting schemes reorder the edge collapsing sequence directly rather than by changing the collapsing cost and then reordering the collapsing sequence indirectly. The proposed reordering mechanisms are designed to achieve the following goals:

- The resolution improvement for a given weighting value in a selected region is predictable.
- The weighting schemes are completely independent of the error metric used, that is, same resolution improvement for a weighting value is obtained no matter which error metric is used.
- A weighting value will result in the same resolution improvement when it is applied to simplified meshes in different resolutions.

2. RELATED WORK

Level-of-detail (LOD) modeling aims to represent a complex mesh with several levels of detail, and from which an appropriate level is selected at run time to represent the original mesh. A number of methods have been proposed in the literature. Most methods simplify the given mesh by using a sequence of primitive collapsing operations, such as edge collapse [11], triangle collapse [12], vertex clustering [13], vertex removal [14], and multi-triangulations [15].

The primitive collapsing operations can be organized in various orders. The simplest way is to perform the operations in arbitrary order. A more sophisticate approach is to perform the operations in the increasing order of collapsing cost, which is analogous to the greedy algorithm. Several error metrics have been proposed to determine the cost of an edge collapsing operation, such as quadric error metrics (QEM) [2], appearance-preserving simplification (APS) [3], image-driven simplification (IDS) [4], and perceptually guided simplification of lit, textured meshes [16]. Each error metric has its own strength and weakness in preserving certain properties of the original mesh. For example, quadric error metrics [2] tends to preserve only the geometric accuracy during the simplification process, appearance-preserving simplification (APS) [3] takes the texture deviation into account, and image-driven simplification [4] aims to preserve the visual fidelity between the simplified mesh and the original mesh. Moreover, these metrics fail to consider semantic or functional features on the models. As a result, it is found in practice that these metrics alone are not able to produce satisfactory results when very low polygon count is the goal.

The first system that allows users to guide the simplification is Zeta proposed by Cignoni *et al.* [5]. Zeta takes a pre-computed sequence of primitive simplifications as an input, and utilizes hyper-triangulation model, which employs vertex decimation as the local mesh reduction operator. Users can selectively refine a model by locally changing error thresholds to extract different approximations that did not appear during the original simplification process. Semisimp proposed by Li and Watson [6] provides three approaches for users to manipulate the simplification results using the simplification hierarchy. It allows users to improve mesh quality by manipulating the simplification orders, vertex positions, and the hierarchical partitioning of mesh during the simplification.

Kho and Garland [8] proposed a user-guided mesh simplification system particularly for meshes derived using QEM [2]. To increase resolution in a selected region, the system multiply quadric errors associated with vertices in the region by the weighting multiplier, and hence postpone the edge collapse operations in the region. The constraint quadrics can be augmented into optimal placement computation to bias the optimal position towards the constrained planes. Pojar *et al.* presented an approach that is very similar to the work of Kho and Garland [9]. A sophisticated Maya plug-in is provided to offer rich interface and great compatibility with other modeling applications. Since the distribution of QEM during simplification can not be described by a simple function, the weighting approach proposed in [9] suffers from the problem that the value of multiplier has no direct relation to the increase in resolution. In consequence, the value of multiplier is chosen in a trial and error basis.

Hussain *et al.* [7] proposed a unified framework, called adaptive simplification model (ADSIMP), for constructing multiresolution meshes based on the simplification hierarchy and hyper-triangulation model [5]. It provides the capability of real-time navigation across continuous LODs of meshes. Two operations, selective refinement and selective simplification, are provided to fine tune the simplified mesh at any level of detail.

A user-assisted simplification method for converting CAD models into the triangle meshes with boundary preserving was proposed by González *et al.* [17]. The method allows users to specify different levels of detail for each sub-object of a CAD model, and ensures the consistency of boundaries between sub-objects. However, the requirement of specifying sub-objects limits its usefulness only to the man-made CAD models.

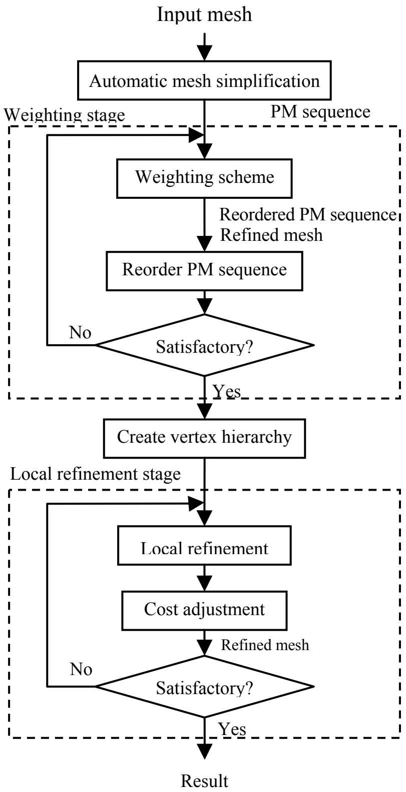


Fig. 1. System overview.

3. VOLUME BASED MESH SEGMENTATION

3.1 Overview

The proposed user-controllable simplification framework allows users to achieve a predictable resolution improvement in selected regions for a simplified mesh derived by any existing error metric, such as QEM [2] or APS [3]. The framework consists of two stages. The first stage employs the proposed weighting schemes that allow users to refine the unsatisfactory regions to the user-expected resolutions. The second stage is a local refinement based on the vertex hierarchy [10], aiming to provide a user-guided fine tune to recover sharp features via vertex splits. Fig. 1 depicts an overview of the framework. At the start-up, we construct a progressive mesh (PM) sequence for the input mesh using an automatic mesh simplification algorithm, such as QEM [2] or APS [3]. If the quality of the simplified mesh is not satisfactory, users can mark the unsatisfactory regions on the original mesh, assign weights to the vertices inside the regions, and apply the weighting scheme to increase the vertex resolution in the selected regions. The weighting scheme can be iteratively applied until a satisfactory result is obtained. The vertex hierarchy is then built according to the reordered PM sequence. Finally, if necessary, users can refine the local features such as sharp edges or corners by iteratively performing vertex splits.

Two weighting schemes are proposed. In the first scheme users assign a constant weighting value to all the vertices in a selected region marked on the original mesh. Each edge collapse associated with the vertex in the selected region are then reordered by comparing it to other edge collapses in PM sequence in such a way that the resulting resolution in the region is about a multiple of the number of the vertices in the region defined by the weighting value. Second scheme allows the user to specify a weighting value to each vertex independently in an unsatisfactory region. The reordering of the edge collapse associated with the vertex is compared only to its dependent edge-collapses in the PM sequence. Such a scheme is applied in per-vertex basis and hence provides a nonuniform weighting effect in the region if the varying value is applied to vertices in the region via surface painting system. After the reordering of edge collapses is completed, the input mesh is simplified according to the new order to a mesh that has same polygon count as before.

As stated in section 1, the weighting scheme proposed here reorders the collapsing sequence directly, rather than indirectly via the weighting of the collapsing cost as in [8, 9]. Such a direct reordering mechanism ensures a predictable improvement of vertex resolution in the selected region, normally by an increase as a multiple of the number of vertices in the region. This effect is usually impossible to be achieved by using previous methods. Moreover, the proposed schemes are quite unique in its capability to be both error-metric and resolution independent. That is, same resolution improvement in the selected region will be obtained for a particular weighting value no matter which error metric is used or whatever the resolution is for the simplified mesh.

Each of the two stages has its own strength and weakness. The weighting scheme reorders the edge collapsing sequence and may greatly alter the simplification result. As a result, the weighting scheme is more effective in overall refinement over the selected region, but can be hardly used to fine tune the local features. On the other hand, the local refinement is restricted by the existing vertex hierarchy; but is effective in performing refinement over local areas to recover sharp features. In the meantime, the local refinement has relatively more control on where to get polygon budget, and hence can be applied to models with low polygon count.

Both the nonuniform weighting scheme and local refinement are based on the vertex hierarchy but with different goals and mechanisms. The nonuniform weighting scheme reorders collapsing order for the edge collapse associated with a vertex according to its relation to its designated ancestor in the vertex hierarchy while the local refinement, refines the mesh around a vertex by splitting the vertex; that is, moving up the active cut on the vertex hierarchy.

3.2 Uniform Weighting Scheme

We consider the weighting value as a multiple value for the expected increase on the vertex resolution in a selected region. That is, given a user-specified weighting value ω and a selected region containing n vertices in the simplified mesh, all the edge collapses in the selected region will be delayed such that approximately ωn vertices will be preserved in the region while maintaining the same total polygon count for the simplified mesh.

Before getting into the detailed reordering scheme, we first define the *order* of an edge collapse. Consider a complete progressive mesh sequence (PM sequence) for simplifying a given original mesh to a vertex, the order of an edge collapse is its order in the PM sequence. For all edge collapses in the selected region, we enumerate them from back to front in the complete PM sequence and in the meantime define the *rank* of the edge collapse according to the enumeration. To make the reordering computation clean, the enumeration starts from 0, that is, the rank of the last edge collapse in the selected region is 0, the last second is 1, and so on.

Consider a selected region R specified on the original mesh and a user-specified weighting value ω assigned to the vertex v in R. Let e be the edge that is collapsed to v, and r and o be the rank and order of edge collapse e, respectively. The new rank $\tilde{r}$ of edge collapse e is computed by

$$\hat{r} = \frac{r}{\omega}. \tag{1}$$

The new collapsing order $\tilde{r}$ of edge collapse e is obtained by the linear interpolation between o_i and o_{i+1}, where $r_i < \tilde{r} \leq r_{i+1}$. That is, for the edge collapse e having new rank $\tilde{r}$, we first find o_i and o_{i+1} such that $r_i < \tilde{r} \leq r_{i+1}$, and then perform the following linear interpolation:

$$\hat{o} = (\hat{r} - r_i) \times o_{i+1} + (r_{i+1} - \hat{r}) \times o_i. \tag{2}$$

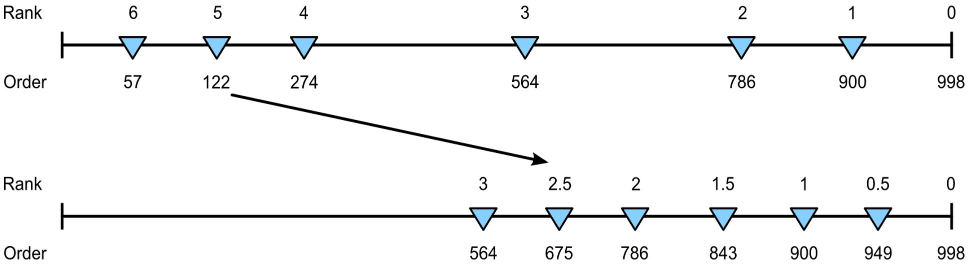


Fig. 2. Reordering edge collapses in the selected region after applying weighting 2 (uniform weighting scheme).

Let's illustrate the reordering process using the example shown in Fig. 2, where the triangle dots on the top horizontal line indicate edge collapses in the selected region and their ranks and orders before the weighting value 2 is applied, while the triangle dots on the bottom horizontal line represent reordered edge collapses and their new ranks and orders. The edge collapse with rank 5 is assigned a new rank 2.5 (= 5/2), and its new order 675 is the result of a linear interpolation between the orders of edge collapses whose ranks are 3 and 2 before weighting. As a result of the reordering, the first of these six edge collapses is reordered to a place where the fourth edge collapse most likely lies. Since the weighting scheme doesn't take the collapsing cost into account, it is apparent that its effectiveness is independent of the error metric employed.

Since the proposed weighting scheme determines the new order for an edge collapse according to where its new rank lies in the original PM sequence, its effectiveness is applied to whole PM sequence. Hence the effectiveness of a particular weighting value works for simplified meshes of different resolution. Take the example shown in Fig. 3, where weighting value is 3 and M_1, M_2, and M_3 represent the termination points of simplified meshes of three different resolutions. We can see that there are one edge collapse remains before the collapsing terminates for M_1. After applying weighting value 3, the number of edge collapses remain becomes 3. Similar results are observed for M_2 and M_3.

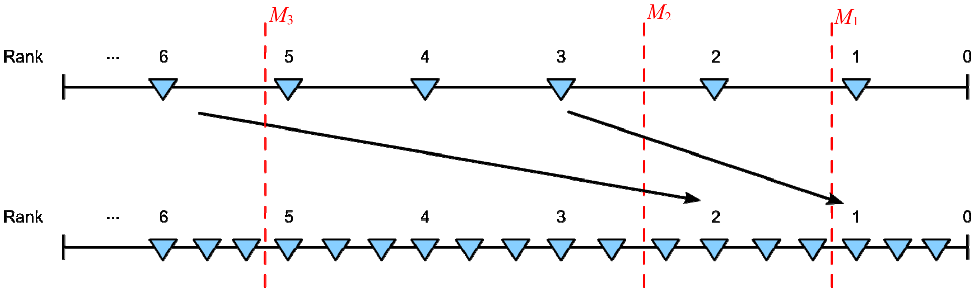


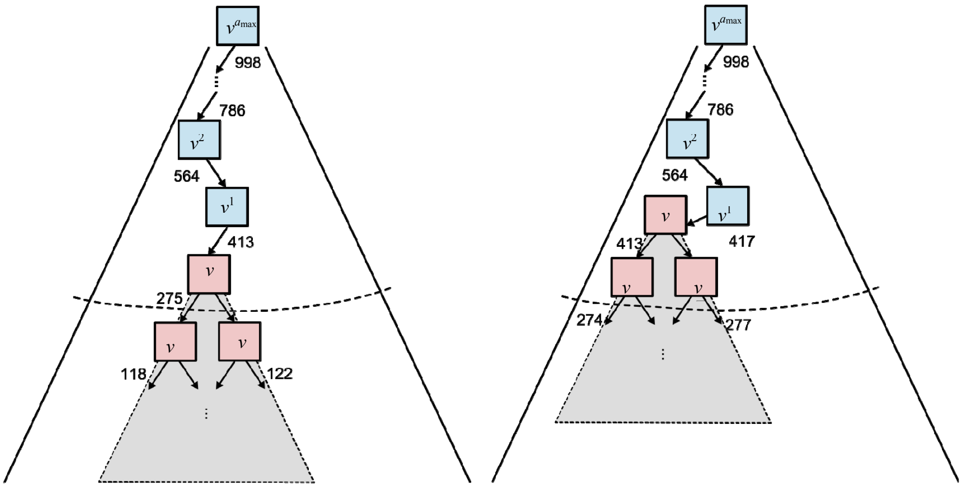
Fig. 3. The effect of applying a weighting value to the selected region with different resolution (uniform weighting scheme).

3.3 Nonuniform Weighting Scheme

In the uniform weighting scheme, the orders of all edge collapses in the selected region are delayed with the same amount of gap in the PM sequence such that an expected resolution increase defined by the weighting value can be achieved. In this weighting scheme, a consistent weighting value is applied to all vertices in the selected region. Such a setting may limit the flexibility that the designers expect to have; for example designers may expect to have different resolution increases for vertices within the unsatisfactory region.

We next propose a weighting scheme in which a weighting value is assigned to each individual vertex in the original mesh and then an increase in vertex resolution indicated by the weighting value will be obtained around the vertex. Thus for a selected region on the original mesh, different weighting values can be applied to vertices in the region by using a surface painting system and, as a result, varying resolution increases will be achieved within the region.

To this end, we formulate the weighting schedule based on the vertex hierarchy formed by the PM sequence; as shown in Fig. 4. Consider the edge e that collapses to the vertex v. If the collapsing order of e is delayed to that of its parent, our goal is to obtain two vertices to replace v. Similarly, four vertices is expected to be obtained if the order of e is delayed to that of its grandparent. Based on the aforementioned observation, when the collapsing order of the edge e associated with a vertex v is delayed to that of v's ith ancestor on the vertex hierarchy, our goal is to obtain 2^i vertices to replace v. Consider an edge e and its collapsed vertex v. Suppose ω is the weight assigned to the vertex v, indicating the number of vertices expected to replace v in current level. We first find i such that $2^i < \omega \leq 2^{i+1}$ and then compute the target collapsing order of e, denoted by $\tilde{o}$, by linearly interpolating the collapsing orders of v's ith and $(i + 1)$th ancestors, as follows,



(a) Before weighting. (b) After apply weighing value of 2.

Fig. 4. Effect to the nonuniform weighting scheme.

$$\hat{o} = \alpha o_{i+1} + \beta o_i, \tag{3}$$

where o_i and o_{i+1} are edge-collapse orders of v's ith and $(i + 1)$th ancestors, respectively, and

$$\alpha = \frac{\omega - 2^i}{2^{i+1} - 2^i},$$

and

$$\beta = \frac{2^{i+1} - \omega}{2^{i+1} - 2^i}.$$

The weighting value ω should be bounded since the highest ancestor for a vertex v is the root of vertex hierarchy. Let's denote the root as the a_{max}th ancestor of the vertex v. The weighting value ω assigned to v should be bounded by $2^{a_{max}}$; that is, ω should be clamped to $\min(\omega, 2^{a_{max}})$.

So far we have described how to reorder the collapsing order of the edge associated with a vertex in the nonuniform weighting scheme. To make it really works, by that we mean 2^i vertices is obtained to replace the vertex v if the collapsing order of the edge e associated with v is delayed to that of v's ith ancestor on the vertex hierarchy, we need also to assign the same weighting value to the descendants of v, at least down to level of 2^{i-1} on the vertex hierarchy. In our interface, after examining the simplified mesh users assign varying weighting value to vertices in the selected regions on the original mesh by using a surface painting tool. For a vertex inside the selected regions, its descendants are around (some of them may be outside the region), and therefore are likely to be assigned with some weighting values. Due to this interface design, as we will see in the result section, the nonuniform weighting scheme may not always achieve the expected resolution increment.

3.4 Local Refinement

The weighting scheme, including previous methods, generally cannot recover sharp features, such as sharp edges and corners. The second stage of our user-controllable simplification framework is a local refinement scheme aiming to provide an effective tool for recovering local sharp features. The proposed refinement operation is similar to the selective refinement and simplification in view-dependent level-of-detail modeling [10]. The selective refinement (simplification) refines (simplifies) a mesh by moving down (up) the active cut of the vertex hierarchy.

Given a simplified mesh with its progressive mesh sequence, normally the result of the first stage, the system constructs the corresponding vertex hierarchy with collapsing cost recorded on each vertex and the active cut associated with the given simplified mesh. To do the local refinement, user selects a set of vertices and the system will perform vertex split on these vertices, and in the meantime do the vertex collapsing on some vertices to maintain the polygon count. Those vertices that have the lowest collapsing cost are the candidate vertices for edge collapsing. Note that the vertex split or collapsing are just the moving down or up of the active cut.

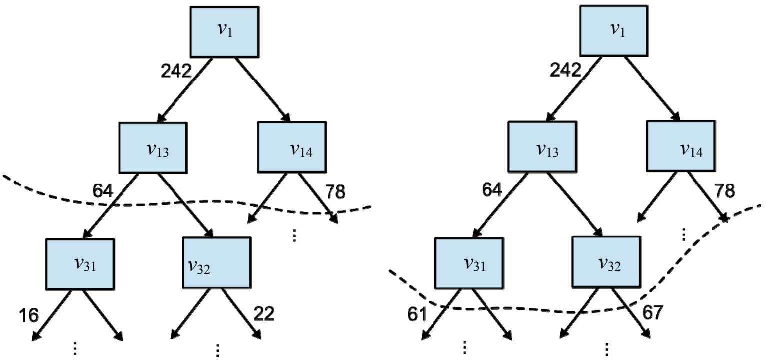
One thing worth mentioning is that the vertex split dependency problem may limit the ability of local refinement since a vertex can be split only if all its neighboring vertices after split are reachable. In our implementation, such problems are overcome by applying the approach proposed in [18]. Another issue needs to be addressed is that, after local refinement, vertices resulting from a vertex split normally have costs lower than their parent. After a sequence of vertex splits applied to a vertex v, the subtree originates from v may have leaf vertices whose costs are relatively lower than that of vertices in the active cut. This implies that the split vertices may soon be collapsed when vertices in other region are split. To prevent this problem, we need to adjust the costs of split vertices such that they have about the same magnitude as the cost of v. Further, the cost difference for vertices in the subtree should be maintained to preserve the local features.

The cost adjustment is done along with the vertex split operations in the local refinement process. Let c_v be the cost of a vertex v to be split, and c_1 and c_2 be the costs of the children of v. Suppose $c_1 \geq c_2$, c_1 and c_2 are adjusted to c^*_1 and c^*_2 as follows:

$$\begin{aligned} c^*_1 &= c_v + \frac{c_1 - c_2}{2}, \\ c^*_2 &= c_v - \frac{c_1 - c_2}{2}. \end{aligned} \tag{4}$$

Note that Eq. (4) ensures that the average cost of the split vertices is the same as their parent and the cost difference between the split vertices is maintained.

As shown in Fig. 5, the costs of v_{31} and v_{32} are adjusted after v_{13} is split, and the average cost of v_{31} and v_{32} are the same as their parent v_{13}. Moreover, the difference between v_{31} and v_{32} remains the same after local refinement.



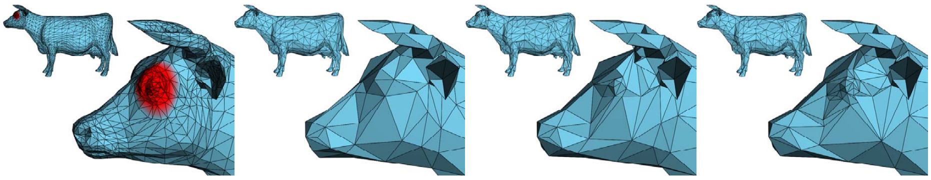
(a) Before the vertex split. (b) After the vertex split.

Fig. 5. Cost adjustment in local refinement process.

4. EXPERIMENTAL RESULTS

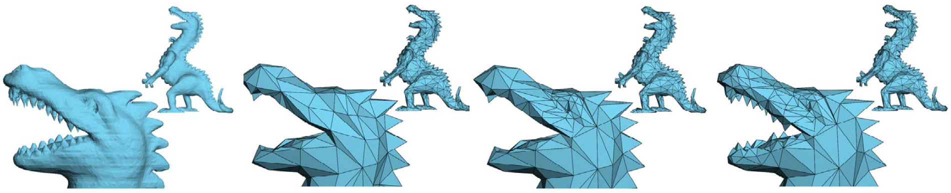
In the implementation, the proposed user-controllable mesh simplification framework supports QEM [2] and APS [3] as the cost measure for the edge collapsing. To preserve the simplification styles of the employed error metric and the consistency of mesh connectivity around the boundary of selected regions, the simplification after applying weighting is executed in the same way as the automatic simplification process with the error metric, except that the edge collapses associated with the weighted vertices are not performed until the delayed orders are encountered.

Several experimental tests are performed to demonstrate the effectiveness of the proposed weighting schemes. First example is a cow model of 5,804 polygons (Fig. 6 (a)), which is simplified to a mesh of 1,160 polygons (20% of the original mesh) by using QEM [2] (Fig. 6 (b)). Different weighting values are applied to the region of left eye using uniform weighting scheme as shown in Fig. 6 (a) by red color. Figs. 6 (c) and (d) depict simplified result and the refined meshes after applying weighting values 2 and 3, respectively. Figs. 7 and 8 illustrate the effectiveness of two-stage user-controllable simplification on the dragon model of 50,000 polygons and male model of 151k polygons. Both models are first simplified to meshes of 1,500 polygons using QEM. For the dragon model, the uniform weighting scheme with weight value of 3 is applied to the regions of eyes, with the resultant mesh shown in Fig. 7 (c). Local refinement is then applied to areas of teeth and nose, producing refined mesh shown in Fig. 7 (d). For the male model, the uniform weighting scheme with weight value of 3 is applied to regions of eyes, lips, and nose, and local refinement is applied to recover sharp features such as eyeballs, eyebrows, and nose. Tables 1 and 2 list the geometry and normal deviations, respectively, before and after the user-controllable simplification for the male model. The errors are measured using MeshDev, which is a mesh comparison tool using attribute deviation metric [19]. Although the mean errors after applying user-controllable simplification are slightly increased, the errors are diffused over the regions that are considered perceptually less important. Fig. 9 visualizes the distributions of geometry and normal deviations for the male model. Noticeable improvements can be found in the selected regions, and the compensative error introduced by the proposed scheme is almost invisible and diffused over the less-important regions.



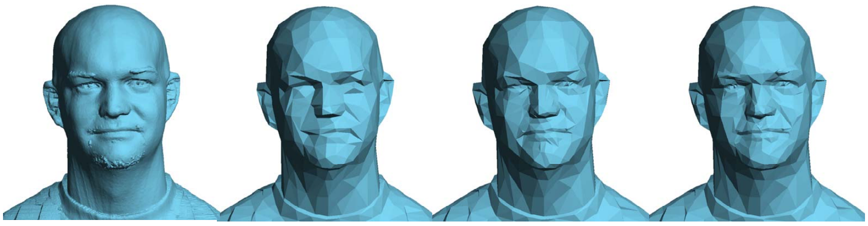
(a) Original mesh. (b) Simplified mesh (8 vertices in the selected region). (c) Weight = 2 (18 vertices in the selected region). (d) Weight = 3 (27 vertices in the selected region).

Fig. 6. Apply uniform weighting on the cow model using different weighting values.



(a) Original mesh. (b) Simplified mesh. (c) After applying weighting scheme. (d) After local refinement.

Fig. 7. Two-stage user-controllable simplification (with uniform weighting) on the dragon model.



(a) Original mesh. (b) Simplified mesh. (c) After applying weighting scheme. (d) After local refinement.

Fig. 8. Two-stage user-controllable simplification (with uniform weighting) on the male model.

Table 1. Geometry deviation of the simplified male model before and after applying the user-controllable simplification.

	Simplified mesh	After applying the proposed method
Minimum	2.758e-8	3.187e-8
Maximum	5.045e-3	5.008e-3
Mean	5.131e-4	5.450e-4
Variance	1.990e-7	1.963e-7

Table 2. Normal deviation of the simplified male model before and after applying the user-controllable simplification.

	Simplified mesh	After applying the proposed method
Minimum	7.984e-4	8.007e-4
Maximum	1.925	1.948
Mean	0.222	0.227
Variance	0.04469	0.04461

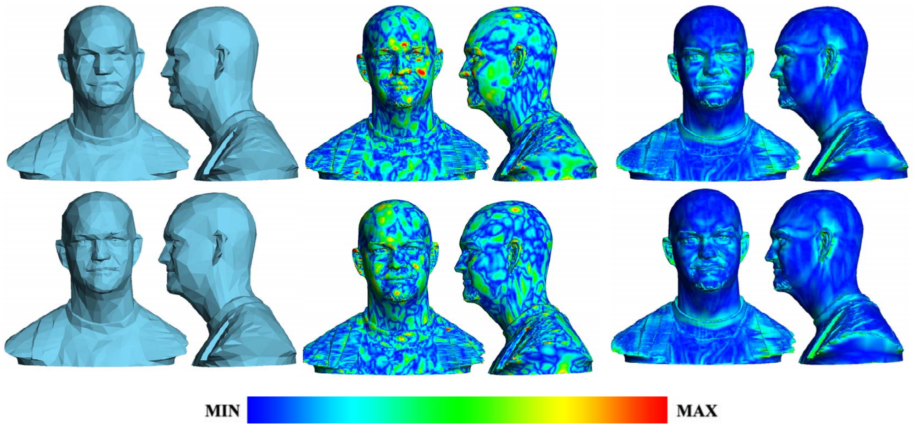


Fig. 9. Visualization of the error distributions for simplified male model before (top) and after apply user-controllable simplification (bottom). The left column are the shaded models, the middle column shows the distributions of geometry deviation, and the right column visualizes the normal deviation. Both deviations are measured by using MeshDev [19].

Fig. 10 illustrates the result of nonuniform weighting scheme and local refinement applied to the buste model of originally 511k polygons. It is first simplified to 1,500 polygons using QEM. Three different levels of nonuniform weights are applied to the model according to the significance in perception; as shown in Fig. 10 (c) on which the green, yellow, and red colors represent the weighting value 2, 3, and 4, respectively. Then, the local refinement is applied to the eyes, nose, and lips to recover crease features.

Fig. 11 compares the effectiveness of the proposed uniform weighting and nonuniform weighting schemes against the one proposed in [8]. The cow model is simplified to 1,160 polygons (20% of the original mesh) using QEM; as shown in Fig. 11 (a). Weighting value 3 is assigned to the left eye as the red region shown in Fig. 6 (a). The proposed uniform and nonuniform weighting schemes yield similar resolution increment for that region, namely increasing from 8 vertices to 27 and 25 vertices, respectively; see Figs. 11 (b) and (c), respectively. The weighting scheme of [8] reorders the edge collapse sequence by directly multiplying the weighting values to the corresponding quadric errors. Since modification of quadric error has no direct link to the resolution improvement, the resolution improvement is not predictable. In this test case, the number of vertices remain unchanged; as shown in Fig. 11 (b). Next, we compare the effectiveness of the proposed uniform weighting, nonuniform weighting schemes, and the weighting scheme in [8] using the buste model. The buste model is first simplified to the meshes of 1,500 polygons. Uniform weighting with values 2 and 3 is applied to the selected regions as shown in Fig. 10. For nonuniform weighting, values similar to that in Fig. 10 are applied to the selected regions. As shown in Fig. 12, the uniform weighting with value 2 may not preserve the eyes well (Fig. 12 (b)) while the uniform weighting with value 3 seems overpreserve the eyes (Fig. 12 (c)). The nonuniform weighting scheme is more capable of adapting to the expectation of users. Again, the weighting scheme of [8] with weighting value of 3 performs badly in this case.

Both of our proposed weighting schemes are independent on the resolution of the given meshes, meaning that similar resolution increment in the selected regions is achieved for the given simplified meshes in different resolutions. Fig. 13 depicts the re-


(a) Original mesh. (b) Simplified mesh. (c) Nonuniform weighting. (d) After applying weighting scheme. (e) After local refinement.

Fig. 10. Two-stage user-controllable simplification (with uniform weighting) on the Buste model.

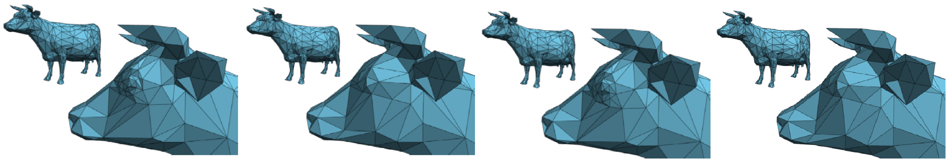

(a) Simplified mesh (8 vertices in the selected region). (b) Uniform weighting (27 vertices in the selected region). (c) Nonuniform weighting (25 vertices in the selected region). (d) Weighting scheme of [8] (8 vertices in the selected region).

Fig. 11. Comparison of the proposed uniform weighting and nonuniform weighting schemes against the weighting scheme in [8].

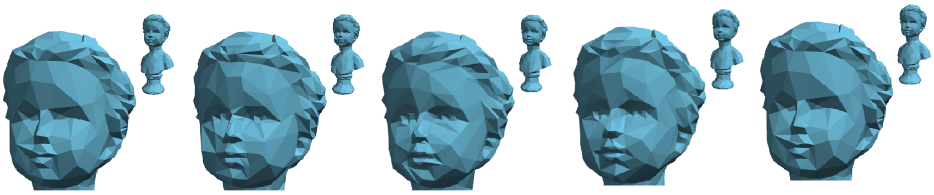

(a) Simplified mesh. (b) Uniform weighting with value 2. (c) Uniform weighting with value 4. (d) Nonuniform weighting. (e) Weighting scheme of [8].

Fig. 12. Comparison of the proposed uniform weighting, nonuniform weighting schemes, and the weighting scheme in [8].

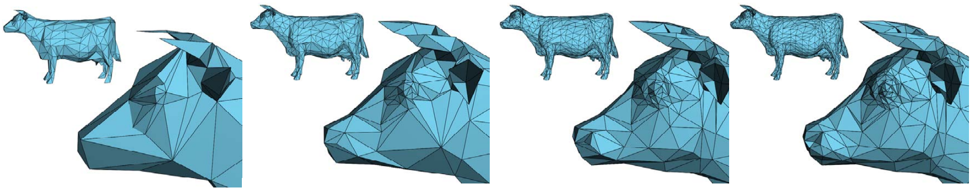

(a) 500 (8.6%) polygons. (b) 1,160 (20%) polygons. (c) 1,739 (30%) polygons. (d) 2,321 (40%) polygons.

Fig. 13. Results of applying the uniform weighting with value 3 to the cow model in different resolutions.

sults of applying the uniform weighting with value 3 to the cow models with polygon counts 500, 1,160, 1,739, and 2,321. The selected region is the same as the one in Fig. 6. The increased number of vertices in the selected region is shown in Table 3, which also includes the performance of the proposed nonuniform weighting scheme and the scheme proposed in [8]. The three numbers in each item of the "vertex count in the selected region" indicate the vertex counts resulting from the uniform weighting scheme (top), the nonuniform weighting scheme (middle), and the weighting scheme of [8] (bottom). We observe that the performance of the uniform and nonuniform weighting scheme is quite close to what we expect. Note that the small inaccuracy in hitting the expected target is due to the dependency problem in the dependency hierarchy that occurs on the boundary of the selected region. On the other hand, the resolution improvement of the weighting scheme of [8] is unpredictable. It is usually hard for users to specify the weight value for an expected resolution improvement.

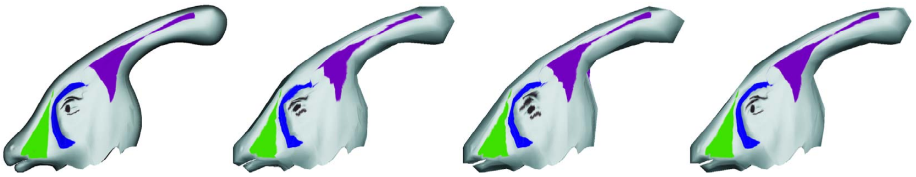
Table 3. Comparison on the resolution improvement obtained by the proposed weighting schemes and the weighting scheme of [8]. The three number in each item of the "vertex count in the selected region" indicate the vertex counts resulting from the uniform weighting scheme (top), the nonuniform weighting scheme (middle), and the weighting scheme of [8] (bottom).

Polygon count of simplified mesh	Vertex count in the selected region		
	Without weighting	Weighting value = 2	Weighting value = 3
500 (8.6%)	4	8 8 3	13 12 3
1,160 (20%)	8	18 16 8	27 25 8
1,740 (30%)	13	26 27 12	39 42 12
2,320 (40%)	22	44 40 23	67 47 23
2,902 (50%)	28	56 49 30	68 61 30

The proposed weighting schemes are also independent on the error metric used in the mesh simplification. Since APS is a texture-deviation error metric, we consider the Parasaur model with texture mapped. The Parasaur model of 7,685 polygons is first simplified to 750 polygons using QEM and APS. Then we apply the uniform weighting scheme with values 2 and 3 to the region of left eye. Table 4 shows the resolution increments in the region of left eye after we apply the uniform weighting scheme with values 2 and 3 to the region. As we can see that the obtained resolution increments are close to what we expect for both metrics. The proposed weighting schemes can be applied to models with texture mapped to reduce the texture distortion. Fig. 14 shows the result of applying the two-stage user controllable simplification scheme to the Parasaur model with texture mapped. Again, the Parasaur model is simplified to a mesh of 750 polygons using APS, on which noticeable texture distortion can be found; as shown in Fig. 14 (b). Figs. 14 (c) and (d) depict a great reduction in texture distortion after applying uniform weighting scheme with weighting value of 3 around the left eye and then local refinement on the texture boundaries.

Table 4. Resolution increment after applying uniform weighting scheme to the mesh simplified using different error metrics.

Error metric	Vertex count in the selected region		
	W/O weighting	Weighting value = 2	Weighting value = 3
QEM	15	33	49
APS	6	15	22



(a) Original mesh. (b) Simplified mesh. (c) After applying uniform weighting scheme. (d) After local refinement.

Fig. 14. Applying two-stage user-controllable simplification to the Parasaur model.

5. CONCLUSIONS AND FUTURE WORK

We have proposed a two-stage user-controllable mesh simplification framework that allows users to improve the quality of simplified meshes derived by any error metric. The first stage employs weighting schemes that allow users to refine a selected region to user-expected resolution. The second stage is a local refinement scheme aiming to provide a user-guided fine-tune to recover local sharp features. Two weighting schemes, namely uniform weighting and nonuniform weighting, are proposed. In uniform weighting scheme, a weight value is applied to all original mesh vertices in a selected region, resulting in a uniform improvement on vertex resolution in the region, while in the nonuniform weighting scheme varying weights are applied to vertices in a selected region and obtain a nonuniform resolution improvement in the region. The proposed weighting scheme differs from the previous approaches in that the weights are used to directly reorder the edge collapsing sequence rather than weighting the collapsing cost. Such a direct reordering mechanism ensures a predictable increase of vertex resolution in the selected region, and is both error-metric and resolution independent. The effectiveness of applying a weighting value to a selected region is the same for meshes derived by any error metric, and for simplified meshes in different resolutions.

In practical modeling applications such as the model design in game industry, it still requires intensive manual effort to create a good level-of-detail model for textured objects. Normally, the texture map for a model is formed by packed texture atlases. When the model is simplified, the method needs to address the texture distortion and texture-continuity problems between atlases induced by simplification. As a future work, we will develop an artist-friendly LOD system that addresses the issues for eliminating the texture distortion [20], enhancing texture-continuity between atlas, and user-controllable simplification.

REFERENCES

1. H. Hoppe, "Progressive meshes," in *Proceedings of ACM SIGGRAPH*, 1996, pp. 99-108.
2. M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of ACM SIGGRAPH*, 1997, pp. 209-216.
3. J. Cohen, M. Olano, and D. Manocha, "Appearance-preserving simplification," in *Proceedings of ACM SIGGRAPH*, 1998, pp. 115-122.
4. P. Lindstrom and G. Turk, "Image-driven simplification," *ACM Transactions on Graphics*, Vol. 19, 2000, pp. 204-241.
5. P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno, "Zeta: A resolution modeling system," *Graphical Models and Image Processing*, Vol. 60, 1998, pp. 305-329.
6. G. Li and B. Watson, "Semiautomatic simplification," in *Proceedings of ACM Symposium on Interactive 3D Graphics*, 2001, pp. 43-48.
7. M. Hussain, Y. Okada, and K. Niijima, "User-controlled simplification of polygonal models," *3D Data Processing, Visualization and Transmission*, 2004, pp. 918-925.
8. Y. Kho and M. Garland, "User-guided simplification," in *Proceedings of ACM Symposium on Interactive 3D graphics*, 2003, pp. 123-126.
9. E. Pojar and D. Schmalstieg, "User-controlled creation of multiresolution meshes," in *Proceedings of ACM Symposium on Interactive 3D Graphics*, 2003, pp. 127-130.
10. H. Hoppe, "View-dependent refinement of progressive meshes," in *Proceedings of ACM SIGGRAPH*, 1997, pp. 189-198.
11. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Proceedings of ACM SIGGRAPH*, 1993, pp. 19-26.
12. B. Hamann, "A data reduction scheme for triangulated surfaces," *Computer Aided Geometric Design*, Vol. 11, 1994, pp. 197-214.
13. J. Rossignac and P. Borrel, "Multi-resolution 3D approximations for rendering complex scenes," *Modeling in Computer Graphics: Methods and Applications*, 1993, pp. 455-465.
14. W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," in *Proceedings of ACM SIGGRAPH*, 1992, pp. 65-70.
15. L. DeFloriani, P. Magillo, and E. Puppo, "Efficient implementation of multi-triangulations," *IEEE Visualization*, Vol. 98, 1998, pp. 43-50.
16. N. Williams, D. Luebke, J. D. Cohen, M. Kelley, and B. Schubert, "Perceptually guided simplification of lit, textured meshes," in *Proceedings of ACM Symposium on Interactive 3D Graphics*, 2003, pp. 113-121.
17. C. González, J. Gumbau, M. Chover, F. Ramos, and R. Quirós, "User-assisted simplification method for triangle meshes preserving boundaries," *Computer Aided De-*

sign, Vol. 41, 2009, pp. 1095-1106.

18. J. Kim and S. Lee, "Truly selective refinement of progressive meshes," *Graphics Interface*, 2001, pp. 101-110.
19. M. Roy, S. Foufou, and F. Truchetet, "Mesh comparison using attribute deviation metric," *International Journal of Image and Graphics*, Vol. 4, 2004, pp. 1-14.
20. C.-C. Chen and J.-H. Chuang, "Texture adaptation for progressive meshes," *Computer Graphics Forum*, Vol. 25, 2006, pp. 343-350.

Tan-Chi Ho (何丹期) received his B.S. degree in Computer Science from National Chiao Tung University, Taiwan, in 2001, and he is currently a Ph.D. student in Department of Computer Science, National Chiao Tung University. His research interests are geometric modeling, GPU based simulation, and visibility pre-processing in computer graphics.

Jung-Hong Chuang (莊榮宏) is a Professor of Computer Science Department at National Chiao Tung University, Taiwan. He received his B.S. degree in applied mathematics from National Chiao Tung University, Taiwan, in 1978, and M.S. and Ph.D. degrees in Computer Science from Purdue University in 1987 and 1990, respectively. Chuang's primary research interests are in the areas of 3D graphics, virtual reality, and geometric modeling. In 3D graphics, he works on the modeling and the real-time rendering techniques for 3D game design, virtual reality and visual simulation.