Mesh Simplification with Hierarchical Shape Analysis and Iterative Edge Contraction

Jingqi Yan, Pengfei Shi, *Senior Member*, *IEEE*, and David Zhang, *Senior Member*, *IEEE*

Abstract—This paper presents a novel mesh simplification algorithm. It decouples the simplification process into two phases: shape analysis and edge contraction. In the analysis phase, it imposes a hierarchical structure on a surface mesh by uniform hierarchical partitioning, marks the importance of each vertex in the hierarchical structure, and determines the affected regions of each vertex at the hierarchical levels. In the contraction phase, it also divides the simplification procedure into two steps: half-edge contraction and optimization. In the first step, memoryless quadric metric error and the importance of vertices in the hierarchical structure are combined to determine one operation of half-edge contraction. In the second step, it repositions the vertices in the half-edge simplified mesh by minimizing the multilevel synthesized quadric error on the corresponding affected regions from the immediately local to the more global. The experiments illustrate the competitive results.

Index Terms—Mesh simplification, object hierarchies, level of detail, shape approximation.

1 Introduction

Polygonal surfaces are commonly used for representing geometric models in a great variety of applications. Advances in imaging devices have made vast and dense sampling data sets of solid objects available: laser range scanners, medical imaging devices, and computer vision systems. Various effective surface reconstruction methods can produce very complex polygonal models from such data sets. While a model with more polygons can capture finer details of the surface, the workload of visualization, process, and transmission increases hugely. Thus, it remains an important problem in visualization and computer graphics to substitute the highly detailed model with faithful level-of-detail models. Mesh simplification is one of effective approaches.

Many impressive algorithms have been developed for mesh simplification in the past 10 years. Most of those algorithms measure the errors caused by simplification operations by immediately local neighborhoods and then perform the simplification operations to minimize the errors on the local regions. Thus, it falls into disorder from the viewpoint of the whole model, which is undesirable to maintain the shape structure of the model and produce better coarser-models. In this paper, we present a two-phase simplification algorithm: shape analysis and edge contraction. It is inspired by Garland's work [9], which pointed out that the quality of simplified models might be improved by decoupling the analysis and synthesis phases of the simplification process.

- J. Yan and P. Shi are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, 1954 Hua Shen Road, Shanghai (200030), China. E-mail: {jqyan, pfshi}@sjtu.edu.cn.
- D. Zhang is with the Center for Multimedia Signal Processing, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: csdzhang@comp.polyu.edu.hk.

Manuscript received 19 June 2001; revised 16 July 2002; accepted 15 May 2003.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 114382.

In the first phase, we partition the original model in a hierarchical way and then impose a uniform hierarchical structure on such a model. The vertices are ranked according to their importance in the structure. The subsequent simplification operations are performed with the guide of this hierarchical structure. Hence, the simplification process is in a stage of order all the time and the shape structure of the whole model is preserved as completely as possible so that the simplified models can't deviate largely from the original model. Furthermore, since an earlier hierarchical shape analysis phase have been performed, for each vertex in the simplified mesh, we can obtain its corresponding affected regions at the different levels and thus reposition it to an optimal position by minimizing the multilevel synthesized quadric error from the immediately local to the more global.

The remainder of this paper is organized as follows: In Section 2, we briefly review previous work related to us. Section 3 describes the detailed procedure for shape analysis. Our scheme for iterative edge contraction is introduced in Section 4. Section 5 illustrates the results of our experiments. Finally, conclusions are drawn in Section 6.

2 Previous Work

Mesh simplification algorithms can be coarsely divided into five categories: vertex decimation [2], [7], [18], [33], [37], vertex clustering [29], [35], region merging [12], [19], [24], [31], subdivision meshes [8], [17], [26], [28], and iterative edge contraction [5], [10], [13], [15], [16], [20], [21], [22], [23], [27], [32], [34], [36]. Because of the large number of published articles on simplification, our review is necessarily incomplete. We will focus on region-merging and edge-contracting simplification algorithms, which are closely related to our work. Some of the surface partitioning algorithms are reviewed.

Surface partitioning algorithms and region merging simplification algorithms. Hinker and Hansen [19] merge quasi-coplanar regions. Maillot et al. [30] partition the mesh by a bucketing of face normals. Eck et al. [8] develop a

Voronoi-based partition. Kalvin and Taylor [24] partition the surface into a set of disjoint face clusters or "Superfaces." Mangan and Whitaker [31] present the curvature-based surface partitioning method by generalizing morphological watersheds to 3D surfaces. Kumar et al. [25] present a normal-based clustering algorithm used for hierarchical back-face culling. Garland et al. [12] describe a process of hierarchical face clustering based on pairwise cluster merging. Once it is divided into several regions, the surface can be simplified by decimating the vertices inside regions and retriangulating the perimeter of the regions [19], [24]. While they can provide the bound on maximum deviation from the original model, region-merging simplification algorithms can't naturally produce a progressive mesh [9].

Simplification algorithms based on edge contraction. Hoppe et al. [23] define an energy function—a measure of distance from the proposed new triangles to a set of sample points on the original mesh—as a quality measure for deciding which edge to collapse. At each step, the element whose elimination causes the lowest increase in the energy function is deleted. Their method can produce high quality results, but may need a very long running time. The enhanced version [20] provides multiresolution management and improves computational efficiency. Later, Popovic and Hoppe [32] extend it to deal with unconnected regions. Guéziec [15], [16] defines and exploits error and tolerance volumes to bound the error locally. The new vertex position is chosen to maintain the enclosed volume of the surface. Ronfard and Rossignac [34] associate a set of planes with each vertex. The error at each vertex is measured by the maximum of squared distances to the planes in its corresponding set. These sets are dynamically merged with the iterative edge contractions. Garland and Heckbert [10] define quadric error metric and store it into a symmetric 4×4 matrix, one matrix per vertex. Moreover, this metric is used both to place the new vertex and to order the list of edge contractions. Recently, they generalized this method to accurately maintain color and texture values [11]. Cohen et al. [5], [6] use edge contraction to produce a mapping between the original mesh and the simplified model. An error box is used to track the greatest deviation between the meshes and this deviation guides which edge is contracted. Gieng et al. [13] present a method using triangle-contracting operations to produce a hierarchy of triangle meshes. Because a triangle can be contracted by contracting two of its edges, their work is also treated as one of the edge-contracting methods. In addition, their method could produce a limited number of intermediate meshes by selecting, at each step, a number of triangles that can be contracted simultaneously. Lindstrom and Turk [27] recently developed a "memoryless" method which does not retain a geometric history during the simplification process.

The important advantage of simplification algorithms based on edge contraction is to use a queue arranged by errors to decide the order of contraction operations. However, the errors are mainly measured by the immediately local regions and, thus, it still falls into disorder from the whole mesh. In these cases, some vertices important for the global shape might be decimated untimely. Unlike those based on edge contraction, region-merging algorithms first perform a procedure for single-level surface partitioning in an approximately global way and then decimate internal vertices prior to corner vertices. In this paper, combining the ideas of such two simplification schemes, a new simplification algorithm is presented. It first performs a procedure of hierarchical surface partitioning and then simplifies the model with edge contraction under the guide of the hierarchical structure; furthermore, in the step of optimization, it optimizes the simplified mesh with a multilevel synthesized quadric error metric. Our normal-based partitioning procedure is from coarse to fine resolution, somewhat similar to the *R-Simp* algorithm [1] based on vertex clustering. However, our simplification procedure is still from fine to coarse resolution and can produce more competitive results in the tested cases.

3 Shape Analysis

In the following sections, we will introduce the detailed procedure of shape analysis: normal-based hierarchical partitioning, connectivity-based repartitioning, and over-segmented region merging.

3.1 Normal-Based Hierarchical Partitioning

We first map a triangulated surface into a unit vector space by the outward-facing normals of triangles. Given a triangulated surface, $S = \{t_i\}_{i=1}^{n}$, where $t_i = (p_{i1}, p_{i2}, p_{i3})$ represents a triangle and p_{i1}, p_{i2}, and p_{i3} are the three vertices of the triangle, respectively. We do a map:

$$\varphi : t_i \in S \rightarrow \mathbf{n}_i, \tag{1}$$

where $\varphi(t_i)$ is a function to calculate the outward-facing unit normal vector of the triangle t_i, that is,

$$\varphi(t_i) = \mathbf{n}_i = \frac{(\mathbf{p}_{i2} - \mathbf{p}_{i1}) \times (\mathbf{p}_{i3} - \mathbf{p}_{i1})}{\|(\mathbf{p}_{i2} - \mathbf{p}_{i1}) \times (\mathbf{p}_{i3} - \mathbf{p}_{i1})\|}, \tag{2}$$

where $\mathbf{p}_{i1}$, $\mathbf{p}_{i2}$, and $\mathbf{p}_{i3}$ are the coordinate vectors of p_{i1}, p_{i2}, and p_{i3}, respectively. In the unit vector space, $\varphi(S)$ is divided into $\varphi(S_1)$, $\varphi(S_2)$, $\dots$, and $\varphi(S_m)$, as well, S is divided into S_1, S_2, $\dots$, and S_m.

Initially, we divide the unit vector space, which is isomorphic to the unit spherical surface, into six clusters. It seems to be similar with the case of using a cube to approximate a sphere, as shown in Fig. 1a. Six outward-facing unit normals are used as the representatives, $\mathbf{H}_j$, $(j = 1, \dots, 6)$, of six clusters. The points, $\varphi(S)$, in the vector space are then divided into six clusters, $\varphi(S_j)$, according to the nearest neighbor principle, i.e., $\forall t_i \in S$, $\varphi(t_i) \in \varphi(S_j)$ if the dot product, $\varphi(t_i) \cdot \mathbf{H}_j$, is maximal.

Next, we divide each nonempty cluster, $\varphi(S_j)$, into four subclusters, $\varphi(S_{jk})$, $(k = 1, 2, 3, 4)$, at the next level of detail. As an example, shown in Fig. 1b, $S_j = EFGH$ is divided into four smaller facets. The new cutpoints of four edges are defined as their normalized middle points. Especially, point *T* is the normalized average of four endpoints of the facet, *EFGH*. The representatives of the subclusters, $\mathbf{H}_{jk}$, are the corresponding outward-facing unit normals of four smaller facets. Then, the elements in $\varphi(S_j)$ are classified: $\forall t_{ji} \in S_j$, $\varphi(t_{ji}) \in \varphi(S_{jk})$, if the dot product, $\varphi(t_{ji}) \cdot \mathbf{H}_{jk}$, is maximal.

3.2 Repartitioning with Connectivity

Since our partitioning performs in the unit vector space, the unconnected triangles with similar outward-facing normals will be classified into the same cluster. In this case, each cluster can't be treated as a patch, which is undesirable to merge oversegmented regions and subsequently simplify

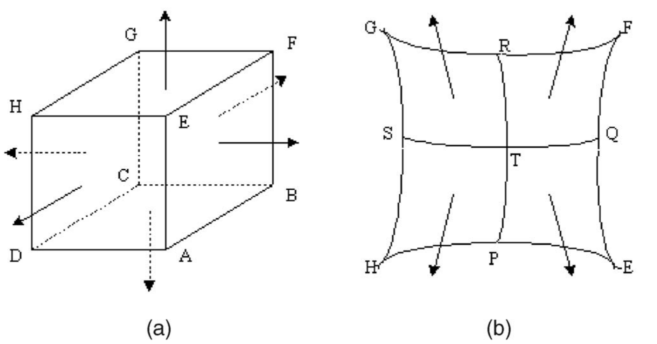


Fig. 1. Normal-based hierarchical partitioning. (a) One-to-six subdivision, like the case of using a cube to approximate a sphere, for the initial level. (b) One-to-dour subdivision for the subsequent levels.

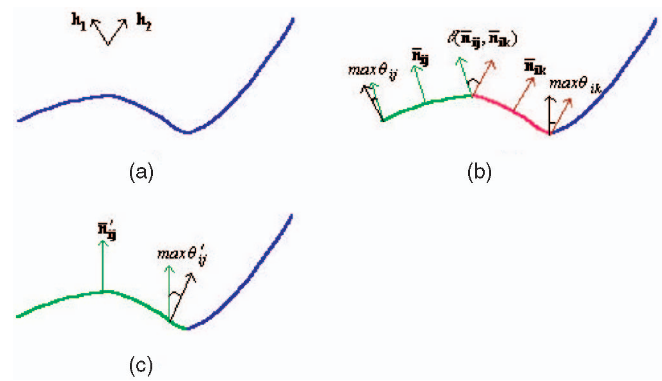


Fig. 2. A 2D case for merging the oversegmented regions. (a) One region (the curved line) and two representatives for subdivision. (b) Three subregions after normal-based partitioning and connectivity-based repartitioning. (c) The first two subregions are merged into one region with new average normal and new maximum normal deviation.

the models with a multilevel synthesized quadric error metric. Therefore, we need to further divide each cluster into connected regions with mesh connectivity.

Select a triangle from the "unvisited" triangles as the seed. Consider edge-sharing triangles of the seed. If one of them is in the same cluster as the seed, the region is grown by adding it. The newly joined triangle is set to "visited." The region goes on growing until all the surrounding triangles belong to different clusters. Reselect a seed triangle and grow it to become a new region by analogy, until all the triangles are repartitioned. As a result, we get a new partition in which the triangles of each cluster are connected together. Thus, each cluster can be treated as a region or a patch.

3.3 Oversegmented Region Merging

During the process of the above-mentioned hierarchical partitioning, it always attempts to divide one region into four subregions to produce the partition of next level. It may result in asynchrony, that is, some regions with small normal discrepancy are subdivided untimely. We use two methods to deal with it. One is to use the maximum inside-cluster normal discrepancy to determine when to subdivide the region. The other is to merge the neighboring regions with small normal discrepancy.

At first, we need the thresholds of different levels. From the description of Section 3.2, we can coarsely evaluate them as:

$$\varepsilon_i = \varepsilon_1 / 2^{i-1}, \tag{3}$$

where ε_i is the threshold of level i and ε_1 is about 0.955 radians (54.7 degree) according to our initial partitioning scheme. Set $max\theta_{ij}$ to be the maximum normal discrepancy of region j at level i. If $max\theta_{ij} < \varepsilon_{i+1}$, region j is not divided from level i to level $i+1$. In addition, we also merge some neighboring regions that are subdivided untimely at the current level. Given two neighboring regions, j and k, at level i, we merge them if they belong to one region at level $i-1$ and if

$$max\theta_{ij} + max\theta_{ik} + \theta(\overline{\mathbf{n}}_j, \overline{\mathbf{n}}_k) < 2\varepsilon_i, \tag{4}$$

where $\theta(\overline{\mathbf{n}}_j, \overline{\mathbf{n}}_k)$ is the angle between the average normals of these two regions. An illustration of a 2D case is shown in Fig. 2. Because the merged regions of current level must be in the same region of previous level, the structure of hierarchies is preserved.

3.4 Open Boundary Edge Partitioning

For open mesh surface, we also partition the boundary edges. Each boundary edge is first mapped into the unit vector space using the unit normal of the plane perpendicular to the plane of the triangle incident to it. Then, like the procedure for partitioning triangles, we can partition the boundary edges hierarchically. At each level, we also need connectivity-based repartition and oversegmented region merging.

3.5 Pseudocode Description for Shape Analysis

We summarize the above-mentioned steps for shape analysis as the format of a pseudocode description, as shown in Fig. 3. The whole shape analysis framework is a hierarchical approach. The total computational complexity is $O(maxlevel \cdot n)$, where n is the total number of triangles. ε_8, the threshold of level 8, calculated by (3), is already equal to 0.00746 radians (about 0.427 degree), so each region at level 8 is much closer to an exact plane. Thus, to set $maxlevel = 8$ is enough for shape analysis of common surfaces. In the implementation, we actually perform the partitioning algorithm to obtain the partitions of $maxlevel - 1$ levels. At level $maxlevel$, each triangle of the original surface is taken as a region.

The function StatisticInformation(), shown in Fig. 3, is especially designed to analyze the useful information at each partitioned level. It will complete two main assignments: ranking the vertices according to their importance in the hierarchical structure and determining the affected regions of each vertex at the hierarchical levels. Both are important for the subsequent phase of iterative edge contraction. We will further introduce them in Section 4.

4 Iterative Edge Contraction

There are many effective simplification schemes such as those reviewed in Section 2. Here, we use the scheme based on edge contraction to simplify the original mesh, but combine some useful ideas of the region-merging approaches. The contraction phase of our algorithm is divided into two steps: half-edge contraction [17] and optimization. Let us begin with the step of half-edge contraction, i.e., one edge is contracted not to an optimal vertex but just to one endpoint.

```
ShapeAnalysis()
{
  For (each level i)
  {
    //***Triangles***//
    NormalBasedPartitioning();
    RepartitionWithConnectivity();
    OverSegmentedRegionMerging();
    //***Open boundary edges***//
    --Omit it here--
    //***Statistics***//
    StatisticInformation();
  }//end for
}

StatisticInformation()
{
  Analysis and record/update
    bvl: The beginning level from which
         one vertex is on the boundaries
         of the regions.
    cvl: The beginning level from which
         one vertex is a corner among
         the regions.
    rtl: The region that one triangle belongs
         to at each level.
    rbl: The region that one open boundary
         edge belongs to at each level.
}
```

Fig. 3. A pseudocode description of the algorithm for shape analysis.

Before the discussion of our half-edge contraction scheme, let us introduce some basics of the traditional edge contraction algorithm [10], [22], [27]. Set $qe(p,\cdot)$ to denote the quadric metric error from a point, p, to one domain. We first show the quadric metric errors from a point to one triangle, $qe(p,t)$, and from a point to one open boundary edge, $qe(p,be)$ as follows:

$$qe(p,t) = A_t \cdot ((\mathbf{p}-\mathbf{p}_t)\cdot \mathbf{n}_t)^2, \tag{5}$$

where A_t is the area of triangle t; $\mathbf{p}_t$ is the coordinate vector of p_t, one vertex of triangle t; and $\mathbf{n}_t$ is the unit normal of triangle t. And,

$$qe(p,be) = L_{be}^2 \cdot ((\mathbf{p}-\mathbf{p}_{be})\cdot \mathbf{n}_{be})^2, \tag{6}$$

where L_{be} is the length of boundary edge be in the open mesh, p_{be} is one of its endpoints, and $\mathbf{n}_{be}$ is the unit normal of the plane perpendicular to the triangle incident to this open boundary edge. Then, we can introduce the quadric metric error of an arbitrary point p to the neighboring domain of p_i as follows:

$$\begin{aligned} lqe(p,p_i) &= qe(p,TD_i) + qe(p,BD_i)) \\ &= \sum_j qe(p,t_{ij}) + \sum_j qe(p,be_{ij}), \\ & t_{ij}\in TD_i, \quad be_{ij}\in BD_i, \end{aligned} \tag{7}$$

where TD_i is the domain consisting of the triangles incident to p_i and BD_i is the domain consisting of the open boundary edges incident to p_i.

Unlike the existing edge contraction schemes, our half-edge contraction scheme considers the importance of the vertices in the hierarchical structure. During the hierarchical partitioning, we use the function, StatisticInformation(), shown in Fig. 3, to obtain the hierarchical information. It

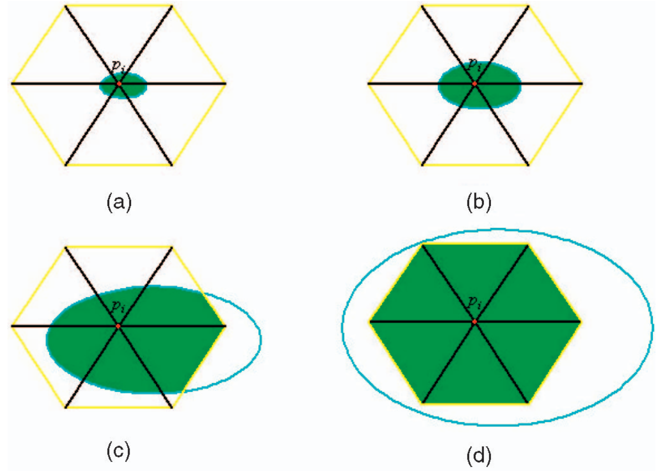


Fig. 4. The affected regions incident to one point, p_i. The yellow lines define the domain incident to p_i in the simplified mesh. The blue lines define the domain incident to p_i in the original mesh at each hierarchical level. The green area illustrates the affected regions incident to p_i at each hierarchical level. (a) At level $maxlevel$. (b) At level $maxlevel-1$. (c) At level 2. (d) At level 1.

TABLE 1
Summary of the Running Time and Other Parameters for Mesh Simplification on the Tested Surface Models by the Presented Algorithm

#Ver	#Tri	Time for Shape Analysis	Time for Half-Edge Contraction	Time for Optimi-zation
The Sphere Model (14,282 vertices, 28,560 triangles)				
1,360	2,716	14.21 (7 levels)	0.83	0.58
366	728		0.85	0.53
98	192		0.86	0.51
26	48		0.86	0.50
8	12		0.87	0.50
The Cylinder Model (2,882 vertices, 4,680 triangles)				
380	640	3.46 (6 levels)	0.22	0.14
186	320		0.22	0.13
96	160		0.22	0.13
48	80		0.22	0.13
24	40		0.23	0.12
The Fandisk Model (7,439 vertices, 12,946 triangles)				
1,618	3,232	7.76 (7 levels)	0.67	0.25
647	1,290		0.69	0.23
323	642		0.69	0.22
129	254		0.69	0.21
64	124		0.69	0.21
The Bunny Model (35,947 vertices, 69,451 triangles)				
3,537	6,945	36.53 (7 levels)	1.94	1.32
1,772	3,472		2.03	1.22
712	1,390		2.08	1.17
359	694		2.11	1.13
181	346		2.12	1.11
The Happy-Buddha Model (543,644 vertices, 1,085,634 triangles)				
54,258	108,562	456.15 (6 levels)	54.76	33.82
26,953	54,280		55.80	31.34
10,659	21,712		56.05	29.98
5,231	10,856		56.42	29.16
2,517	5,428		56.60	28.75

The time in seconds (not including the time to I/O operations) is reported on a 1.2GHz Pentium IV machine with 512M memories.

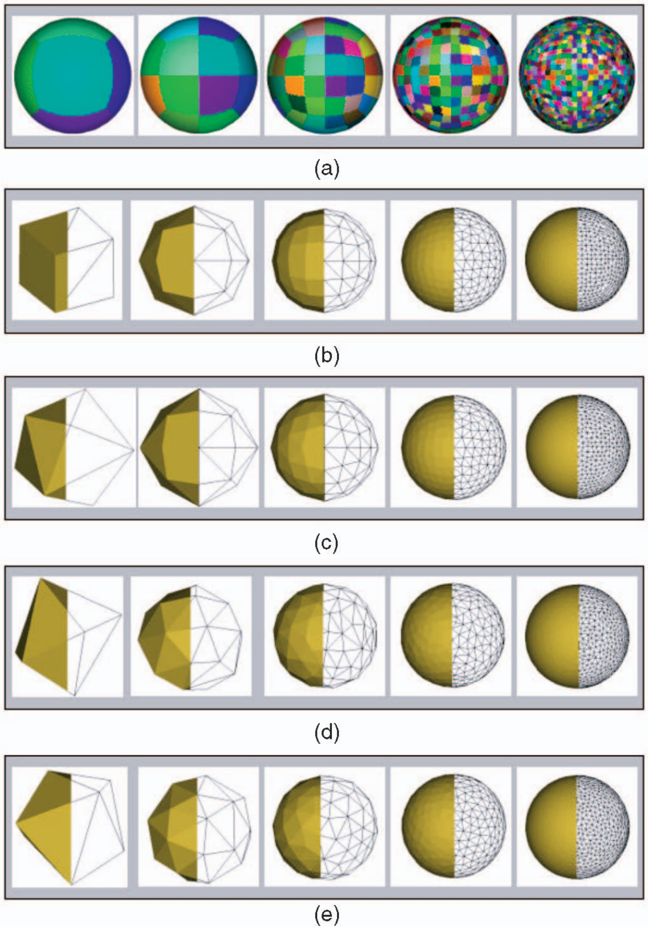


Fig. 5. The results (first five levels) of hierarchical surface partitioning and the results (12, 48, 192, 728, and 2,716 triangles, respectively) of mesh simplification for the Sphere model. (a) The results after hierarchical partitioning. (b) The results of half-edge contraction with the guide of the hierarchical structure. (c) The results of multilevel quadric metric synthesized optimization. (d) The results of Qslim algorithm. (e) The results of MEC algorithm.

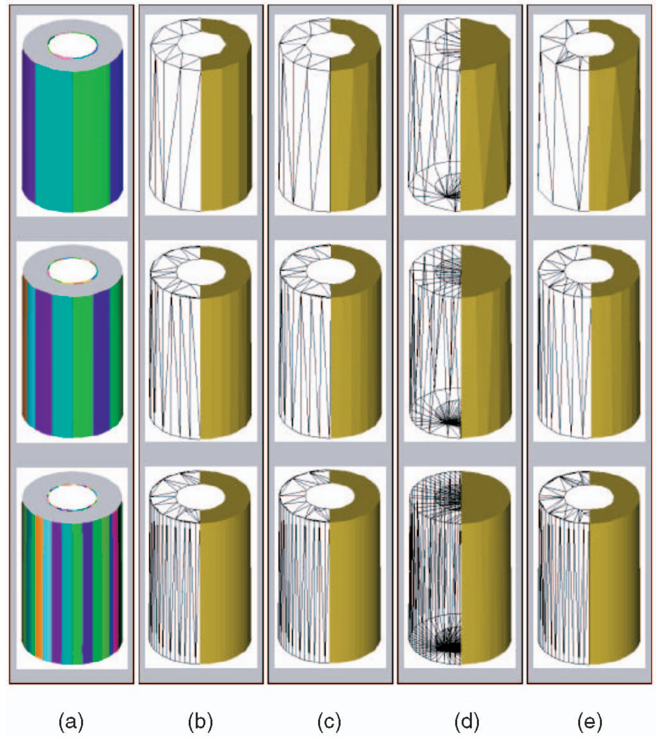


Fig. 7. The results (levels 2, 3, and 4) of hierarchical partitioning (a) (note that the results of hierarchical boundary partitioning are also shown) and the results (80, 160, and 320 triangles, respectively) of mesh simplification for the open Cylinder model. (b) Half-edge contraction with the guide of the hierarchical structure. (c) After multilevel quadric metric synthesized optimization. (d) Qslim algorithm (because of the serious self-intersection, the invisible triangles are not eliminated). (e) MEC algorithm.

will give the information on bvl, cvl, rtl, and rbl. Since rtl and rbl are used in the step of optimization, we explain them in the next paragraph. bvl_i denotes the beginning level from which vertex p_i is on the boundaries of the regions. cvl_i denotes the beginning level from which vertex p_i is a corner among the regions. bvl and cvl mark the importance of each vertex in the hierarchical structure. Vertices with lower bvl

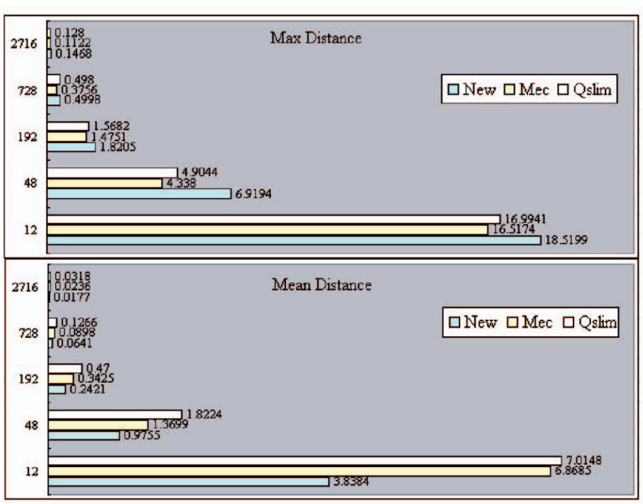


Fig. 6. The bar chart for comparing the results on the Sphere model by the Qslim, MEC, and new algorithm in the *max* and *mean* distances.

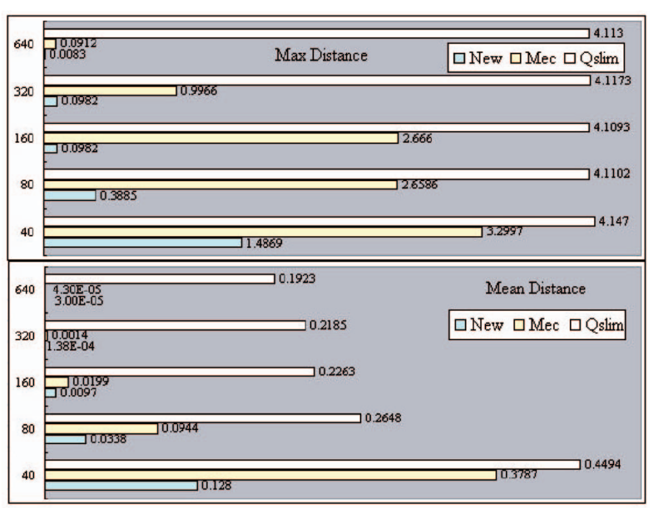


Fig. 8. The bar chart for comparing the results on the Cylinder model by the Qslim, MEC, and new algorithm in the *max* and *mean* distances.

Fig. 9. The results (first four levels) of hierarchical partitioning and the results (40, 80, 160, and 320 triangles, respectively) of mesh simplification for the open Cylinder30 (rotate the Cylinder model by 30° angle). (a) The results after hierarchical partitioning. Note that the results of hierarchical boundary partitioning are also shown. (b) The results of half-edge contraction with the guide of the hierarchical structure. (c) The results of multilevel quadric metric synthesized optimization.

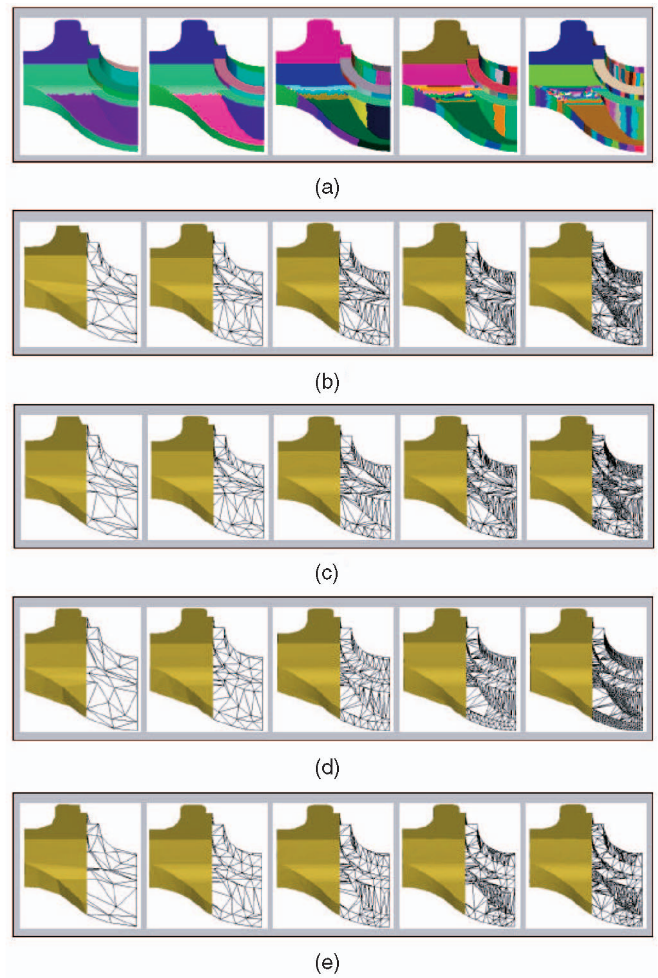


Fig. 10. The results (first five levels) of hierarchical surface partitioning and the results (124, 254, 642, 1,290, and 3,232 triangles, respectively) of mesh simplification for the Fandisk model. (a) The results after hierarchical partitioning. (b) The results of half-edge contraction with the guide of the hierarchical structure. (c) The results of multilevel quadric metric synthesized optimization. (d) The results of Qslim algorithm. (e) The results of MEC algorithm.

and cvl are more important in the hierarchical structure and these values are used to affect the simplification operations. Consider one operation of half-edge contraction, $\tau : (p_i, p_j) \rightarrow \overline{p}$. We determine $\overline{p}$ and the cost for operation τ as follows:

$$\overline{p} = \begin{cases} p_i, \text{if} \begin{cases} bvl_i < bvl_j \\ bvl_i = bvl_j \text{ and } cvl_i < cvl_j \\ bvl_i = bvl_j \text{ and } cvl_i = cvl_j \text{ and } lqe(p_i, p_j) \leq lqe(p_j, p_i) \end{cases} \\ p_j, \text{otherwise} \end{cases} \tag{8}$$

$$c_\tau = lqe(\overline{p}, p_i) + lqe(\overline{p}, p_j). \tag{9}$$

Then, for all edges in the mesh, we use minheap to build a queue by the cost of contraction operation. Iteratively, popup the top of minheap and perform its corresponding contraction: Replace p_i and p_j with $\overline{p}$, delete the degenerated triangles, update $lqe(p, \overline{p})$, update the costs to contract the edges incident to $\overline{p}$, and, finally, update the queue. Note that the operations for updating are similar with memoryless edge collapse (MEC, [22], [27]), that is, $lqe(p, \overline{p})$ is updated by recomputing it according to the updated neighborhood.

In the step of optimization, we reposition the vertices of the mesh achieved by endpoint simplification procedure to discount the mean distance between the simplified mesh and the original mesh. The optimizer takes into consideration the multilevel synthesized quadric metric error. rtl_{lj}, obtained from the phase of hierarchical shape analysis (see Fig. 3), denotes which region one triangle in the original mesh, t_j, belongs to at level l. In the open mesh, rbl_{lj} denotes which region one open boundary edge in the original mesh, be_j, belongs to at level l. Set RTD_{il} to denote the domain consisting of the triangles (of the original mesh) in the regions incident to p_i at level l. It is also required that the triangles in RTD_{il} should fall inside the domain consisting of the triangles incident to p_i in the simplified mesh. An example is shown in Fig. 4. In this figure, the blue lines define the corresponding domain in the original mesh at

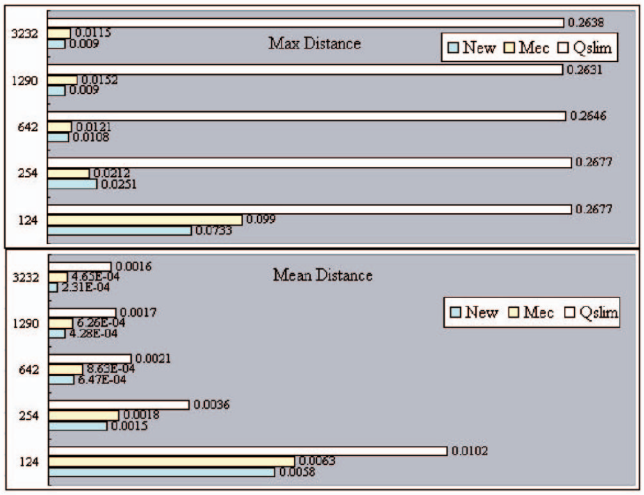


Fig. 11. The bar chart for comparing the results on the Fandisk model by the Qslim, MEC, and new algorithm in the *max* and *mean* distances.

each level, the yellow lines define the corresponding domain in the simplified mesh, and the green area illustrates the domain of RTD_{il} incident to p_i at level l. In the implementation, RTD_{il} is obtained from the two-dimension array of rtl plus the judgment of validness. The multilevel synthesized quadric metric error of one vertex in the simplified mesh, p_i, is defined as follows:

$$\begin{aligned} sqe(p, p_i) &= \sum_i qe(p, RTD_{il}) + \sum_i qe(p, RBD_{il}) \\ &= \sum_i \sum_j qe(p, t_j) + \sum_i \sum_j qe(p, be_j), \qquad (10) \\ & t_j \in RTD_{il}, be_j \in RBD_{il}, \end{aligned}$$

where RBD_{il} denotes the domain consisting of the open boundary edges (in the original mesh) in the regions incident to p_i at level l. In a closed mesh, RBD_{il} is empty. Equation (10) can be simplified, like the work in [10], in the following fashion:

$$sqe(p, p_i) = \mathbf{p}^T \mathbf{A} \mathbf{p} + 2\mathbf{b}^T \mathbf{p} + c, \qquad (11)$$

where $\mathbf{A}$ is a symmetric 3×3 matrix, $\mathbf{b}$ is a 3×1 vector, and c is a scalar. Then, we can obtain the optimal position of p_i by $\mathbf{p} = -\mathbf{A}^{-1}\mathbf{b}$.

TABLE 2
Comparisons of Simplification Algorithms on the Tested Surface Models (the Distances Are Measured by Metro3.1 for Windows)

Distance		Max			Mean			RMS	
Algorithm	Qslim	Mec	New	Qslim	Mec	New	Qslim	Mec	New
Sphere (The original mesh has 28,560 triangles)									
12 tri.	16.9941	16.5174	18.5199	7.0148	6.8685	3.8384	8.0207	7.9359	4.9603
48 tri.	4.9044	4.3380	6.9194	1.8224	1.3699	0.9755	2.1196	1.6003	1.3364
192 tri.	1.5682	1.4751	1.8205	0.4700	0.3425	0.2421	0.5531	0.4025	0.3192
728 tri.	0.4980	0.3756	0.4998	0.1266	0.0898	0.0641	0.1502	0.1058	0.0821
2,716 tri.	0.1280	0.1122	0.1468	0.0318	0.0236	0.0177	0.0379	0.0279	0.0226
Cylinder (The original mesh has 4,680 triangles)									
40 tri.	4.1470	3.2997	1.4869	0.4494	0.3787	0.1280	0.7363	0.6041	0.1862
80 tri.	4.1102	2.6586	0.3885	0.2648	0.0944	0.0338	0.6449	0.1672	0.0477
160 tri.	4.1093	2.6660	0.0982	0.2263	0.0199	0.0097	0.6589	0.1166	0.0135
320 tri.	4.1173	0.9966	0.0982	0.2185	0.0014	1.38e-4	0.6831	0.0265	0.0023
640 tri.	4.1130	0.0912	0.0083	0.1923	4.3e-5	3.0e-5	0.6441	0.0016	2.40e-4
Cylinder30 (The original mesh has 4,680 triangles)									
40 tri.	/	/	1.4869	/	/	0.1497	/	/	0.2134
80 tri.	/	/	0.3885	/	/	0.0335	/	/	0.0477
160 tri.	/	/	0.0982	/	/	0.0104	/	/	0.0145
320 tri.	/	/	0.0982	/	/	1.58e-4	/	/	0.0023
640 tri.	/	/	0.0040	/	/	2.5e-5	/	/	1.77e-4
Fandisk (The original mesh has 12,946 triangles)									
124 tri.	0.2677	0.0990	0.0733	0.0102	0.0063	0.0058	0.0212	0.0111	0.0103
254 tri.	0.2677	0.0212	0.0251	0.0036	0.0018	0.0015	0.0160	0.0032	0.0027
642 tri.	0.2646	0.0121	0.0108	0.0021	8.63e-4	6.47e-4	0.0152	0.0016	0.0013
1,290 tri.	0.2631	0.0152	0.0090	0.0017	6.26e-4	4.28e-4	0.0146	0.0013	9.42e-4
3,232 tri.	0.2638	0.0115	0.0090	0.0016	4.65e-4	2.31e-4	0.0148	0.0011	6.14e-4
Bunny (The original mesh has 69,451 triangles)									
346 tri.	0.8510	0.5375	0.4970	0.0782	0.0614	0.0482	0.1028	0.0794	0.0641
694 tri.	0.6688	0.3947	0.3208	0.0380	0.0282	0.0246	0.0504	0.0363	0.0315
1,390 tri.	0.1900	0.3947	0.1864	0.0203	0.0148	0.0129	0.0264	0.0196	0.0178
3,472 tri.	0.1147	0.1870	0.1856	0.0088	0.0066	0.0058	0.0115	0.0086	0.0079
6,945 tri.	0.1249	0.1307	0.1399	0.0049	0.0038	0.0033	0.0064	0.0050	0.0047
Happy-Buddha (The original mesh has 1,085,634 triangles)									
5,428 tri.	0.2373	0.1911	0.2914	0.0106	0.0130	0.0088	0.0146	0.0176	0.0122
10,856 tri.	0.1980	0.2047	0.1843	0.0056	0.0051	0.0042	0.0083	0.0073	0.0060
21,712 tri.	0.2025	0.1438	0.1837	0.0031	0.0027	0.0024	0.0055	0.0039	0.0035
54,280 tri.	0.1852	0.1431	0.1512	0.0015	0.0011	8.26e-4	0.0033	0.0021	0.0017
108,562 tri.	0.1664	0.1272	0.1131	8.02e-4	6.11e-4	5.25e-4	0.0019	0.0015	0.0013

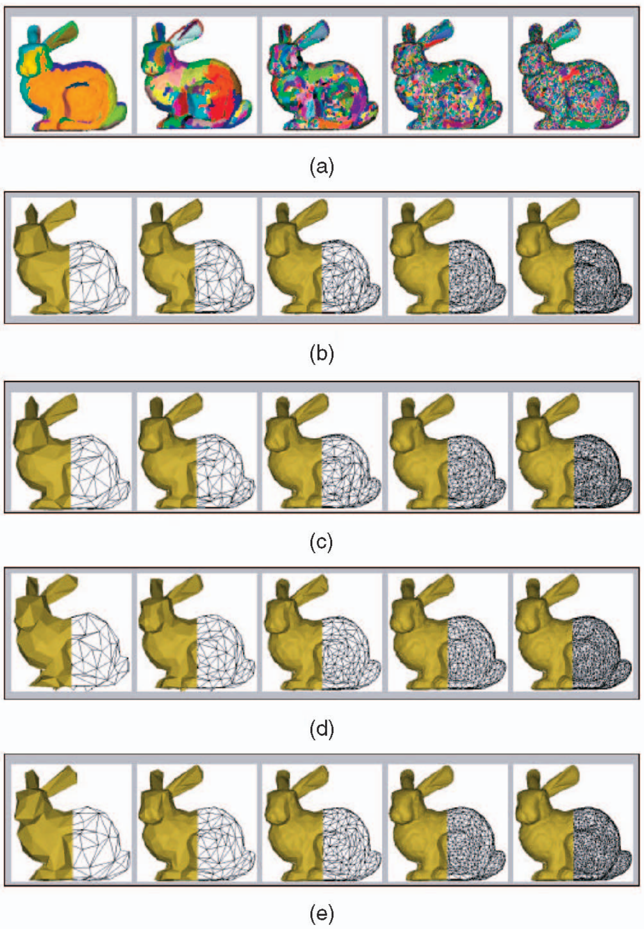


Fig. 12. The results (first five levels) of hierarchical surface partitioning and the results (346, 694, 1,390, 3,472, and 6,945 triangles, respectively) of mesh simplification for the Bunny model. (a) The results after hierarchical partitioning. (b) The results of half-edge contraction with the guide of the hierarchical structure. (c) The results of multi-level quadric metric synthesized optimization. (d) The results of Qslim algorithm. (e) The results of MEC algorithm.

5 Experiments And Comparisons

In our experiments, we used some data sets: Sphere, Cylinder, Fandisk, Bunny, and Happy-Buddha to demonstrate the performance of our simplification algorithm. All tests were performed on a 1.2GHz Pentium IV Intel processor with 512Mbytes memory. All the errors between the simplified mesh and the original mesh were measured by Metro tools [4] (Metro 3.1 for Windows under the default mode), which had been used to evaluate many mesh simplification methods [3], [27]. In each kind of geometrical errors (*max*, *mean*, and *RMS* distances), we chose the bigger one from two values, one from the original mesh to the simplified mesh and the other from the simplified mesh to the original mesh, computed by Metro tools. In the experiments, we also compared our results with those by Quadric Metric Error [10] (Qslim, ran Garland's Version 2.0 implementation under the default mode) and Memoryless Edge Collapse [22], [27] (MEC, ran our implementation directly without the guide of the hierarchical structure).

The running time and other parameters were summarized in Table 1. From Table 1, our simplification algorithm slightly increased the cost of time for shape analysis and multilevel synthesized optimization. We pictured the results of shape

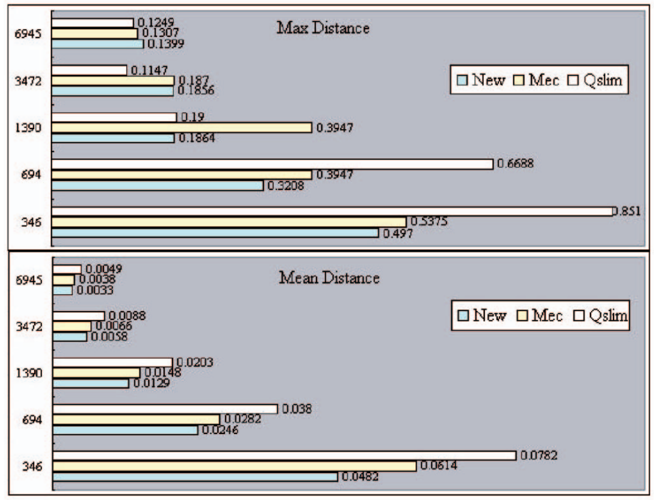


Fig. 13. The bar chart for comparing the results on the Bunny model by the Qslim, MEC, and new algorithm in the *max* and *mean* distances.

analysis for all tested models in Figs. 5a, 7a, 10a, 12a, and 14a. In these pictures, the color of each region was assigned randomly. The half-edge simplified results by our algorithm shown in Figs. 5b, 7b, 10b, 12b, and 14b and those optimized results were shown in Figs. 5c, 7c, 10c, 12c, and 14c. In addition, we also pictured the simplified results by Qslim and MEC algorithms in Figs. 5d, 7d, 10d, 12d, and 14d and Figs. 5e, 7e, 10e, 12e, and 14e, respectively. From the comparisons of those images, one could obtain the initial impression that our simplified results were significantly better in the shape structure than those of other two simplification algorithms. Furthermore, we would compare them in the geometrical errors.

The *max*, *mean*, and *RMS* distances between the simplified mesh and the original mesh were summarized in Table 2. For the intuitive comparison, the bar charts of *max* and *mean* distances (the case in *RMS* distances was much similar than that in *mean* distances) were shown in Figs. 6, 8, 11, 13, and 14. From these figures, one could see that there was no consistent winner in the *max* distance among three algorithms and that our algorithm was the winner of the most cases in the *mean* and *RMS* distances among three algorithms. Compared with the Qslim algorithm, our algorithm commonly provided 40-60 percent reduction in the *mean* and *RMS* distances and, compared with the Mec algorithm, our algorithm commonly provided 20-40 percent reduction in the *mean* and *RMS* distances. It should be pointed out that the implementation of the Qslim algorithm might not perform strict checking for the self-intersection during iterative edge contraction. Hence, the Qslim algorithm might produce some local but large deviations (measured from the simplified mesh to the original mesh) such as those in the simplified Cylinder models (see Fig. 7d), the simplified Fandisk models (see the left-bottom part in Fig. 10d), and the simplified Bunny models (see the bottom part in Fig. 12d).

Since it was performed in the vector space, our hierarchical partitioning approach was not rotationally invariant. The choice of coordinate system might affect the performance of our algorithm. The methods of "body frame" [38] and "oriented bounding box (OBB)" [14] could be used to partly solve this problem. However, it should also be pointed out that the problem was not critical for the whole simplification algorithm because the maximum

Fig. 14. The results (first three levels) of hierarchical surface partitioning and the results (5,428, 10,856, and 21,712 triangles, respectively) of mesh simplification for the Happy-Buddha model. (a) The results after hierarchical partitioning. (b) The results of half-edge contraction with the guide of the hierarchical structure. (c) The results of multilevel quadric metric sysnthesized optimization. (d) The results of Qslim algorithm. (e) The results of MEC algorithm.

normal deviation in all the clusters at level i was limited to ε_i and the oversegmented regions at each level were merged. In fact, we fixed the representatives of the initial partitioning aligned to the positive and negative directions of the axes in all the tested cases. For example, we took the Cylinder model and rotated it by 30° around X-axis to get the Cylinder30 model. The partitioned results at the first four levels, the half-edge simplified meshes, and the optimized results were pictured in Figs. 9a, 9b, and 9c, respectively. The *mean*, *max*, and *RMS* distances were also given in Table 2. Compared with the results for the Cylinder model, while the concrete region that one triangle belonged to might be changed, the performance of the simplification algorithm changed little in the geometrical errors between the simplified mesh and the original mesh.

6 Conclusion

This paper addressed the problem for mesh simplification. We presented a novel algorithm that divided the simplification process into two phases: hierarchical shape analysis and edge contracting mesh simplification. In the phase of shape analysis, we proposed a new normal-based algorithm to build the uniform hierarchies of surfaces. In the next phase, we used iterative edge-contracting algorithm to

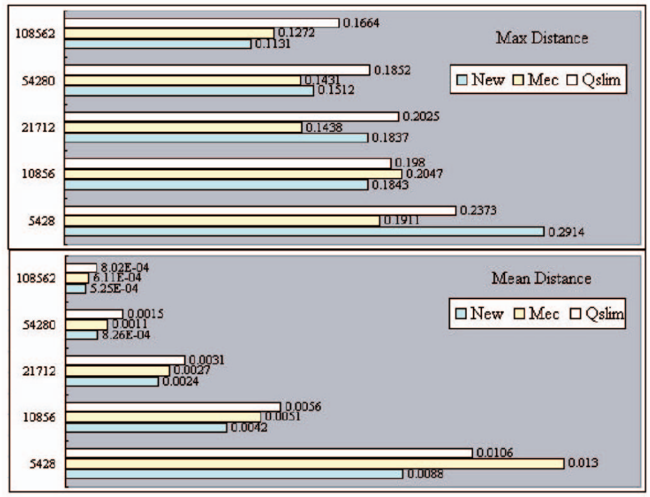


Fig. 15. The bar chart for comparing the results on the Happy-Buddha model by the Qslim, MEC, and new algorithm in the *max* and *mean* distances.

simplify the highly detailed meshes under the guide of the hierarchical structure. The positions of the vertices in the simplified meshes were optimized with the multilevel synthesized quadric metric. In the tested models, our algorithm produced competitive results with respect to the *max*, *mean*, and *RMS* errors. However, our current normal-based surface partitioning method was sensitive to noise. For hugely noisy meshes, it would build a very large number of patches even at the first level of the hierarchy. Thus, many vertices had similar importance in the hierarchical structure. In this case, the performance of the presented algorithm would decrease. In the future, it might be expected to improve the performance of this kind of two-phase simplification algorithms owing to the advance of hierarchical surface partitioning methods.

Acknowledgments

The authors would like to thank all reviewers for their detailed comments and many useful suggestions. The research was partly supported by UGC/CERG Hong Kong Government, Central Fund of The Hong Kong Polytechnic University, and the National Nature and Science Fund in China (No. 69931010, No. 60273200).

References

[1] D. Brodsky and B. Watson, "Model Simplification through Refinement," *Proc. Graphics Interface 2000,* pp. 221-228, May 2000.

[2] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, "Multi-resolution Decimation Based on Global Error," *The Visual Computer,* vol. 13, no. 5, pp. 228-246, 1997.

[3] P. Cignoni, C. Montani, and R. Scopigno, "A Comparison of Mesh Simplification Algorithms," *Computers & Graphics,* vol. 22, no. 1, pp. 37-54, 1998.

[4] P. Cignoni, C. Montani, and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces," *Computer Graphics Forum,* vol. 17, no. 2, pp. 167-174, June 1998.

[5] J. Cohen, M. Olano, and D. Manocha, "Simplifying Polygonal Models Using Successive Mappings," *Proc. Visualization '97,* pp. 395-402, Oct. 1997.

[6] J. Cohen, M. Olano, and D. Manocha, "Appearance-Preserving Simplification," *SIGGRAPH '98 Conf. Proc.,* pp. 115-122, 1998.

[7] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright, "Simplification Envelopes," *SIGGRAPH '96 Conf. Proc.*, H. Rushmeier, ed., pp. 119-128, Aug. 1996.

[8] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," *SIGGRAPH '95 Conf. Proc.*, R. Cook, ed., pp. 173-182, Aug. 1995.

[9] M. Garland, "Multiresolution Modeling: Survey & Future Opportunities," *Eurographics '99, State of the Art Report (STAR)*, 1999.

[10] M. Garland and P.S. Heckbert, "Surface Simplification Using Quadric Error Metrics," *SIGGRAPH '97 Conf. Proc.*, T. Whitted, ed., pp. 209-216, Aug. 1997.

[11] M. Garland and P.S. Heckbert, "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics," *Proc. IEEE Visualization '98*, pp. 263-269, 1998.

[12] M. Garland, A. Willmott, and P.S. Heckbert, "Hierarchical Face Clustering on Polygonal Surfaces," *Proc. ACM Symp. Interactive 3D Graphics*, Mar. 2001.

[13] T.S. Gieng, B. Hamann, K.I. Joy, G.L. Schussman, and I.J. Trotts, "Constructing Hierarchies for Triangle Meshes," *IEEE Trans. Visualization and Computer Graphics*, vol. 4, no. 2, pp. 145-161, Apr.-June 1998.

[14] S. Gottschalk, M. Lin, and D. Manocha, "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," *Proc. SIGGRAPH '96*, pp. 171-180, 1996.

[15] A. Guéziec, "Surface Simplification with Variable Tolerance," *Proc. Second Ann. Int'l Symp. Medical Robotics and Computer Assisted Surgery*, pp. 132-139, Nov. 1995.

[16] A. Guéziec, "Locally Toleranced Surface Simplification," *IEEE Trans. Visualization and Computer Graphics*, vol. 5, no. 2, pp. 168-189, Apr.-June 1999.

[17] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder, "Normal Meshes," *SIGGRAPH '00 Conf. Proc.*, pp. 95-102, 2000.

[18] B. Hamman, "A Data Reduction Scheme for Triangulated Surfaces," *Computer Aided Geometric Design*, vol. 11, pp. 197-214, 1994.

[19] P. Hinker and C. Hansen, "Geometric Optimization," *Proc. Visualization '93*, pp. 189-195, 1993.

[20] H. Hoppe, "Progressive Meshes," *SIGGRAPH '96 Conf. Proc.*, H. Rushmeier, ed., pp. 99-108, Aug. 1996.

[21] H. Hoppe, "View-Dependent Refinement of Progressive Meshes," *SIGGRAPH '97 Conf. Proc.*, T. Whitted, ed., pp. 189-198, Aug. 1997.

[22] H. Hoppe, "New Quadric Metric for Simplifying Meshes with Appearance Attributes," *Proc. IEEE Visualization '99*, pp. 59-66, 1999.

[23] H. Hoppe, T. DeRose, T. Dunchamp, J. McDonald, and W. Stuetzle, "Mesh Optimization," *SIGGRAPH '93 Conf. Proc.*, pp. 19-25, 1993.

[24] A. Kalvin and R. Taylor, "Superfaces: Polygonal Mesh Simplification with Bounded Error," *IEEE Computer Graphics and Applications*, vol. 16, pp. 64-77, May 1996.

[25] S. Kumar, D. Manocha, W. Garrett, and M. Lin, "Hierarchical Back-Face Culling," *Computer and Graphics*, vol. 23, no. 5, pp. 681-692, 1999.

[26] A. Lee, H. Moreton, and H. Hoppe, "Displaced Subdivision Surfaces," *SIGGRAPH '00 Conf. Proc.*, pp. 85-94, 2000.

[27] P. Lindstrom and G. Turk, "Evaluation of Memoryless Simplification," *IEEE Trans. Visualization and Computer Graphics*, vol. 5, no. 2, pp. 98-115, Apr.-June 1999.

[28] M. Lounsbery, T. DeRose, and J. Warren, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type," *ACM Trans. on Graphics*, vol. 16, no. 1, pp. 34-73, 1997.

[29] K.L. Low and T.S. Tan, "Model Simplification Using Vertex-Clustering," *Proc. ACM Symp. Interactive 3D Graphics*, pp. 75-82, 1997.

[30] J. Maillot, H. Yahia, and A. Verroust, "Interactive Texture Mapping," *SIGGRAPH '93 Conf. Proc.*, pp. 27-34, 1993.

[31] A.P. Mangan and R.T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation," *IEEE Trans. Visualization and Computer Graphics*, vol. 5, no. 4, pp. 308-221, Oct.-Dec. 1999.

[32] J. Popovic and H. Hoppe, "Progressive Simplicial Complexes," *SIGGRAPH '97 Conf. Proc.*, T. Whitted, ed., pp. 217-224, Aug. 1997.

[33] K.J. Renze and J.H. Oliver, "Generalized Unstructured Decimation," *IEEE Computer Graphics and Applications*, vol. 16, no. 6, pp. 24-32, Nov. 1996.

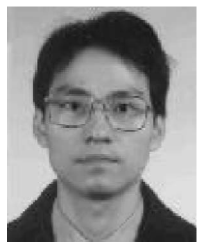
[34] R. Ronfard and J. Rossignac, "Full-Range Approximation of Triangulated Polyhedra," *Proc. Eurographics '96, Computer Graphics Forum*, vol. 15, no. 3, pp. 67-76, Aug. 1996.

[35] J. Rossignac and P. Borrel, "Multi-Resolution 3D Approximations for Rendering Complex Scenes," *Modeling in Computer Graphics*, B. Falcidieno and T. Kunii, eds., pp. 455-465, Springer-Verlag, 1993.

[36] P.V. Sander, J. Snyder, S.J. Gortler, and H. Hoppe, "Texture Mapping Progressive Meshes," *SIGGRAPH '01 Conf. Proc.*, 2001.

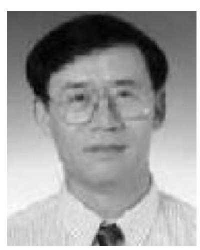
[37] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen, "Decimation of Triangle Meshes," *Computer Graphics (SIGGRAPH '92 Proc.)*, E.E. Catmull, ed., vol. 26, pp. 65-70, July 1992.

[38] T.L. Yong and A.G. Requicha, "Algorithms for Computing the Volume and Other Integral Properties of Solids. 1. Known Methods and Open Issues," *Comm. ACM*, vol. 25, no. 9, pp. 635-641, Sept. 1982.



Jingqi Yan received the BS degree (1996) in automatic control and the MS (1999) and PhD (2002) degrees in pattern recognition and intelligent system from Shanghai Jiao Tong University, Shanghai, China. He is presently a researcher in the Institute of Image Processing and Pattern Recognition at Shanghai Jiao Tong University. His interests include geometric modeling, computer graphics, scientific visualization, and image processing.

Pengfei Shi received the bachelor's and master's degrees in electrical engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 1962 and 1965, respectively. In 1980, he joined the Institute of Image Processing and Pattern Recognition (IPPR), SJTU. During the past 23 years, he has worked in the area of image analysis, pattern recognition, and visualization. He has published more than 60 papers. He is at present the director of the Institute of IPPR at SJTU and a professor of pattern recognition and intelligent system on the Faculty of Electronic and Information Engineering. He is a senior member of the IEEE.



David Zhang graduated in computer science from Peking University in 1974 and received the MSc and PhD degrees in computer science and engineering from the Harbin Institute of Technology (HIT) in 1983 and 1985, respectively. From 1986 to 1988, he was a postdoctoral fellow at Tsinghua University and became an associate professor at Academia Sinica, Beijing, China. He received his second PhD degree in electrical and computer engineering from the University of Waterloo, Ontario, Canada, in 1994. Currently, he is a professor at the Hong Kong Polytechnic University. He is founder and director of two Biometrics Technology Centers supported by the Government of the Hong Kong SAR (UGC/CRC) and the National Nature Scientific Foundation (NSFC) of China, respectively. He is also founder and editor-in-chief of the *International Journal of Image and Graphics* (IJIG). His research interests include automated biometrics-based authentication, pattern recognition, biometric technology and systems. As a principal investigator, he has finished many biometrics projects since 1980. So far, he has published more than 200 papers and seven books, including He is a senior member of the IEEE.