The Visual Computer

Manifold-guaranteed out-of-core simplification of large meshes with controlled topological type

Yong-Jin Liu,
Matthew Ming-Fai Yuen,
Kai Tang

Department of Mechanical Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
E-mail: {liuyj,meymf,mektang}@ust.hk

Published online: 5 November 2003

In this paper, a simple and efficient algorithm is proposed for manifold-guaranteed out-of-core simplification of large meshes with controlled topological type. By dual-sampling the input large mesh model, the proposed algorithm utilizes a set of Hermite data (sample points with normals) as an intermediate model representation, which allows the topological structure of the mesh model to be encoded implicitly and thus makes it particularly suitable for out-of-core mesh simplification. Benefiting from the construction of an in-core signed distance field, the proposed algorithm possesses a set of features including manifoldness of the simplified meshes, toleration of nonmanifold mesh data input, topological noise removal, topological type control and, sharp features and boundary preservation. A novel, detailed implementation of the proposed algorithm is presented, and experimental results demonstrate that very large meshes can be simplified quickly on a low-cost off-the-shelf PC with tightly bounded approximation errors and with time and space efficiency.

Key words: Out-of-core mesh simplification – Large data – Two-manifold meshes – Controlled topological type

Correspondence to: Y.-J. Liu

1 Introduction

Recent advances in laser rangefinder technology have contributed to creating and presenting digital models of cultural heritage artifacts with fine details via the technology of virtual reality and multimedia: we refer to the outstanding work of, e.g., the Digital Michelangelo project at Stanford University (Levoy et al. 2000), the Pieta project at the IBM T.J. Watson Research Center (Bernardini et al. 2002), the Minerva project in the Visual Computing Group (Rocchini et al. 2001), and the acquisition of archaeological relics by the Museum of Terra Cotta Warriors and Horses, China (Zheng and Zhong 1999).

Aimed at creating a 3D archive of highly detailed digital models of cultural artifacts, most high-end 3D scanners sample artworks at a very high resolution and thus lead to huge datasets, For example, as shown in Fig. 1, the original digital David model that stands 5 m tall without its pedestal has 28 184 526 vertices and 56 230 343 triangles and needs 565 MB capacity to store the compressed binary file. Although representing models at their full resolution in high-end workstations suits certain applications, such huge datasets introduce severe problems in any real-time mesh modeling and virtual heritage systems, especially at the steps of data retrieval, editing, and interactive presentation through networks.

To create and manage a virtual heritage system in an inexpensive server, accessing and editing it using a standard PC and taking into account the limited bandwidth of the network, it is necessary to reduce the model complexity to a manageable size using a low-cost PC; an out-of-core mesh simplification is the only solution to this task.

Currently few approaches exist for high-quality out-of-core simplification; most promising schemes utilize the vertex clustering method (Rossignac and Borrel 1992; Lindstrom 2000; Lindstrom and Silva 2001) and its variations (Shaffer and Garland 2001; Garland and Shaffer 2002). A serious disadvantage of these vertex-clustering-based methods is that the simplified output meshes are in general nonmanifold. Numerous model-processing operators, however, require a manifold-guaranteed model as input. An example of dressed David is illustrated in Fig. 2; in this case the mesh-processing operators used such techniques as cut and paste, smoothing, editing, parameterization, texture mapping, and collision detection, which are all based on the manifold meshes. If a vertex clustering method is adopted for large mesh simplification, for downstream mesh processing either a specially designed operator (Gueziec et

Digital Object Identifier (DOI) 10.1007/s00371-003-0222-2

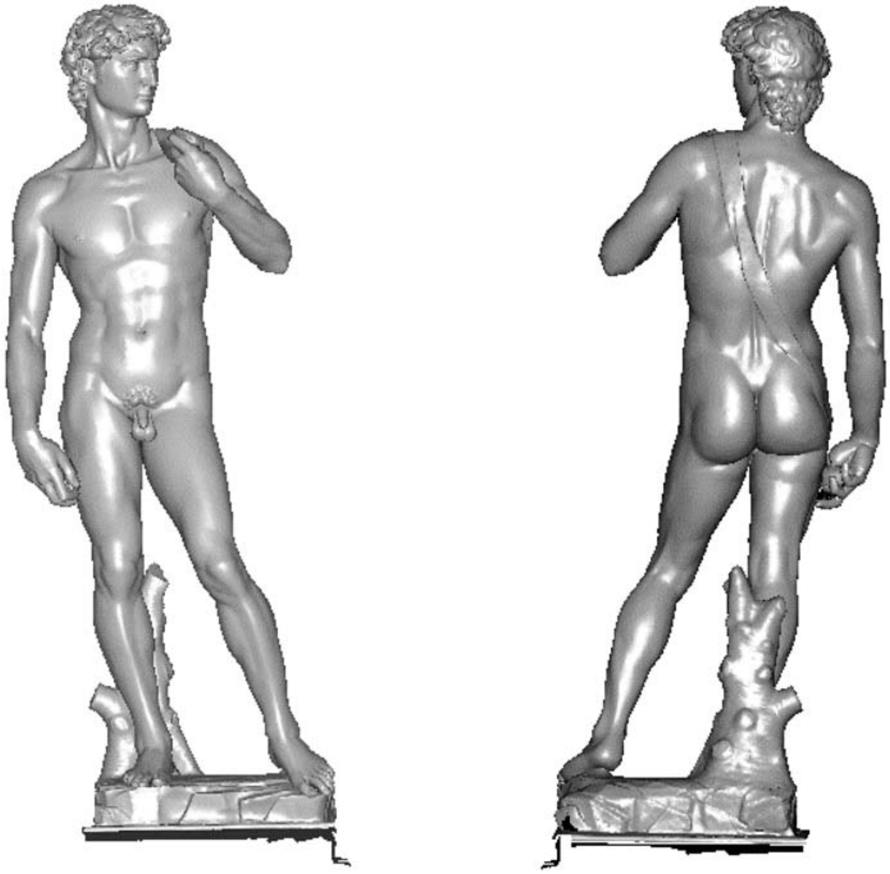

Fig. 1. A simplified digital Michelangelo's David model processed by an off-the-shelf PC. The model is bounded in a cube 2150 mm × 1234 mm × 5424 mm. The original mesh data have 28 184 526 vertices and 56 230 343 triangles and need 565 MB capacity to store the compressed binary file. The simplified mesh data have 56 523 vertices and 111 431 triangles and need 0.99 MB capacity to store the compressed file. Due to the topological noise removal and the manifoldness, the modest-sized model presents high-resolution details

al. 2001) converting the model from nonmanifold to manifold or a set of operators based on a non-manifold data structure has to be applied (Hubeli and Gross 2001; Popovic and Hoppe 1997; Ying and Zorin 2001); due to the nature of nonmanifold data structures, these operators need to be designed and used with great care and thus introduce additional complexity to the chosen mesh modeling system.

In this paper, a rather different approach is proposed for high-quality out-of-core mesh simplification (OoCS). In addition to providing comparable time and space complexity with other OoCS methods, the proposed approach possesses two distinct features: (1) it is manifold property guaranteed for the simplified meshes with toleration of nonmanifold mesh data input and (2) it explicitly controls topological type and thus is attractive when taking downstream applications into consideration.

The paper is organized as follows. The related work is presented in Sect. 2. Our new out-of-core large mesh simplification algorithm is presented in Sect. 3, and a novel implementation of this algorithm is presented in Sect. 4. A detailed discussion, including time and space complexity, approximation error analysis, and key features, is presented in Sect. 5. Finally, concluding remarks are presented in Sect. 6.

Fig. 2. Manifold mesh processing for a dressed David: the tools used, including cut and paste, smoothing, editing, parameterization, texture mapping, and collision detection, are all working on the manifold mesh generated by our proposed algorithm

2 Related work

A mesh simplification operation is designed to reduce the model complexity with faithful approximations. Over the last decade, in-core mesh simplification has been well studied (Cignoni et al. 1998a; Garland 1999a). Any in-core method reads the whole original mesh into the main memory with a certain data structure from which the complete topological information can be inferred. Given a mesh model with a very large size, the in-core method confronts the limited core memory as the bottleneck.

Out-of-core methods utilizing external memory offer possible solutions to the large mesh simplification task. Hierarchical methods can be adapted to partition the mesh into blocks stored in external memory and simplify the model by fitting in turn each block of the model either into the main memory (Bernardini et al. 1999; Cignoni et al. 2003) or in a view-dependent manner (El-Sana and Chiang 2000; Hoppe 1998). These hierarchical methods can support levels-of-detail processing and also preserve manifoldness if the input is a manifold.

Another class of out-of-core simplification is based on the vertex clustering method (Rossignac and Borrel 1993). Compared with the hierarchical methods, the clustering algorithm is easy to implement in the external memory and can achieve high time efficiency. Lindstrom (2000) first introduced a uniform vertex-clustering-based OoCS method and further improved it using constant main memory independent of both the input and output mesh complexity (Lindstrom and Silva 2001). Observing that the uniform clustering algorithm produced models that are not optimal in areas of low curvature variation, Shaffer and Garland (2001) proposed an adaptive vertex clustering technique based on BSP trees. Most recent adaptive techniques (Fei et al. 2002; Garland and Shaffer 2002) can produce high-quality simplified models to fit in-core. While vertex clustering schemes can offer great time and space efficiency, the output meshes often tend to be nonmanifold objects that are not suitable for a great diversity of existing mesh-processing tools that take manifold meshes as input. If a nonmanifold-to-manifold converting algorithm is used (Gueziec et al. 2001), due to its difficult implementation, which requires significant effort to enumerate all possible topological cases, the complexity of the whole mesh-processing system is increased dramatically.

A radically different approach to out-of-core mesh simplification is proposed in this paper. The proposed algorithm is based on the idea of downsampling Hermite data and contouring the zero set of an in-core signed distance field. Experimental results show that the proposed algorithm can produce high-quality simplified meshes with time and space efficiency.

3 Manifold-guaranteed simplification algorithm

The proposed algorithm is based on a simple and natural observation: a very large model with high resolution is usually greatly oversampled. By implicitly encoding the topological information, the proposed simplification algorithm downsamples the original model into an appropriate size suitable for processing on a client machine.

It is important to note that the given mesh model M is a set of structured sample points S, i.e., $M = (S, T)$, where T encodes the model's structure information serving as the topological realization of M, and $S \in \boldsymbol{R}^3$ is the set of point positions used to determine the geometric realization of M.

For a huge model, the data structure from which the complete topological information can be inferred cannot be built in the main memory, which is why in-core methods are difficult to adapt to perform out-of-core simplification. For an efficient out-of-core performance, two key constraints should be assured:

1. No topological information should be explicitly maintained in the main memory, and the in-core data structure should be independent of the input model size.
2. To minimize random access to data stored in the external memory and to minimize the needed capacity of external memory, minimum (local) topological information should be kept in the external memory.

The vertex clustering solutions (Lindstrom 2000; Lindstrom and Silva 2001; Shaffer and Garland 2001) receive high time and space efficiency by easily satisfying the above constraints: (1) the model space partition with respect to vertex clustering can be carried out efficiently, e.g., by constructing a hash table (Lindstrom 2000) or a BSP tree (Shaffer and Garland 2001), and the mesh topological structure neednot be kept explicitly in the core memory; (2) by operating on a triangle soup (Lindstrom 2000), minimum local topological information is kept in the external memory by means of representing each triangle as a triplet of vertex coordinates.

With a view toward satisfying the above two constraints, a new out-of-core algorithm is proposed for downsampling a very large model to a manifold-guaranteed simplified model. Note that a mesh M is a set of structured sample points S with T. For oversampling data S, as with the vertex clustering solutions, we regularly downsample the point data in the model space by clustering to satisfy constraint 1. The key difference between the proposed solution and the vertex clustering ones is how we represent the minimum local topological information in the external memory to satisfy constraint 2.

Due to the slow speed of the I/O performance, obviously only a small portion of topological information T of M should be obtained from the external memory via random access. Lindstrom (2000) used a triangle soup representation to minimize random access to the external memory. Observing that each triangle is represented as a triplet of vertex positions in an ordered (clockwise or counterclockwise) sequence, from a geometric point of view it actually offers the normal and orientation information of model M. Then we propose to represent the original model $M = (S, T)$ in a more compact form than the triangle soup, using a set of Hermite data, i.e., a set of sample points with their normals,

$$H = (S', N) = \{(x_i, y_i, z_i, nx_i, ny_i, nz_i) | i \in \#S'\},$$

where S' is an adequate sampling of M. In this representation, the structure information T in M is replaced by the oriented normal information N in H. A major part of the information in T is lost in H; this loss is, however, reasonable since for an efficient out-of-core simplification only a small portion of the complete information in T can be used and others must be abandoned. The key advantages of our new model representation are the following:

1. This Hermite data model is rather easily downsampled by model space partition, as well as by any scheme used in the vertex clustering methods.
2. The complete structural information of any downsampled Hermite dataset H' can be explicitly rebuilt either by (a) first building a signed distance field F from H' (Hoppe et al. 1992) and (b) secondly from F extracting an optimized mesh M' interpolating H' (Liu and Yuen 2003). The output simplified mesh M' is strictly manifold-guaranteed and is optimized in areas of low curvature variation with controllable face number and controllable triangle shape.

Since the isosurface and its polygonization are guaranteed to be manifold, we can further take the model topological type, characterized by its *genus*, into consideration. If the genus-reducing simplification is performed directly on the existing meshes, additional attention has to be paid to avoid a nonmanifold structure. Recently, Wood et al. (2002) demonstrated that controlled genus simplification can be achieved efficiently and robustly by volumetric modification rather than by mesh surgery. By taking full advantage of the signed distance field, our proposed approach can efficiently incorporate the function of volumetric modification to achieve out-of-core simplification of large meshes with controlled topological type.
Like the algorithmic structure in Garland and Shaffer (2002), our proposed algorithm can be regarded as a two-phase approach outlined as follows (Fig. 3):
Phase I. Out-of-core Hermite data downsampling.

1. Vertex binary data generation and model bounding box parameter calculation.
2. Model space partition based on bounding box parameters and the specified RAM capacity.
3. Dual Hermite data generation and downsampling by random access to vertex binary data.

Phase II. In-core optimized mesh generation and simplification.

4. Fast signed distance field generation based on prior knowledge from the input mesh data.
5. (Optional) Interactive volumetric modification to achieve controlled model topological type.
6. Simplified mesh generation by optimized isosurface polygonization.

By adopting the scheme in Garland and Shaffer (2002) to pass quadric error information between the two phases, the proposed algorithm can produce high-quality approximation. Given the two distinct features of manifoldness and controlled topological type, further in-core simplification can produce even better quality approximations than those from other OoCS methods, in which many triangles may be wasted by suffering from topological noises and nonmanifold structure. A novel, detailed implementation of the proposed algorithm is presented in the next section, followed by a detailed comparison with other methods in Sect. 5.

4 Algorithm implementation

The proposed algorithm takes the huge mesh data with the widely used indexed mesh representation as input, even without model bounding box parameters: the mesh data are formatted by a vertex position list followed by a face list consisting of triplets of vertex indices. The input mesh data are stored in the external memory and accessed via standard disk I/O managed by the operating system.
It is worth noting that in addition to indexed mesh representation, another representation for mesh models exists. Lindstrom (2000) proposed to use a compressed triangle soup representation by which random access to the external memory is avoided and the simplification speed thus potentially increased by a factor of more than ten.

4.1 Vertex binary data generation

The proposed algorithm performs a single pass over the original input mesh data. When the vertex position list is read in main memory line by line, vertex binary data with a fixed length record for each vertex are synchronously generated on the fly.

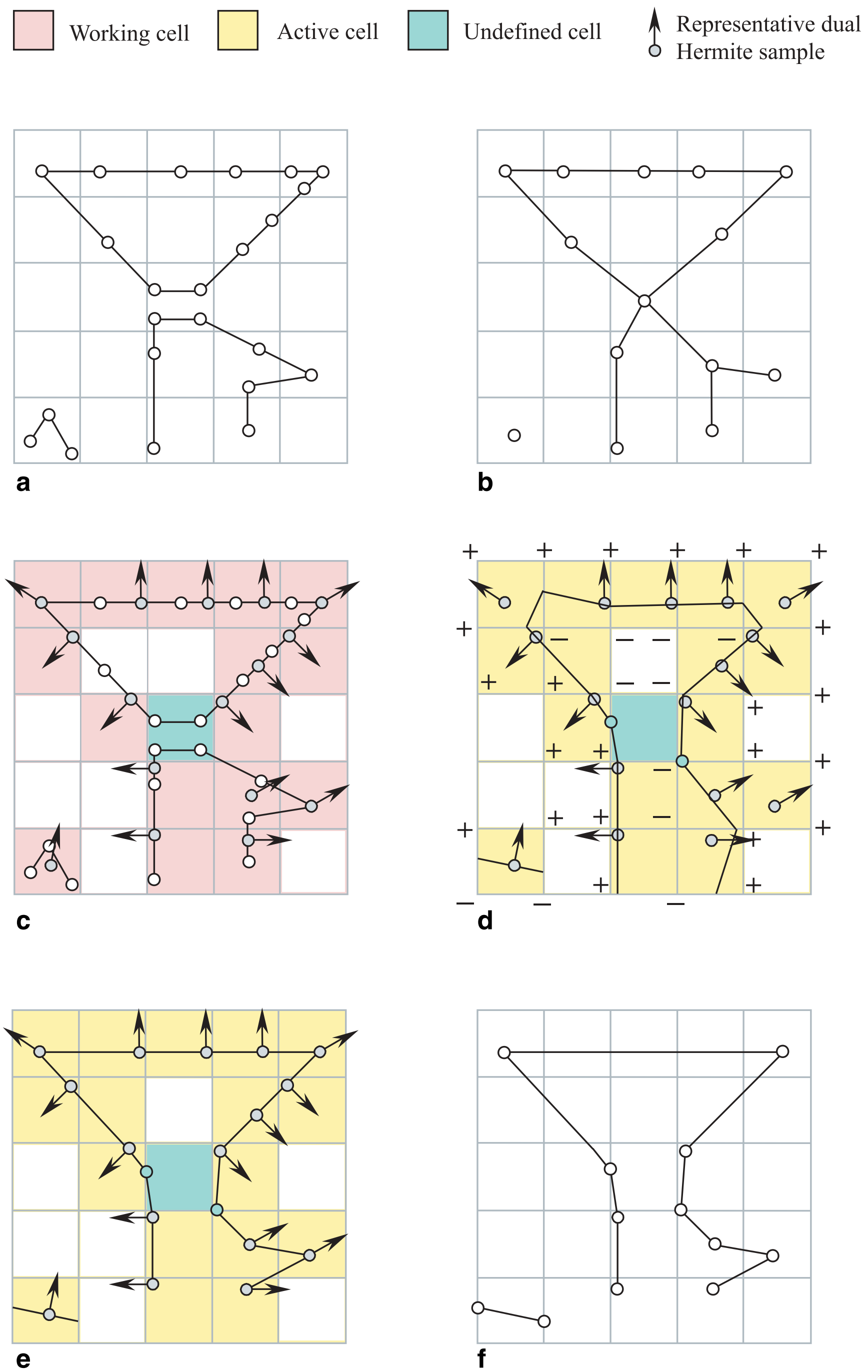

Fig. 3. The proposed manifold-guaranteed out-of-core simplification. **a** The input polygonal curve data; **b** the non-manifold structure generation by vertex clustering; **c** dual Hermite data generation and downsampling; **d** Curve generation by marching squares; **e** optimal polygonal curve generation by interpolating Hermite data; **f** further polygonal curve optimization

By generating such vertex binary data in external memory, the operation to look up the vertex coordinates corresponding to the indices in a face record can be efficiently performed by repositioning the file pointer based on the byte offset. Our current implementation on the Visual C++ platform utilizes the class *CFile* in the Microsoft Foundation Class Library to handle normal file I/O operations.

4.2 Model space partition

The model bounding box parameters can be recovered immediately after vertex position list scanning. These bounding parameters are used in uniformly decomposing the model space into a user-specified number of rectilinear grid cells that are hashed into a table stored in core memory. It is in general not known in advance how much RAM is available in the client machine, and thus any promising algorithm should be capable of (at least roughly) estimating, based on the specified RAM capacity, to what extent the model space needs to be decomposed.

Note that the proposed algorithm is designed to simplify the model to a manageable size suitable for real-time processing on the client machine; that is, the whole simplified model should be read in the main memory with any compact data structure from which the complete topological information can be inferred. Consider the medium-sized directed-edge data structure (Campagna et al. 1998): given a triangle mesh with n vertices and m triangles, a total of $16n + 36m \approx 44m$ bytes is required. Therefore, around 4–6 million triangles can be processed on a system with 256 MB RAM.

The proposed algorithm simplifies the mesh data by downsampling the Hermite data followed by zero set contouring to guarantee the manifold property. Note that statistically there are $\Theta(N^{2/3})$ active cells in an N-cell volume dataset (Chiang et al. 1998), and each active cell may contain a representative Hermite sample. Therefore, to output a modest-size simplified mesh model, say, 3 million triangles interpolating the downsampled Hermite data, the model space should be partitioned into about 1840 million cells.

The majority of the core memory is used in the construction of the cluster cells. Since relatively few cells will be occupied by representative samples, a hash table is adopted in our current implementation for the model space partition.

4.3 Dual Hermite data generation and downsampling

The algorithm performs a single pass over the input mesh data by scanning the vertex position list and the face list in tandem. When each face element is scanned, by looking up the corresponding vertex coordinates in the vertex binary data the barycenter $\boldsymbol{p}$ and the normal $\boldsymbol{n}$ of the face are calculated and serve as the dual Hermite sample $h = (\boldsymbol{p}, \boldsymbol{n})$ of the face element; this is equivalent to applying a barycenter dual mesh operator to the input mesh model (Taubin 2001) and taking the samples as the vertices of the barycenter dual mesh with their normals.

Each face element contributes to a dual Hermite sample h. For each sample h, a hash key from the cell that s falls in is constructed, and by looking up the hash table, the representative sample h' in the cell is updated.

The huge input mesh model may be represented with highly varying triangle size, and thus there may exist large triangles each of which spans a number of cells. In this case, we perform an adaptive 1–4 split to subdivide the large triangle, as illustrated in Fig. 4. The recursive split is terminated if the largest edge length of the current triangle is smaller than the one and a half cell sizes. This uniformly dual Hermite sampling scheme can offer better approximation of the downstream simplified meshes.

Note that each Hermite sample $h = (\boldsymbol{p}, \boldsymbol{n})$ corresponds to a tangent plane on the mesh model, i.e., a face in the original mesh. By dual sampling, the mesh (piecewise linear surface) can be considered as a local combination of the tangent planes associated with the Hermite data. For sample point clustering in a cell using a representative sample, it is equivalent to approximating a set of tangent planes using a single tangent plane. We adopt the quadratic error metric (Garland 1999b) as a measure of approximation error to find an optimal representative Hermite sample in each working cell.[1]

Given a working cell, every Hermite sample $h = (\boldsymbol{p}, \boldsymbol{n})$ falling in it contributes to a quadric $\boldsymbol{Q} = (\boldsymbol{n}\boldsymbol{n}^T, -\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{p}, (\boldsymbol{n}^T\boldsymbol{p})^2)$. Given a triangle $t =$

[1] We use the term "working cell" to identify a cell that contains Hermite samples, and the term "active cell" is preserved for isosurface contouring.

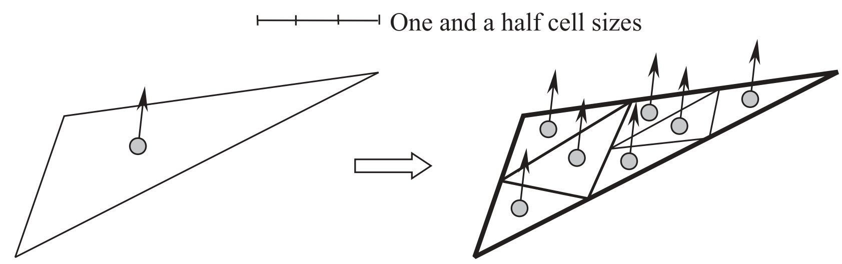

Fig. 4. Adaptive 1–4 split of large triangles for uniform Hermite data generation

(p_1, p_2, p_3), the unit normal vector of t is $\boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v}/(2\Delta)$, where $\boldsymbol{u} = p_2 - p_1$, $\boldsymbol{v} = p_3 - p_2$, $\Delta = \|\boldsymbol{u} \times \boldsymbol{v}\|/2$ is the area of t. Rather than using the unit normal vector $\boldsymbol{n}$ in $\boldsymbol{Q}$, we prefer to use an area-weighted normal $2\Delta \cdot \boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v}$ in a weighted quadric $\boldsymbol{wQ} = 4\Delta^2\boldsymbol{Q}$, which leads to better error measures (by square area weighting) and less computational complexity (without explicitly calculating Δ). By summing the quadrics of all the Hermite samples inside a working cell, $\sum \boldsymbol{wQ} = (\boldsymbol{A}, \boldsymbol{b}, c)$, the position $\boldsymbol{p}'$ of the representative Hermite sample h' is readily obtained by solving $\boldsymbol{Ap}' = -\boldsymbol{b}$. For robust inversion of $\boldsymbol{A}$, we adopt the approach in Lindstrom (2000) by performing a singular value decomposition of $\boldsymbol{A}$.

The normal $\boldsymbol{n}'$ of the representative sample h' determines the orientation of the tangent plane associated with h'. For each working cell, we determine the $\boldsymbol{n}'$ by averaging the area-weighted normals of all the Hermite samples falling in the cell: $\boldsymbol{n}' = \sum_i 2\Delta_i \boldsymbol{n}_i/\#n = \sum_i \boldsymbol{u}_i \times \boldsymbol{v}_i/\#n$, where $\#n$ is the number of samples. After accumulating all the Hermite samples, the normal $\boldsymbol{n}'$ is finally normalized. Experiments show that this simple scheme works well, while attention needs to be paid to some special cases presented below.

The (nonnormalized) averaged normal in a working cell may be close to null vector, e.g., in CAD models thin shell structures contributing to opposite normals can be contained in a cell. For such working cells, we identify them by setting a small threshold for the magnitude of the nonnormalized averaged normal and mark them by undefined cells. The manifold structure inside undefined cells can be recovered in a subsequently zero set contouring process (Figs. 3c–f and 7); we refer to this function as topology auto-repair.

4.4 Fast in-core signed distance field generation

Given a set of downsampled Hermite data H' linked with the grid, we first build a discrete signed distance field F using the grid and then generate an in-core optimized mesh from F that interpolates the data H'. In traditional volumetric methods, usually a signed distance field is built first and then, for surface modeling purposes, the isosurface is extracted by tracking active cells. Both steps are computationally expensive: the distance field needs to be determined by evaluating every cell in the grid and active cell selection needs to use time-consuming search operators, even in an optimal isosurface extraction algorithm (Cignoni et al. 1997).

In the proposed algorithm, a particular method that is fast and simple is carried out based on a priori knowledge from the input mesh data: the active cells are tracked first and then the local discrete signed distance fields around the active cells are determined. More specifically, during face list scanning, every face element is read into main memory in turn. It is easy to determine the active cells associated with each face: they are the cells interacting with that face. In this way, the active cell selection can be achieved with linear time complexity accompanied by face list scanning. Due to uniform downsampling, the sign distance field defined on the active cells can be determined in constant time by looking up the representative samples in the neighboring active cells using the signed distance function proposed in Hoppe et al. (1992), which assigns to each node $\boldsymbol{v}$ of the grid a value $f(\boldsymbol{v})$:

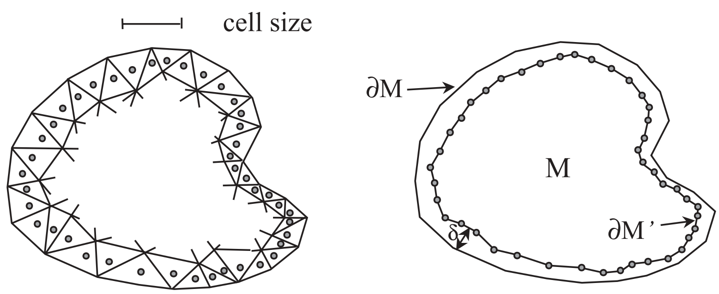

Fig. 5. The shrinkage of reconstructed boundary $\partial M'$: the offset δ of this shrinkage is less than one cell size due to uniform Hermite sampling

1. Among the representative samples in adjacent active cells of $\boldsymbol{v}$, find the sample $h = (\boldsymbol{p}, \boldsymbol{n})$ nearest to $\boldsymbol{v}$.
2. Set $f(\boldsymbol{v}) = (\boldsymbol{v} - \boldsymbol{p})^T \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ is normalized.

It is worth noting that identifying active cells by input mesh data also helps preserve boundaries; we will return to this point in the next section.

4.5 Simplified mesh generation by optimized isosurface polygonization

Given the active cells, the Marching Cubes algorithm (Lorensen and Cline 1987) is adopted to polygonalize the isosurface, and afterwards an optimization algorithm (Liu and Yuen 2003) is adopted to output an optimized mesh that interpolates the downsampled Hermite data. For Marching Cubes, the active cell polygonization can be efficiently achieved by looking up a case table that enumerates all possible topological states of a cell and by solving the ambiguity problem with the complementary cases (Montani et al. 1994; Van Gelder and Wilhelms 1994).

Given a set of downsampled Hermite samples (Fig. 3c) and an initial mesh from Marching Cubes (Fig. 3d), the algorithm proposed in Liu and Yuen (2003) constructs an optimized mesh that interpolates the Hermite samples (Fig. 3e) and can further optimize the mesh in areas of low curvature variation with controllable face number and controllable triangle shape to a degree desired by the user (Fig. 3f). With a linear time preprocessing accomplished in the Marching Cubes process, Liu and Yuen's algorithm is designed in optimal $O(n \log n)$ complexity, where n is the amount of downsampled Hermite data.

The necessity of interpolating downsampled Hermite data is due to the following reasons. During the Hermite data downsampling process, each representative Hermite sample is associated with a quadric as the error metric. By interpolating Hermite samples, the approximation error information accumulated in the out-of-core downsampling phase can be passed to the next in-core optimization phase to generate high-quality simplified meshes.

Interpolating Hermite samples also helps preserve sharp features and the boundary. The idea of preserving sharp features by interpolating Hermite samples has been proposed and demonstrated in recent work (Ju et al. 2002; Kobbelt et al. 2001; Liu and Yuen 2003). As illustrated in Fig. 5, since the Hermite samples can only be generated near the boundary ∂M but not on the boundary, the reconstructed boundary $\partial M'$ is shrunk from ∂M; by uniform Hermite sampling and active cell identification using input mesh data, the offset δ of this shrinkage is less than one cell size. Two 3D examples of boundary and sharp feature preservation are illustrated in Fig. 6.

4.6 Explicit topological type control by volumetric modification

Since the isosurface and its polygonization are guaranteed to be manifold, the model topological type can be unambiguously determined by evaluating its

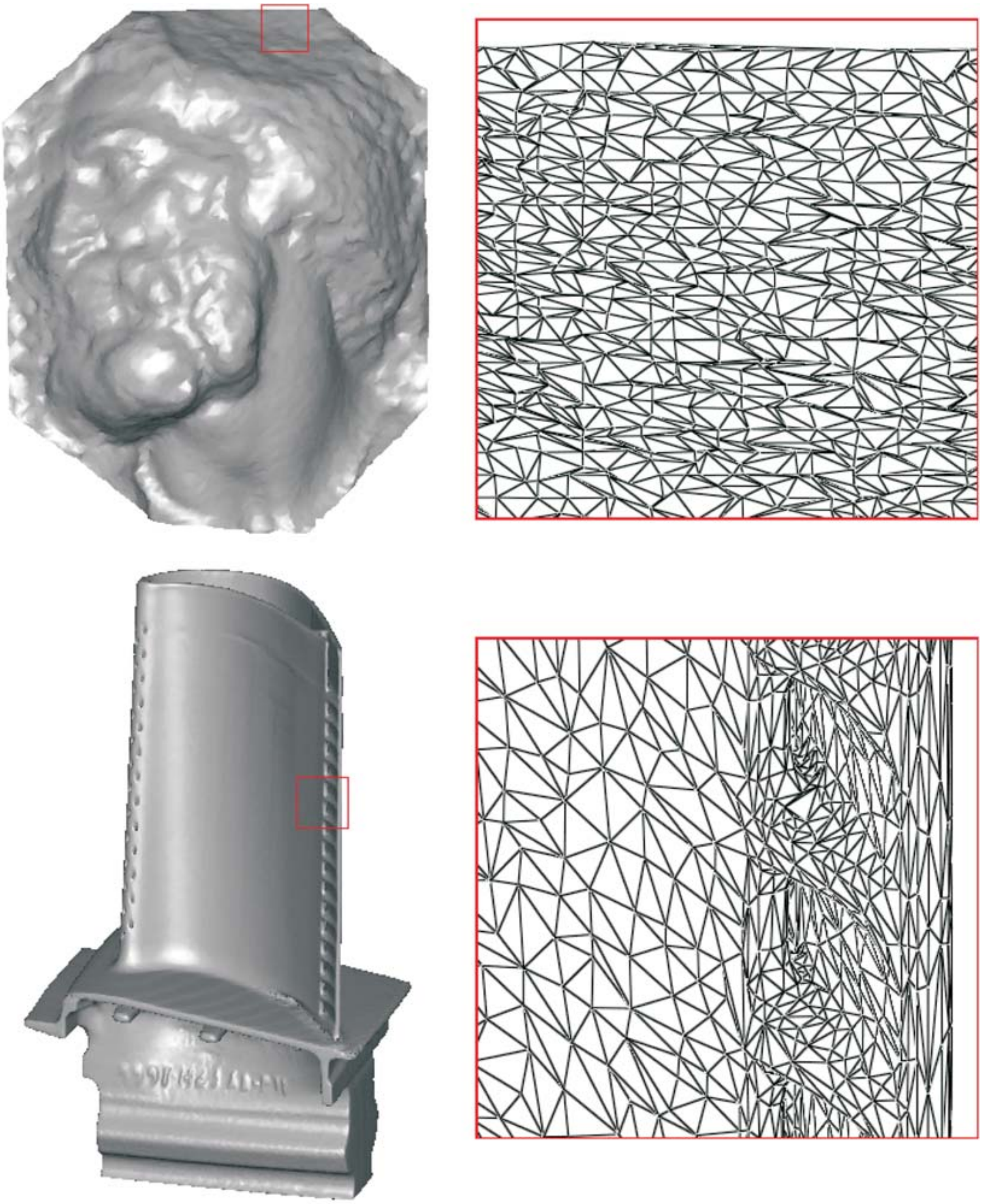

Fig. 6. Boundary and sharp feature preservation by interpolating the Hermite samples and identifying the active cells using input mesh data

genus.[2] A model with genus n is homeomorphic to a sphere with n handles, which can be identified by cycles in a Reeb graph (Fomenko and Kunii 1997): given a characteristic function f defined on the manifold M, the Reeb graph is the quotient

[2] For manifold with boundary, we analyze its topological type by capping all its connected boundaries.

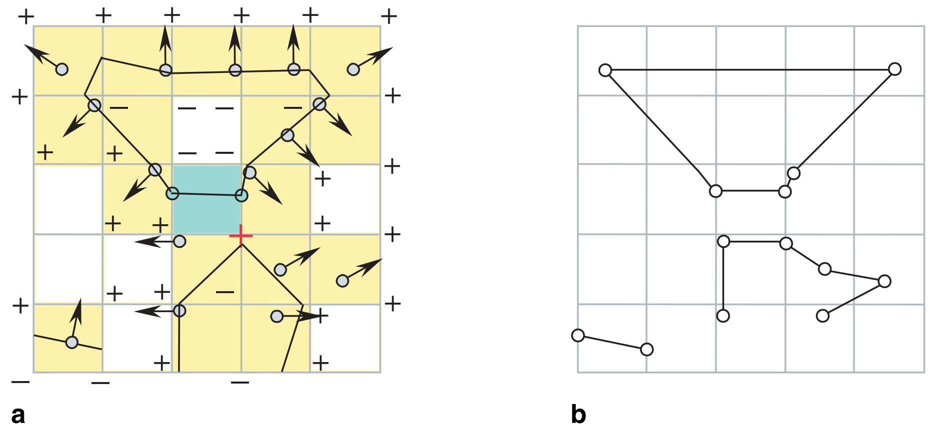

Fig. 7. Topological type control by grid value modification. **a** Curve generation by marching squares with one grid node value modified, in comparison with Fig. 3d; **b** optimized polygonal curve generation with modified topological type

space of M with the equivalence relation $\sim$ defined by $x_1 \sim x_2$ iff $f(x_1) = f(x_2)$ and x_1, x_2 are in the same connected component of $f^{-1}(f(x_1))$. In the case of volumetric representation, the Reeb graph induced from a height function can be constructed by making an axis-aligned sweep through the volume (Wood et al. 2002). After identification of all topological handles with the Reeb graph, rather than modifying the existing meshes with special attention paid to the nonmanifold structures that may have occurred, the change of topological type can be efficiently achieved by modifying the grid node value directly (Fig. 7).

Given a model with n handles, different handles may have different geometric significance that can be classified by defining a measure of handle size as the length of the minimal essential loop of the handle on the model surface (Wood et al. 2002). In our study, we consider two classes of handles. We consider small handles, e.g, less than the cell size, as topological noises that are eliminated by Hermite data clustering. The remaining large handles we regard as the topological features inherent in the model. The choice of which feature handles to remove is subjective, and user intervention is inevitable. We thus implement our algorithm in an interactive way to locate and remove any feature handles; this function offers an explicit control of model topological type and works as an option to users. See Fig. 8 for an example.

5 Discussion and comparison with other methods

5.1 Time and space complexity

The proposed algorithm works in two phases. In the first phase of out-of-core Hermite sample clustering, most main memory is used to construct the fine grid, which is independent of the size of the input mesh data. The grid is implemented as a hash table. A hash key is held for each working cell that points to a representative sample associated with a quadric. By adopting the same clustering scheme, this phase must run with exactly the same time and space complexity as the vertex clustering algorithms proposed in Lindstrom (2000), Lindstrom and Silva (2001), and Shaffer and Garland (2001).

In the second phase of in-core optimized mesh generation, our algorithm needs additional memory for zero set polygonization and optimization. Based on a special design utilizing a priori knowledge of the input data, the time complexities of isosurface polygonization and optimization are $O(n)$ and $O(n \log n)$, respectively, where n is the number of downsampled Hermite samples. In terms of adaptive optimization, the algorithm in Shaffer and Garland (2001) also needs additional memory for BSP tree construction, which is dependent on the size of the simplified output model.

Table 1. The performance of the proposed algorithm running on an off-the-shelf PC with a Pentium III 500-MHz processor, 512 MB RAM, and 12 GB hard disk on Microsoft Windows 2000 operating system. The algorithm is implemented on the Visual C++ platform

Model	Input mesh data		Output mesh data		RAM used (MB)	Total time* (h:m:s)
Happy Buddha	Vertex number: Triangle number:	543 644 1 085 634	Vertex number: Triangle number:	74 219 148 101	142	6:51 (1:55‡)
Blade	Vertex number: Triangle number:	882 954 1 765 388	Vertex number: Triangle number:	94 055 188 338	158	11:32
St. Matthew_Face	Vertex number: Triangle number:	3 382 866 6 755 412	Vertex number: Triangle number:	14 074 27 546	146	26:47
David	Vertex number: Triangle number:	28 184 526 56 230 343	Vertex number: Triangle number:	56 523 111 431	180	1:55:13

* The running time includes all disk I/O operation time

‡ If we read the vertex position list into the RAM, instead of generating vertex binary data in external memory, the running time is reduced to 1:55, most of which contributes to Marching Cubes and mesh optimization using Liu and Yuen's algorithm (2003)

Table 2. Approximation error test on the Happy Buddha model using the Metro v.3.1 (Cignoni et al. 1998). Our algorithm is performed with a $184 \times 447 \times 184$ grid to generate an optimized mesh with 131 500 triangles, and then the mesh is further simplified to 60 000 and 10 000 triangles

	Triangle number of output meshes	RMS error (%)	
		Qslim 2.0	Our model
Happy Buddha model with 1 085 634 triangles	131 500	0.034	0.041
	60 000	0.065	0.065
	10 000	0.404	0.339

Table 1 summarizes our experimental results. All tests are performed on an off-the-shelf PC with a Pentium III processor running at 500 MHz with 512 MB RAM and a 12 GB hard disk on Microsoft Windows 2000. The proposed algorithm is fully implemented on the Visual C++ platform.

5.2 Approximation error

The proposed algorithm uses Hermite data H dual-sampled from the input mesh to represent the model. The simplification is carried out by downsampling the data H into a simplified version H' using a uniformly partitioned grid. Noting that polygonalization is a sampling process, we can evaluate the approximation error as the two-sided Hausdorff distance between H and H'. Since the model space is uniformly partitioned and the data uniformly dual sampled, the approximation error by sample clustering is strictly bounded by the diagonal length of the cell. By passing the approximation error information with the quadric between two phases, the proposed algorithm can produce highly detailed simplified meshes with comparable quality to the in-core quadric error-metric-based algorithm (Garland and Heckbert 1997). As a comparison with Qslim (Garland 1999b), the numerical data of the RMS approximation error is presented in Table 2 using the Metro tool (Cignoni et al. 1998b), which adopts a more rigorous error metric based on surface sampling and point-to-surface distance computation. Given the properties of manifoldness and controlled topological type, our proposed algorithm may generate even better quality simplified meshes than the previously reported OoCS algorithms (see the last row of Table 2; their visual difference is illustrated in Fig. 9). As a summary, the original OoCS algorithm (Lindstrom 2000) runs fastest with a relatively large approximation error. The algorithm (Garland and Shaffer 2002) offers an excellent balance of efficiency and quality. Our proposed algorithm output simplified meshes with quality comparable to that of Garland and Shaffer's algorithm (2002) and with distinct features of manifoldness and controlled topological type.

5.3 Key features

To summarize, the proposed algorithm possesses the following features:

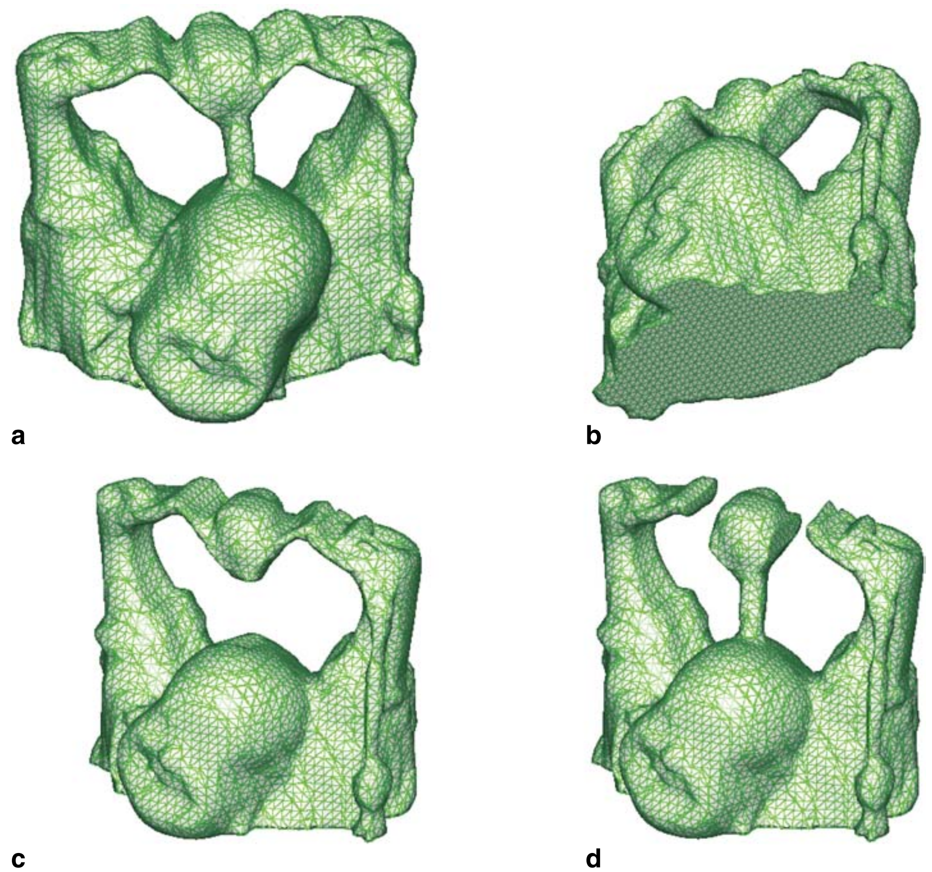

Fig. 8. Volumetric modification for explicit topological type control; the modified signed distance field is visualized indirectly by on-the-fly isosurface contouring. **a** Genus-2 modeling; **b** another view of the model in **a**; **c** Genus-1 modeling; **d** Genus-0 modeling

1. Manifoldness. Due to the topology consistency by applying a modified lookup table (Montani et al. 1994), the output mesh is guaranteed to be strictly a manifold. Downstream mesh processing operations, such as cut and paste, smoothing, editing, parameterization, texture mapping, and collision detection, will benefit from this property (Fig. 2).
2. Toleration of nonmanifold mesh input. Since we represent the mesh model concisely as Hermite sample data, nonmanifold input meshes (i.e., with dangling faces) are also tolerated by the proposed algorithm.
3. Topological noise removal. The proposed algorithm is volume based. By clustering all the samples falling into the same cell, the tiny topological handles whose size is less than the cell size are automatically removed.
4. Explicitly topological type control. With the help of volumetric representation, explicitly topological type control of feature handles can be simply and robustly achieved by volumetric modification (Fig. 8).
5. Sharp feature and boundary preservation. As demonstrated by recent work (Ju et al. 2002; Kobbelt et al. 2001; Liu and Yuen 2003), the Hermite data, serving as the bridge between volumetric representation and surface representation, can accurately identify and process sharp features on the geometric model. By uniformly sampling the Hermite data and identifying the active cells using input mesh data, the proposed algorithm

Fig. 9. The Happy Buddha model simplification. The left model is simplified by an in-core QSlim algorithm (Garland 1999b) applied to the original Stanford mesh data; the simplified model is with 10 000 triangles of which many are wasted by the nonmanifold structure. The right model is simplified by the proposed algorithm; the simplified model is with 10 000 triangles and is manifold

also preserves the boundary on the model with a controlled shrinkage (Fig. 6).

6 Conclusion

In this paper, a simple and efficient algorithm for out-of-core mesh simplification with controlled topological type is proposed. The algorithm adopts a Hermite data format to represent the geometric model, which serves as the basis for integrating various surface modeling techniques, i.e., dual Hermite sampling, samples clustering, isosurface generation, and mesh optimization, into a single framework with time and space efficiency and with a set of distinct features. The experimental results have demonstrated that very large meshes can be simplified with high quality on a low-cost off-the-shelf PC.

Acknowledgements. The authors would like to thank Stanford Computer Graphics Lab and GVU Geometry Group for making their models available to us. The authors also gratefully acknowledge the insightful comments of the three anonymous reviewers.

References

1. Bernardini F, Mittleman J, Rushmeier H, Taubin G (1999) Case study: scanning Michelangelo's Florentine Pieta. In: Proceedings of ACM SIGGRAPH '99, Los Angeles, August 1999, course notes 4.
2. Bernardini F, Martin I, Mittleman J, Rushmeier H, Taubin G (2002) Building a digital model of Michelangelo's Florentine Pieta. IEEE Comput Graph Appl 22(1):59–67
3. Campagna S, Kobbelt L, Seidel HP (1998) Directed edges: a scalable representation for triangle meshes. ACM J Graph Tools 3(4):1–12
4. Chiang YJ, Silva CT, Schroeder WJ (1998) Interactive out-of-core isosurface extraction. In: Proceedings of Visualization '98, Research Triangle Park, NC, October 1998, IEEE, pp 167–174
5. Cignoni P, Marino P, Montani C, Puppo E, Scopigno R (1997) Speeding up isosurface extraction using interval trees. IEEE Trans Visual Comput Graph 3(2):158–170
6. Cignoni P, Montani C, Scopigno R (1998a) A comparison of mesh simplification algorithms. Comput Graph 22(1):37–54
7. Cignoni P, Rocchini C, Scopigno R (1998b) Metro: measuring error on simplified surfaces. Comput Graph Forum 17(2):167–174
8. Cignoni P, Montani C, Rocchini C, Scopigno R (2003) External memory management and simplification of huge meshes. IEEE Trans Visualizat Comput Graph 9(4):525–537
9. El-Sana J, Chiang YJ (2000) External memory view-dependent simplification. In: Proceedings of Eurographics '00, Switzerland, Aug 2000. Comput Graph Forum 19(3):139–150
10. Fei G, Cai K, Guo B, Wu E (2002) An adaptive sampling scheme for out-of-core simplification. Comput Graph Forum 21(2):111–119
11. Fomenko A, Kunii T (1997) Topological modeling for visualization. Springer, Berlin Heidelberg New York
12. Garland M, Heckbert P (1997) Surface simplification using quadratic error metrics. In: Proceedings of SIGGRAPH '97, Las Angeles, Aug 1997, pp 209–16
13. Garland M (1999a) Multiresolution modeling: survey and future opportunities. In: Proceedings of Eurographics '99, Italy, Sep 1999, State of the art report, pp 111–131
14. Garland M (1999b) Quadric-based polygonal surface simplification. Ph.D. thesis, Carnegie Mellon University, Department of Computer Science, Pittsburgh
15. Garland M, Shaffer E (2002) A multiphase approach to efficient surface simplification. In: Proceedings Visualization '02, Boston, Oct 2002. IEEE Press, New York, pp 117–124
16. Gueziec A, Taubin G, Lazarus F, Hom B (2001) Cutting and stitching: converting sets of polygons to manifold surfaces. IEEE Trans Visualizat Comput Graph 7(2):136–151
17. Hoppe H (1998) Smooth view-dependent level-of-detail control and its application to terrain rendering. In: Proceedings of Visualization '98, North Carolina, Oct 1998. IEEE Press, New York, pp 35–42
18. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. In: Proceedings of SIGGRAPH '92, Chicago, July 1992, pp 71–78
19. Hubeli A, Gross M (2001) Multiresolution methods for nonmanifold models. IEEE Trans Visualizat Comput Graph 7(3):207–221
20. Ju T, Losasso F, Schaefer S, Warren J (2002) Dual contouring of Hermite data. In: Proceedings of SIGGRAPH '02, San Antonio, July 2002, pp 339–46
21. Kobbelt L, Botsch M, Schwanecke U, Seidel H (2001) Feature sensitive surface extraction from volume data. In: Proceedings of SIGGRAPH '01, New Orleans, Aug 2001, pp 57–66
22. Levoy M, Rusinkiewcz S, Ginzton M, Ginsberg J, Pulli K, Koller D, Anderson S, Shade J, Curless B, Pereira L, Davis J, Fulk D (2000) The digital Michelangelo project: 3D scanning of large statues. In: Proceedings of SIGGRAPH '00, New Orleans, July 2000, pp 131–144
23. Lindstrom P (2000) Out-of-core simplification of large polygonal models. In: Proceedings of SIGGRAPH '00, New Orleans, July 2000, pp 259–262
24. Lindstrom P, Silva C (2001) A memory insensitive technique for large model simplification. In: Proceedings of Visualization '01, location, day month 2001. IEEE Press, pp 121–126
25. Liu YJ, Yuen MMF (2003) Optimized triangle mesh reconstruction from unstructured points. Vis Comput 19(1):23–37
26. Lorensen W, Cline H (1987) Marching cubes: a high resolution 3D surface construction algorithm. In: Proceedings of SIGGRAPH '87, Anaheim California, July 1987. Comput Graph, 21(4):163–169
27. Montani C, Scateni R, Scopigno R (1994) A modified look-up table for implicit disambiguation of Marching Cubes. Vis Comput 10(6):353–355
28. Popovic J, Hoppe H (1997) Progressive simplicial complexes. In: Proceedings of SIGGRAPH '97, Los Angeles, Aug 1997, pp 217–224
29. Rocchini C, Cignoni P, Montani C, Pingi P, Scopigno R (2001) A low cost 3D scanner based on structured light. In: Proceedings of Eurographics '01, Manchester, UK, Sep 2001. Comput Graph Forum 20(3):299–308
30. Rossignac J, Borrel P (1993) Multi-resolution 3D approximation for rendering complex scenes. In: Falcidieno B, Kunii TL (eds) Geometric modeling in computer graphics. Berlin New York Springer-Verlag, pp 455–465
31. Shaffer E, Garland M (2001) Efficient adaptive simplification of massive meshes. In: Proceedings of Visualization '01, San Diego, Oct 2001. IEEE Press, New York, pp 127–134
32. Taubin G (2001) Dual mesh resampling. In: Proceedings of Pacific Graphics '01, Tokyo Japan, Oct 2001. IEEE Press, New York, pp 180–188
33. Van Gelder A, Wilhelms J (1994) Topological considerations in isosurface generation. ACM Trans Graph 13(4):337–375
34. Wood Z, Hoppe H, Desbrun M, Schroder P (2002) Isosurface topology simplification. Technical Report MSR-TR-2002-28, Microsoft Research, Redmond, WA
35. Ying L, Zorin D (2001) Nonmanifold subdivision. In: Proceedings of Visualization '01, San Diego, Oct 2001. IEEE Press, New York, pp 325–569
36. Zheng J, Zhong L (1999) Virtual recovery of excavated relics. IEEE Comput Graph Appl 19(3):6–11

MR. YONG-JIN LIU is currently a Ph.D. candidate at Hong Kong University of Science and Technology. He received his B.Eng. (1998) in mechanoelectronic engineering from Tianjin University, PRC, and his M.Phil. (1999) in mechanical engineering from Hong Kong University of Science and Technology, Hong Kong. His research interests include computer graphics, geometric modeling, reverse engineering, and design automation.

DR. MATTHEW M.F. YUEN is a professor at HKUST. He obtained his Ph.D. in mechanical engineering from Bristol University, UK, in 1977. He joined the Hong Kong University of Science and Technology (HKUST) as one of the founding members of the Department of Mechanical Engineering in 1992. He had served as the associate dean of engineering, the founding director of the CAD/CAM Facility, and the founding director of the EPACK Laboratory at HKUST. He now holds the concurrent positions of professor of mechanical engineering, and director of the Technology Transfer Center at HKUST. He is also the vice president of the HKUST R&D Corporation. Professor Yuen has published more than 150 papers in international journals and conferences. He has won Best Paper awards from both the Institution of Mechanical Engineers and the American Society of Mechanical Engineers. He was also a commonwealth fellow and a United Nations Industrial Development Organisation fellow. He is a fellow of the Institution of Mechanical Engineers.

DR. KAI TANG is currently a faculty member in the Department of Mechanical Engineering at the Hong Kong University of Science and Technology. Before joining HKUST in 2001 he worked for more than 13 years in the CAD/CAM and IT industries. His research interests focus on designing efficient and practical algorithms for solving real-world computational, geometric, and numerical problems. Dr. Tang received his Ph.D. in computer engineering from the University of Michigan in 1990, his M.Sc. in information and control engineering in 1986, also from the University of Michigan, and his B.Sc. in mechanical engineering from Nanjing Institute of Technology in China in 1982.