Graphical Models **64,** 169–182 (2002)
doi:10.1006/gmod.2002.0576

A Novel Volume Constrained Smoothing Method for Meshes

Xinguo Liu

Microsoft Research Asia, 3F Sigma Center, No. 49 Zhichun Road, Haidian District Beijing 100080, China
E-mail: i-xgliu@microsoft.com

Hujun Bao

State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China
E-mail: bao@cad.zju.edu.cn

Heung-Yeung Shum

Microsoft Research Asia, 3F Sigma Center, No. 49 Zhichun Road, Haidian District Beijing 100080, China
E-mail: hshum@microsoft.com

and

Qunsheng Peng

State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China
E-mail: peng@cad.zju.edu.cn

Received September 15, 2001; accepted May 14, 2002

In general, mesh smoothing is performed by minimizing the discrete energy function for the surface. One of the major problems in mesh smoothing is to prevent the mesh from shrinking. In this paper, we propose a novel volume constraint to address the shrinking problem in mesh smoothing. Our key observation is that the mesh can be efficiently smoothed patch by patch in a signal processing manner, and then a local volume preserving constraint can be easily imposed to the energy minimization problem associated with the small patch, called the smoothing stencil, so as to effectively avoid the mesh shrinkage. In our implementation, the smoothing stencil is the 1-ring neighboring region of an edge or a triangle. And the constrained minimization problem is solved by a 2-step approximation method for efficiency. A series of examples demonstrate that the proposed smoothing method can be applied to remove noise from a mesh or remove rough detail from an original mesh to generate a smooth model in object reconstruction and geometry modeling.

Key Words: curves and surfaces; geometry modeling; object reconstruction; mesh generation; surface fairing and smoothing.

1524-0703/02 $35.00

1. INTRODUCTION

In the computer graphics community, triangular meshes have become more and more popular for their simplicity and flexibility in modeling complex shapes. With the development of 3D scanning and surface reconstruction techniques, creating triangle meshes of high complexity is not a hard task [1–4], but how to effectively process such meshes remains a challenging problem. For example, 3D scanning systems commonly give rise to noisy meshes due to the factors related to measurement; thus, some practical smoothing techniques need development to generate fairing surfaces by removing noise or undesired rough features from triangular meshes. Hoppe *et al.* proposed a mesh optimization method [5] to optimize the reconstructed triangular mesh by removing noises, reconstructing crease features, and saving the number of vertices and triangles. Since it needs minimization of an extremely complex energy–cost function, the efficiency of the mesh optimization method is very poor. In recent years, a number of smoothing methods have been introduced for the fairing purpose and used for fair surface design [12–14]. The basic idea of surface smoothing is to minimize the surface energy or to remove the high frequency content from surfaces.

Based upon a discrete approximation of the Laplacian operator on a mesh, Taubin [6] proposed a signal processing approach to address the fairing problem for discrete meshes. In this approach the Laplacian operator is repeatedly performed over the mesh to remove the noises. Because of the linear complexity in both time and memory, this method can smooth large meshes very quickly. More detailed examination on the filter design for fairing meshes is further discussed in [7]. In the discrete fairing method for arbitrary triangular meshes [8], Kobbelt derived a similar linear smoothing operator to that of Taubin under a specified parameterization for the local mesh patch, which was then successfully applied in their multiresolution modeling scheme [9]. The Laplacian operator is also called an umbrella operator. We refer to the above smoothing method based on the Laplacian operator as the Laplacian smoothing method.

Laplacian smoothing can be thought of as forward time integration of the heat equation on an irregular mesh. The stability criterion for the forward integration requires that the time step be less than 1. In order to avoid this limitation, Desbrun *et al.* developed an implicit fairing approach to smooth meshes more efficiently and stably [10], which allows for a very large time step at the cost of solving a linear system. Designing a nonuniform relaxation procedure, Guskov *et al.* generalized basic signal processing tools such as up-sampling, down-sampling, and filters to irregular triangular meshes [11]. Since the weights in the relaxation procedure depend on the mesh geometry as well as the mesh connectivity, it is suitable for processing irregular meshes and is therefore superior to previous works. The relaxation procedure is adopted to build a set of multiresolution signal processing tools for meshes together with a multiresolution mesh hierarchy, including multiresolution filtering, editing, enhancement, and so on.

However, applying the standard Laplacian operator to mesh fairing may result in serious shrinkage, which is the major problem of many mesh smoothing methods. Taubin attenuates such shrinkage in his signal processing approach by applying both the Laplacian and the second order Laplacian operators alternatively, which is referred to as the $\lambda|\mu$ method by Desbrun *et al.* [10]. However, the resulting mesh heavily depends on the choice of two mesh-related constants λ and μ. In the implicit fairing method [10], the scale-dependent Laplacian operator and scale-based volume preservation approach are adopted to reduce the degree

of shrinkage and distortion. They also introduced a curvature flow operator in discrete differential geometry to remove small scale details and to prevent the distortion. Using anisotropic curvature evolution, Clarenz *et al.* proposed a multiscale method to smooth discretized surfaces while simultaneously preserving the geometric features such as edges and corners [15]. In 1999, Vollmer *et al.* presented an improved Laplacian smoothing method [16] to attenuate the shrinkage, where the basic idea is to move the vertices of the smoothed mesh back toward their previous locations by some distance. The constrained mesh fairing method proposed in [17] prevents the mesh from further shrinkage by keeping the centroid of all triangles unchanged during the smoothing procedure.

In this paper, we propose a novel volume-constrained smoothing method for triangular meshes, which preserves exactly the mesh volume during the smoothing process, so as to prevent the mesh from shrinking. In this method, the local volume is examined and preserved. The advantage of local volume preserving over other volume preserving methods is that it tends to preserve the shape of the surface, and it can process open meshes for which the global volume is not well defined. In the next sections, we will give a brief introduction to the Laplacian smoothing method after introducing some concepts and notations related to mesh smoothing. In Section 3 we will describe our constrained smoothing operator and propose an efficient 2-step approach to solve the constrained energy minimization problem. At last, in Section 4, we give multiple examples to show the efficiency of our novel smoothing method and finally draw conclusions and describe some future work.

2. LAPLACIAN SMOOTHING

2.1. Mesh and Notations

Throughout the rest of this paper, we consider the smoothing problem on a triangular mesh, $\mathcal{M}$, which is usually denoted as a *triple* $\mathcal{M} = (\mathcal{V}, \mathcal{K}, \mathcal{P})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the *vertex set* and $\mathcal{K}$ is an *abstract simplicial complex*, i.e., a set of subsets of $\mathcal{V}$, which contains all the *adjacency* information or the *connectivity* information of the mesh. The subsets in $\mathcal{K}$ are called *simplices* and come in three types, *vertices* $v = \{i\} \in \mathcal{K}$, *edges* $e = \{i, j\} \in \mathcal{K}$ and *faces* $t = \{i, j, k\} \in \mathcal{K}$, so that any nonempty subset of a simplex of K is again a simplex of $\mathbf{K}$. $\mathcal{P}$ is a set of 3D points $\mathcal{P} = \{\mathbf{p}_i = (x_i, y_i, z_i) \in \mathcal{R}^3 \mid i \in \mathcal{V}\}$, which is a map from $\mathcal{V}$ to $\mathcal{R}^3$, called the *geometry realization* of the mesh.

Two vertices $\{i\}$ and $\{j\}$ are *neighbors* if $\{i, j\} \in \mathcal{K}$. The 1-ring neighborhood of a vertex $\{i\}$ is the set $n(i) = \{j \mid \{i, j\} \in \mathcal{K}\}$. The out-degree is defined as the number of its 1-ring neighborhood $|n(i)|$.

2.2. Laplacian Smoothing

Surface fairing techniques are generally based on constrained energy minimization. For a parametric surface $\mathcal{S} : P = P(u, v)$, the most popular energy functions are the membrance energy and thin-plate energy [19]:

$$E_{memb}(\mathcal{S}) = \frac{1}{2} \oint_S \left(P_u^2 + P_v^2\right) d\mathcal{S},$$

$$E_{thin}(\mathcal{S}) = \frac{1}{2} \oint_S \left(P_{uu}^2 + 2P_{uv}^2 + P_{vv}^2\right) d\mathcal{S}.$$

Their respective derivatives correspond to the Laplacian and the second-order Laplacian operator:

$$L(P) = P_{uu} + P_{vv}, \tag{1}$$

$$L^2(P) = L \circ L(P) = P_{uuuu} + P_{uuvv} + 2P_{vvvv}. \tag{2}$$

For a discrete mesh, the Laplacian at each vertex $\{i\}$ can be approximated linearly using the umbrella operator [6, 8, 9]:

$$L(i) = \frac{1}{|n(i)|} \sum_{j \in n(i)} (\mathbf{p}_j - \mathbf{p}_i). \tag{3}$$

The general form of Laplacian's linear approximation is

$$L(i) = \omega_{i,i}\mathbf{p}_i + \sum_{j \in n(i)} \omega_{i,j}\mathbf{p}_j, \tag{4}$$

where $\omega_{i,i}$ and $\omega_{i,j}$ are constant combination weights and $\omega_{i,j} = 0$ if $i \neq j$ and vertex $\{i\}$ and vertex $\{j\}$ are not adjacent. One way to smooth a discrete mesh is through a *diffusion process* [10]:

$$\frac{\partial \mathcal{P}}{\partial t} = \lambda L(\mathcal{P}). \tag{5}$$

A sequence of meshes $\mathcal{P}^k$ can be constructed by integrating the diffusion equation over time using the simple forward *Euler* scheme:

$$\mathcal{P}^{k+1} = (I + \lambda\, dt L)\mathcal{P}^k. \tag{6}$$

At each diffusion step, the small disturbances or noises will disperse into its neighborhood. From a signal processing standpoint, small disturbances or noises correspond to the high frequency component of the surface. As k grows, we will get an increasingly smoother mesh. As we pointed out above, the major problem with the Laplacian method is that the main shape of the mesh will degrade more and more as it gets smoother. A shrinkage example from Laplacian smoothing is shown in Fig. 2. Taubin [6] used signal processing analysis to show that a combination of the two derivatives in (1) and (2), i.e., $(\lambda + \mu)L - \lambda\mu L^2$, can provide *Gaussian* filtering that minimizes the shrinkage.

2.3. Local Smoothing Method

Another way to smooth the discrete mesh is to directly perform surface energy minimization. Adhering to the idea of signal processing, we propose a local minimization method for the surface energy function. Consider a small patch $\mathcal{N}$, centered at an edge $e = \{i, j\}$ as shown in Fig. 1a. The associated energy function is defined as

$$\mathcal{E} = \frac{1}{2}\left\{ L^2(i) + L^2(j) + \sum_{b \in \mathcal{B}} L^2(b) \right\}, \tag{7}$$

where $\mathcal{B} = n(i) \bigcup n(j) \backslash \{i, j\}$ denotes the set of vertices at the patch boundary. The small patch $\mathcal{N}$ is called the *smooth stencil*. Keeping the position of the stencil's boundary fixed,

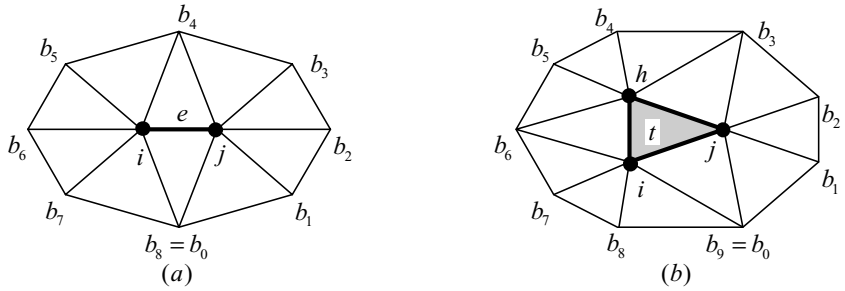

FIG. 1. Smoothing stencils: (a) edge stencil, and (b) triangle stencil. In our local smoothing method, the stencils are smoothed through the volume constrained energy minimization, providing a lower pass mesh filtering.

the above energy $\mathcal{E}$ is a function of $\mathbf{p}_i$ and $\mathbf{p}_j$, and can be represented in the following matrix form

$$\mathcal{E}(\mathbf{p}_i, \mathbf{p}_j) = \frac{1}{2}\|\mathbf{K}_i\mathbf{p}_i + \mathbf{K}_j\mathbf{p}_j - \mathbf{Q}\|^2 = \frac{1}{2}\|\mathbf{KP} - \mathbf{Q}\|^2, \tag{8}$$

where $\mathbf{K}_i$ and $\mathbf{K}_j$ are the coefficient matrices consisting of the combination weights in the Laplacian's approximation such as the umbrella operator in (3) and (4). $\mathbf{Q}$ consists of the Laplacian responses after subtracted by the corresponding weighted sums of $\mathbf{p}_i$ and $\mathbf{p}_j$. $\mathbf{P} = (\mathbf{p}_i\ \mathbf{p}_j)^T$, $\mathbf{K} = (\mathbf{K}_i\ \mathbf{K}_j)$. Taking the edge stencil in Fig. 1a as an example, the coefficient matrices are

$$\mathbf{K}_i = \begin{pmatrix} k_{0,i} \\ \vdots \\ k_{7,i} \\ k_{8,i} \\ k_{9,i} \end{pmatrix}, \quad \mathbf{K}_j = \begin{pmatrix} k_{0,j} \\ \vdots \\ k_{7,j} \\ k_{8,j} \\ k_{9,j} \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_7 \\ \mathbf{q}_8 \\ \mathbf{q}_9 \end{pmatrix},$$

where $k_{s,v} = \omega_{b_s,v}$, $k_{8,v} = \omega_{i,v}$, $k_{9,v} = \omega_{j,v}$, $\mathbf{q}_s = k_{s,i}\mathbf{p}_i + k_{s,j}\mathbf{p}_j - L(b_s)$, $\mathbf{q}_8 = k_{8,i}\mathbf{p}_i + k_{8,j}\mathbf{p}_j - L(i)$, $\mathbf{q}_9 = k_{9,i}\mathbf{p}_i + k_{9,j}\mathbf{p}_j - L(j)$, for $v \in \{i, j\}$ and $0 \le s \le 7$.

By minimizing the energy function defined on the stencil, we obtain the following local smoothing result:

$$\mathbf{P} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\mathbf{Q}. \tag{9}$$

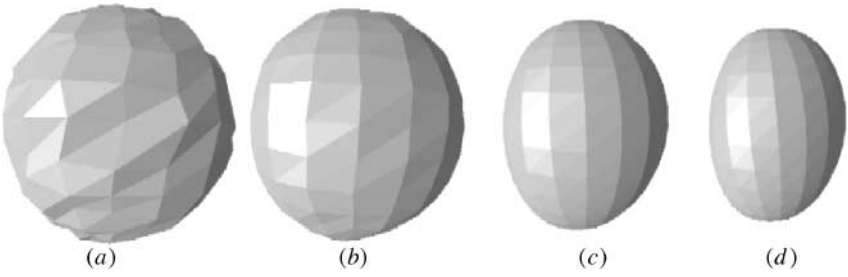

FIG. 2. A sphere smoothed by the unstrained Laplacian method. (a) The original mesh; (b), (c), and (d) Smoothed mesh after one, six, and ten times of Laplacian iterations with $\lambda\, dt = 1$. This example shows that the main shape degrades substantially.

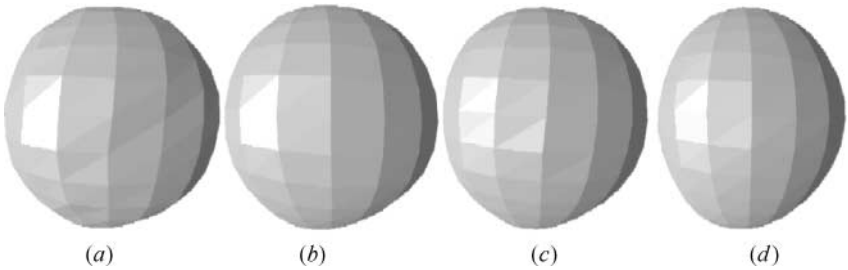

FIG. 3. Smooth results comparison. The original mesh is shown in Fig. 2a. (a) and (b) are smoothed meshes by the volume-constrained method with respectively one and six iterations; (c) and (d) are smoothed meshes by the local smoothing method with respectively one and six iterations. (a) and (b) show that the volume-constrained method can preserve the mesh's main shape very well, while there is significant shape degradation in (c) and (d), which shows that the local smoothing method still suffers from shrinkage.

Applying the above energy minimization method to all edge stencils in turn, we can obtain a smoothed mesh, as seen in the example of a sphere mesh in Figs. 3c and 3d. This example shows a similar smoothing result as that in Fig. 2; i.e., the smoothed meshes all suffer from great shrinkage.

3. VOLUME CONSTRAINED SMOOTHING

The advantage of the local smoothing method in Section 2.3 is that it can easily be modified to address the shrinkage problem by introducing a constraint of volume preservation. The major observation is that the volume V between the original mesh and the smoothed mesh can be easily obtained and then used to measure the shrinkage. Imposing the constraint of $V = 0$ onto the stencil energy minimization problem will provide an anti-shrinking mesh smoothing method. As shown in Fig. 1, the volume V is

$$V = \sum_{\{r,s,t\}\in\mathcal{N}} \left\{(\mathbf{p}_r \otimes \mathbf{p}_s)\cdot\mathbf{p}_t - \left(\mathbf{p}_r^0 \otimes \mathbf{p}_s^0\right)\cdot\mathbf{p}_t^0\right\},$$

where $\mathbf{p}_r^0$, $\mathbf{p}_s^0$, and $\mathbf{p}_t^0$ are respectively the original positions of vertex $\{r\}$, $\{s\}$, and $\{t\}$, and $\mathbf{p}_b^0 = \mathbf{p}_b$ for any vertex $b \in \mathcal{B}$, since the stencil's boundary is fixed during the smoothing process. Separating $\mathbf{p}_i$ and $\mathbf{p}_j$ from other points, we have

$$V = \Omega(\mathbf{p}_i, \mathbf{p}_j) - \Omega\left(\mathbf{p}_i^0, \mathbf{p}_j^0\right),$$

where the *volume function* Ω is a function from $\mathcal{R}^3 \times \mathcal{R}^3$ to $\mathcal{R}$,

$$\Omega(x, y) = \mathbf{n}_i \cdot x + \mathbf{n}_j \cdot y + (x \otimes \mathbf{n}_{ij}) \cdot y, \tag{10}$$

and $\mathbf{n}_i$, $\mathbf{n}_j$, and $\mathbf{n}_{ij}$ are three vectors in $\mathcal{R}^3$, determined by the positions of the stencil boundary vertices. Again, taking the edge stencil in Fig. 1a as an example, these three vectors are respectively:

$$\mathbf{n}_i = \sum_{s=4}^{7} \mathbf{p}_{b_s} \otimes \mathbf{p}_{b_{s+1}}, \qquad \mathbf{n}_{ij} = \mathbf{p}_{b_0} - \mathbf{p}_{b_4}, \qquad \mathbf{n}_j = \sum_{s=0}^{3} \mathbf{p}_{b_s} \otimes \mathbf{p}_{b_{s+1}}.$$

Keeping $V = 0$, we get a constrained energy minimization problem:

$$\begin{aligned} \min \quad & \mathcal{E}(\mathbf{p}_i, \mathbf{p}_j) = \frac{1}{2}\|\mathbf{KP} - \mathbf{Q}\|^2, \\ \text{subject to} \quad & \Omega(\mathbf{p}_i, \mathbf{p}_j) = \Omega\big(\mathbf{p}_i^0, \mathbf{p}_j^0\big). \end{aligned} \tag{11}$$

This is a typical constrained optimization problem, which can be converted to an unconstrained nonlinear one using the well-known penalty approach, the Lagrange multiplier method [27], or the elimination method (see Appendix A for details). However, general methods may be time-consuming when smoothing large meshes. Additionally, it is very difficult to obtain the global minimum for a nonlinear optimization problem. Therefore we develop a 2-step approximation method to efficiently solve the minimization problem in (11).

3.1. Two-Step Approximation

The basic idea of the 2-step method is that we first minimize the stencil energy function without volume preservation, then in the second step we translate the edge $\{i, j\}$ simultaneously by a vector to restore the volume of the original mesh while minimizing the energy increment.

The first step can be achieved by the local smoothing method discussed in Section 2.3. Let $\mathbf{P}^1 = (\mathbf{p}_i^1\ \mathbf{p}_j^1)^T$ be the smoothing result in the first step. According to (9), we have:

$$\mathbf{P}^1 = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\mathbf{Q}. \tag{12}$$

In the second step, we translate $\mathbf{p}_i^1$ and $\mathbf{p}_j^1$ simultaneously to restore the mesh volume. Let $\mathbf{w}$ be the translation vector; then the volume between the two meshes is

$$\begin{aligned} \Delta V &= \Omega\big(\mathbf{p}_i^1 + \mathbf{w}, \mathbf{p}_j^1 + \mathbf{w}\big) - \Omega\big(\mathbf{p}_i^1, \mathbf{p}_j^1\big) \\ &= \big(\mathbf{n}_i + \mathbf{n}_j + \big(\mathbf{p}_i^1 - \mathbf{p}_j^1\big) \otimes n_{ij}\big) \cdot \mathbf{w} \\ &= \mathbf{n} \cdot \mathbf{w}, \end{aligned}$$

where $\mathbf{n} = \mathbf{n}_i + \mathbf{n}_j + (\mathbf{p}_i^1 - \mathbf{p}_j^1) \otimes \mathbf{n}_{ij}$. After translation in the second step, the energy over the stencil becomes

$$\begin{aligned} \mathcal{E}(\mathbf{p}_i, \mathbf{p}_j) &= \mathcal{E}\big(\mathbf{p}_i^1 + \mathbf{w}, \mathbf{p}_j^1 + \mathbf{w}\big) \\ &= \frac{1}{2}\|(\mathbf{k}_i + \mathbf{k}_j)\mathbf{w} - (\mathbf{Q} - \mathbf{KP}^1)\|^2 \\ &= \frac{1}{2}\|(\tilde{\mathbf{K}}\mathbf{w} - \tilde{\mathbf{Q}})\|^2, \end{aligned}$$

where $\tilde{\mathbf{K}} = \mathbf{K}_i + \mathbf{K}_j$ and $\tilde{\mathbf{Q}} = \mathbf{Q} - \mathbf{KP}^1$. Therefore, the volume preservation in the second step can be formulated by the following constrained optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|(\tilde{\mathbf{K}}\mathbf{w} - \tilde{\mathbf{Q}})\|^2, \\ \text{subject to} \quad & \mathbf{n} \cdot \mathbf{w} = \Omega\big(\mathbf{p}_i^0, \mathbf{p}_j^0\big) - \Omega\big(\mathbf{p}_i^1, \mathbf{p}_j^1\big). \end{aligned} \tag{13}$$

The minimizer of (13) is

$$\mathbf{w} = \frac{1}{\alpha}(\tilde{\mathbf{K}}^T\tilde{\mathbf{Q}} - \beta\mathbf{n}),$$

where

$$\alpha = \tilde{\mathbf{K}}^T\tilde{\mathbf{K}} \qquad \text{and} \qquad \beta = \frac{1}{\|\mathbf{n}\|^2}\left\{(\tilde{\mathbf{K}}^T\tilde{\mathbf{Q}})\cdot\mathbf{n} - \alpha\left(\Omega\left(\mathbf{p}_i^0, \mathbf{p}_j^0\right) - \Omega\left(\mathbf{p}_i^1, \mathbf{p}_j^1\right)\right)\right\}.$$

In (11), the constraint is nonlinear with respect to the arguments $\mathbf{p}_i$ and $\mathbf{p}_j$, so it is somewhat difficult to obtain the minimizer. However, the constraint in (13) is linear with respect to the argument $\mathbf{w}$! Therefore, its minimizer can be easily found using Lagrange multipliers [27]. This is the motivation for translating the vertex $\{i\}$ and vertex $\{j\}$ simultaneously by a vector $\mathbf{w}$.

The smoothing result of the above volume-constrained smoothing method for the same mesh model in Fig. 2a is shown in Figs. 3a and 3b, where the results are obtained respectively by one and six iterations of the constrained smoothing method. Compared with the Laplacian smoothing results in Fig. 2 and the local smoothing results in Figs. 3c and 3d, these results preserve the mesh's main shape very well. And we compared the volume curves of the volume-constrained method, local smoothing method, and Laplacian method in Fig. 4, which shows that the volume-constrained method is the best for anti-shrinking mesh smoothing and the shrinkage of the local smoothing method is less than that of the Laplacian method.

The surface energy will increase at the second step (if $\mathbf{w} \neq 0$), so it is possible for the 2-step method to increase the surface energy. Suppose that the smoothing stencil is already the most smooth, i.e., the original positions of vertex $\{i\}$ and $\{j\}$ are the minimizer of (11). In this case the surface will be less smooth after processing by our 2-step method (if the vertex position is changed) (see Fig. 5). However, the experiments show that the surface energy decreased during the whole process in most cases. In practice, we can reject the results at those stencils where the surface energy tends to increase. It is clear that when two adjacent stencils are processed in different order, the final mesh may be different. Although the order affects the smoothing result, the difference is not noticeable for meshes with many vertices.

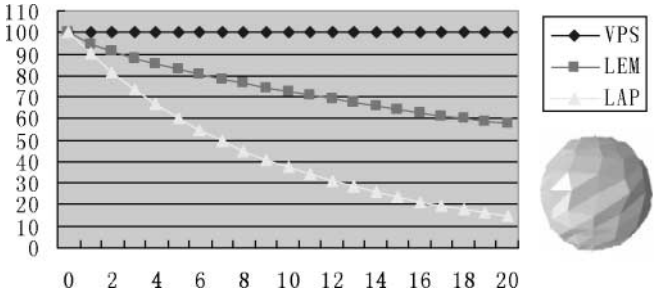

FIG. 4. The volume curves of different smoothing methods. VPS, LEM, and LAP respectively denote the volume-constrained method, local smoothing method, and Laplacian method. The vertical axis is the percentage of the smoothed mesh's volume to the original mesh's volume, the horizontal axis is the number of smoothing iterations. The original mesh is shown in Fig. 2a.

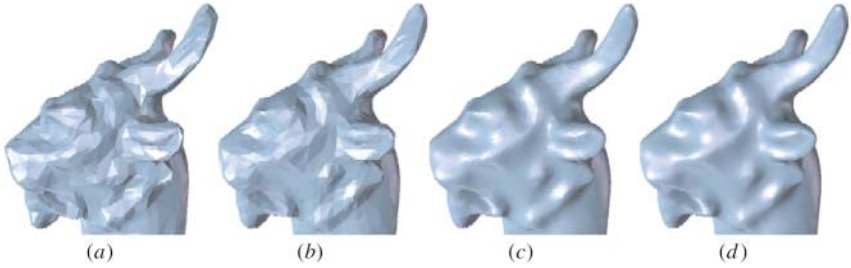

FIG. 5. Example of volume-constrained smoothing. (a) The original coarse mesh. (b) is the smoothed meshes by the volume-constrained method with five iterations. (c) Adaptively subdivide the mesh and smooth it again for five iterations. (d) Repeat the operations in (c).

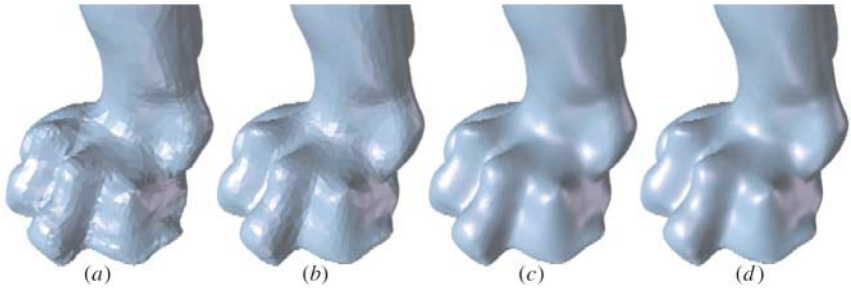

FIG. 6. Example of volume-constrained smoothing using *triangle stencil*. (a) The original coarse mesh. (b) is the smoothed meshes by the volume-constrained method with five iterations. (c) Adaptively subdivide the mesh and smooth it again for five iterations. (d) Repeat the operations in (c).

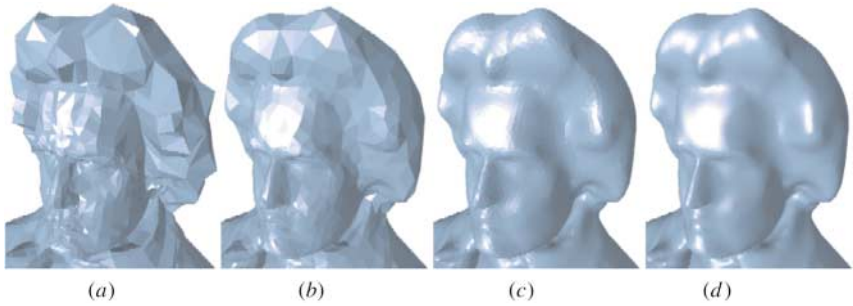

FIG. 7. Another example of the volume-constrained smoothing method. (a) The original coarse mesh; (b), (c), and (d) are the smoothed meshes after performing the same operations as in Figs. 5 and 6.

3.2. Triangle Stencil

Our volume-constrained smoothing method is applicable not only on the edge stencil, but also on some other stencils, for example the triangle stencil as shown in Fig. 1b. For the triangle stencils, a similar energy minimization problem to the edge stencils can be formulated as we show using the triangle stencil in Fig. 1b as an example.

The energy function corresponding to (8) is

$$\mathcal{E}(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_h) = \frac{1}{2}\|\mathbf{K}_i\mathbf{p}_i + \mathbf{K}_j\mathbf{p}_j + \mathbf{K}_h\mathbf{p}_h - \mathbf{Q}\|^2 = \frac{1}{2}\|\mathbf{KP} - \mathbf{Q}\|^2,$$

where $\mathbf{P} = (\mathbf{p}_i\ \mathbf{p}_j\ \mathbf{p}_h)^T$,

$$\mathbf{K}_i = \begin{pmatrix} k_{0,i} \\ \vdots \\ k_{8,i} \\ k_{9,i} \\ k_{10,i} \\ k_{11,i} \end{pmatrix}, \quad \mathbf{K}_j = \begin{pmatrix} k_{0,j} \\ \vdots \\ k_{8,j} \\ k_{9,j} \\ k_{10,j} \\ k_{11,j} \end{pmatrix}, \quad \mathbf{K}_h = \begin{pmatrix} k_{0,h} \\ \vdots \\ k_{8,h} \\ k_{9,h} \\ k_{10,h} \\ k_{11,h} \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_8 \\ \mathbf{q}_9 \\ \mathbf{q}_{10} \\ \mathbf{q}_{11} \end{pmatrix},$$

and $k_{s,v} = \omega_{b_s,v}$, $k_{9,v} = \omega_{i,v}$, $k_{10,v} = \omega_{j,v}$, $k_{11,v} = \omega_{h,v}$, $\mathbf{q}_s = k_{s,i}\mathbf{p}_i + k_{s,j}\mathbf{p}_j + k_{s,h}\mathbf{p}_h - L(b_s)$, $\mathbf{q}_9 = k_{9,i}\mathbf{p}_i + k_{9,j}\mathbf{p}_j + k_{9,h}\mathbf{p}_h - L(i)$, $\mathbf{q}_{10} = k_{10,i}\mathbf{p}_i + k_{10,j}\mathbf{p}_j + k_{10,h}\mathbf{p}_h - L(j)$, $\mathbf{q}_{11} = k_{11,i}\mathbf{p}_i + k_{11,j}\mathbf{p}_j + k_{11,h}\mathbf{p}_h - L(h)$ for $v \in \{i, j, h\}$, $0 \le s \le 8$.

The triangle stencil's *volume function* corresponding to (10) is

$$\Omega(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{n}_i \cdot \mathbf{x} + \mathbf{n}_j \cdot \mathbf{y} + \mathbf{n}_h \cdot \mathbf{z} + (\mathbf{x} \otimes \mathbf{n}_{ij}) \cdot \mathbf{y} + (\mathbf{y} \otimes \mathbf{n}_{jh}) \cdot \mathbf{z} + (\mathbf{z} \otimes \mathbf{n}_{hi}) \cdot \mathbf{x} + (\mathbf{x} \otimes \mathbf{y}) \cdot \mathbf{z},$$

where

$$\mathbf{n}_{ij} = \mathbf{p}_{b_0}, \qquad \mathbf{n}_{jh} = \mathbf{p}_{b_3}, \qquad \mathbf{n}_{hi} = \mathbf{p}_{b_6},$$

$$\mathbf{n}_i = \sum_{s=6}^{8} \mathbf{p}_{b_s} \otimes \mathbf{p}_{b_{s+1}}, \qquad \mathbf{n}_j = \sum_{s=0}^{2} \mathbf{p}_{b_s} \otimes \mathbf{p}_{b_{s+1}}, \qquad \mathbf{n}_h = \sum_{s=3}^{5} \mathbf{p}_{b_s} \otimes \mathbf{p}_{b_{s+1}}.$$

And the volume between the smoothed mesh and the original mesh is

$$V = \Omega(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_h) - \Omega\left(\mathbf{p}_i^0, \mathbf{p}_j^0, \mathbf{p}_h^0\right).$$

At last the same 2-step approach in Section 3.1 is used to smooth the stencil and preserve the local mesh volume. Figure 6 shows the smoothing results for a feline claw mesh model using the triangle stencil. One can also try a stencil smaller than the edge stencil in the same way, such as the vertex stencil $\{i\} \cup n(i)$ centered at vertex $\{i\}$. However the vertex stencil is so small that the volume constraint to preserve the volume may be too strong to smooth the stencil. Larger stencils containing too many variables are not preferred since they lead to solving a larger linear system at each local minimization process, which will hurt the efficiency. And the simple simultaneous translation of all inner vertices in a large stencil is also not reasonable for preserving the volume. Therefore we have currently implemented only two kinds of stencils, the edge stencil and the triangle stencil.

4. CONCLUSION, DISCUSSION AND FUTURE WORK

We presented a new approach to smooth a triangular mesh for surface reconstruction and discrete fair surface design. Adhering to the idea of signal processing, the mesh is smoothed stencil by stencil through a local energy minimization process. The volume-constrained smoothing method has been implemented with the edge stencil and the triangle stencil and tested on many triangular mesh models, including a very coarse model like the noised sphere shown in Fig. 2a and many very dense models as shown in Figs. 5, 6, and 7. In the latter three figures, the meshes are subdivided adaptively. In short, the subdivision scheme consists of three steps. First, according to a user-specified length value δ, edges longer than δ are uniformly subdivided such that each subedge is less than δ. Second, cast some points inside each face such that they are distributed as evenly as possible and the distance between nearby points is less than δ. At last, subdivide the faces using the new points. Experimental results show that the volume-constrained smoothing method generates very smooth models without any shrinkage due to its exact volume preservation and the two-step approach is an effective solution to the constrained minimization problem.

Although the volume for open meshes is undefined, our volume-constrained smoothing method can handle them naturally since it uses the local volume instead of the global volume. Certainly, the boundary curves of an open mesh need to be processed separately. Similar to local volume-constrained method, a local area-constrained method can be designed to smooth boundary curves such that the hole formed by the boundary will remain the same size.

Currently the constrained smoothing method is implemented for single resolution meshes. One topic of future work is to combine it with a multiresolution mesh representation for multiresolution modeling purposes [9, 11, 18, 19]. We observed that when the mesh becomes denser, the function of the volume constraint becomes smaller. When the mesh is extremely dense, even the standard Laplacian can smooth the mesh quite well without any noticeable main shape shrinkage and distortion. Therefore, taking advantage of the multiresolution mesh representation, we can perform the volume-constrained smoothing method for coarse mesh levels to smooth the mesh and preserve the main shape, while using the standard Laplacian smoothing method or the implicit fairing method [10] for fine mesh levels for efficiency. For an irregular mesh, there are many simplification methods to build the multiresolution representation [20–26], among which the method proposed by Lindstrom *et al.* [25, 26] should be the most suitable for our purpose, since it also preserves the mesh volume for decimation.

In our constrained smoothing method, the volume is preserved, but it may be transferred to another neighboring area. Note that other operators can also be adopted in the first step of the 2-step approach. Another area of future work is to find an alternative approach for the first step to avoid this kind of volume transfer.

APPENDIX A

According to the volume constraint, we can eliminate a variable to convert the constrained energy minimization in (11) into a normal minimization problem without constraint. First the derivatives of $\mathcal{E}(\mathbf{p}_i, \mathbf{p}_j)$ with respect to $\mathbf{p}_i$ and $\mathbf{p}_j$ can be easily obtained:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i} = \mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q}), \qquad \frac{\partial \mathcal{E}}{\partial \mathbf{p}_i} = \mathbf{K}_j^T(\mathbf{KP} - \mathbf{Q}). \tag{14}$$

Denote $F(\mathbf{p}_i, \mathbf{p}_j) = \Omega(\mathbf{p}_i, \mathbf{p}_j) - \Omega(\mathbf{p}_i^0, \mathbf{p}_j^0)$; then the volume constraint becomes $F(\mathbf{p}_i, \mathbf{p}_j) = 0$. Similarly, the derivatives of $F(\mathbf{p}_i, \mathbf{p}_j)$ with respect to $\mathbf{p}_i$ and $\mathbf{p}_j$ can be easily obtained also:

$$\frac{\partial F}{\partial \mathbf{p}_i} = \mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j, \quad \frac{\partial F}{\partial \mathbf{p}_j} = \mathbf{n}_j - \mathbf{n}_{ij} \otimes \mathbf{p}_i. \tag{15}$$

In typical cases, the derivatives are not 0. Suppose $(\frac{\partial F}{\partial \mathbf{p}_i})_x$ is not 0 without loss of any generality (for a vector $v \in \mathcal{R}^3$, $(v)_x$, $(v)_y$, and $(v)_z$ are used to denote its x, y, and z components, respectively). Then according to the volume constraint $F(\mathbf{p}_i, \mathbf{p}_j) = 0$, the x component of $\mathbf{p}_i$ can be represented by $\mathbf{p}_j$ and the other two components of $\mathbf{p}_i$. We denote it as:

$$(\mathbf{p}_i)_x = \Psi((\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j). \tag{16}$$

Then, substituting $(\mathbf{p}_i)_x$ from (16), the energy function (8) becomes

$$\tilde{\mathcal{E}}((\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j) = \mathcal{E}(\Psi((\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j), (\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j),$$

and the energy minimization in (11) is converted into a normal optimization problem without constraint:

$$\min_{(\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j} \tilde{\mathcal{E}}((\mathbf{p}_i)_y, (\mathbf{p}_i)_z, \mathbf{p}_j). \tag{17}$$

The minimizer in (17) can be obtained using the conjugate gradient method (see [28] for details). This method requires the derivatives of $\tilde{\mathcal{E}}$ with respect to $(\mathbf{p}_i)_y$, $(\mathbf{p}_i)_z$, and $\mathbf{p}_j$, which can be obtained according to (14), (15), and (16):

$$\begin{aligned}
\frac{\partial \tilde{\mathcal{E}}}{\partial (\mathbf{p}_i)_y} &= \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_y + \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\partial \Psi}{\partial (\mathbf{p}_i)_y} = \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_y - \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\left(\frac{\partial F}{\partial \mathbf{p}_i}\right)_y}{\left(\frac{\partial F}{\partial \mathbf{p}_i}\right)_x} \\
&= \left(\mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q})\right)_y - \left(\mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q})\right)_x \frac{(\mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j)_y}{(\mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j)_x}, \\
\frac{\partial \tilde{\mathcal{E}}}{\partial (\mathbf{p}_i)_z} &= \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_z + \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\partial \Psi}{\partial (\mathbf{p}_i)_z} = \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_z - \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\left(\frac{\partial F}{\partial \mathbf{p}_i}\right)_z}{\left(\frac{\partial F}{\partial \mathbf{p}_i}\right)_x} \\
&= \left(\mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q})\right)_z - \left(\mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q})\right)_x \frac{(\mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j)_z}{(\mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j)_x}, \\
\frac{\partial \tilde{\mathcal{E}}}{\partial \mathbf{p}_j} &= \frac{\partial \mathcal{E}}{\partial \mathbf{p}_j} + \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\partial \Psi}{\partial \mathbf{p}_j} = \frac{\partial \mathcal{E}}{\partial \mathbf{p}_j} - \left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_x \frac{\frac{\partial F}{\partial \mathbf{p}_j}}{\left(\frac{\partial F}{\partial \mathbf{p}_i}\right)_x} \\
&= \mathbf{K}_j^T(\mathbf{KP} - \mathbf{Q}) - \left(\mathbf{K}_i^T(\mathbf{KP} - \mathbf{Q})\right)_x \frac{\mathbf{n}_i + \mathbf{n}_{ij} \otimes \mathbf{p}_j}{(\mathbf{n}_i - \mathbf{n}_{ij} \otimes \mathbf{p}_j)_x}.
\end{aligned}$$

It is clear that, although (17) is a normal minimization problem without constraint, the computation cost of the above direct solution is too great to be used for large mesh smoothing. Additionally, the returned solution is not stable because of the possibility of multiple local minima. This is why we propose the 2-step method approach to preserve the volume.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their valuable comments. Timely proofreading by Steve Lin is greatly appreciated. Hujun Bao and Qunsheng Peng were supported in part by National Natural Science Foundations of China (Grants 69925204, 60021201, and 60133020) for Distinguished Young Scholars and Innovative Research Groups.

REFERENCES

1. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, Surface reconstruction from unorganized points, in *SIGGRAPH 92 Conference Proceedings, 1992*, pp. 71–78.
2. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, Piecewise smooth surface reconstruction, in *SIGGRAPH 94 Conference Proceedings, 1994*, pp. 295–302.
3. N. Amenta, M. Bern, and M. Kamvysselis, A new Voronoi-based surface reconstruction algorithm, *SIGGRAPH 98 Conference Proceedings, 1998*, pp. 415–421.
4. B. Curless and M. Levoy, A volumetric method for building complex models from range images, in *SIGGRAPH 96 Conference Proceedings, 1996*, pp. 303–312.
5. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, Mesh optimazation, in *SIGGRAPH 93 Conference Proceedings, 1993*, pp. 19–26.
6. G. Taubin, A signal processing approach to fair surface design, in *SIGGRAPH 95 Conference Proceedings, 1995*, pp. 351–358.
7. G. Taubin, T. Zhang, and G. Golub, Optimal surface smoothing as filter design, in *Proceedings of the 4th European Conference on Computer Vision, Cambridge, UK, 1996*, pp. 283–292.
8. L. Kobbelt, Discrete fairing, in *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, 1997*, pp. 101–131.
9. L. Kobbelt, S. Campagna, J. Vorsatz, and H. P. Seidel, Interactive multiresolution modeling on arbitrary meshes, in *SIGGRAPH 98 Conference Proceedings, 1998*, pp. 105–114.
10. M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, in *SIGGRAPH 99 Conference Proceedings, 1999*, pp. 317–324.
11. I. Guskov, W. Sweldens, and P. Schroder, Multiresolution signal processing for meshes, in *SIGGRAPH 99 Conference Proceedings, 1999*, pp. 325–334.
12. W. Welch and A. Witkin, Varational surface modeling, in *SIGGRAPH 92 Conference Proceedings, 1992*, pp. 157–166.
13. W. Welch and A. Witkin, Free-form shape design using triangulated surfaces, in *SIGGRAPH 94 Conference Proceedings, 1994*, pp. 247–256.
14. H. P. Morton and C. H. Sequin, Functional optimization for fair surface designmm, in *SIGGRAPH 92 Conference Proceedings, 1992*, pp. 167–176.
15. U. Clarenz, U. Diewald, and M. Rumpf, Nonlinear anisotropic diffusion in surface processing, in *Proceedings of IEEE Visualization 2000*, 2000, pp. 397–405.
16. J. Vollmer, R. Mencl, and H. Muller, Improved Laplacian smoothing of noisy surface meshes, in *EUROGRAPHICS 99 Conference Proceedings, 1999*, pp. 131–138.
17. X. Liu, H. Bao, P. Heng, T. Wong, and Q. Peng, Constrained fairing for meshes, *Comput. Graphics Forum* **20**(2), 2001, 115–124.
18. D. Zorin, P. Schroder, and W. Sweldens, Interactive multiresolution mesh editing, in *SIGGRAPH 97 Conference Proceedings, 1997*, pp. 259–268.
19. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, Multiresolution analysis of arbitrary meshes, in *SIGGRAPH 95 Conference Proceedings, 1995*, pp. 173–182.
20. M. Garland and P. S. Heckbert, Surface simplification using quardric error metrics, in *SIGGRAPH 97 Conference Proceedings, 1997*, pp. 209–218.
21. H. Hoppe, Progressive meshes, in *SIGGRAPH 96 Conference Proceedings, 1996*, pp. 99–108.

22. J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright, Simplification envelopes, in *SIGGRAPH 96 Conference Proceedings, 1996*, pp. 119–128.
23. A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, Multiresolution decimation based on global error, *Visual Comput.* **13**(5), 1997, 228–246.
24. W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, Decimation of triangle meshes, in *SIGGRAPH 92 Conference Proceedings, 1992*, pp. 65–70.
25. P. Lindstrom and G. Turk, Fast and memory efficient polygonal simplification, in *IEEE Visualization 98 Conference Proceedings, 1998*, pp. 279–286.
26. P. Lindstrom and G. Turk, Evaluation of memoryless simplification, *IEEE Trans. Visual. Comput. Graphics* **5**(2), 1999, 98–115.
27. A. Schabak and H. Werner, *Numerische Mathematik*, Springer-Verlag, Berlin/New York, 1993.
28. H. William, A. Saul, T. William, and P. Brian, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, UK, 1992.