Vertex-Based Diffusion for 3-D Mesh Denoising

Ying Zhang and A. Ben Hamza

Abstract—**We present a vertex-based diffusion for 3-D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3-D mesh smoothing strategy to fully preserve the geometric structure of the data. Illustrating experimental results demonstrate a much improved performance of the proposed approach in comparison with existing methods currently used in 3-D mesh smoothing.**

Index Terms—**3-D mesh smoothing, partial differential equations.**

I. Introduction

THE great challenge in image processing and computer graphics is to devise computationally efficient and optimal algorithms for recovering images and 3-D models contaminated by noise and preserving their geometrical structure. With the increasing use of scanners to create 3-D models, which are usually represented as triangle meshes in computer graphics and geometric-aided design, there is a rising need for robust and efficient 3-D mesh denoising techniques to remove undesirable noise from the data.

In recent years, various partial differential equations (PDE)-based methods have been proposed to tackle the problem of 2-D image denoising with a good preservation of features [1]–[10]. Much of the appeal of PDE-based methods lies in the availability of a vast arsenal of mathematical tools which, at the very least, act as a key guide in achieving numerical accuracy, as well as stability. Partial differential equations or gradient descent flows are generally a result of variational problems [11]. The 3-D mesh denoising problem, however, has received much less attention [12]–[16]. The most commonly used mesh smoothing method is Laplacian flow, which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices [12]. Although the Laplacian smoothing flow is simple and fast, it produces, however, the shrinking effect and an oversmoothing result. The most recent mesh denoising techniques include the mean, median, and bilateral filters [17]–[19], which are all adopted from image processing literature. Also, a number of anisotropic diffusion methods for triangle meshes and implicit surfaces have been proposed recently. Desbrun *et al.* [20], [21] introduce a weighted Laplacian smoothing technique by choosing new edge weights based on curvature flow operators. This denoising method avoids the undesirable edge equalization from Laplacian flow and helps to preserve curvature for constant curvature areas. However, recomputing new edge weights after each iteration results in more expensive computational cost. Clarenz *et al.* [22] propose a multiscale surface smoothing method based on the anisotropic curvature evolution problem. By discretizing nonlinear partial differential equations, this method aims to detect and preserve sharp edges by two user defined parameters which are a regularization parameter for filtering out high frequency noisy and a threshold for edge detection. This multiscale method was also extended to the texture mapped surfaces [23] in order to enhance edge type features of the texture maps. Different regularization parameters and edge detection threshold values, however, need to be defined by users onto noisy surfaces and textures respectively before the smoothing process. Bajaj *et al.* [24] present a unified anisotropic diffusion for 3-D mesh smoothing by treating discrete surface data as a discretized version of a 2-D Riemannian manifold and establishing a PDE diffusion model for such a manifold. This method helps enhancing sharp features while filtering out noise by considering three-ring neighbors of each vertex to achieve nonlinear approach of smoothing process. Tasdizen *et al.* [25], [26] introduce a two-step surface smoothing method by solving a set of coupled second-order PDEs on level set surface models. Instead of filtering the positions of points on a mesh, this method operates on the normal map of a surface and manipulates the surface to fit the processed normals. All the surfaces normals are processed by solving second-order equations using implicit surfaces. In [27], Hildebrandt *et al.* present a mesh smoothing method by using a prescribed mean curvature flow for simplicial surfaces. This method develops an improved anisotropic diffusion algorithm by defining a discrete shape operator and principal curvatures of simplicial surfaces.

Motivated by the outperformance, in tackling the 2-D image denoising problem, of Laplacian smoothing by anisotropic diffusion [1], [2], [5], we propose in this paper a vertex-based nonlinear flow for 3-D mesh smoothing by solving a discrete partial differential equation [28]. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3-D mesh smoothing strategy to fully preserve the geometric structure of the data.

The rest of this paper is organized as follows. In the next section, we briefly recall some basic concepts of 3-D mesh data,

Manuscript received June 15, 2006; revised November 8, 2006. This work was supported in part by Natural Sciences and Engineering Research Council of Canada under Discovery Grant N00929. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Philippe Salembier.

The authors are with the Concordia Institute for Information Systems Engineering Concordia University, Montréal, QC H3G 1M8 Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2007.891787

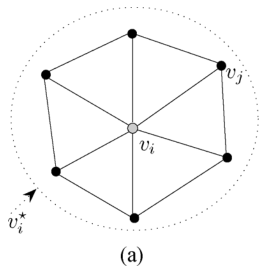

Fig. 1. (a) Vertex neighborhood $\boldsymbol{v}_i^\star$; (b) triangle neighborhood $\boldsymbol{t}_i^\star$.

and we introduce the vertex differential operators, then a general formulation of 3-D mesh smoothing problem is stated. In Section III, we briefly review some recent 3-D mesh denoising techniques that are closely related to our proposed approach. In Section IV, a vertex-based nonlinear diffusion for 3-D mesh smoothing is introduced. In Section V, we provide experimental results to demonstrate a much improved performance of the proposed method in 3-D mesh smoothing. Finally, some conclusions are included in Section VI.

II. Problem Formulation

In computer graphics and geometric-aided design, 3-D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles. Each edge $e_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$ connects a pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$. Two distinct vertices $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are adjacent (written $\boldsymbol{v}_i \sim \boldsymbol{v}_j$) if they are connected by an edge $e_{ij} \in \mathcal{E}$. The neighborhood (also referred to as a ring) of a vertex $\boldsymbol{v}_i$ is the set $\boldsymbol{v}_i^\star = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is simply the cardinality of $\boldsymbol{v}_i^\star$. We denote by $\mathcal{T}(\boldsymbol{v}_i^\star)$ the set of triangles of the ring $\boldsymbol{v}_i^\star$, and by $\boldsymbol{t}_i^\star$ the set of all triangles sharing a vertex or an edge with a triangle $\boldsymbol{t}_i \in \mathcal{T}$ of a mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$. Fig. 1(a) depicts an example of a neighborhood $\boldsymbol{v}_i^\star$, where the degree of the vertex $\boldsymbol{v}_i$ is $d_i = 6$, and the number of triangles of the set $\mathcal{T}(\boldsymbol{v}_i^\star)$ is also equal to 6. An illustration of $\boldsymbol{t}_i^\star$ is provided in Fig. 1(b).

Given a triangle $\boldsymbol{t}_j \in \boldsymbol{t}_i^\star$, we denote by $A(\boldsymbol{t}_j)$ and $\boldsymbol{n}(\boldsymbol{t}_j)$ the area and the unit normal of $\boldsymbol{t}_j$, respectively. The normal $\boldsymbol{n}_i$ at a vertex $\boldsymbol{v}_i$ is obtained by averaging the normals of its neighboring triangles and is given by

$$\boldsymbol{n}_i = \frac{1}{d_i} \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} \boldsymbol{n}(\boldsymbol{t}_j). \tag{1}$$

The mean edge length $\bar{\ell}$ of the mesh $\mathbb{M}$ is given by

$$\bar{\ell} = \frac{1}{|\mathcal{E}|} \sum_{e_{ij} \in \mathcal{E}} \|e_{ij}\| \tag{2}$$

where $\|e_{ij}\| = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$, and $\|e_{ij}\| = 0$ otherwise.

A. Laplacian Matrix of a Triangle Mesh

The Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by $L = D - A$, where $A = (a_{ij})$ is the adjacency matrix between the vertices, that is $a_{ii} = 0$ and $a_{ij} = 1$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$, and D is an $m \times m$ diagonal matrix whose (i, i) entry is d_i [29]. The normalized Laplacian matrix $\mathcal{L}$ is given by [29]

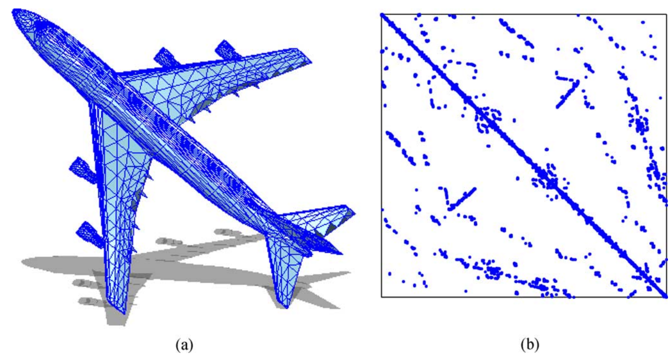

Fig. 2. (a) Three-dimensional triangle mesh and (b) its Laplacian matrix.

$$\mathcal{L} = D^{-1/2} L D^{-1/2}$$

and may be viewed as an operator defined on the space of functions $\varphi : \mathcal{V} \to \mathbb{R}$ as follows:

$$\mathcal{L}\varphi(\boldsymbol{v}_i) = -\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\varphi(\boldsymbol{v}_j)}{\sqrt{d_j}} - \frac{\varphi(\boldsymbol{v}_i)}{\sqrt{d_i}} \right), \quad \forall \boldsymbol{v}_i \in \mathcal{V}.$$

Fig. 2 illustrates an example of a 3-D triangle mesh and its Laplacian matrix.

B. Vertex Differential Operators

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, we define the vertex gradient operator $\nabla \boldsymbol{v}_i$ as

$$\nabla \boldsymbol{v}_i = \left\{ \frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} : \boldsymbol{v}_j \in \boldsymbol{v}_i^\star \right\}.$$

We also define the vertex Laplace operator as

$$\operatorname{div}(\nabla \boldsymbol{v}_i) = \Delta \boldsymbol{v}_i = \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right)$$

where $\operatorname{div}(\cdot)$ is the divergence operator. Note the analogy between the vertex Laplace operator and the normalized Laplacian matrix $\mathcal{L}$ defined as an operator.

C. Mesh Smoothing Model

In all real applications, measurements are perturbed by noise. In the course of acquiring, transmitting, or processing a 3-D model for example, the noise-induced degradation often yields a resulting vertex observation model, and the most commonly used is the additive one

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\eta} \tag{3}$$

where the observed vertex $\boldsymbol{v}$ includes the original vertex $\boldsymbol{u}$, and the random noise process $\boldsymbol{\eta}$ which is usually assumed to be Gaussian with zero mean and standard deviation σ.

Mesh smoothing refers to the process of recovering a 3-D model contaminated by noise. The challenge of the problem of interest lies in recovering the vertex $\boldsymbol{u}$ from the observed vertex

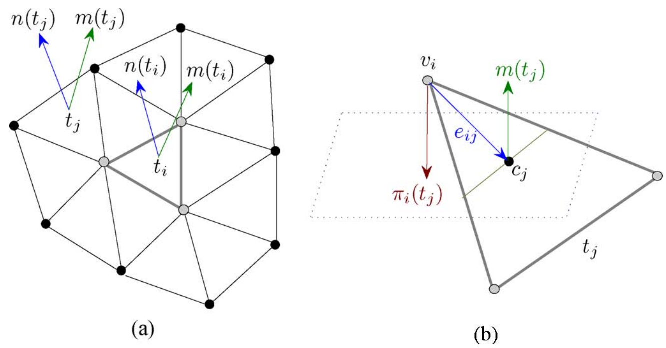

Fig. 3. Illustration of the mesh mean filter algorithm.

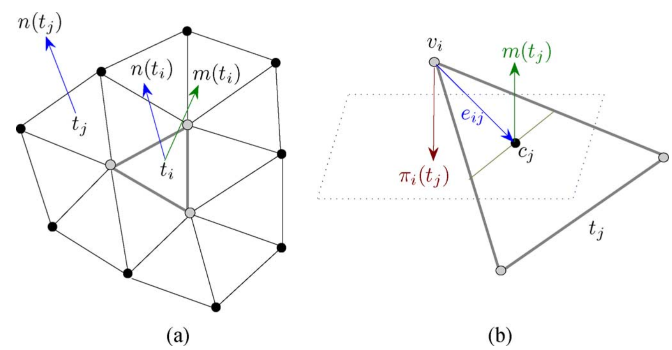

Fig. 4. Illustration of the mesh median filter algorithm.

$\boldsymbol{v}$, and furthering the estimation by making use of any prior knowledge/assumptions about the noise process $\boldsymbol{\eta}$.

The PDE-based smoothing approach is commonly formulated in a continuous domain which enjoys a large arsenal of analytical tools and, hence, offers a greater flexibility. Laplacian smoothing is the most commonly used mesh smoothing method which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices using the following update rule

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \left(\frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{d_i} \right). \tag{4}$$

It is worth pointing out that the Laplacian flow given by (4) is the discrete form of the isotropic heat equation $\boldsymbol{v}_t = \Delta \boldsymbol{v}$ applied to each vertex of the triangle mesh, where we assume that all vertices have the same degree.

Although the Laplacian smoothing flow is simple and fast, it tends, however, to produce a shrinking effect and an oversmoothing result. Motivated the good performance of anisotropic diffusion, we propose in Section IV a vertex-based flow defined by a nonlinear partial differential equation.

III. Related Work

In this section, we will review some representative methods for 3-D mesh smoothing that are closely related to our proposed approach, and we briefly show their mathematical foundations and algorithmic methodologies as well as their limitations.

A. Mean Filter for Averaging Face Normals

The mean filter procedure is depicted in Fig. 3 and is applied in three successive steps [17].

Step 1) Compute the area weighted average face normal $\mathbf{m}(\boldsymbol{t}_i)$ for each mesh triangle $\boldsymbol{t}_i$

$$\boldsymbol{m}(\boldsymbol{t}_i) = \frac{1}{\sum\limits_{\boldsymbol{t}_j \in \boldsymbol{t}_i^\star} A(\boldsymbol{t}_j)} \sum_{\boldsymbol{t}_j \in \boldsymbol{t}_i^\star} A(\boldsymbol{t}_j) \boldsymbol{n}(\boldsymbol{t}_j). \tag{5}$$

Step 2) Normalize the averaged normal $\boldsymbol{m}(\boldsymbol{t}_i)$

$$\boldsymbol{m}(\boldsymbol{t}_i) \leftarrow \frac{\boldsymbol{m}(\boldsymbol{t}_i)}{\|\boldsymbol{m}(\boldsymbol{t}_i)\|}.$$

Step 3) Update each vertex $\boldsymbol{v}_i$ in the mesh as follows:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{1}{\sum\limits_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} A(\boldsymbol{t}_j)} \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} A(\boldsymbol{t}_j) \boldsymbol{\pi}_i(\boldsymbol{t}_j)$$

where $\boldsymbol{\pi}_i(\boldsymbol{t}_j) = \langle \boldsymbol{e}_{ij}, \boldsymbol{m}(\boldsymbol{t}_j) \rangle \boldsymbol{m}(\boldsymbol{t}_j)$, and $\boldsymbol{e}_{ij} = \boldsymbol{c}_j - \boldsymbol{v}_i$ is the vector from vertex $\boldsymbol{v}_i$ to the centroid $\boldsymbol{c}_j$ of the triangle $\boldsymbol{t}_j$. Note that by definition of the inner product, the vector $\boldsymbol{\pi}_i(\boldsymbol{t}_j)$ is the projection of the vector $\boldsymbol{e}_{ij}$ onto the direction of the normal $\boldsymbol{t}_j$.

B. Angle Median Filtering for Face Normals

For each triangle $\boldsymbol{t}_i \in \mathcal{T}$, denote by $\Theta_i = \{\theta_{ij} = \angle(\boldsymbol{n}(\boldsymbol{t}_i), \boldsymbol{n}(\boldsymbol{t}_j)) : \boldsymbol{t}_j \in \boldsymbol{t}_i^\star\}$ the set of angles between $\boldsymbol{n}(\boldsymbol{t}_i)$ and $\boldsymbol{n}(\boldsymbol{t}_j)$, where $\boldsymbol{n}(\boldsymbol{t}_i)$ is the normal of $\boldsymbol{t}_i$ and $\boldsymbol{n}(\boldsymbol{t}_j)$ is the normal of $\boldsymbol{t}_j$. As illustrated in Fig. 4, instead of computing the average face normal in Step 1) of the mean filter, in the angle median filtering method [17], we first compute the median angle $\hat{\theta}_i = \mathrm{median}(\Theta_i) = \angle(\boldsymbol{n}(\boldsymbol{t}_i), \boldsymbol{n}(\hat{\boldsymbol{t}}_j))$ where $\hat{\boldsymbol{t}}_j$ is the triangle where the median angle is achieved, and then we replace the weighted average normal $\boldsymbol{m}(\boldsymbol{t}_i)$ by $\boldsymbol{n}(\hat{\boldsymbol{t}}_j)/\|\boldsymbol{n}(\hat{\boldsymbol{t}}_j)\|$..

The mean and median filtering methods show better performance than the Laplacian flow. These two methods, however, require a large number of iterations to reach stable results.

C. Weighted Laplacian Filter

Instead of using unit edge costs, the weighted Laplacian smoothing method [20] chooses edge weights based on the approximation to the curvature normal. The edge weights w_{ij} are given by $w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$, where α_{ij} and β_{ij} are the angles $\angle \boldsymbol{v}_i \boldsymbol{v}_{j-1} \boldsymbol{v}_j$ and $\angle \boldsymbol{v}_i \boldsymbol{v}_{j+1} \boldsymbol{v}_j$ depicted in Fig. 5. Then, the update rule of the weighted Laplacian smoothing procedure is given by

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{1}{\sum\limits_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} w_{ij}} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} w_{ij} (\boldsymbol{v}_j - \boldsymbol{v}_i). \tag{6}$$

The improved edge weights are used to compensate for the irregularities of the triangle mesh and to help avoid the edge equalization.

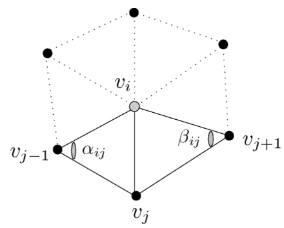

Fig. 5. Illustration of the angles α_{ij} and β_{ij}.

D. Anisotropic Geometric Diffusion

Motivated by the good performance of anisotropic diffusion in image processing, Clarenz *et al.* [22] proposed an anisotropic geometric diffusion for 3-D mesh processing using a diffusion tensor defined from the principal directions and principal curvatures of the deformed surface. The performance of the geometric diffusion is, however, heavily dependent of two user-defined parameters, and also sensitive to the estimation of higher order derivatives.

E. Bilateral Mesh Denoising

Similar to the mean and angle median filters, the bilateral 3-D mesh denoising method [18] was also adopted from the bilateral filtering technique used in image denoising. This algorithm filters each vertex $\boldsymbol{v}_i$ of the mesh in the normal direction using local neighborhoods according the following update rule:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{n}_i \left(\frac{\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} (w_{ij}^c\, w_{ij}^s) \langle \boldsymbol{n}_i, \boldsymbol{v}_i - \boldsymbol{v}_j \rangle}{\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} w_{ij}^c\, w_{ij}^s} \right)$$

where $\boldsymbol{n}_i$ is the vertex normal, w_{ij}^c is the standard Gaussian filter $w_{ij}^c = e^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2/2\sigma_c^2}$ with parameter σ_c, and w_{ij}^s is feature-preserving weight function $w_{ij}^s = e^{-\langle \boldsymbol{n}_i, \boldsymbol{v}_i - \boldsymbol{v}_j \rangle^2/2\sigma_s^2}$ with parameter σ_s. Bilateral mesh denoising algorithm is parameter-dependent and requires the user to assign the two parameters σ_c and σ_s interactively. The lack of object information, however, might affect the smoothing result.

IV. Proposed Method

The proposed vertex-based method for 3-D mesh smoothing is motivated by the good performance of anisotropic diffusion in 2-D image denoising, and it is defined by the following nonlinear vertex-based partial differential equation:

$$\boldsymbol{v}_t = \operatorname{div}(g(|\nabla \boldsymbol{v}|)\nabla \boldsymbol{v}) \tag{7}$$

where g is Cauchy weight function (see Fig. 6) given by

$$g(x) = \frac{1}{1 + \frac{x^2}{c^2}} \tag{8}$$

and c is a constant tuning parameter that needs to be estimated.

Intuitively, the smoothing effect of the proposed flow may be explained as follows: in flat regions of a 3-D mesh where the vertex gradient magnitudes are relatively small, (7) is reduced to the heat equation which tends to smooth more but the smoothing effect is unnoticeable. And around the sharp features of the 3-D mesh where the vertex gradient magnitudes are large, the diffusion flow given by (7) tends to smooth less and, hence, leads to

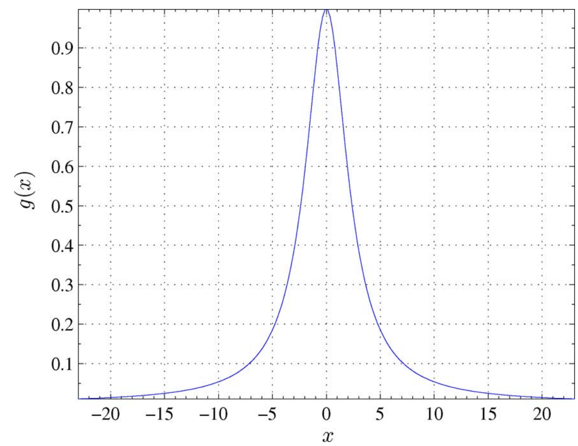

Fig. 6. Cauchy weight function with $c = 2.3849$.

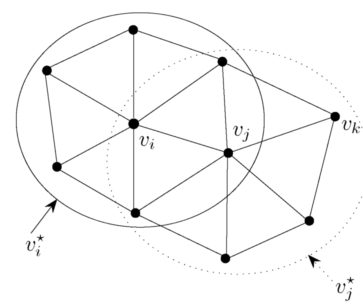

Fig. 7. Illustration of two neighboring rings.

a much better preservation of the mesh geometric structures. It can be shown (see [31]) that the 95% asymptotic efficiency on the standard Gaussian distribution is obtained with the tuning constant $c = 2.3849$. This tuning value is used in all the experimental results of Section V.

In discrete form, it is easy to show that the proposed vertex-based flow is reduced to the following update rule:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right) \Big(g(|\nabla \boldsymbol{v}_i|) + g(|\nabla \boldsymbol{v}_j|) \Big) \tag{9}$$

where the gradient magnitudes are given by

$$|\nabla \boldsymbol{v}_i| = \left(\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \left\| \frac{\boldsymbol{v}_i}{\sqrt{d_i}} - \frac{\boldsymbol{v}_j}{\sqrt{d_j}} \right\|^2 \right)^{1/2} \tag{10}$$

and

$$|\nabla \boldsymbol{v}_j| = \left(\sum_{\boldsymbol{v}_k \in \boldsymbol{v}_j^\star} \left\| \frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_k}{\sqrt{d_k}} \right\|^2 \right)^{1/2}. \tag{11}$$

Note that the update rule of the proposed method requires the use of two neighboring rings as depicted in Fig. 7.

V. Experimental Results

This section presents simulation results where the mean filtering [17], angle median filtering [17], weighted Laplacian [20], [21], geometric diffusion [22], bilateral filtering [18], and the proposed method are applied to noisy 3-D models obtained by adding Gaussian noise to the original three models shown

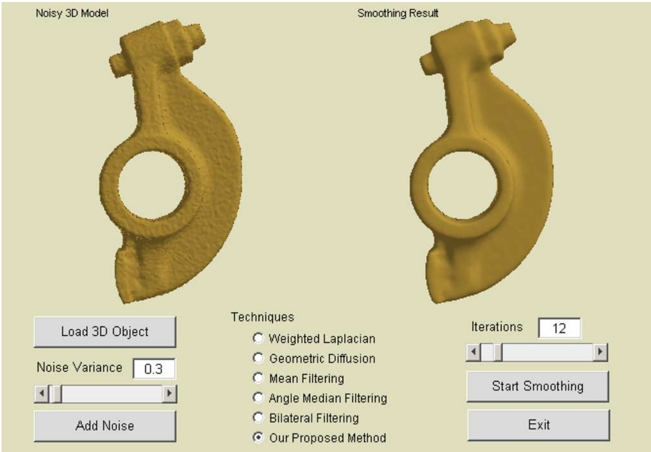

Fig. 8. Graphical user Interface for 3-D mesh denoising.

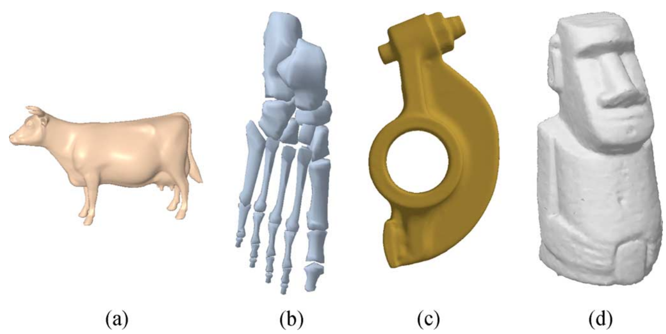

Fig. 9. Original 3-D models used for experimentation: (a) cow, (b) foot bones, (c) rocker arm, and (d) moai statue.

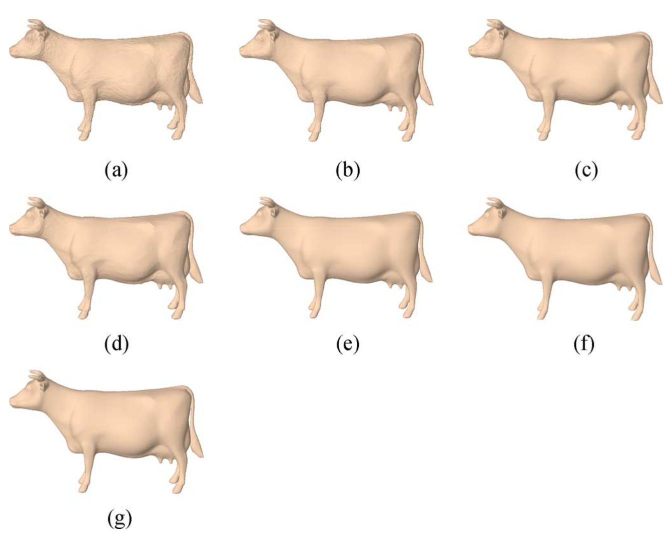

Fig. 10. Three-dimensional mesh smoothing results. (a) Noisy 3-D cow model with 92 864 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by the geometric diffusion, (f) smoothed by bilateral mesh flow, and (g) smoothed by the proposed approach. The number of iteration times is 18 for each case.

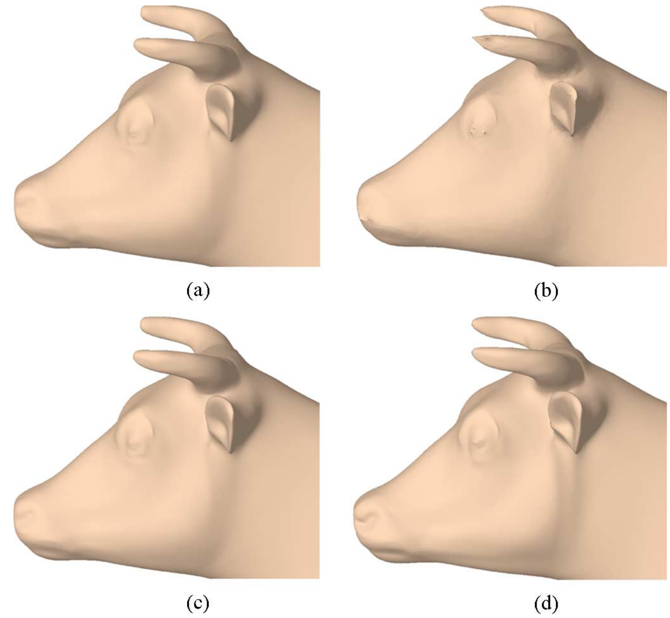

Fig. 11. Three-dimensional mesh smoothing results after zooming on the head of the 3-D cow model. (a) Smoothed by the geometric diffusion, (b) smoothed by bilateral mesh flow, (c) smoothed by the proposed approach. (d) Original model.

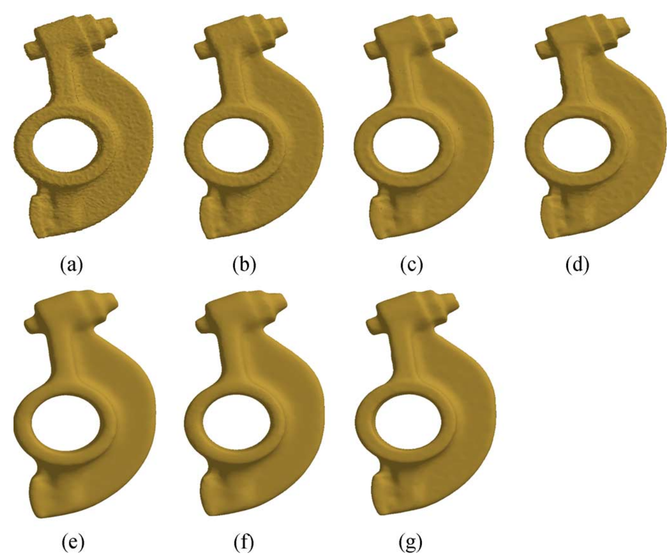

Fig. 12. Three-dimensional mesh smoothing results. (a) Noisy 3-D rocker arm model with 80,354 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by the geometric diffusion, (f) smoothed by bilateral mesh flow, and (g) smoothed by the proposed approach. The number of iteration times is 12 for each case.

in Fig. 9(a)–(c) (courtesy Stanford University, Avalon, and Cyberware). The standard deviation of the noise was set to 2% of the mean edge length, that is $\sigma = 0.02\,\bar{\ell}$, where $\bar{\ell}$ is given by (2). We also test the performance of these denoising techniques on an original noisy laser-scanned 3-D model (moai statue) shown in Fig. 9(d) (courtesy Max-Planck Institute).

A. *Qualitative Evaluation of the Proposed Method*

For ease of visualization, we designed a user-friendly Graphical User Interface (GUI) to test the performance of the proposed technique with different 3-D models, and to also perform a comparison with the most prevalent methods used in 3-D mesh smoothing. Fig. 10(a) depicts a noisy 3-D cow model, and Fig. 10(b)–(g) shows the denoising results using the weighted

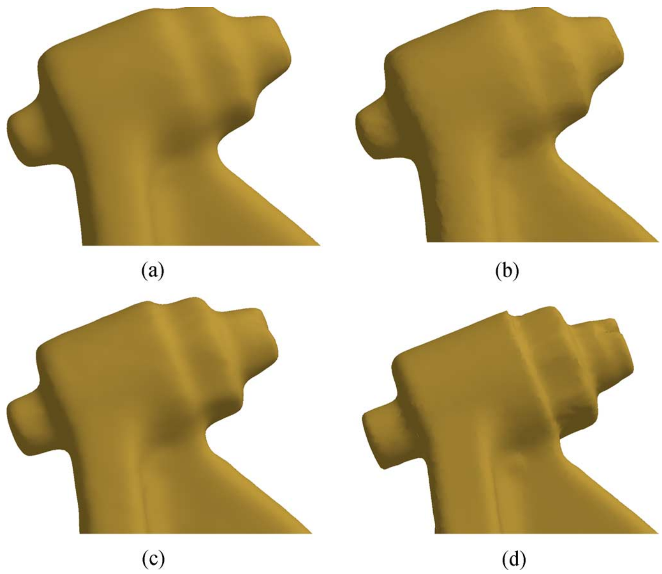

Fig. 13. Three-dimensional mesh smoothing results after zooming on the head of the 3-D rockerarm model. (a) Smoothed by geometric diffusion, (b) smoothed by bilateral mesh flow, and (c) smoothed by the proposed approach. (d) Original model.

Laplacian flow, mean filtering, angle median filtering, geometric diffusion with parameters $\lambda = 25$, $\epsilon = 0.06$ and $t = 3 \times 10^{-4}$, bilateral filtering, and the proposed method, respectively. These results clearly show that our method outperforms all the mesh filtering techniques used for comparison. Moreover, the proposed method is simple and easy to implement. One main advantage of the proposed algorithm is that it requires only few iterations to smooth out the noise, whereas the geometric diffusion, the mean and the angle median filters require substantial computational time. On the other hand, the bilateral filtering technique is also computationally fast, but has a poor smoothing performance in comparison with the proposed method as illustrated in Fig. 11, where we use the zoom tool to enlarge the view of the 3-D cow model's head in order to clearly show the better performance of our proposed algorithm. In particular, the geometric structures and the fine details around the eye and the ear of the 3-D cow model are very well preserved by our method. Note that the bilateral filter produces pointy horns, whereas the proposed method preserves the structure of the original horns pretty well. Also, note that the geometric diffusion produces slightly better results that the weighted Laplacian method, but also tends to smooth out some geometric features. More experimental results showing the better performance of the proposed algorithm are presented in Figs. 12–15.

In all the experiments, we observe that the proposed technique is able to suppress noise while preserving important geometric structure of the 3-D models in a very fast and efficient way. This better performance is in fact consistent with a large number of 3-D models used for experimentation.

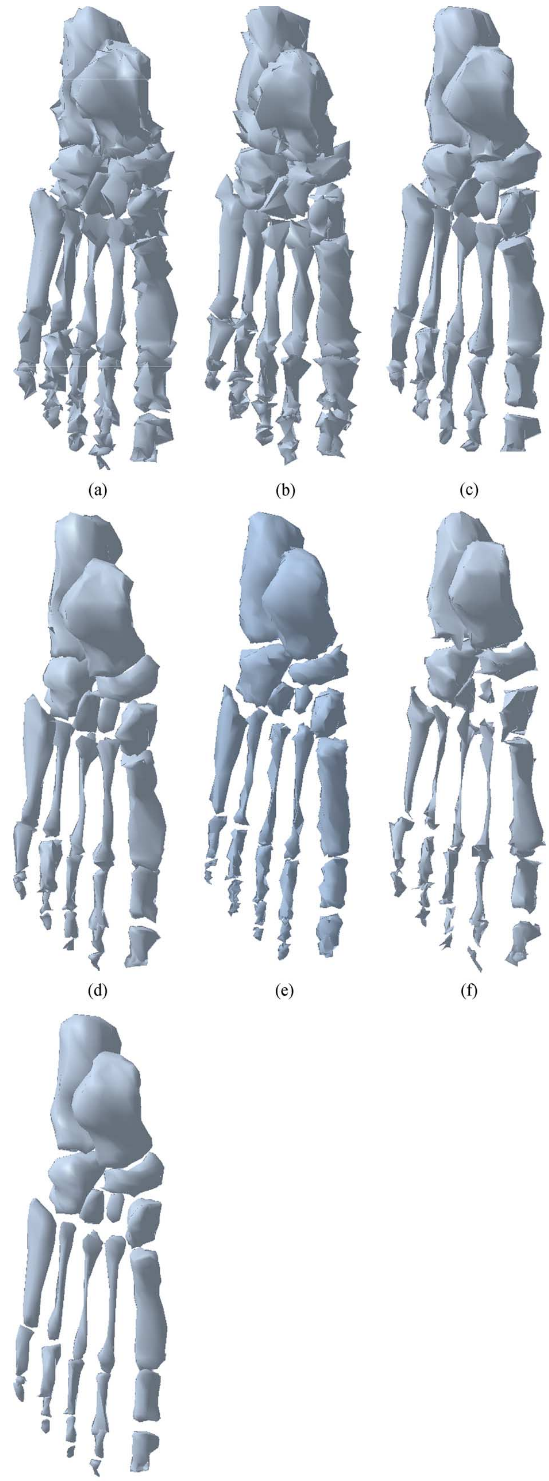

Fig. 14. Three-dimensional mesh smoothing results. (a) Noisy 3-D foot bones model with 4 204 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by the geometric diffusion, (f) smoothed by bilateral mesh flow, and (g) smoothed by the proposed approach. The number of iteration times is 3 for each case.

B. *Quantitative Evaluation of the Proposed Method*

Let $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ and $\widehat{\mathbb{M}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}}, \widehat{\mathcal{T}})$ be the original model and the smoothing result model with vertices $\mathcal{V} = \{\boldsymbol{v}_i\}_{i=1}^m$ and $\widehat{\mathcal{V}} = \{\hat{\boldsymbol{v}}_i\}_{i=1}^m$, edges $\mathcal{E} = \{e_{ij}\}$ and $\widehat{\mathcal{E}} = \{\hat{e}_{ij}\}$, and triangles $\mathcal{T} = \{\boldsymbol{t}_i\}_{i=1}^n$ and $\widehat{\mathcal{T}} = \{\hat{\boldsymbol{t}}_i\}_{i=1}^n$ respectively. To quantify the better performance of the proposed approach in comparison with the mean, angle median, weighted Laplacian, geometric diffusion, and bilateral filters, we computed the L^2 vertex-position and the L^2 face-normal error metrics [17].

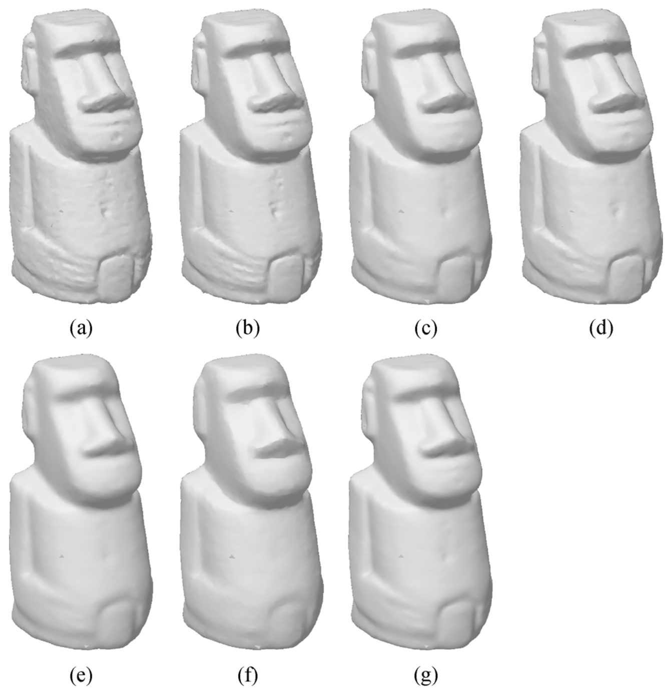

Fig. 15. Three-dimensional mesh smoothing results. (a) Noisy 3-D moai statue model with 19 985 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by the geometric diffusion, (f) smoothed by bilateral mesh flow, and (g) smoothed by the proposed approach. The number of iteration times is 5 for each case.

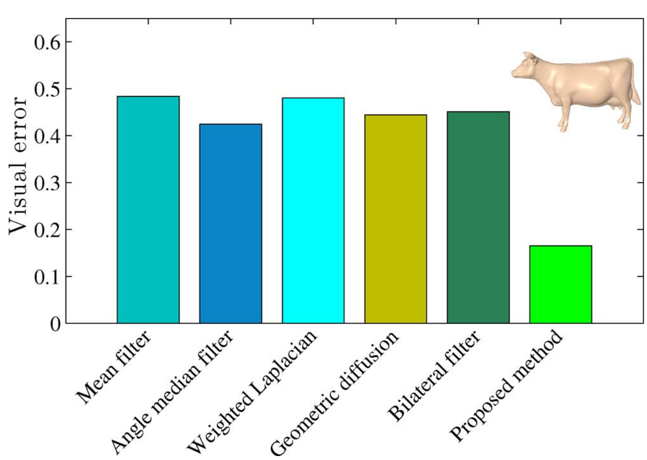

Fig. 16. Visual errors for the 3-D cow model.

The L^2 vertex-position error metric [17] is given by

$$E_1 = \frac{1}{3A(\widehat{\mathbb{M}})} \sum_{\hat{\boldsymbol{v}}_i \in \widehat{\mathcal{V}}} A(\hat{\boldsymbol{v}}_i) d(\hat{\boldsymbol{v}}_i, \mathbb{M})$$

where $A(\widehat{\mathbb{M}}) = \sum_{\hat{\boldsymbol{t}}_i \in \widehat{\mathcal{T}}} A(\hat{\boldsymbol{t}}_i)$, $A(\hat{\boldsymbol{v}}_i) = \sum_{\hat{\boldsymbol{t}}_j \in \widehat{\mathcal{T}}(\hat{\boldsymbol{v}}_i)} A(\hat{\boldsymbol{t}}_j)$, and $d(\hat{\boldsymbol{v}}_i, \mathbb{M})$ is the distance between $\hat{\boldsymbol{v}}_i$ and a triangle of $\mathbb{M}$ closest to $\hat{\boldsymbol{v}}_i$.

The L^2 face-normal error metric [17] is given by

$$E_2 = \frac{1}{A(\widehat{\mathbb{M}})} \sum_{\hat{\boldsymbol{t}}_i \in \widehat{\mathcal{T}}} A(\hat{\boldsymbol{t}}_i) \|\boldsymbol{n}(\boldsymbol{t}_i) - \boldsymbol{n}(\hat{\boldsymbol{t}}_i)\|$$

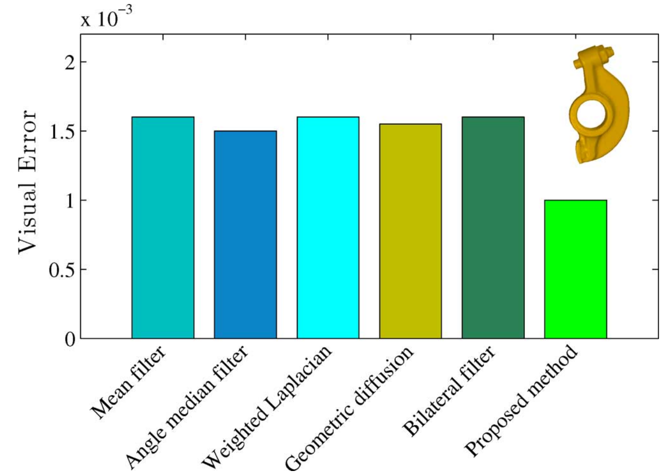

Fig. 17. Visual errors for the 3-D rocker arm model.

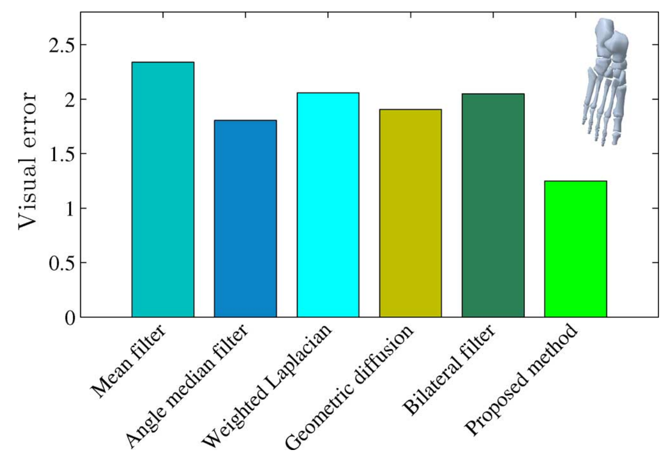

Fig. 18. Visual errors for the 3-D foot bones model.

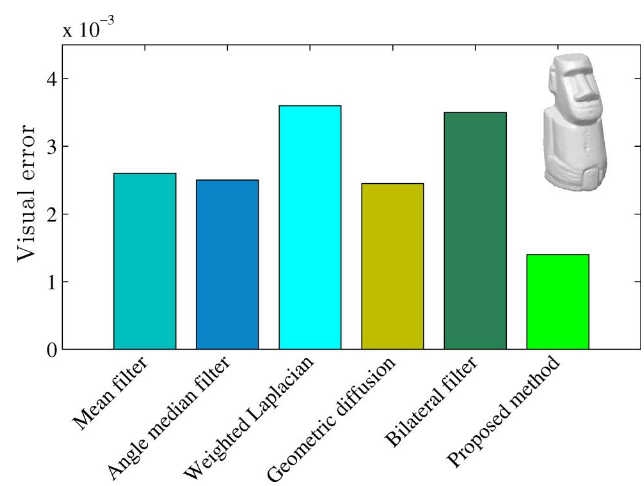

Fig. 19. Visual errors for the 3-D moai statue model.

where $\boldsymbol{n}(\boldsymbol{t}_i)$ and $\boldsymbol{n}(\hat{\boldsymbol{t}}_i)$ are the unit normals of $\boldsymbol{t}_i$ and $\hat{\boldsymbol{t}}_i$ respectively, and $A(\hat{\boldsymbol{t}}_i)$ is the area of $\hat{\boldsymbol{t}}_i$.

We also computed a visual error metric [30] given by

$$E_3 = \frac{1}{2m} \left(\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\|^2 + \sum_{i=1}^{m} \|\mathcal{I}(\boldsymbol{v}_i) - \mathcal{I}(\hat{\boldsymbol{v}}_i)\|^2 \right) \quad (12)$$

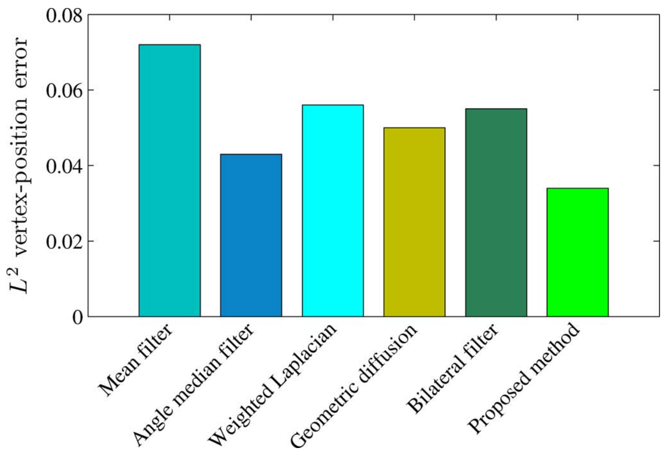

Fig. 20. L^2 vertex-position errors for the 3-D cow model.

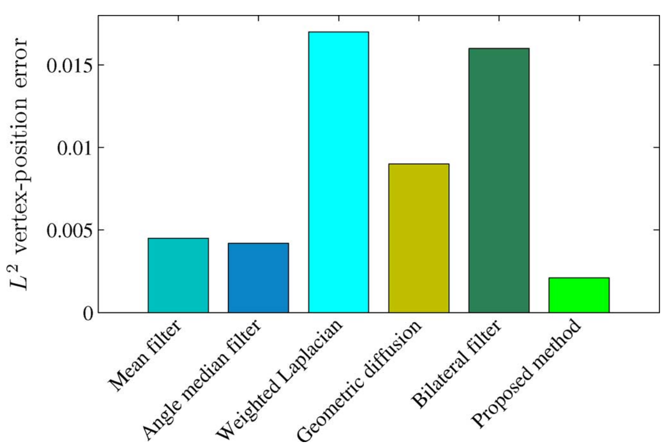

Fig. 21. L^2 vertex-position errors for the 3-D rocker arm model.

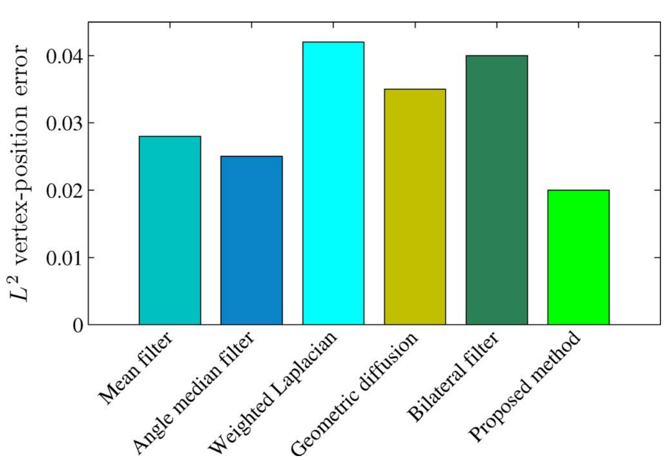

Fig. 22. L^2 vertex-position errors for the 3-D foot bones model.

where $\mathcal{I}$ is the geometric Laplacian operator defined as

$$\mathcal{I}(\boldsymbol{v}_i) = \boldsymbol{v}_i - \frac{1}{d_i} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \boldsymbol{v}_j.$$

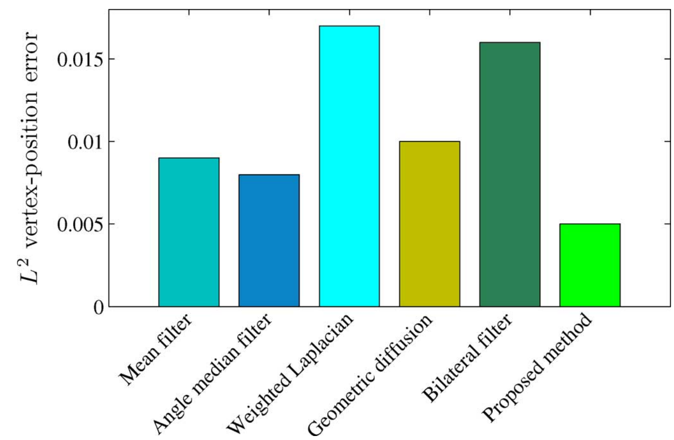

Fig. 23. L^2 vertex-position errors for the 3-D moai statue model.

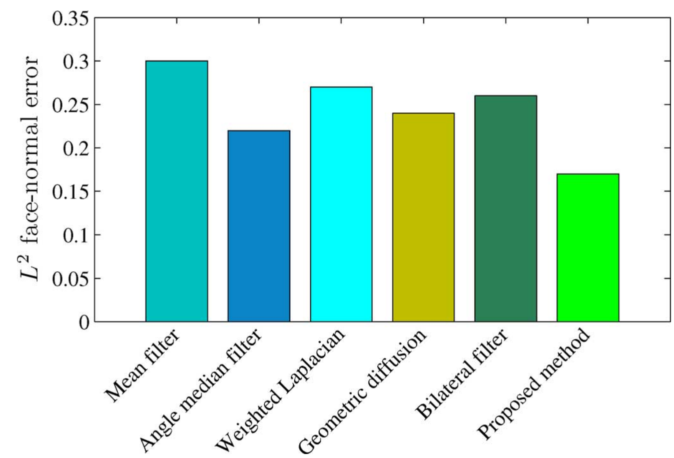

Fig. 24. L^2 face-normal errors for the 3-D cow model.

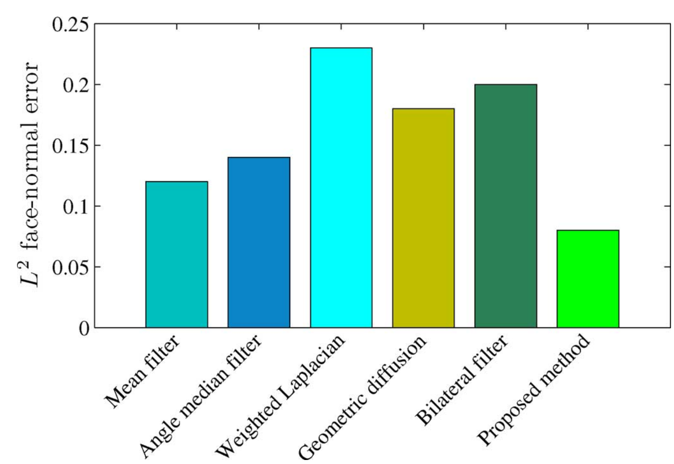

Fig. 25. L^2 face-normal errors for the 3-D rocker arm model.

The values of visual error metric for all the experiments are depicted in Figs. 16–27, which clearly show that the proposed method gives the best results indicating the consistency with the subjective comparison.

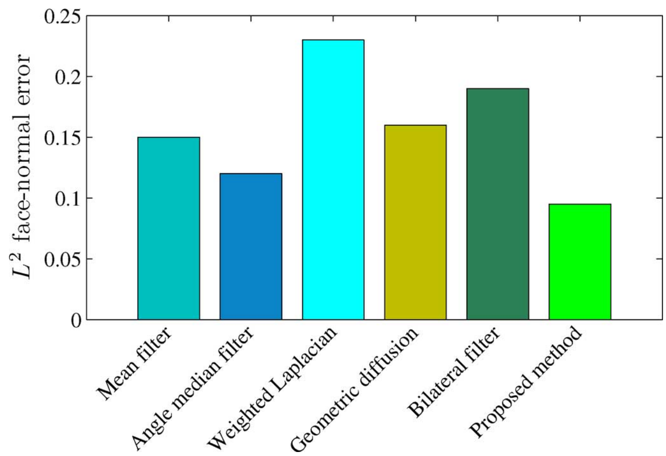

Fig. 26. L^2 face-normal errors for the 3-D foot bones model.

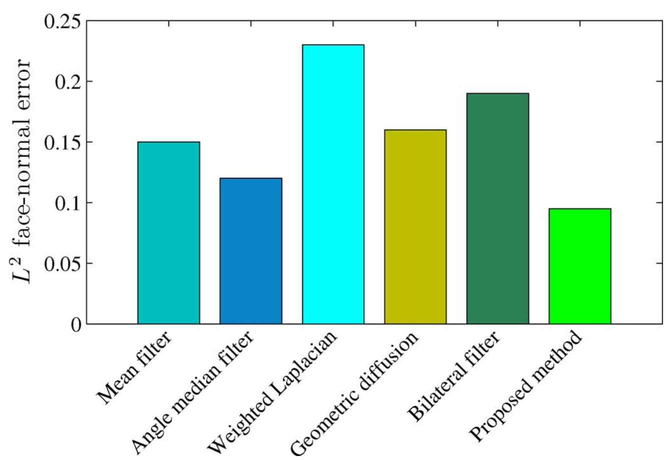

Fig. 27. L^2 face-normal errors for the 3-D moai statue model.

VI. Conclusion

In this paper, we introduced a vertex-based anisotropic diffusion for 3-D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3-D mesh smoothing strategy to fully preserve the geometric structure of the 3-D mesh data. The experimental results clearly showed a much improved performance of the proposed approach in comparison with the current methods used in 3-D mesh smoothing. For future work, we plan to incorporate the curvature information, as well as additional regularization terms into the proposed model.

Acknowledgment

The authors would like to thank the anonymous reviewers for helpful and very insightful comments.

References

[1] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
[2] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
[3] Y. L. You, W. Xu, A. Tannenbaum, and M. Kaveh, "Behavioral analysis of anisotropic diffusion in image processing," *IEEE Trans. Image Process.*, vol. 5, no. 11, pp. 1539–1553, Nov. 1996.
[4] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, Feb. 1997.
[5] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart, Germany: Teubner-Verlab, 1998.
[6] A. Yezzi, "Modified curvature motion for image smoothing and enhancement," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 345–352, Mar. 1998.
[7] P. Kornprobst, R. Deriche, and G. Aubert, "Image sequence analysis via partial differential equations," *J. Math. Imag. Vis.*, vol. 11, no. 1, pp. 5–26, 1999.
[8] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "A variational model for image classification and restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 460–472, May 2000.
[9] M. Cetin and W. C. Karl, "Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 623–631, Apr. 2001.
[10] Y. Bao and H. Krim, "Smart nonlinear diffusion: A probabilistic approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 63–72, Jan. 2004.
[11] M. Giaquinta and S. Hildebrandt, *Calculus of Variations I: The Lagrangian Formalism*. New York: Springer-Verlag, 1996.
[12] G. Taubin, "A signal processing approach to fair surface design," in *Proc. SIGGRAPH*, 1995, pp. 351–358.
[13] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, "Polyhedral surface smoothing with simultaneous mesh regularization," in *Geometric Modeling and Processing Conf.*, Hong Kong, Apr. 2000, pp. 229–237.
[14] G. Taubin, "Linear Anisotropic Mesh Filtering," Res. Rep. RC2213 IBM, 2001.
[15] S. Petitjean, "A survey of methods for recovering quadrics in triangle meshes," *ACM Comput. Surv.*, vol. 34, no. 2, pp. 211–262, 2002.
[16] Y. Shen and K. E. Barner, "Fuzzy vector median-based surface smoothing," *IEEE Trans. Vis. Comput. Graph.*, vol. 10, no. 3, pp. 252–265, Mar. 2004.
[17] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh smoothing via mean and median filtering applied to face normals," in *Proc. Geometric Modeling and Processing*, 2002, pp. 124–131.
[18] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *Proc. ACM SIGGRAPH*, 2003, pp. 950–953.
[19] T. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature preserving mesh smoothing," in *Proc. SIGGRAPH*, 2003, pp. 943–949.
[20] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. SIGGRAPH*, 1999, pp. 317–324.
[21] M. Desbrun, M. Meyer, P. Schröder, and A. Bar, "Anisotropic feature-preserving denoising of height fields and bivariate data," *Graph. Interface*, pp. 145–152, 2000.
[22] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," presented at the IEEE Visualization, 2000.
[23] ——, "Processing textured surfaces via anisotropic geometric diffusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 248–261, Feb. 2004.
[24] C. L. Bajaj and G. Xu, "Anisotropic diffusion of subdivision surfaces and functions on surfaces," *ACM Trans. Graphics*, vol. 22, no. 1, pp. 4–32, 2003.
[25] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface smoothing via anisotropic diffusion of normals," in *Proc. IEEE Visualization*, 2002, pp. 125–132.
[26] ——, "Geometric surface processing via normal maps," *ACM Trans. Graphics*, vol. 22, no. 4, pp. 1012–1033, 2003.
[27] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Comput. Graph. Forum*, vol. 23, no. 3, 2004.
[28] Y. Zhang and A. Ben Hamza, "PDE-based smoothing for 3D mesh quality improvement," presented at the IEEE Int. Conf. Electro/Information Technology, 2006.
[29] F. R. Chung, *Spectral Graph Theory*. Providence, RI: Amer. Math. Soc., 1997.
[30] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proc. SIGGRAPH*, 2000, pp. 279–286.
[31] W. J. Rey, *Introduction to Robust and Quasi-Robust Statistical Methods*. Berlin, Germany: Springer, 1983.

Ying Zhang received the B.Eng. degree in system engineering from Tianjin University, China. She is currently pursuing the M.A.Sc. degree in electrical and computer engineering at Concordia University, Montréal, QC, Canada.

From 1999 to 2003, she was a Programmer and System Analyst at Tianjin Tiancai. She is currently a Software Designer at Engenuity Technologies, Inc. Her research interests are computer graphics and artificial intelligence.

A. Ben Hamza received the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, in 2003, where he worked on computational imaging, 3-D object recognition, and information theory.

He is currently an Assistant Professor in the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montréal, QC, Canada. Prior to joining CIISE, he was a Postdoctoral Research Associate at Duke University, Durham, NC, affiliated with both the Department of Electrical and Computer Engineering and the Fitzpatrick Center for Photonics and Communications Systems. His current research interests include multivariate process control, computer graphics, multimedia security, and multisensor data processing.