SIViP (2011) 5:191–201
DOI 10.1007/s11760-010-0154-z

ORIGINAL PAPER

Multivariate kernel diffusion for surface denoising

Khaled Tarmissi · A. Ben Hamza

Received: 28 April 2009 / Revised: 10 January 2010 / Accepted: 11 January 2010 / Published online: 27 January 2010

Abstract In this paper, we introduce a 3D mesh denoising method based on kernel density estimation. The proposed approach is able to reduce the over-smoothing effect and effectively remove undesirable noise while preserving prominent geometric features of a 3D mesh such as curved surface regions, sharp edges, and fine details. The experimental results demonstrate the effectiveness of the proposed approach in comparison to existing mesh denoising techniques.

Keywords Mesh denoising · Kernel density estimation · Anisotropic diffusion

1 Introduction

Recent advances in computer and information technology have increased the use of 3D models in many fields including medicine, the media, art and entertainment. With the increasing use of 3D scanners to create 3D models, which are usually represented as triangle meshes, there is a rising need for robust mesh denoising techniques to remove inevitable noise in the measurements. Even with high-fidelity scanners, the acquired 3D models are usually contaminated by noise, and therefore a reliable mesh denoising technique is often required.

In recent years, a variety of techniques have been proposed to tackle the 3D mesh denoising problem [1–5]. The most commonly used mesh denoising method is the so-called Laplacian flow which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices [1]. 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 [6–8] which are all adopted from the image processing literature. Also, a number of anisotropic diffusion methods for triangle meshes and implicit surfaces have been proposed recently. Desbrun et al. [9,10] 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, re-computing new edge weights after each iteration results in more expensive computational cost. Clarenz et al. [11] 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 [12] 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. [13] present a unified anisotropic diffusion for 3D mesh smoothing by treating discrete surface data as a discretized version of a 2D Riemannian manifold and establishing a partial differential equation (PDE) diffusion model for such a manifold. This method helps enhancing sharp features while filtering out noise by considering 3-ring neighbors of each vertex to achieve non-linear approach of smoothing process. Tasdizen et al. [14,15] introduce a two-step surface smoothing method by solving a set of coupled

K. Tarmissi · A. Ben Hamza (✉)
Concordia Institute for Information Systems Engineering,
Concordia University, Montréal, QC, Canada
e-mail: hamza@ciise.concordia.ca

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 [16], 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. Peng et al. [17] have successfully applied locally adaptive Wiener filtering to 3D meshes. Delouille et al. [18] proposed wavelet-based approaches to denoise a signal defined on an irregular bivariate grid that represent the denoised data in the wavelet domain by a few scaling coefficients present at the coarsest scale together with the detail coefficients. Another multiscale approach is proposed in Le Faucheur et al. [19], where the authors presented a Bayesian shrinkage framework for spherical wavelets with interscale dependency and intrascale smoothing considerations and showed how local consistency can help outperform uniform shrinkage rules. El Ouafdi et al. [20] proposed a stochastic diffusion-based approach for mesh denoising using normalized transition probability and diffusion tensor.

Roughly speaking, mesh denoising techniques can be defined as the requirement to adjust vertex positions without changing the connectivity of the 3D mesh, and may be classified into two main categories: one-step or two-step approaches. The one-step approaches directly update vertex positions using the original vertex coordinates and a neighborhood around the current vertex, and sometimes face normals too. On the other hand, the two-step approaches first adjust face normals and then update vertex positions using some error minimization criterion based on the adjusted normals. In many cases, a single pass of a one-step or two-step approach does not yield a satisfactory result, and therefore iterated operations are performed. In this paper, we present a 3D mesh denoising method based on kernel density estimation. A preliminary work on this approach was presented in [21]. The proposed technique falls into the category of one-step approaches. The main idea is to use Laplacian smoothing algorithm combined with Gaussian kernel density estimators in order to reduce the over-smoothing problem and remove the noise effectively while preserving the nonlinear features of the 3D mesh such as curved surface regions, sharp edges, and fine details.

The rest of this paper is organized as follows. In the next section, we briefly recall some basic concepts of 3D mesh data, and we introduce the vertex differential operators, then a general formulation of 3D mesh denoising problem is stated. In Sect. 3, a kernel-based nonlinear diffusion is introduced. In Sect. 4, we provide experimental results to demonstrate a much improved performance of the proposed method in 3D mesh denoising. Finally, we conclude and point out some future work directions in Sect. 5.

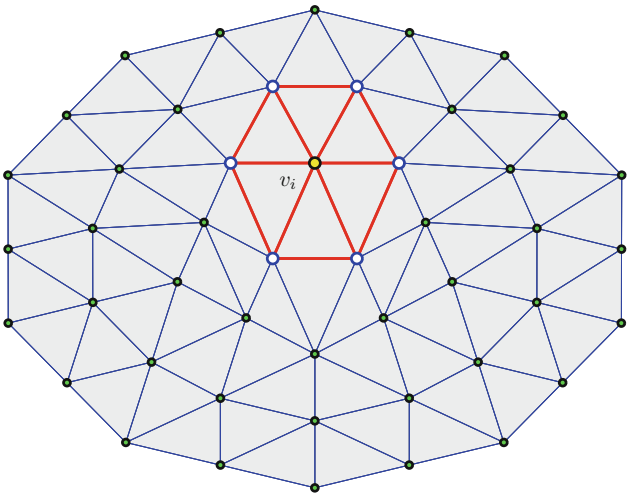

Fig. 1 Illustration of vertex neighborhood $\boldsymbol{v}_i^\star$

2 Problem formulation

In computer graphics and geometric-aided design, triangle meshes have become the de facto standard representation of 3D objects. A triangle mesh $\mathbb{M}$ may be defined as $\mathbb{M} = (\mathcal{V}, \mathcal{E})$ or $\mathbb{M} = (\mathcal{V}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, 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 (denoted by $\boldsymbol{v}_i \sim \boldsymbol{v}_j$ or simply $i \sim j$) if they are connected by an edge, i.e. $e_{ij} \in \mathcal{E}$. 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 $\boldsymbol{v}_i^\star$. The highlighted red-colored sub-mesh in Fig. 1 displays a vertex neighborhood $\boldsymbol{v}_i^\star$, where the degree d_i of the vertex $\boldsymbol{v}_i$ is equal to 6.

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

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

where $\|e_{ij}\| = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|$, and $|\mathcal{E}|$ is the cardinality of the set of edges.

2.1 Laplacian matrix of a triangle mesh

The Laplacian matrix of a triangle mesh $\mathbb{M}$ 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.

The normalized Laplacian matrix $\mathcal{L}$ is given by [23]

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

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

$$\mathcal{L}\varphi(\boldsymbol{v}_i) = -\sum_{j\sim i} \frac{1}{\sqrt{d_i}} \left(\frac{\varphi(\boldsymbol{v}_j)}{\sqrt{d_j}} - \frac{\varphi(\boldsymbol{v}_i)}{\sqrt{d_i}} \right), \quad \forall \boldsymbol{v}_i \in \mathcal{V}. \tag{3}$$

2.2 Vertex differential operators

We define the vertex gradient operator as

$$\nabla \boldsymbol{v}_i = \left\{ \frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} : \boldsymbol{v}_j \in \boldsymbol{v}_i^\star \right\}. \tag{4}$$

We also define the vertex Laplace operator as

$$\Delta \boldsymbol{v}_i = \sum_{j\sim i} \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right). \tag{5}$$

Note the analogy between the vertex Laplace operator and the normalized Laplacian matrix $\mathcal{L}$ defined as an operator.

2.3 Mesh denoising model

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

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\eta}, \tag{6}$$

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 3D model contaminated by noise. The challenge of the problem of interest lies in recovering the vertex $\boldsymbol{u}$ from the observed vertex $\boldsymbol{v}$, and furthering the estimation by making use of any prior knowledge/assumptions about the noise process $\boldsymbol{\eta}$, as well as the unknown original mesh.

Generally, 3D mesh denoising methods may be classified into two major categories: isotropic and anisotropic. The former techniques filter the noisy data independently of direction [1,3], while the latter methods modify the diffusion equation to make it nonlinear or anisotropic in order to preserve the sharp features of a 3D mesh [16]. Most of these nonlinear methods were inspired by anisotropic-type diffusions in the image processing literature.

3 Proposed mesh denoising approach

The proposed method is inspired by the good performance of anisotropic diffusion in image denoising. In [22], we defined a vertex-based anisotropic diffusion as follows

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

where g is a redescending function of the vertex gradient magnitude. This function is chosen to allow more smoothing in homogenous regions of 3D mesh, and less smoothing around sharp features. That is, the function g satisfies $g(x) \to 0$ when $x \to \infty$ so that the diffusion is "stopped" across sharp details of the mesh. More specifically, the smoothing effect of a vertex-based anisotropic diffusion may be explained as follows: in flat regions of a 3D mesh where the vertex gradient magnitudes are relatively small, Eq. (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 3D mesh where the vertex gradient magnitudes are large, the diffusion flow given by Eq. (7) tends to smooth less and hence leads to a much better preservation of the mesh geometric structures. Note that if g is a tensor, then the approach given Eq. (7) is anisotropic, whereas if g is a function then the approach would be non-homogeneous.

In discrete form, it can be easily shown that the vertex-based anisotropic diffusion may be reduced to the following update rule

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{j\sim i} \varphi(\boldsymbol{v}_i - \boldsymbol{v}_j) \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right), \tag{8}$$

where the φ-function is defined as $\varphi(\boldsymbol{v}_i - \boldsymbol{v}_j) = g(|\nabla \boldsymbol{v}_i|) + g(|\nabla \boldsymbol{v}_j|)$, and the gradient magnitudes are given by

$$|\nabla \boldsymbol{v}_i| = \left(\sum_{j\sim i} \left\| \boldsymbol{v}_i/\sqrt{d_i} - \boldsymbol{v}_j/\sqrt{d_j} \right\|^2 \right)^{1/2}, \tag{9}$$

and

$$|\nabla \boldsymbol{v}_j| = \left(\sum_{k\sim j} \left\| \boldsymbol{v}_j/\sqrt{d_j} - \boldsymbol{v}_k/\sqrt{d_k} \right\|^2 \right)^{1/2}. \tag{10}$$

Kernel density estimates are output as smooth curves with the amount of smoothing governed by a bandwidth value used during calculation [24]. Densities are calculated by placing kernels over the distribution of data points. Kernels that overlap one another increase density values in shared areas of the distribution. For univariate and multivariate data, the Gaussian kernel is the most commonly used one. In particular, for 3D data, the standardized Gaussian kernel function (see Fig. 2) is given by

$$K(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{3}{2}}} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2} \right), \quad \forall \boldsymbol{x} \in \mathbb{R}^3. \tag{11}$$

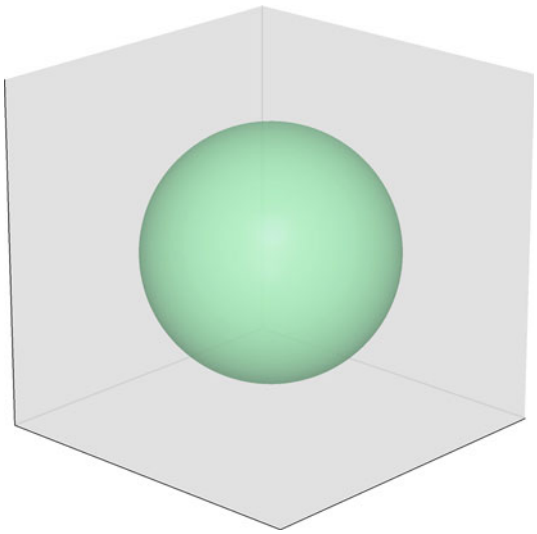

Fig. 2 3D level surface of the Gaussian kernel function $K(\boldsymbol{x})$

Motivated by kernel density estimation as an important data analytic tool that provides a very effective way of showing structure in a set of a data [24], we propose a mesh kernel flow. This mesh denoising flows updates iteratively each mesh vertex according to the following rule

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{j\sim i} \mathcal{K}_{H_i}(\boldsymbol{v}_i - \boldsymbol{v}_j) \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right) \tag{12}$$

where the φ-function is given by

$$\mathcal{K}_{H_i}(\boldsymbol{v}_i - \boldsymbol{v}_j) = \det(H_i)^{-1/2} K\left(H_i^{-1/2}(\boldsymbol{v}_i - \boldsymbol{v}_j) \right) \tag{13}$$

and H_i is a symmetric positive semi-definite matrix. This matrix defines a covariance matrix around the neighborhood

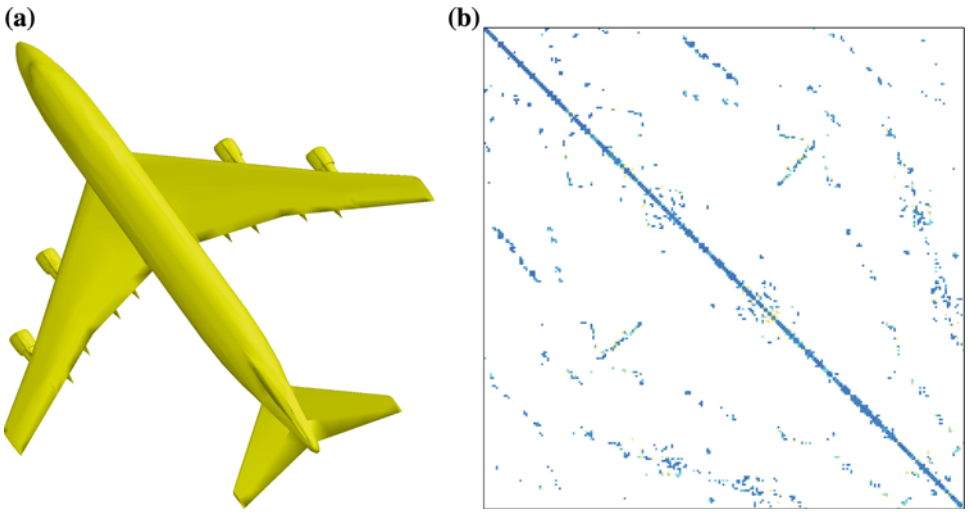

Fig. 3 **a** 3D object with $m = 3403$ vertices and its **b** mesh neighborhood weighting kernel matrix

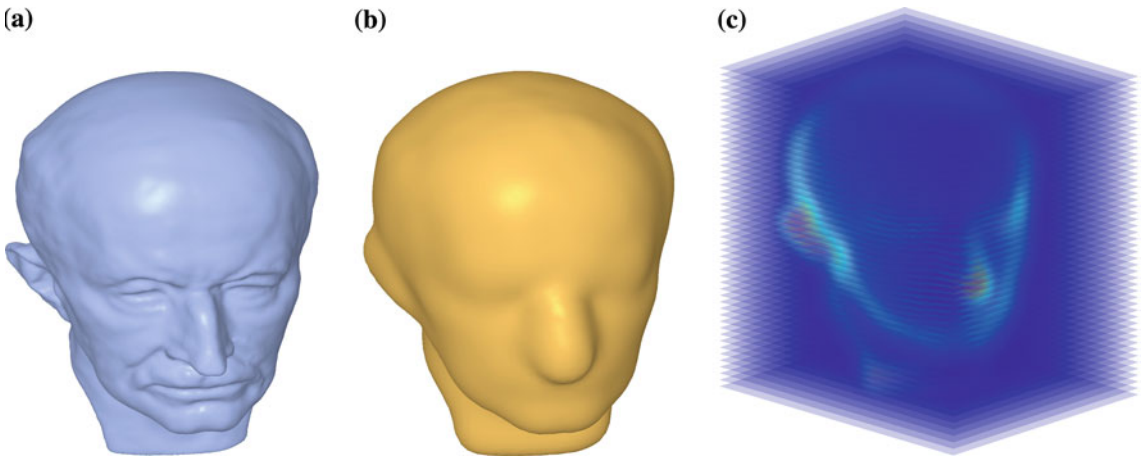

Fig. 4 **a** 3D object, **b** mesh KDE, and **c** horizontal slices of the mesh KDE

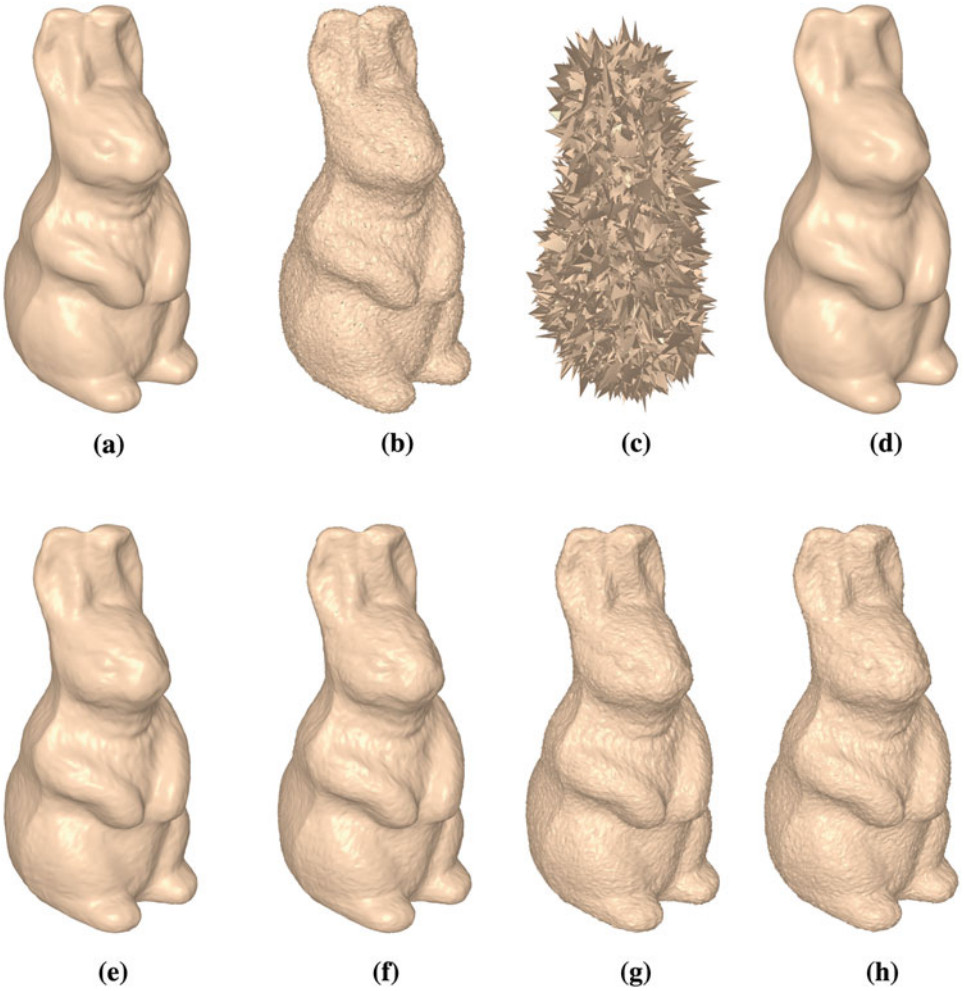

Fig. 5 Output results of our proposed mesh denoising approach for different values of the regularization parameter. The number of iterations is set to 5: **a** original model, **b** noisy model, **c** $\lambda = 0.1$, **d** $\lambda = 0.4$, **e** $\lambda = 0.5$, **f** $\lambda = 0.6$, **g** $\lambda = 0.75$, **h** $\lambda = 0.9$

of vertex v_i, and it is given by

$$H_i = \sum_{j\sim i}(\boldsymbol{v}_j - \boldsymbol{c}_i)(\boldsymbol{v}_j - \boldsymbol{c}_i)^T, \quad \text{where } \boldsymbol{c}_i = \frac{1}{d_i}\sum_{j\sim i}\boldsymbol{v}_j. \tag{14}$$

It is worth pointing out that H_i is also called the bandwidth matrix in the context of kernel smoothing and it measures the amount of smoothing. Also, note that the choice of the kernel function appears to have very little effect on the quality of the proposed denoising approach. However, the selection of the bandwidth matrix is widely recognized to have more effect on the performance of kernel density estimation. In this paper, we use the trivariate Gaussian density as a kernel function, and the data-driven neighborhood covariance as a bandwidth matrix. Note that the use of kernel density estimation in our proposed approach is motivated by the use of a Gaussian diffusion function in nonlinear diffusion, where the flow is maximum when the gradient magnitude is close to the standard deviation. Specifically, the bandwidth matrix in our approach is analogous to the standard deviation of the Gaussian diffusion function in nonlinear diffusion. In other words, the bandwidth matrix essentially drives the diffusion behavior in Eq. (12).

The neighborhood weighting kernel $\mathcal{K}_{H_i}$ may be expressed in matrix form as $\mathcal{K} = (\kappa_{ij})$, which will be referred to as mesh neighborhood weighting kernel matrix. Each element κ_{ij} of this $m \times m$ sparse matrix is given by the right-hand side of Eq. (13). Thus, the mesh neighborhood weighting kernel matrix may be written as

$$\mathcal{K} = (\kappa_{ij}) = \begin{cases} \det(H_i)^{-1/2}(2\pi)^{-3/2} & \text{if } \boldsymbol{v}_i = \boldsymbol{v}_j \\ \det(H_i)^{-1/2}\, K\left(H_i^{-1/2}(\boldsymbol{v}_i - \boldsymbol{v}_j)\right) & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{o.w.} \end{cases} \tag{15}$$

Note that the value of the the first row of Eq. (15) results directly from Eq. (11) when $\boldsymbol{x} = 0$, that is $K(\mathbf{0}) = (2\pi)^{-3/2}$. Therefore, like many iterative methods the update flow of the proposed approach can be easily implemented in terms of matrix-vector products. Figure 3 displays a 3D object and its mesh neighborhood weighting kernel matrix.

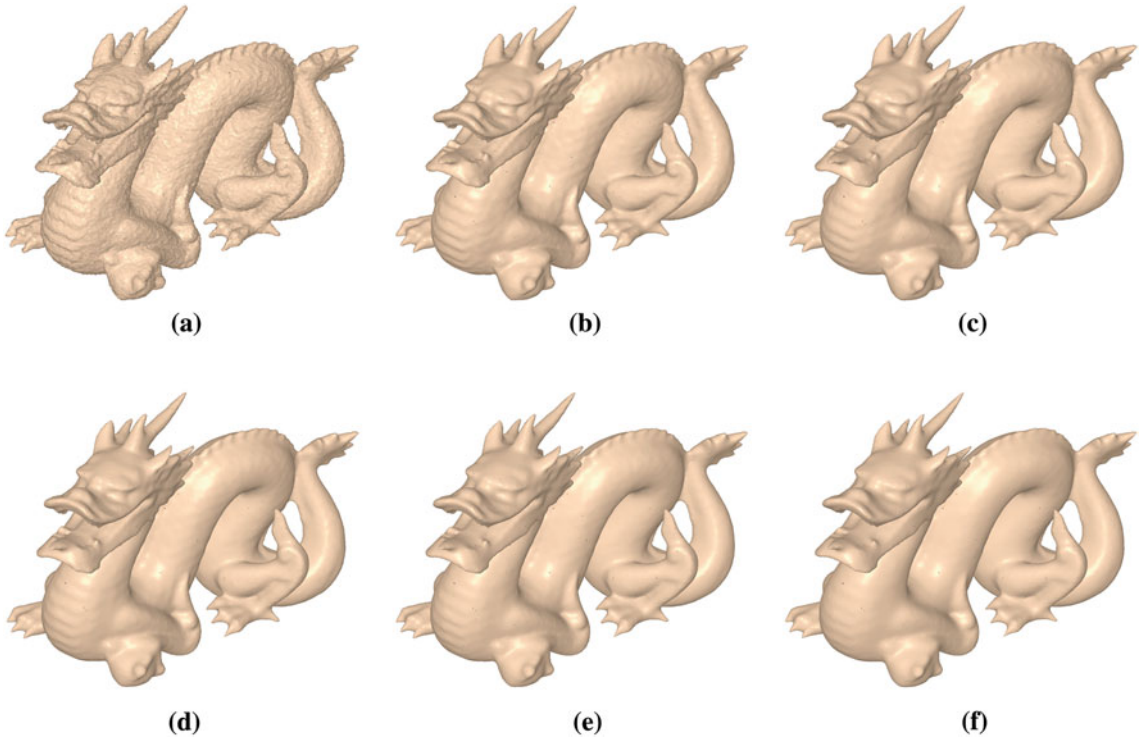

Fig. 6 Output results of our proposed mesh denoising approach at different iteration numbers. The regularization parameter is set to $\lambda = 0.8$: **a** noisy model, **b** 3 iterations, **c** 5 iterations, **d** 6 iterations, **e** 8 iterations, **f** 12 iterations

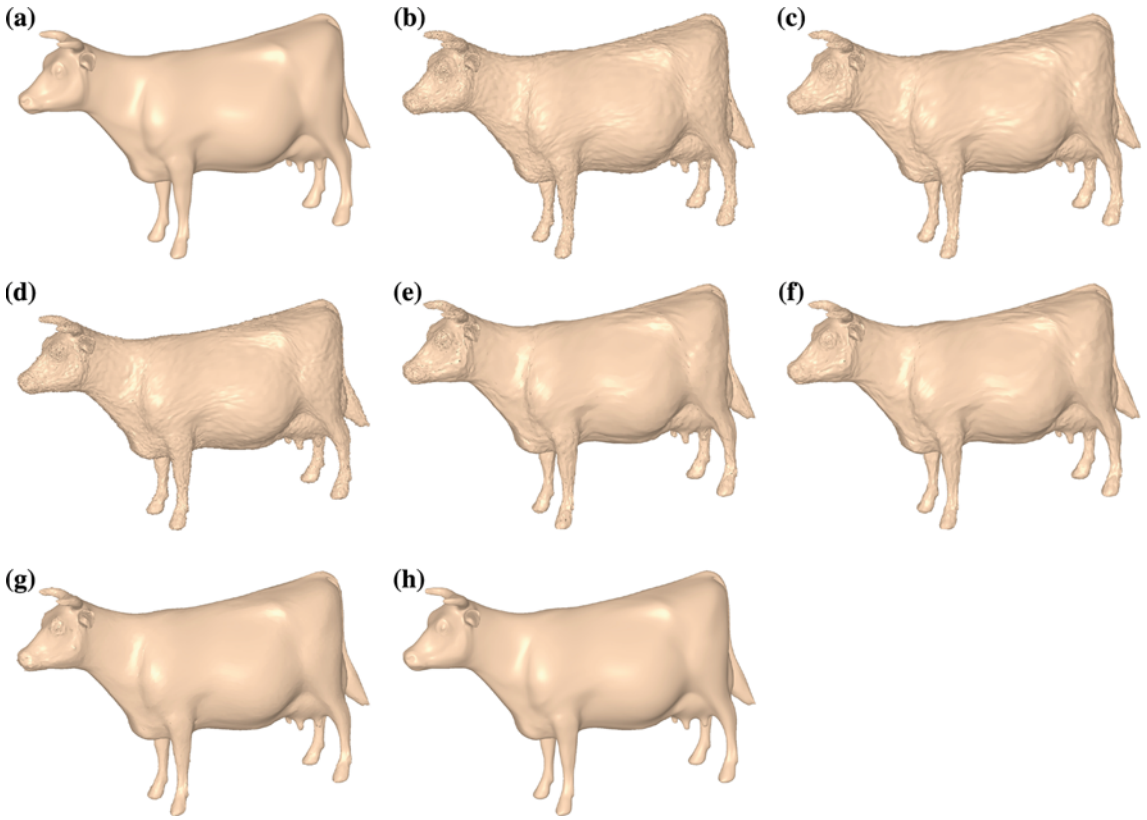

Fig. 7 Denoising results for the 3D cow model: **a** original model, **b** noisy model, **c** Laplacian flow, **d** weighted Laplacian flow, **e** mean filtering, **f** angle median filtering, **g** bilateral mesh flow, **h** our proposed approach. The number of iterations is set to 6 in each case

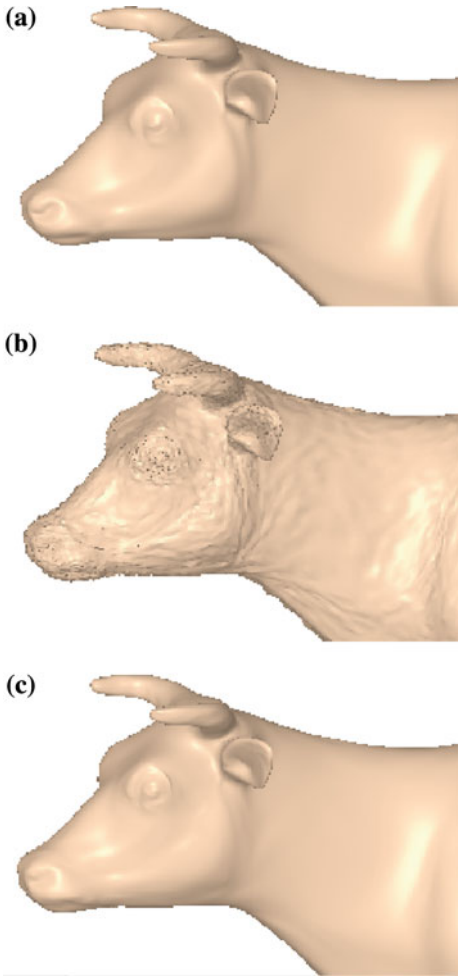

Fig. 8 Zoomed portion of the 3D cow model: **a** original model, **b** noisy model, **c** output result of our proposed approach with $\lambda = 0.8$. The number of iterations is set to 6

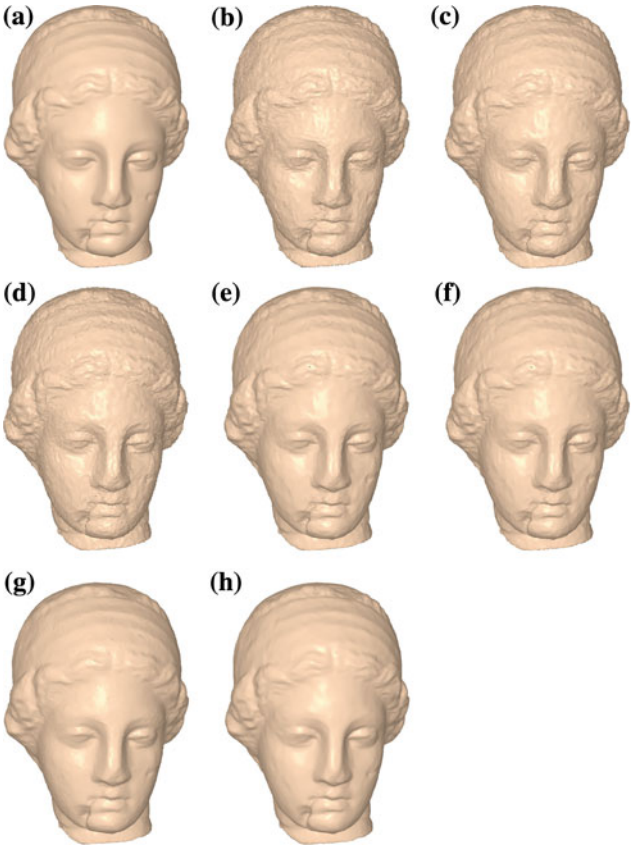

Fig. 9 Denoising results for the 3D igea model: **a** original model, **b** noisy model, **c** Laplacian flow, **d** weighted Laplacian flow, **e** mean filtering, **f** angle median filtering, **g** bilateral mesh flow, **h** our proposed approach. The number of iterations is set to 6

Next we show how kernel density estimation can be used for 3D mesh reconstruction. Given m mesh vertices $\boldsymbol{v}_i$, let $\boldsymbol{v}$ be a 3D vector whose ith realization is $\boldsymbol{v}_i$. Thus, the mesh kernel density estimate (KDE) may be written in the general form

$$\hat{f}(\boldsymbol{v}) = \frac{1}{m}\sum_{i=1}^{m} \det(H)^{-1/2}\, K\left(H^{-1/2}(\boldsymbol{v}-\boldsymbol{v}_i)\right) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{K}_H(\boldsymbol{v}-\boldsymbol{v}_i), \tag{16}$$

where $H = \sum_{i=1}^{m}(\boldsymbol{v}_i - \boldsymbol{c})(\boldsymbol{v}_i - \boldsymbol{c})^T$ is the mesh covariance matrix which controls the smoothness of the resulting density estimate, and $\boldsymbol{c} = (1/m)\sum_{i=1}^{m}\boldsymbol{v}_i$ is the mesh centroid. The mesh KDE is a trivariate volumetric function which may be graphically visualized by plotting a level surface (also called implicit surface or isosurface) of $\hat{f}$ as shown in Fig. 4b. This figure displays an isosurface of the mesh KDE using the vertices of the 3D object shown in Fig. 4a. The horizontal slices of the mesh KDE are also depicted in Fig. 4c.

4 Experimental results

This section presents experimental results where the mean filtering [6], angle median filtering [6], Laplacian flow [1], weighted Laplacian flow [9,10], geometric diffusion [11], bilateral filtering [7], and the proposed method are applied to noisy 3D models obtained by adding Gaussian noise to the original 3D models. 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 Eq. (1).

In practical applications, the covariance matrix H_i given by Eq. (14) may become singular. To circumvent this singularity problem and also to ensure the stability of the proposed algorithm, we use a regularized covariance matrix as follows

$$H_i = H_i + \lambda\, I \tag{17}$$

where I is a 3×3 identity matrix and λ is a positive regularization parameter.

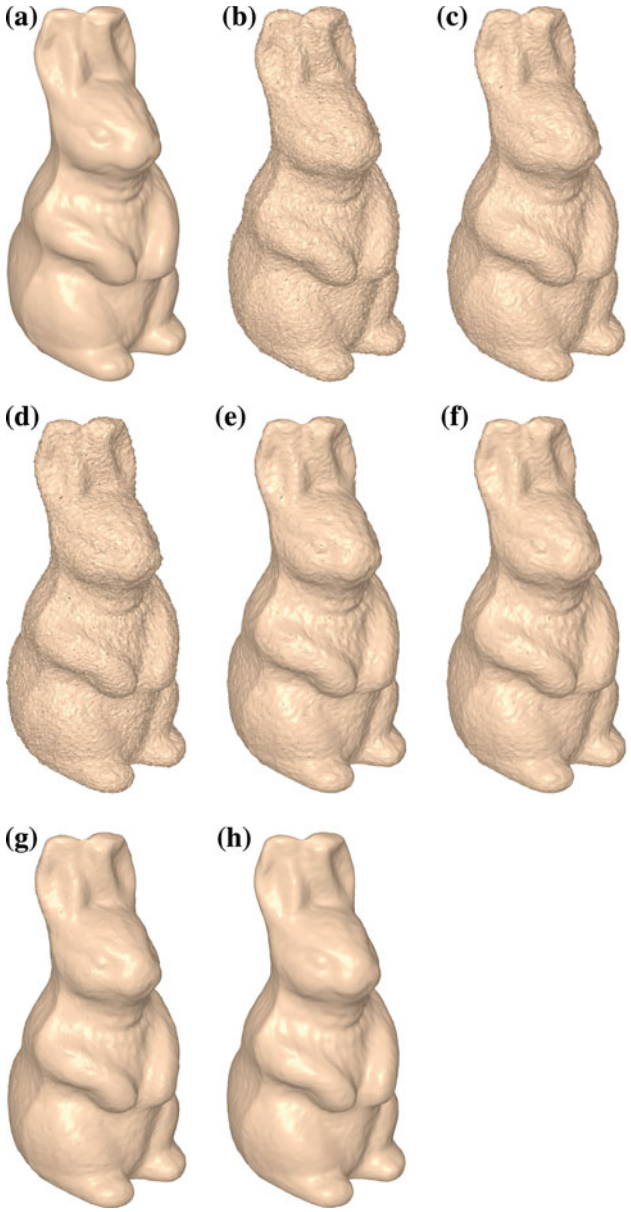

Fig. 10 Denoising results for the 3D rabbit model: **a** original model, **b** noisy model, **c** Laplacian flow, **d** weighted Laplacian flow, **e** mean filtering, **f** angle median filtering, **g** bilateral mesh flow, **h** our proposed approach. The number of iterations is set to 3

Figure 5 displays the mesh denoising results obtained by our proposed method for different values of the regularization parameter λ, where the number of iterations was to set to 5. As can be seen in Fig. 5, the value $\lambda = 0.4$ gives the best denoising result for the 3D rabbit model. And as a the value of λ increases, the rabbit model becomes much more noisier. Also, we noticed through experimentation that a smaller value of λ often tends to produce a geometrically distorted shape of the 3D object as shown in Fig. 5c. Therefore, the regularization parameter should be tuned to be small enough to capture the intrinsic shape of a 3D object and large enough not to recapture noise.

Figure 6 depicts the output results of the proposed approach at different iteration numbers. These results show that, using the proposed approach, the noise can be removed with just a small number of iterations and that the sharp features are well preserved when the regularization parameter is appropriately chosen.

4.1 Qualitative evaluation of the proposed method

Figure 7c–h shows the denoising results obtained via Laplacian flow, weighted Laplacian flow, mean filtering, angle median filtering, 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 weighted Laplacian flow, the mean and the angle median filters require substantial computational time.In Fig. 8, we use the zoom tool to enlarge the view of the 3D 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 3D cow model are very well preserved by our method. More experimental results showing the better performance of the proposed algorithm are presented in Figs. 9 and 10.

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

4.2 Quantitative evaluation of the proposed method

Let $\mathbb{M}$ and $\widehat{\mathbb{M}}$ be the original model and the smoothing result model with vertex sets $\mathcal{V} = \{\boldsymbol{v}_i\}_{i=1}^m$ and $\widehat{\mathcal{V}} = \{\hat{\boldsymbol{v}}_i\}_{i=1}^m$ respectively. To quantify the performance of the proposed approach, we computed the visual error metric [25] given by

$$E = \frac{1}{2m}\left(\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\|^2 + \sum_{i=1}^{m} \|\mathcal{I}(\boldsymbol{v}_i) - \mathcal{I}(\hat{\boldsymbol{v}}_i)\|^2\right), \quad (18)$$

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

$$\mathcal{I}(\boldsymbol{v}_i) = \boldsymbol{v}_i - \frac{1}{d_i}\sum_{j\sim i} \boldsymbol{v}_j. \quad (19)$$

Intuitively, the error metric given by Eq. (18) captures the visual difference between the original model and the denoised one by taking into account geometric closeness and local smoothness difference. More specifically, the first term of

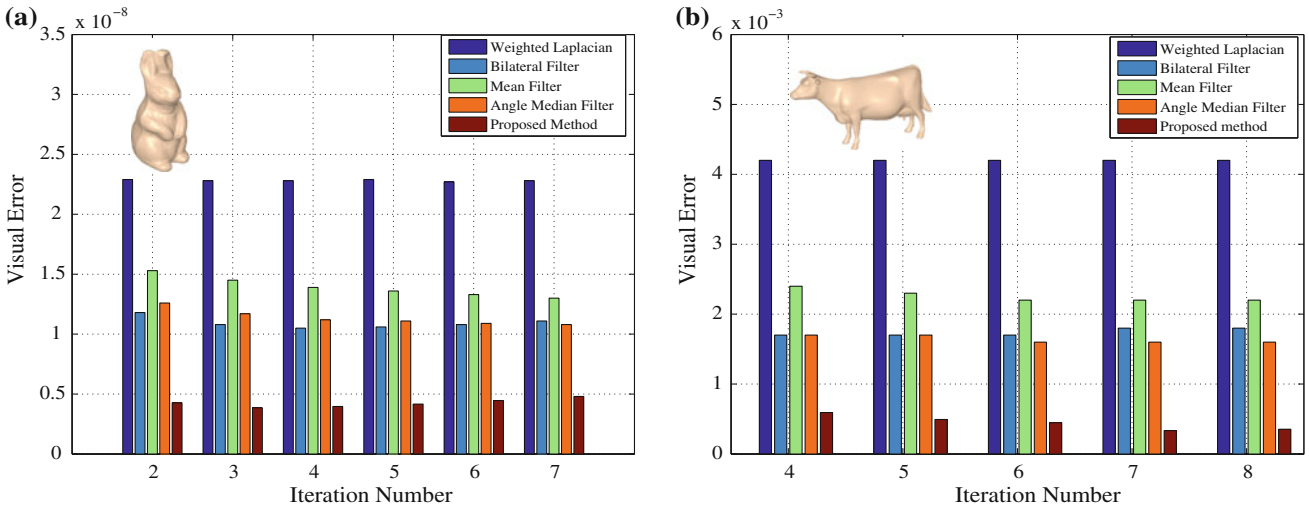

Fig. 11 Visual error comparison results between the proposed approach and other methods for the **a** rabbit and **b** cow models

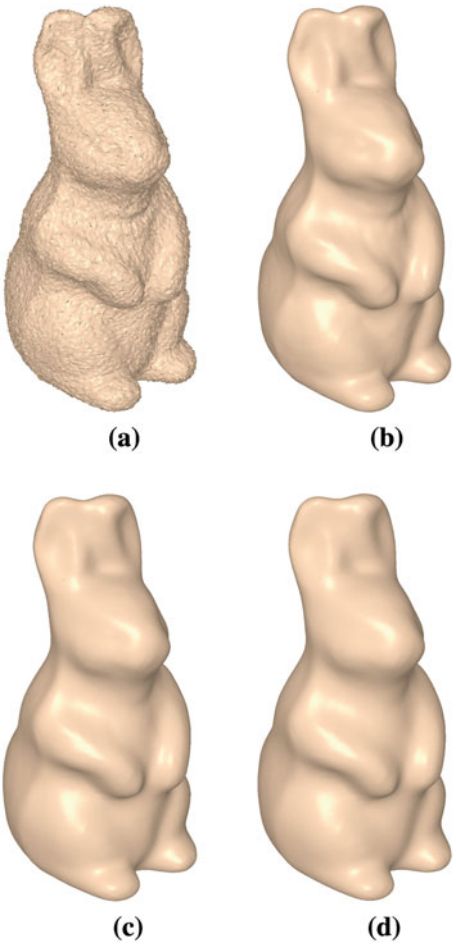

Fig. 12 Denoising results for the 3D rabbit model using $\lambda = 0.4$ at higher iteration numbers: **a** noisy model, **b** 10 iterations, **c** 20 iterations, **d** 30 iterations

this visual metric measures how close the vertices in both models are, whereas the second term captures the object smoothness which basically represents the visual properties of the human eye. The values of the visual error metric for some experiments are depicted in Fig. 11a and b which clearly show that the proposed method gives the best results, indicating the consistency with the subjective comparison.

Unlike the Laplacian flow which tends to produce spherically shaped outputs at higher iteration numbers [1], the proposed approach is, however, experimentally shown to stabilize as depicted in Figs. 12 and 13. It is apparent from these figures that the output results at iterations 20 are 30 are visually indistinguishable, indicating that the proposed algorithm produces a stable solution.

4.3 Choice of the regularization parameter

A mentioned earlier, the regularization parameter should be chosen appropriately in order to obtain satisfactory mesh denoising results. This parameter may be estimated experimentally using the visual error as shown in Fig. 14a and b, which display the plots of the visual error vs. the regularization parameter for different iteration numbers of the proposed approach.

5 Conclusions

In this paper, we introduced a simple and fast 3D mesh denoising technique using the concept of multivariate kernel density estimation. The main idea behind our proposed approach is to use a regularized bandwidth matrix of the kernel density in order to avoid over-smoothing and to fully preserve the geometric structure of the 3D mesh data, while effectively removing undesirable noise. The experimental results showed that our proposed technique is robust, accurate, and has a low computational cost compared to existing methods.

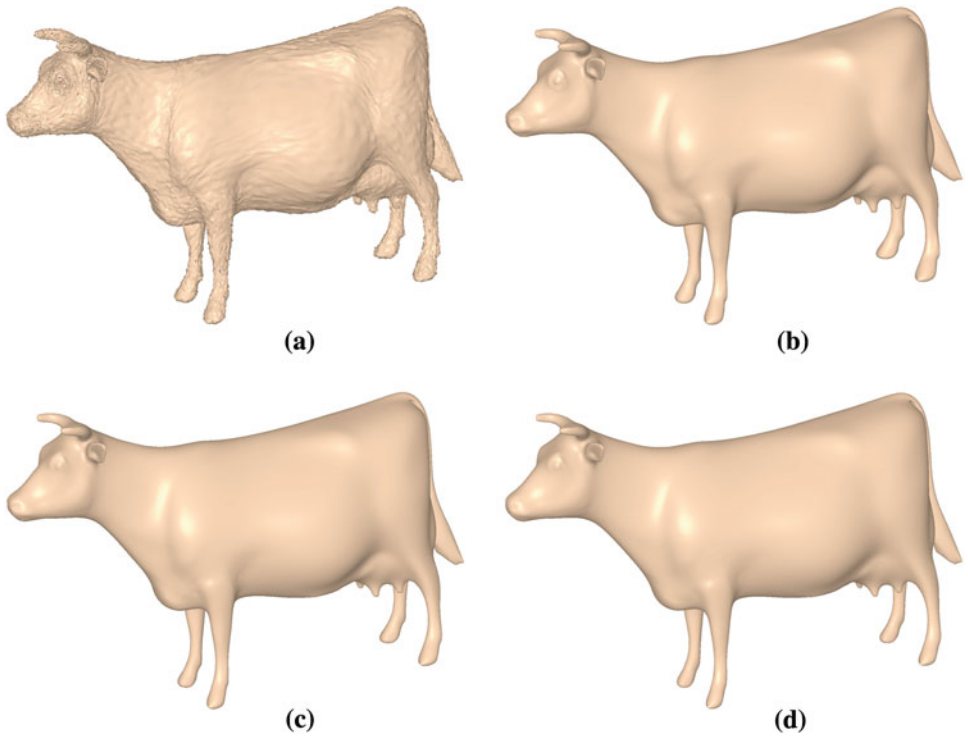

Fig. 13 Denoising results for the 3D cow model using $\lambda = 0.8$ at higher iteration numbers: **a** noisy model, **b** 10 iterations, **c** 20 iterations, **d** 30 iterations

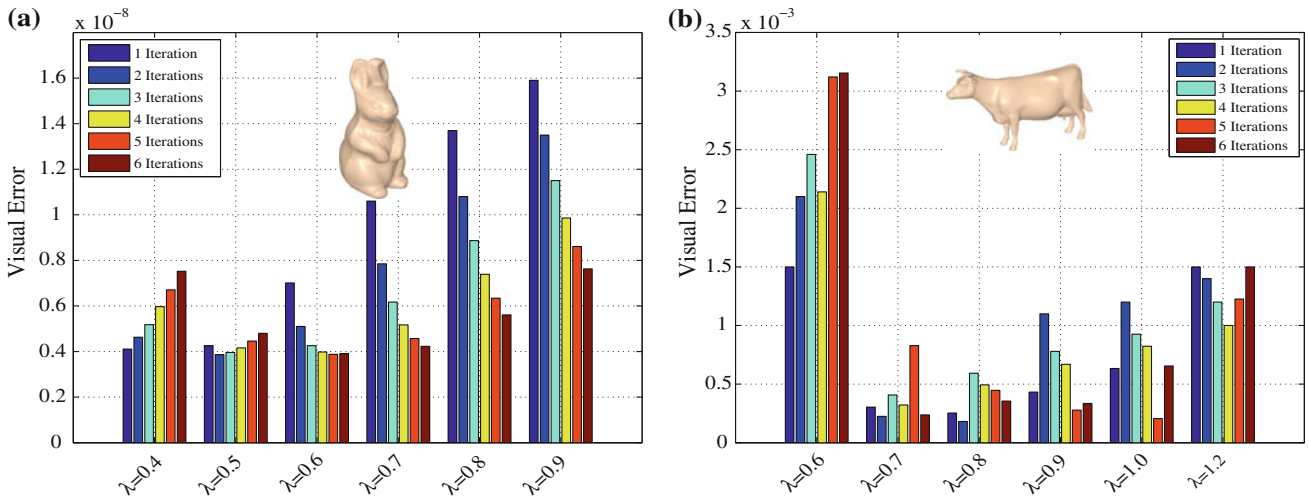

Fig. 14 Visual error versus regularization parameter with different number of iterations for the **a** rabbit and **b** cow models

Future work will be focused on exploring the use of the local structure tensor instead of the covariance matrix. Also, theoretically we hope to develop more rigorous way of finding the optimal regularization parameter of the covariance matrix.

References

1. Taubin, G.: A signal processing approach to fair surface design. In: Proceedings of the ACM SIGGRAPH, pp. 351–358 (1995)
2. Ohtake, Y., Belyaev, A.G., Bogaevski, I.A.: Polyhedral surface smoothing with simultaneous mesh regularization. In: Proceedings of the Geometric Modeling and Processing, pp. 229–237 (2000)
3. Taubin, G.: Linear anisotropic mesh filtering. IBM Research Report, RC2213 (2001)
4. Petitjean, S.: A survey of methods for recovering quadrics in triangle meshes. ACM Comput. Surv. **34**(2), 211–262 (2002)
5. Shen, Y., Barner, K.E.: Fuzzy vector median-based surface smoothing. IEEE Trans. Visual. Comput. Graph. **10**(3), 252–265 (2004)
6. Yagou, H., Ohtake, Y., Belyaev, A.: Mesh smoothing via mean and median filtering applied to face normals. In: Proceedings of the Geometric Modeling and Processing—Theory and Applications, pp. 124–131 (2002)
7. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. In: Proceedings of the ACM SIGGRAPH, pp. 950–953 (2003)

8. Jones, T., Durand, F., Desbrun, M.: Non-iterative, feature preserving mesh smoothing. In: Proceedings of the ACM SIGGRAPH, pp. 943–949 (2003)
9. Desbrun, M., Meyer, M., Schröder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of the ACM SIGGRAPH, pp. 317–324 (1999)
10. Desbrun, M., Meyer, M., Schröder, P., Bar, A.: Anisotropic feature-preserving denoising of height fields and bivariate data. In: Graphics Interface, pp. 145–152 (2000)
11. Clarenz, U., Diewald, U., Rumpf, M.: Anisotropic geometric diffusion in surface processing. In: Proceedings of the IEEE Visualization, pp. 397–405 (2000)
12. Clarenz, U., Diewald, U., Rumpf, M.: Processing textured surfaces via anisotropic geometric diffusion. IEEE Trans. Image Processing **13**(2), 248–261 (2004)
13. Bajaj, C.L., Xu, G.: Anisotropic diffusion of subdivision surfaces and functions on surfaces. ACM Trans. Graph. **22**(1), 4–32 (2003)
14. Tasdizen, T., Whitaker, R., Burchard, P., Osher, S.: Geometric surface smoothing via anisotropic diffusion of normals. In: Proceedings of the IEEE Visualization, pp. 125–132 (2002)
15. Tasdizen, T., Whitaker, R., Burchard, P., Osher, S.: Geometric surface processing via normal maps. ACM Trans. Graph. **22**(4), 1012–1033 (2003)
16. Hildebrandt, K., Polthier, K.: Anisotropic filtering of non-linear surface features. Comput. Graph. Forum **23**(3), 391–400 (2004)
17. Peng, J., Strela, V., Zorin, D.: A simple algorithm for surface denoising. In: Proceedings of the IEEE Visualization, pp. 107–112 (2001)
18. Delouille, V., Jansen, M., von Sachs, R.: Second-generation wavelet denoising methods for irregularly spaced data in two dimensions. Signal Processing **86**(7), 1435–1450 (2006)
19. Le Faucheur, X., Vidakovic, B., Tannenbaum, A.: Bayesian spherical wavelet shrinkage: applications to shape analysis. In: Proceedings of the SPIE, vol. 6763 (2007)
20. El Ouafdi, A.F., Ziou, D., Krim, H.: A smart stochastic approach for manifold smoothing. Comput. Graph. Forum **27**(5), 1357–1364 (2008)
21. Tarmissi, K., Ben Hamza, A.: Geometric mesh denoising via multivariate kernel diffusion. In: Proceedings of the MIRAGE'09. Lecture Notes in Computer Science, vol. 5496, pp. 23–33 (2009)
22. Zhang, Y., Ben Hamza, A.: Vertex-based diffusion for 3D mesh denoising. IEEE Trans. Image Processing **16**(4), 1036–1045 (2007)
23. Chung, F.R.: Spectral Graph Theory. American Mathematical Society, Providence (1997)
24. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
25. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of the ACM SIGGRAPH, pp. 279–286 (2000)