Mesh Smoothing via Mean and Median Filtering Applied to Face Normals

Hirokazu Yagou [*] Yutaka Ohtake [†] Alexander G. Belyaev [*,†]

[*] Shape Modeling Lab, University of Aizu, Aizu-Wakamatsu 965-8580 Japan

[†] Computer Graphics Group, Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany

m5051134@u-aizu.ac.jp ohtake@mpi-sb.mpg.de belyaev@u-aizu.ac.jp

Abstract

In this paper, we introduce iterative mean and median filtering schemes for smoothing noisy 3D shapes given as triangle meshes. Our main idea consists of applying mean and median filtering schemes to mesh normals and then update the mesh vertex position in order to fit the mesh to the modified normals.

We also give a quantitative evaluation of the proposed mesh filtering schemes and compare them with conventional mesh smoothing procedures such as the Laplacian smoothing flow and the mean curvature flow.

We demonstrate that our mean and median mesh filtering methods outperform the conventional Laplacian and mean curvature flows in terms of accuracy and resistance to over-smoothing.

1 Introduction

In many computer graphics applications, polygonal meshes deliver a simple and flexible way to represent and handle complex geometric objects. Dense triangle meshes are standard output of modern shape acquisition techniques such as laser scanning and isosurfacing volumetric data. The surface of a computer graphics model reconstructed from real-world data is often corrupted by noise. An important problem is to surpress noise while preserving desirable geometric features of the model. Many powerful noise supressing techniques were proposed for signal and image processing needs. However that techniques were developed for regularly sampled data and cannot be directly extended to meshes. In this paper, we introduce iterative mean and median filtering schemes for smoothing noisy 3D shapes approximated by triangle meshes. The main idea of our approach consists of applying mean and median filtering to mesh normals and then updating the mesh vertex positions in order to fit the mesh to the modified normals.

In image processing, the mean and median filters are simple and very effective tools for noise suppressing. In its simplest form, the mean filter replaces every pixel with the arithmetic mean of the pixels contained in a window around the pixel. The basic idea of the median filtering consists of simultaneous replacing every pixel of an image with the median of the pixels contained in a window around the pixel. Mean filtering is usually used for suppressing Gaussian noise while median filtering is a powerful tool for removing impulsive noise [1, 8]. Recently iterative mean and median filtering schemes and their modifications became very popular because of their close connection with PDE methods in image processing [11].

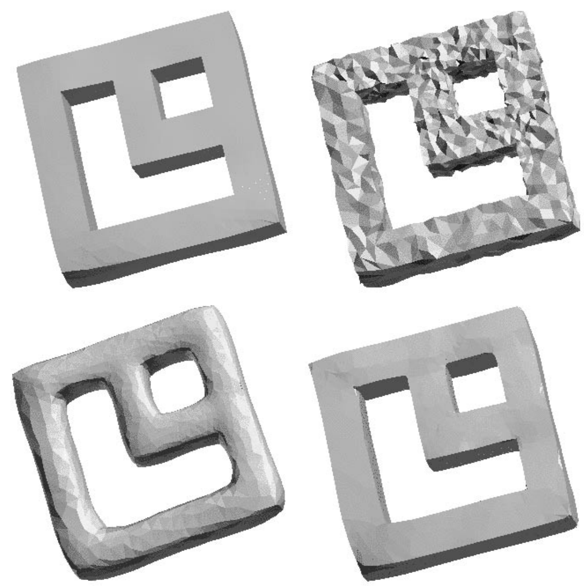

Figure 1. Top left: a triangle mesh representing a two-holed torus with sharp edges. Top right: noise is added. Bottom left: after smoothing by iterative mean filter. Bottom right: after smoothing by iterative weighted median filter.

Fig. 1 illustrates how iterative mean and median filtering schemes developed in this paper smoothing a polygonal model corrupted by additive random noise. Note that iterative mean filtering does not produce mesh shrinkage and

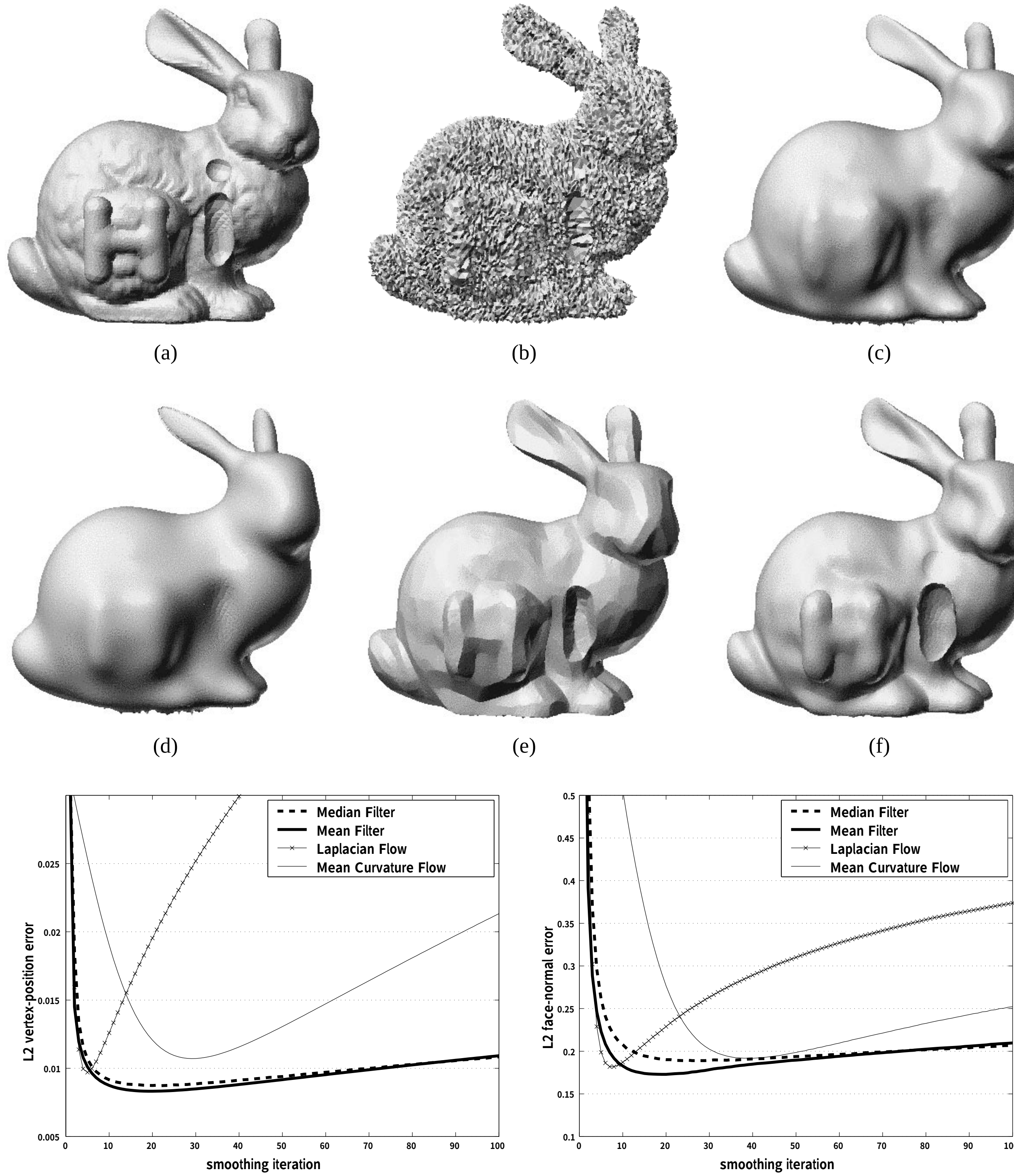

Figure 2. (a) Stanford Bunny with word "Hi" embossed. (b) Noise is added. (c)-(f) Oversmoothing by Laplacian flow (c), mean curvature flow (d), iterative median filter (e), iterative mean filter (f). For each smoothing method, the number of iterations used is equal to $10\times$(optimal number of iterations) where the optimal number of iterations is defined according to minimal value of vertex-based L^2 error. Bottom left: graphs of vertex-based L^2 error between original and smoothed models. Bottom right: graphs of normal-based L^2 error between original and smoothed models.

note how good sharp features are restored by iterative median filtering.

We also give a quantitative evaluation of the proposed mesh filtering schemes and compare them with conventional mesh smoothing procedures such as the Laplacian smoothing flow [12, 7] and the mean curvature flow [4, 3]. We use two L^2 error metrics introduced in [9] and comparing two close meshes by measuring deviations between corresponding mesh vertices and normals. It turns out that the mean and median filtering methods proposed in the paper produce significantly smaller oversmoothing then the conventional Laplacian and mean curvature flows whereas the best results obtained via mean and median filtering are at least no worse than that produced by the Laplacian and mean curvature flows. See Fig. 2 for details.

The paper is organized as follows. Section 2 describes two conventional methods for smoothing triangle meshes: the Laplacian [12, 7] and the mean curvature [4, 3] flows. In Section 3, we introduce our iterative mean filter. In Section 4, we present several median filtering schemes. Vertex-based and normal-based L^2 error metrics are described in Section 5. We compare the considered smoothing methods in Section 6 and conclude in Section 7.

2 Laplacian and Mean Curvature Flows

In this section, two conventional methods of polygonal surface smoothing are considered: the Laplacian flow [12, 7] and the mean curvature flow [4, 3].

Consider a discrete mesh evolution process each step of which updates mesh vertices according to

$$P_{new} \leftarrow P_{old} + \lambda \mathbf{D}(P_{old}). \tag{1}$$

where $\mathbf{D}(P)$ is a displacement vector, and λ is a step-size parameter.

The Laplacian mesh smoothing flow is obtained from (1) if the displacement vector $\mathbf{D}(P)$ is defined by the so-called umbrella operator [7]

$$\mathcal{U}(P) = \frac{1}{n} \sum_{i \in \mathcal{N}_1(P)} Q_i - P, \tag{2}$$

where P is a mesh vertex, $\mathcal{N}_1(P) = \{Q_0, Q_1, \ldots, Q_{n-1}\}$ is the 1-ring of mesh vertices neighboring with P, as seen in Fig. 3.

The explicit vertex updating scheme corresponding to the mean curvature flow is given by (1) where the displacement vector $\mathbf{D}(P)$ is equal to the mean curvature vector [4, 3]

$$H\mathbf{n}(P) = \frac{3}{4A} \sum_{i \in \mathcal{N}_1(P)} (\cot \alpha_i + \cot \beta_i)(Q_i - P). \tag{3}$$

Here A is the sum of the areas of the triangles surrounding P, α_i and β_i are the angles opposite to the edge Q_iP, as seen in Fig. 4.

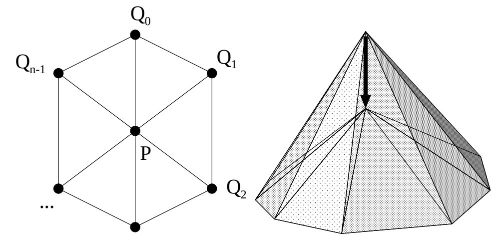

Figure 3. Left: 1-ring of neighbors of vertex P. Right: updating vertex position by umbrella operator.

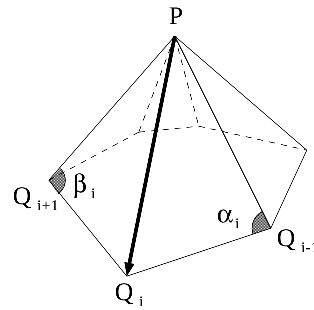

Figure 4. Angles α_i and β_i are used to estimate the mean curvature vector at P.

For closed meshes, in order to eliminate mesh shrinking and following [4] we keep the volume of the evolving mesh constant by rescaling the mesh after each step of the mesh evolution process.

3 Mean Filter for Averaging Face Normals

Consider an oriented triangle mesh. Let T be a mesh triangle, $\mathbf{n}(T)$ be the unit normal of T, $A(T)$ be the area of T, and $C(T)$ be the centroid of T. Denote by $\mathcal{N}(T)$ the set of all mesh triangles that have a common edge or vertex with T. One iteration of the iterative mesh mean filtering scheme consists of the following three successive steps.

Step 1. For each mesh triangle T, compute the triangle normal $\mathbf{n}(T)$ and perform the following area-weighted averaging normals:

$$\mathbf{m}(T) = \frac{1}{\sum A(S)} \sum_{S \in \mathcal{N}(T)} A(S)\mathbf{n}(S). \tag{4}$$

See the left image of Fig. 5.

Step 2. For each mesh triangle T, normalize the averaged normals $\mathbf{m}(T)$:

$$\mathbf{m}(T) \longleftarrow \frac{\mathbf{m}(T)}{\|\mathbf{m}(T)\|}.$$

Step 3. For each mesh vertex P, perform the vertex updating procedure

$$P_{\text{new}} \longleftarrow P_{\text{old}} + \frac{1}{\sum A(T)} \sum A(T)\mathbf{v}(T) \qquad (5)$$

$$\text{with} \quad \mathbf{v}(T) = \left[\overrightarrow{PC} \cdot \mathbf{m}(T)\right] \mathbf{m}(T), \qquad (6)$$

where the sums are taken over all triangles T adjacent to P, $\mathbf{v}(T)$ is the projection of the vector $\overrightarrow{PC}$ onto the $\mathbf{m}(T)$ direction, as exposed by the right image of Fig. 5.

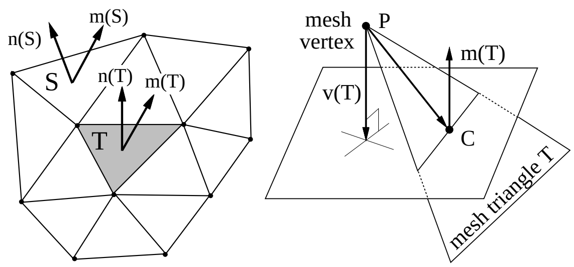

Figure 5. Left: mesh triangles T and S, their normals $\mathbf{n}(T)$ and $\mathbf{n}(S)$, and "diffused normals". $\mathbf{m}(T)$ and $\mathbf{m}(S)$. Right: updating mesh vertex position.

Area weighted averaging normals in **Step 1** and normalizing **Step 2** define a new unit vector field $\{\mathbf{m}\}$ defined at the mesh triangles. The mesh updating **Step 3** attempts to find a mesh whose normals are close to the unit vector field $\{\mathbf{m}\}$.

Now the complete smoothing procedure consists of applying $\mathbf{Step\,1} + \mathbf{Step\,2} + \mathbf{Step\,3}$ a sufficient number of times. It turns out that the mesh evolution process $(\mathbf{Step\,1} + \mathbf{Step\,2} + \mathbf{Step\,3})^n$ converges quickly as $n \to \infty$ and in practice $10 - 30$ iterations is quite enough to achieve a steady-state.

The iterative mean filter considered above is a simplified version of a nonlinear diffusion of normals proposed in [10] and used for crease enhancing. A similar mesh smoothing method was very recently proposed in [14].

4 Median Filtering Face Normals

The median filtering procedure described in this section differs from the mean filtering scheme of Section 3 by Step 1 only.

Instead of averaging mesh normals $\mathbf{n}(S)$ let us apply the classical median filtering [6] to the angles $\varphi(S,T) = \angle\big(\mathbf{n}(S), n(T)\big)$ between $\mathbf{n}(S)$ and $\mathbf{n}(T)$. Let $\mathbf{n}(S_{\text{median}})$ correspond to the median angle, then we set $\mathbf{m}(T) = \mathbf{n}(S_{\text{median}})$. We call this variation of mesh median filtering by *angle median filtering*.

Another median filtering procedure is obtained if we apply the classical median filtering to estimated directional curvatures $k(S,T) = \varphi(S,T)/|C(T)C(S)|$, where $C(T)$ and $C(S)$ are the centroids of the triangles T and S, respectively. Let us call this procedure by *curvature median filtering*.

The median filtering schemes enhance shape creases. According to our experiments, the angle median filtering scheme has a stronger crease enhancing effect than the the curvature median filtering scheme, see Fig. 6.

Weighted median filtering. The weighted median filtering scheme [1] described below is a simple and useful modification of the basic median filter.

Consider a set of samples $(x_0, \ldots, x_{n-1})$ and positive weights $(w_0, \ldots, w_{n-1})$. The output of the weighted median filter $\hat{x}$ is defined by

$$\hat{x} = Median(w_0 \odot x_0, \ldots, w_{n-1} \odot x_{n-1}), \qquad (7)$$

where

$$w_i \odot x_i = \underbrace{x_i, x_i, \ldots, x_i}_{w_i \text{ times}}. \qquad (8)$$

It is evident that elements with large weights are more frequently selected by the weighted median filter.

Let us divide the set of neighboring triangles $\mathcal{N}(T)$ of a given triangle T in two subsets: $\mathcal{N}_e(T)$, the set of mesh triangles sharing an edge with T, and $\mathcal{N}_v(T)$, the set of mesh triangles touching T at one vertex. We assign weight 2 to the triangles of $\mathcal{N}_e(T)$ and weight 1 to the triangles of $\mathcal{N}_v(T)$, as seen in Fig. 7.

5 L^2-Error Estimation

In order to compare the proposed mean and median mesh smoothing schemes with conventional smoothing methods we introduce two error metrics.

Consider an ideal mesh M and a mesh M' obtained from M by adding noise and applying several iterations of a smoothing process. Consider a vertex P' of the smoothed mesh M'. Let us set $\operatorname{dist}(P', M)$ equal to the distance between P' and a triangle of the ideal mesh M closest to P'. Our vertex-based L^2 error metric is given by

$$\varepsilon_v = \frac{1}{3A(M')} \sum_{P' \in M'} A(P') \operatorname{dist}(P', M)^2 \qquad (9)$$

where $A(P')$ is the sum of areas of all triangles of M' incident with P' and $A(M')$ is the total area of M'.

Our normal-based L^2 error metric is defined in a similar way. Consider a triangle T' of the mesh M' and let us find a triangle T of M closest to T'. Let $\mathbf{n}(T)$ and $\mathbf{n}(T')$ be the orientation unit normals of T and T' respectively. The normal-based error metric is given by

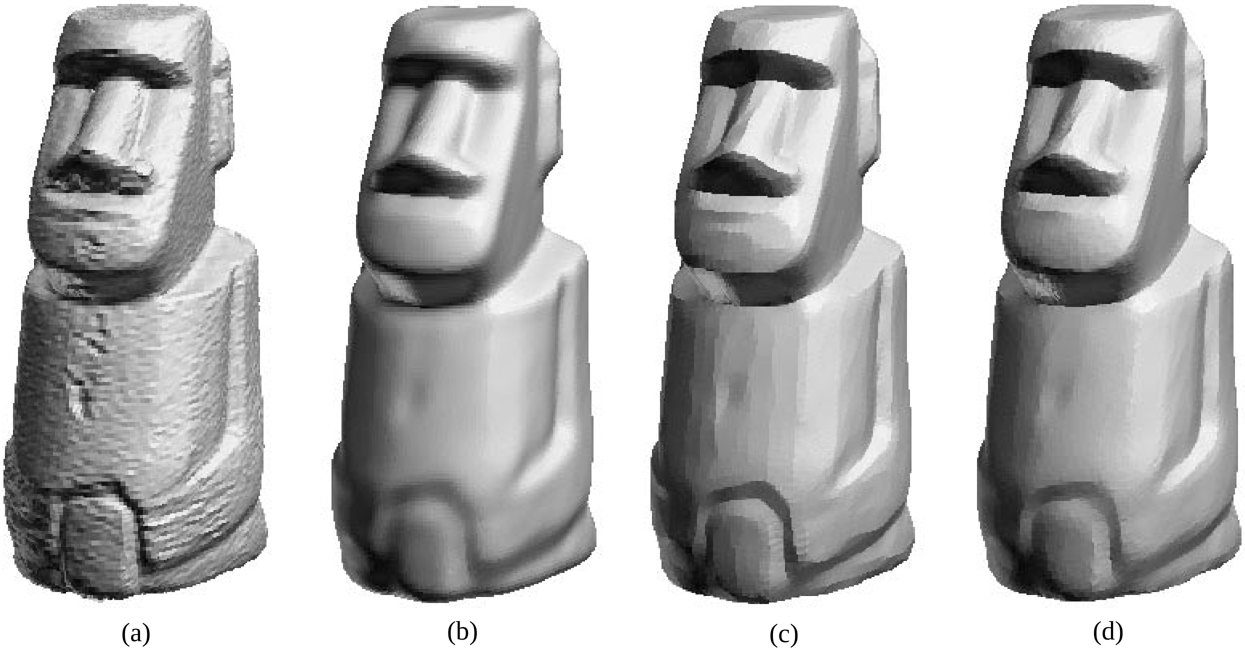

Figure 6. (a) A Moai statue model digitized by a 3-D laser scanning system (Minolta Vivid 700). (b) Smoothed by fifty iterations of the mean filter. (c) Smoothed by fifty iterations of the angle median filter. (d) Smoothed by fifty iterations of the curvature median filter.

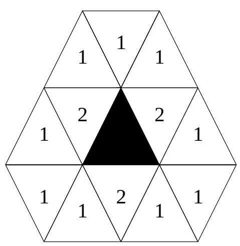

Figure 7. Allocating weights to triangles from $\mathcal{N}(T)$.

$$\varepsilon_f = \frac{1}{A(M')} \sum_{T' \in M'} A(T') |\mathbf{n}(T) - n(T')|^2 \qquad (10)$$

where $A(T')$ is the area of T'.

6 Comparison of Smoothing Methods

The developed mean and median filtering schemes show better performance with respect to the vertex-based and normal-based L^2 error metrics than the Laplacian and mean curvature flows.

According to our experiments, iterative mean filtering outperforms slightly iterative median filtering for uniform meshes without sharp features, as seen in Fig. 8. However for highly nonuniform meshes with sharp creases the iterative median filtering scheme demonstrate a better performance than the iterative mean filtering scheme, as seen in Fig. 9.

A comparison of the mean filter and simple/weighted and angle/curvature median filtering schemes is presented by Fig. 10. The tested two-holed torus model has sharp features and therefore median filtering is preferable. The best smoothing effect according to visual appearance and the error metrics is achieved by the weighted median filters.

7 Conclusion and Future Work

In this paper, we have presented new methods for triangle mesh denoising: iterative mean and median filtering schemes. We have also compared the proposed mesh filtering schemes with conventional mesh smoothing procedures such as the Laplacian smoothing flow and the mean curvature flow. The comparison has demonstrated that the proposed methods outperformed the conventional ones in terms of accuracy and resistance to oversmoothing.

One interesting direction for future research consists of developing an error metric corresponding to our visual perception of 3D shapes better than the proposed vertex-based and normal-based L^2 error metrics. A comparison of the proposed methods with advanced mesh smoothing techniques developed recently [12, 5, 15, 2, 13, 14] also remains a task for future research.

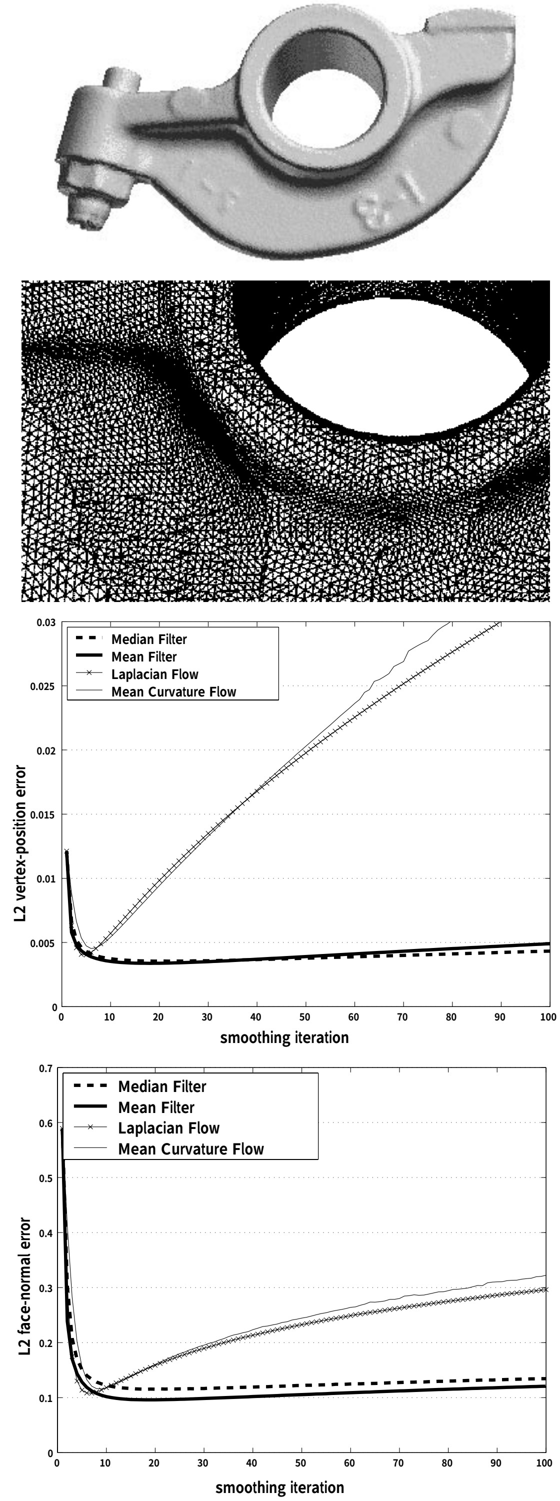

Figure 8. Rocker-arm model is represented by a relatively uniform mesh. Mean filtering is the best choice for the model with noise added.

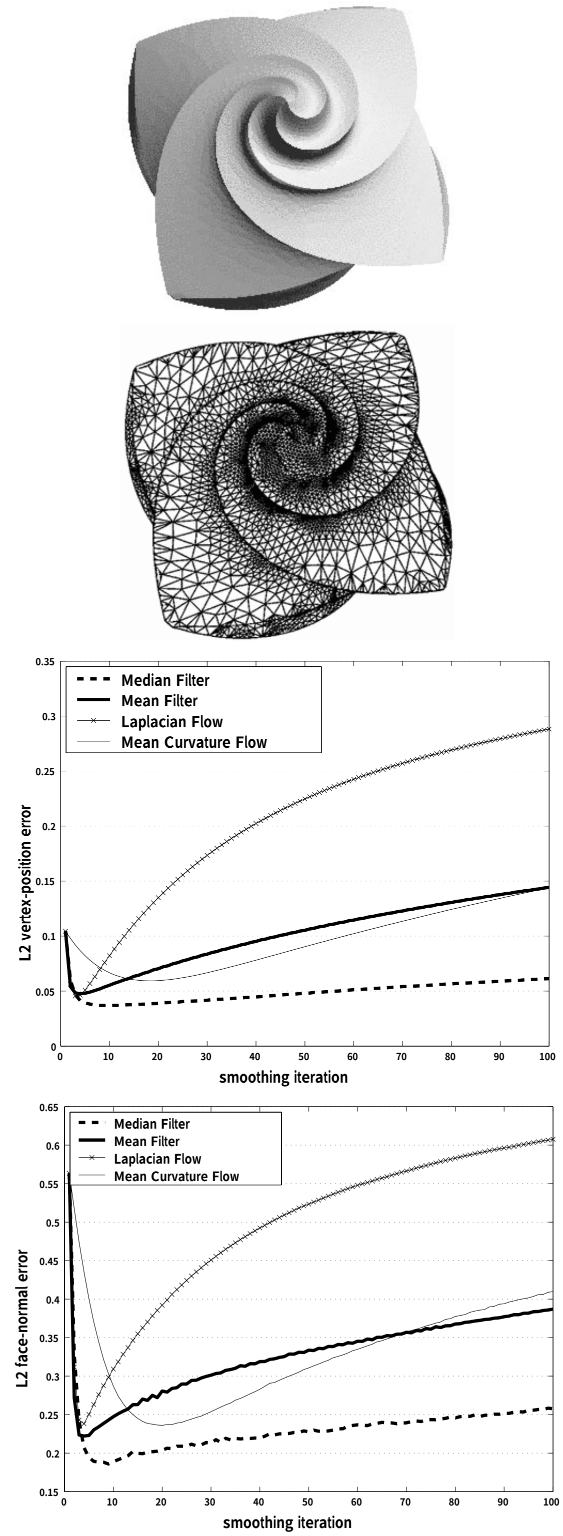

Figure 9. Flower model is represented by a highly non-uniform mesh and has sharp features. Angle median filtering demonstrates the best performance for smoothing the model with noise added.

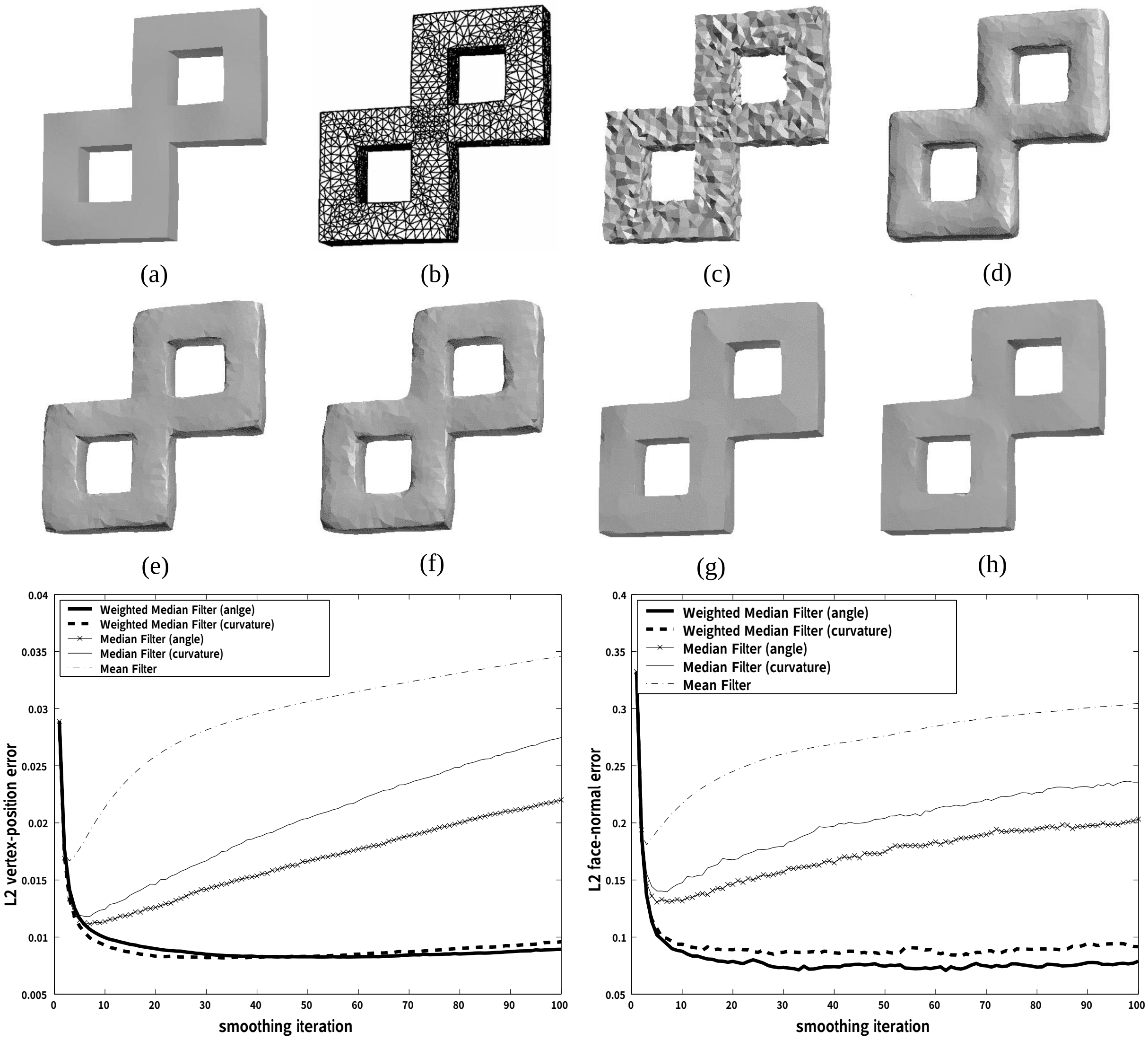

Figure 10. Smoothing a noisy two-holed torus. vertex-based L^2 error. (a) Original flat-shaded model. (b) Original wireframe model. (c) Noisy is added. (d) Smoothed by mean filtering. (e) Smoothed by angle median filtering. (f) Smoothed by curvature median filtering. (g) Smoothed by weighted angle median filtering. (h) Smoothed by weighted curvature median filtering.

References

[1] G. Arce and J. Paredes. Image enhancement and analysis with weighted medians. In S. K. Mitra and G. L. Sicuranza, editors, *Nonlinear Image Processing*. Academic Press, 2001.

[2] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *IEEE Visualization 2000*, pages 397–405, October 2000.

[3] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Discrete differential-geometry operators in nD. Available on WWW at http://www.multires.caltech.edu/pubs/pubs.htm.

[4] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics (Proceedings of SIGGRAPH 99)*, pages 317–324, 1999.

[5] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Computer Graphics (Proceedings of SIGGRAPH 99)*, pages 325–334, 1999.

[6] R. Klette and P. Zamperori. *Handbook of Image Processing Operators*. John Wiley & Sons, 1996.

[7] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Computer Graphics (SIGGRAPH 98 Proceedings)*, pages 105–114, 1998.

[8] N. Nikolaidis and I. Pitas. *3-D Image Processing Algorithms*. John Wiley & Sons, 2001.

[9] Y. Ohtake. *Mesh Optimization and Feature Extraction*. PhD thesis, University of Aizu, Japan, March 2002.

[10] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Mesh regularization and adaptive smoothing. *Computer-Aided Design*, 33(4):789–800, 2001.

[11] G. Sapiro. *Geometric Partial Diffrential Equations and Image Analysis*. Cambridge University Press, 2001.

[12] G. Taubin. A signal processing approach to fair surface design. In *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 351–358, 1995.

[13] G. Taubin. Dual mesh resampling. In *Proceedings of Pacific Graphics ’01*, pages 180–188, October 2001.

[14] G. Taubin. Linear anisotropic mesh filtering. IBM Research Report RC22213 (W0110-051), IBM Thomas J. Watson Research Center, October 2001.

[15] J. Vollmer, R. Mencl, and H. Muller. Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum (Proc. Eurographics 1999)*, 18(3):131–138, 1999.