Mesh Median Filter for Smoothing 3-D Polygonal Surfaces

Hirokazu Yagou* Alexander Belyaev† Daming Wei‡

*,† ,‡ Shape Modeling Lab, The University of Aizu, Aizu-Wakamatsu City 965-8580 Japan
† Computer Graphics Group, Max-Planck-Institut für Infomatik, 66123 Saarbrücken, Germany

Abstract

In this paper, we introduce a new mesh filtering method: a mesh median filter. *This is an application of the median filter to smoothen 3-D noisy shapes given by triangle meshes. An algorithm of the mesh median filter is realized by applying the median filter to face normals on triangle meshes and updating mesh vertex positions to make them fit to the filtered normals. As an advanced modification of the mesh median filter, we further introduce a* weighted mesh median filter. *The weighted mesh median filter has a reinforced feature preservation effect. The weighted mesh median filter with positive weighting has the smoothing effect, and the one with negative weighting has the enhancing effect. The two kinds of mesh median filters are compared with two conventional mesh filtering methods: the Laplacian smoothing flow and the mean curvature flow. Experimental results demonstrate that the mesh median filter does not induce oversmoothing.*

Keywords: *triangulated surfaces, triangle meshes, 3-D mesh smoothing, median filter, weighted median filter*

Technical Area: *3-D Computer Graphics*

1 Introduction

Triangulated surfaces reconstructed from real-world data usually contain undesirable noise. It is an important requirement to smooth the noise on a triangulated surface while preserving geometric features of the surface.

Let us consider the definition of noise on a triangulated surface. On a triangle mesh with no additive noise, vertices of each triangle exist at proper positions, especially touching on the mesh surface. If a triangle mesh is added some noise, the vertices are disarranged and their positions separates from the mesh. We define the noise as mesh vertices separating from their proper positions on the mesh surface. Thus, a noise suppression process is equivalent to the correction of mesh vertex positions. To perform such noise suppression, concepts based on the heat diffusion on a surface and the differential geometry approach have been used in previous work [2, 3, 5, 10]. However, noise suppression based on these approaches usually distort sharp geometric features as shown in Fig. 1-(c) and (d).

In signal and image processing, a nonlinear filter usually has a feature-preserving effect. The *median filter* [4] is one of such nonlinear filters. In this study, we apply the median filter to suppress undesirable noise on 3-D shapes given by triangle meshes, and investigate its feature-preserving effect. Fig. 1 shows an example how the mesh median filter works; it suppresses noise while preserves the sharp feature.

To discuss how to apply the median filter to triangle meshes, we consider how it works in image processing. The median filter in image processing [4] is applied to intensity values of image pixels. At a filtering process, a local neighborhood centered at the filtered pixel is considered. We collect all intensity values from the local neighborhood; select the median value; and set it to the center pixel. Therefore, to apply the median filter to face normals on triangle meshes, we consider the direction of a face normal as the intensity value of a image pixel, and update the direction at a smoothing process.

In this paper, a *weighted mesh median filter* is also introduced as an advanced modification of the mesh median filter. The weighted mesh median filter has a strengthened feature-preserving effect. Positive weighting increases the low-pass filtering (smoothing) effect, and negative weighting boosts the high-pass filtering (enhancing) effect.

This paper is organized as follows. Section 2 describes frameworks of the two conventional mesh smoothing methods. In section 3, an algorithm of the mesh median filter is presented, and then the weighted mesh median filter is depicted in section 4. Experimental results are discussed in section 5, and they are discussed in section 6. This paper is

*m5051134@u-aizu.ac.jp
†belyaev@mpi-sb.mpg.de, belyaev@u-aizu.ac.jp
‡dm-wei@u-aizu.ac.jp

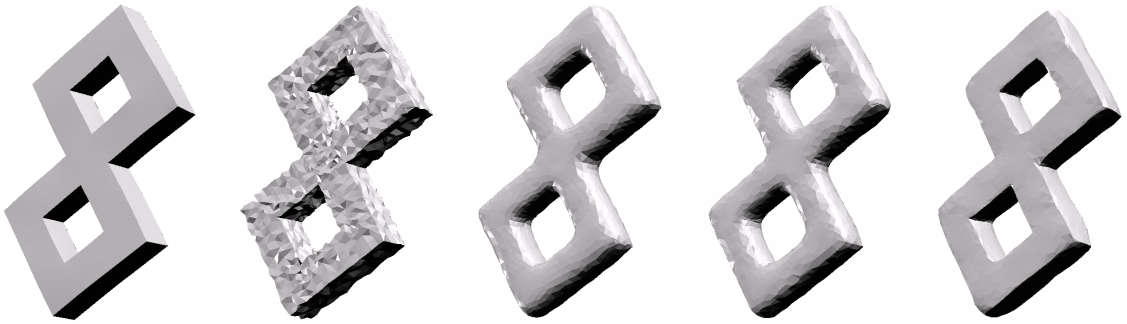

Figure 1. (a) A two-holed torus model. (b) Noise is added. (c) Smoothed by the Laplacian smoothing flow. (d) Smoothed by the mean curvature flow. (e) Smoothed by the mesh median filter.

concluded in section 7.

2 Background

In this section, two conventional methods of polygonal surface smoothing are considered: the Laplacian smoothing flow [5, 10] and the mean curvature flow [2, 3]. The Laplacian smoothing is developed from a two-dimensional heat equation, and the mean curvature flow is formulated based on concepts of the differential geometry.

Consider a discrete mesh evolution process at which mesh vertex positions are updated 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 smoothing flow is obtained from Eq. (1) if the displacement vector $\mathbf{D}(P)$ is defined by the so-called umbrella operator [5]

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

where P is a mesh vertex, and $\mathcal{N}_1(P) = \{Q_0, Q_1, \ldots, Q_{n-1}\}$ is the 1-ring of mesh vertices neighboring on P, as seen in Fig. 2.

For the explicit vertex updating scheme corresponding to the mean curvature flow, the displacement vector $\mathbf{D}(P)$ in Eq. (1) is equal to the mean curvature vector [2, 3]

$$H\mathbf{n}(P) = \frac{3}{2A} \sum_{i \in \mathcal{N}_1(P)} (\cot \alpha_i + \cot \beta_i)(Q_i - P). \tag{3}$$

Here α_i and β_i are the angles opposite to the edge Q_iP, as seen in Fig. 4.

In order to eliminate mesh shrinking, we keep the volume of the evolving mesh constant by rescaling the mesh after each step of the mesh evolution process [3].

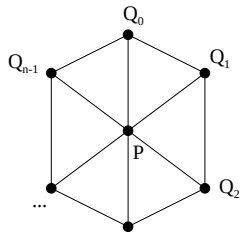

Figure 2. 1-ring of neighbors of vertex P.

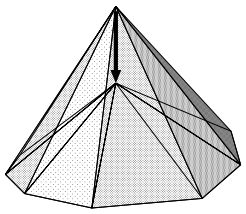

Figure 3. Updating vertex position by umbrella operator.

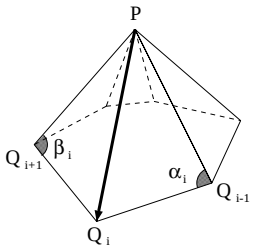

Figure 4. Angles α_i and β_i are used to estimate the mean curvature vector at P.

3 Mesh Median Filter

An implementation of the mesh median filter is discussed in this section. Consider an oriented triangle mesh. Let T be a mesh triangle; $\mathbf{n}(T)$ be the unit normal; $A(T)$ be the area of T; and $C(T)$ be the centroid of T. $\mathcal{N}(T)$ indicates the set of all triangles touching T with a edge or a vertex. One cycle process of the mesh median filter consist of the following two consecutive steps.

Step 1. In the local neighborhood $\mathcal{N}(T)$, an angle $\theta_i = \angle(\mathbf{n}(T), \mathbf{n}(U_i))$ is considered over all triangles adjacent to the T . The classical median filter is applied to those angles θ_i. Let θ_i be the median angle in the $\mathcal{N}(T)$, then face normal of the center triangle $\mathbf{n}(T)$ is replaced by $\mathbf{n}(U_i)$. We define the replaced normal as $\mathbf{m}(T)$. This replacement operation is performed throughout all triangles of a mesh.

Step 2. Consider all face normals on a mesh have been already modified in the step 1. For each mesh vertex P, its position is updated by

$$P_{\text{new}} \longleftarrow P_{\text{old}} + \frac{1}{\sum A(T)} \sum A(T)\mathbf{v}(T) \tag{4}$$

$$\text{with} \quad \mathbf{v}(T) = \left[\overrightarrow{PC} \cdot \mathbf{m}(T)\right] \mathbf{m}(T) \tag{5}$$

where the summations are taken over all triangles T adjacent to P, and $\mathbf{v}(T)$ is the projection of a vector $\overrightarrow{PC}$ onto the direction of $\mathbf{m}(T)$, as shown by the right image of Fig. 5. The application of the median filter to face normals on a tri-

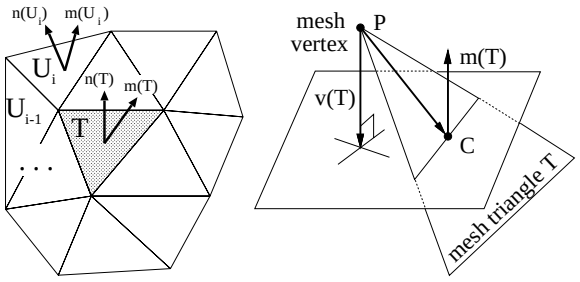

Figure 5. Left: a center triangle T and neighboring triangles U_i in a local neighborhood $\mathcal{N}(T)$. $\mathbf{n}(T)$ and $\mathbf{n}(U_i)$ are original face normals and $\mathbf{m}(T)$ and $\mathbf{m}(U_i)$ are filtered ones. Right: a visual representation of Eq. 5.

angle mesh (**Step 1**) defines a new unit vector field, and then the renewal of mesh vertex positions (**Step 2**) try to find a new mesh whose normals are close to the new unit vector field. The complete process of the mesh median filter is composed of the application of **Step 1** and **Step 2** in a sufficient number of iterations.

4 Weighted Mesh Median Filter

In this section, we first discuss about the principle of the weighted median filter, and then it is applied to 3-D mesh smoothing.

Consider a set of samples $(x_0, x_1, \ldots, x_{n-1})$ and positive weights $(w_0, \ldots, w_{n-1})$. The output of the weighted median filter $\hat{x}$ is defined by

$$\hat{x} = Median(w_0 \odot x_0, \ldots, w_{n-1} \odot x_{n-1}), \tag{6}$$

where

$$w_i \odot x_i = \underbrace{x_i, x_i, \ldots, x_i}_{w_i \text{ times}}. \tag{7}$$

It is evident that elements with large weights are more frequently selected by the weighted median filter [1].

To apply the weighted median filter to triangle meshes, we divide a set of all triangle adjacent to a triangle T into two subsets: $\mathcal{N}_e(T)$ and $\mathcal{N}_v(T)$. The $\mathcal{N}_e(T)$ is a set of mesh triangles sharing an edge with the T, and the $\mathcal{N}_v(T)$ is a set of mesh triangles touching the T with a vertex. We assign a weight 2 to all triangle of the $\mathcal{N}_e(T)$ and a weight 1 to all triangles of the $\mathcal{N}_v(T)$, as shown in Fig. 6. The weighted

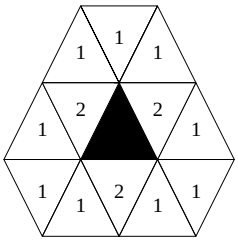

Figure 6. Allocate weights to all triangles of $\mathcal{N}(T)$ except the center triangle.

median filter is applied to all angles $\theta_i = \angle(\mathbf{n}(T), \mathbf{n}(U_i))$ in $\mathcal{N}(T)$ based on the weight allocation illustrated in Fig. 6.

The weighted mesh median filter has a better feature-preserving effect than the mesh median filter does. The positive weighting increases the smoothing effect of the mesh median filter. In flip side, the negative weighting boosts the enhancing effect of the mesh median filter, but the smoothing effect weakens. In this case, a weight -2 is allocated to all triangles of $\mathcal{N}_e(T)$. An operation of the negative weighting is as follows:

$$\begin{aligned} (-w_i) \odot x_i &= w_i \odot (-x_i), \\ &= \underbrace{-x_i, -x_i, \ldots, -x_i}_{w_i \text{ times}}. \end{aligned}$$

A difference between the positive weighting and the negative weighting is demonstrated in Fig. 8.

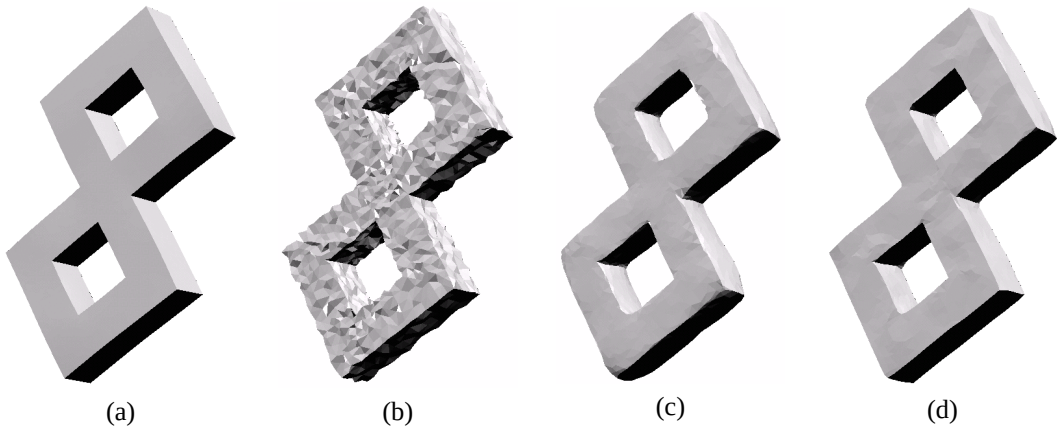

Figure 7. Smoothing results of a two-holed torus. (a): Original model. (b): Noise is added. (c): smoothed by the mesh median filter. (d): smoothed by the weighted mesh median filter.

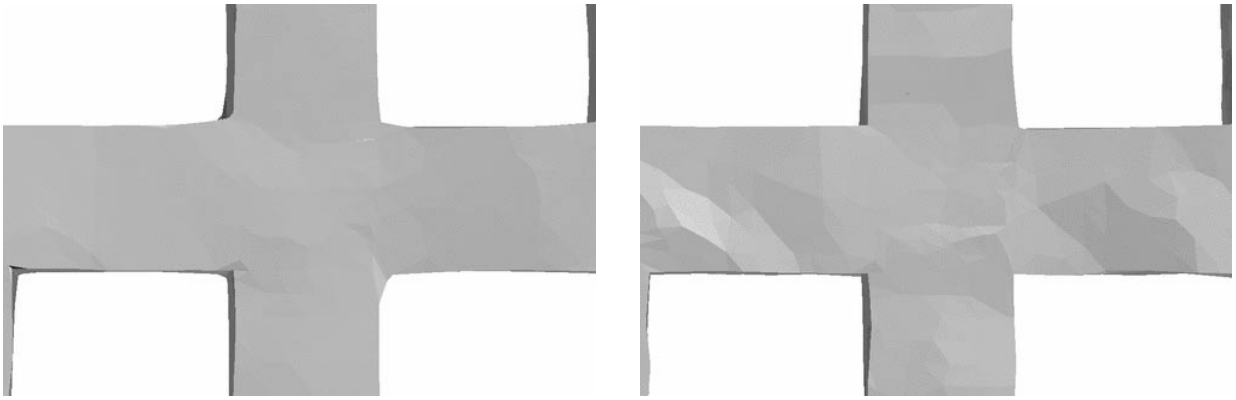

Figure 8. Top: the results of the weighted mesh median filter with **positive weights**. Bottom: the results of the weighted mesh median filter with **negative weights**.

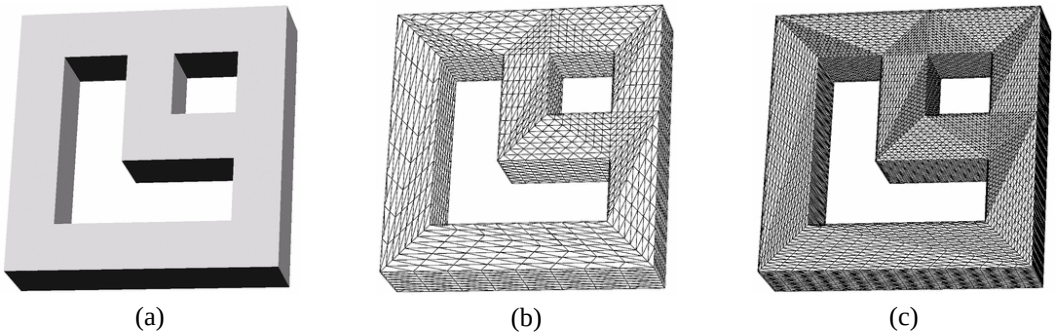

Figure 9. A mesh model is used at a bench-mark test. (a) The original model. (b) A wireframe representation of (a). (c) Subdivided linearly.

5 Experimental Results

We evaluate the two kinds of mesh median filters with the following three points of view: a speed of processing; a resistance to oversmoothing; and a effect of preserving sharp geometric features.

First, we check processing speeds of the two kinds of median filters through a speed-benchmark test. A personal computer used in the test is equipped with a 1.7-gigahertz CPU (Pentium 4) and a 512-megabyte RAM. A triangle mesh, as seen in Fig. 9, is used for the benchmark test; the mesh model is constructed by 2174 vertices and 4352 triangle faces. We measure how much time does it cost to complete a process of 20-iteration smoothing (case 1). Further, the mesh model is linearly subdivided to be constructed by 8702 vertices and 17408 triangle faces. The same experiment is performed for the subdivided mesh (case 2). Tables in Fig. 10 shows the results of the benchmark test.

METHOD	TIME (msec)		(Case 2)/(Case 1)
	Case 1	Case 2	
(a)	15	47	3.100
(b)	46	203	4.413
(c)	469	1938	4.132
(d)	672	2672	3.976

PROPERTY	QUANTITY		(Case 2)/(Case 1)
	Case 1	Case 2	
Vertex	2174	8702	4.002
Face	4352	17408	4.000

Figure 10. Top: the results of the speed-benchmark test. (a) Laplacian smoothing. (b) Mean curvature flow. (c) Mesh median filtering. (d) Weighted mesh median filtering. The bottom table shows how many vertices and faces are increased through the subdivision operation.

Second, we perform an experiment to inspect the resistance to oversmoothing of the mesh median filter. In this experiment, a moai statue model digitized by a 3-D laser scanning system (Minolta VIVID 700) is smoothed by the Laplacian smoothing, the mean curvature flow, and the mesh median filter. The number of smoothing iterations is 200 for the Laplacian smoothing and the mean curvature flow. The step size of these methods is similarly 0.2. The mesh median filter is applied by 400 iterations. For the moai model, 20-iteration smoothing is suitable. As shown in Fig. 11, the mesh median filter has a strong resistance to the oversmoothing.

We then verify the effect of preserving geometric features through another experiment. A monk statue model digitized by the VIVID 700 is used in this experiment. This model has several big hollows on its surface, as seen in Fig. 12. However, these hollows does not exist on the captured object. We apply smoothing methods until suppressing those hollows and then check a smoothing effect for other parts. If geometric features are well preserved after the smoothing operation, we consider that the smoothing method has the feature-preserving effect. In this experiment, the Laplacian smoothing, the mean curvature flow, the mesh median filter, and the weighted mesh median filter are used.

Fig. 12 show the experimental results. To fill up the hollows, all smoothing methods are applied by 100 iterations. The step size of the Laplacian smoothing is 0.2, and the one of the mean curvature flow is 0.1.

6 Discussion

The top table in Fig. 10 shows that the Laplacian smoothing and the mean curvature flow faster work than mesh median filters. However, they degrade sharp features of the mesh model.

The mean curvature flow, the mesh median filter, and the weighted mesh median filter are nonlinear methods. After the subdivision operation, their processing times are increased to be approximately four times as many as the previous ones. The increasing rate is nearly similar to ones of vertices and faces. The Laplacian smoothing, a linear method, has the least increasing rate of the processing time.

As shown in Fig. 11, it is evident that the mesh median filter more stably performs than the conventional smoothing methods do. Geometric features of the moai statue model, smoothed by the mesh median filter, is not blurred at the filtering process. The mesh median filter gives us such smoothing results even if we set the large number of smoothing iterations.

We must set the step size when using the Laplacian smoothing and the mean curvature flow. If a wrong step size is set to these smoothing methods, the oversmoothing occurs in few iterations. However, the mesh median filter does not require such step size, and the oversmoothing is restrained at a smoothing process.

Fig. 12 shows that the mesh median filter is effective to suppress big noise. In fact, the mesh median filter suppresses needless hollows on the monk statue model while preserves the model's geometric features. The weighted mesh median filter has a better feature-preserving effect than the mesh median filter does, as seen Fig. 7 and 12.

7 Conclusion

In this paper, we applied the *mesh median filter* and the *weighted mesh median filter* to 3-D mesh smoothing. The mesh median filter is constructed by a combination of 1) the application of the median filter to face normals on triangle meshes and 2) the evolution of mesh vertex positions to make them fit to the filtered normals. Numerical experiments show that the two median filters are satisfactory in noise reduction and feature preservation in 3-D mesh smoothing applications.

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] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for Triangulated 2-Manifold. Available on WWW at http://www.multires.caltech.edu/pubs/pubs.htm.

[3] 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.

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

[5] 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.

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

[7] Y. Ohtake, A. G. Belyaev, and I. A. Bogavski. Polyhedral Surface Smoothing with Modified Laplacian and Curvature Flow. *The Journal of Three Dimensional Images*, 13(3): 19–24, 1999.

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

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

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

[11] G. Taubin. Linear Anisotropic Mesh Filtering. IBM Research Report RC22213 (W0110-051), IBM, October 2001.

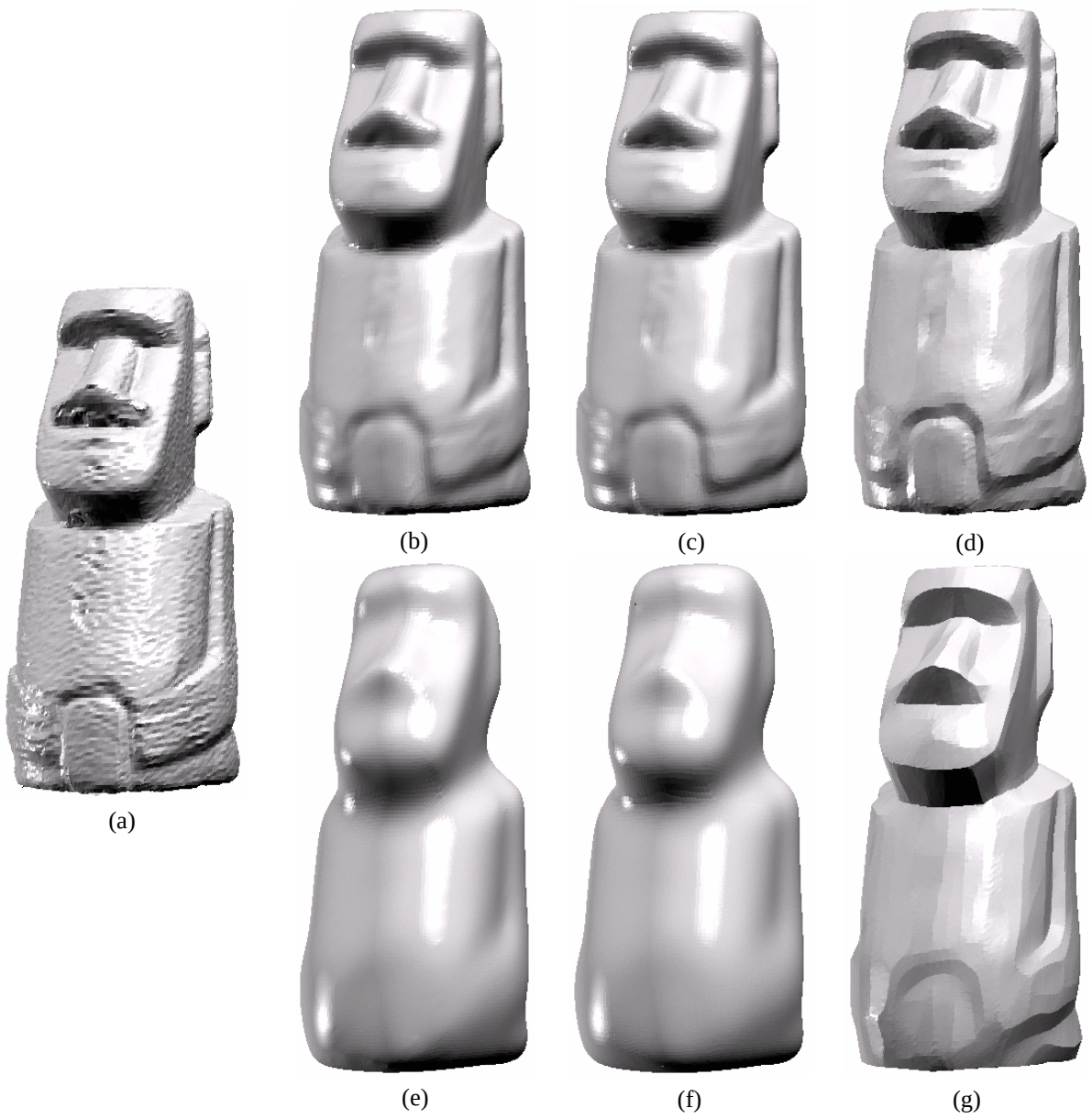

Figure 11. (a): A moai statue model with real-world noise. (b), (c), and (d) are the results of 20-iteration smoothing. (b): Smoothed by the Laplacian smoothing. (c): Smoothed by the mean curvature flow. (d): Smoothed by the mesh median filter. (e) and (f) are the results of 200-iteration smoothing. (e): Smoothed by the Laplacian smoothing. (f): Smoothed by the mean curvature flow. (g) is the results of 400-iteration smoothing. It was smoothed by the mesh median filter.

Figure 12. (a) A monk statue model. (b) Needless hollows exist on the triangulated surface. (c) Smoothed by the Laplacian smoothing. (d) Smoothed by the mean curvature flow. (e) Smoothed by the mesh median filter. (f) Smoothed by the weighted mesh median filter.