Feature-Preserving Mesh Denoising via Bilateral Normal Filtering

Kai-Wah Lee, Wen-Ping Wang
Computer Graphics Group
Department of Computer Science, The University of Hong Kong
kwlee@cs.hku.hk, wenping@cs.hku.hk

Abstract

In this paper, we propose a feature-preserving mesh denoising algorithm which is effective, simple and easy to implement. The proposed method is a two-stage procedure with a bilateral surface normal filtering followed by integration of the normals for least squares error (LSE) vertex position updates. It is well-known that normal variations offer more intuitive geometric meaning than vertex position variations. A smooth surface can be described as one having smoothly varying normals whereas features such as edges and corners appear as discontinuities in the normals. Thus we cast feature-preserving mesh denoising as a robust surface normal estimation using bilateral filtering. Our definition of "intensity difference" used in the influence weighting function of the bilateral filter robustly prevents features such as sharp edges and corners from being washed out. We will demonstrate this capability by comparing the results from smoothing CAD-like models with other smoothing algorithms.

1. Introduction

The ease-of-use and affordable 3D scanning devices [12], [13] have been widely used in various domain of applications ranging from reverse engineering to character modeling in animation production. These models are often represented as triangular meshes as hardware graphics cards are optimized for triangle rendering. Despite of using high-fidelity scanning, undesirable noise is inevitably introduced from various sources such as local measurements, limited sampling resolution and algorithmic errors. Models extracted from CT or MRI [14] volumetric data also result in detailed meshes with significant amount of noise. Thus, detailed noisy models need to be smoothed or denoised before any subsequent mesh processing such as simplification [15] and compression [16] could be successfully applied.

Mesh denoising has been an active research area since the pioneer work done by Taubin [1]. Research efforts were initially focused in surface fairing in which surface smoothness is enhanced at the expense of sharp features being blurred [1], [3], [4]. Attentions are then turned to smoothing with features preserved. Feature-preserving filtering techniques used in 2D image denoising are extended to address the denoising problems with 3D mesh. Mostly notable techniques are anisotropic diffusion [7], robust estimation [18] and bilateral filtering [9]. Researches [10], [18] have shown that these three techniques do have strong connections with each other. In this paper, we choose to first apply bilateral filtering on the surface normals and then evolve the mesh with least squares error with respect to the filtered normal field. Though recent researches of bilateral filtering on 3D mesh [5], [6] have delivered promising results in general, preservations of sharp edges and corners on CAD-like models are still far from satisfactory. Since surface variations are best described with the first-order normal variation, we propose to apply the bilateral filtering on the surface normals first and evolve the mesh with least squares error update with respect to the filtered normal field. Our major contribution is the formulation of the intensity difference on the normal field so that bilateral filtering on surface normals can be properly applied.

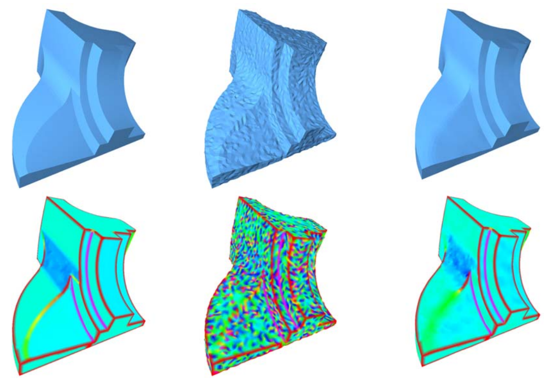


Figure 1. Denoising a model: The leftmost column is the original noise-free model. Gaussian noise is added in the middle column. The mesh is smoothed with our proposed algorithm. The top row shows the mesh details while the bottom row displays the mean curvature visualization of the mesh.

The rest of the paper is organized as follows: Section 2 will present a brief overview of related work in the literature. The mesh notation used in this paper together with the working principle of bilateral filtering will be explained in Section 3. In Section 4, we will explain the application of bilateral filtering on surface normals together with our novel formulation of intensity difference. We will also discuss how to update the mesh vertex positions based on the smoothed surface normals. We will outline our algorithm in pseudo-code. Results and comparison with other smoothing algorithms are done in Section 5 and finally conclusions are drawn in Section 6.

2. Related Work

Mesh denoising has been an on-going research problem and a wide variety of algorithms have been proposed. Taubin [1] pioneered a signal processing approach to mesh smoothing based on the definition of Laplacian on mesh. He proposed an $\lambda|\mu$ algorithm which uses alternative signed smoothing to prevent shrinkage problem associated with Laplacian smoothing. Desbrun [3] then proposed a geometric diffusion algorithm which performs smoothing in the normal direction and inhibits vertex shift in flat regions. The rate of smoothing is determined by the mean curvature. Guskov et al. [4] introduced a smoothing application from the design of a general signal processing framework based on subdivision. These algorithms mentioned so far are isotropic in nature. Thus, noise and salient features such as edges and corners are indiscriminately smoothed. To address this shortcoming, anisotropic diffusion schemes have been recently proposed.

Taubin [2] proposed a two-phase linear isotropic mesh filtering method. In his approach, surface normals are first filtered by applying a rotation determined by the weighted sum of neighboring surface normals. Then the vertex positions are updated by solving a system of linear equations using the least squares error method. He refers this as anisotropic Laplacian smoothing as the weights applied to the neighboring vertices are matrices rather than scalars. Similarly, other researchers like Yagou, Ohtake and Belyaev [11] perform median smoothing on the surface normals first and compute a mesh evolution to match the new surface normal field.

Imaging denoising is a major research area in image processing and computer vision. Recently, Fleishman et al. [6] and Jones et a. [5] have independently extended the bilateral filtering [9] from image denoising to mesh denoising and achieve satisfactory results. Fleishman approached the smoothing problem by iteratively moving the vertices in the normal direction with an offset determined from bilateral filtering of the heights of neighboring vertices over the tangent plane. On the other hand, Jones et al. compute the projections of current vertices on neighboring tangent planes and apply bilateral filtering to vertex predictions to obtain a robust estimate of vertex positions. The latest work from Hildebrandt et al. [19] not only considered sharp edges preservation but also protected and recovered non-linear surface features through their proposed prescribed mean curvature flow.

Tasdizen [8] pointed out that surface normals play an important role in surface denoising as surface features are best described with the first-order surface normals.

3. Basic Concepts

In this section, we will first define the notation used in this paper for mesh representation. Then the working principle of bilateral filtering on 2D image denoising is introduced.

3.1. Mesh Representation

Geometrically, a triangle mesh is a piecewise linear surface consisting of triangular faces pasted together along their edges. The mesh geometry can be denoted by a tuple (K,V), where K is a simplicial complex specifying the connectivity of the mesh simplices and $V=\{\mathbf{v}_1,\ldots,\mathbf{v}_m\}$ is the set of vertex positions defining the shape of the mesh in $\mathbf{R}^3$. The three different types of simplex are: 0-simplex, 1-simplex and 2-simplex. A 0-simplex, represented by $\mathbf{v}=\{\mathbf{v}_i\}$, is a vertex, a 1-simplex, $\mathbf{e}=\{\mathbf{v}_i,\mathbf{v}_j\}$, is an edge and a 2-simplex, $\mathbf{f}=\{\mathbf{v}_i,\mathbf{v}_j,\mathbf{v}_k\}$, is a face. In this paper, we represent the vertices, edges and faces by their corresponding indices. Besides, the terms face and triangle, surface and mesh are used interchangeably.

3.2. Bilateral Filtering

The bilateral filter is a nonlinear, feature preserving image filter, proposed by Smith and Brady [17], and separately by Tomasi and Manduchi [9]. Although, the filter is initially designed to be an alternative to anisotropic diffusion [7], recent researches demonstrate that it has close connections with robust estimation and anisotropic diffusion [10], [18].

Following the formulation of Tomasi and Manduchi [9], the bilateral filtering for an image $I(\mathbf{u})$, at coordinate $\mathbf{u}=(x,y)$, is defined as:

$$\widehat{I}(\mathbf{u})=\frac{\sum_{\mathbf{p}\in N(\mathbf{u})} W_c\left(\|\mathbf{p}-\mathbf{u}\|\right)W_s\left(|I(\mathbf{u})-I(\mathbf{p})|\right)I(\mathbf{p})}{\sum_{\mathbf{p}\in N(\mathbf{u})} W_c\left(\|\mathbf{p}-\mathbf{u}\|\right)W_s\left(|I(\mathbf{u})-I(\mathbf{p})|\right)}, \qquad (1)$$

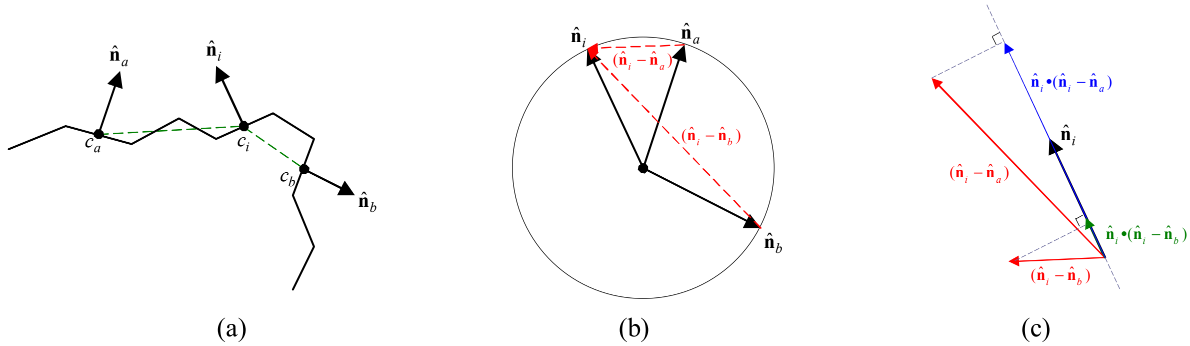


Figure 2. (a) Noisy mesh. (b) Normals within a unit circle. (c) Projections of the normal differences $(\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_a)$ and $(\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_b)$ along the $\hat{\mathbf{n}}_i$ direction.

where $N(\mathbf{u})$ is the neighborhood of $\mathbf{u}$ and defined to be the set of points $\{\mathbf{q_i} : \|\mathbf{u} - \mathbf{q_i}\| < \rho = \lceil 2\sigma_c \rceil\}$. The spatial smoothing function is a standard Gaussian filter $W_c(x) = e^{-x^2/2\sigma_c^2}$ with the standard deviation σ_c and the influence function is also chosen to be a standard Gaussian filter $W_s(x) = e^{-x^2/2\sigma_s^2}$ with the standard deviation σ_s.

The output of the filter is the weighted average of the input where the weight of each pixel is computed using a standard Gaussian function W_c in the spatial domain multiplied by an influence function W_s in the intensity domain that decreases the weight of pixels with large intensity differences. Therefore, the value at a pixel $\mathbf{u}$ is influenced mainly by pixels that are spatially close and have a similar intensity. Since large intensity differences are regarded as image features and penalized by the influence function W_s, so smoothing across features are inhibited.

4. The Algorithm

In this section, we will first show how to apply bilateral filtering on surface normals and our formulation of the intensity difference. Then, we will introduce the least squares error update of the vertex positions with respect to the filtered normal field.

4.1. Surface Normal Filtering

We extend the bilateral filtering applied on 2D images to filter the surface normals of 3D triangular meshes. For a mesh face i with unit surface normal $\hat{\mathbf{n}}_i$ and centroid at point $\mathbf{c}_i$, the bilateral filtered normal $\overline{\mathbf{n}}_i$ at the face i is defined as:

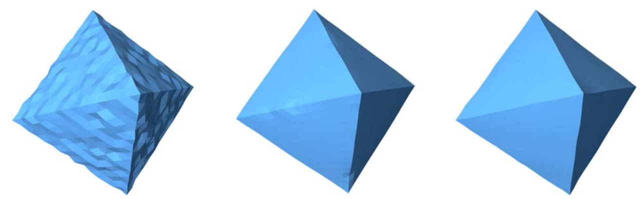


Figure 3. Left: Noisy pyramind model. Middle: Model smoothed with Fleishman's bilteral filtering on vertex positions (note the corrupted edges). Right: Model smoothed with our proposed algorithm. Sharp edges are preserved.

$$\overline{\mathbf{n}}_i = \frac{\sum_{j \in N(i)} W_c(\|\mathbf{c}_j - \mathbf{c}_i\|) W_s(d_{ij}) \hat{\mathbf{n}}_j}{\sum_{j \in N(i)} W_c(\|\mathbf{c_j} - \mathbf{c_i}\|) W_s(d_{ij})}, \tag{2}$$

where $N(i) = \{ j : \|\mathbf{c}_j - \mathbf{c}_i\| < \rho = \lceil 2\sigma_c \rceil \}$ is the set of neighborhood faces j of face i with unit surface normal $\hat{\mathbf{n}}_j$ and d_{ij} is the "intensity difference" between the two face normals $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$.

The intensity difference is defined to be the projection of the normal difference vector $(\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_j)$ on the surface normal $\hat{\mathbf{n}}_i$, i.e.

$$d_{ij} = \hat{\mathbf{n}}_i \bullet (\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_j) \tag{3}$$

In Figure 2a, face neighborhoods are considered if the Euclidean distance between the face centers are within a defined radius. In Figure 2b and 2c, normal differences and their projections along the face normal $\hat{\mathbf{n}}_i$ are computed.

Though the formulation is simple, it provides an effective measure of the degree of dispersion of the neighborhood face normals $\hat{\mathbf{n}}_j$ at the face normal $\hat{\mathbf{n}}_i$. Figure 3 illustrates the advantage of applying bilateral filtering on surface normals over Fleishman's approach in which bilateral filter is used to determine the vertex positions from the heights of the neighboring vertices defined over the tangent plane. Sharp edges along the pyramid model are properly preserved when using our proposed algorithm.

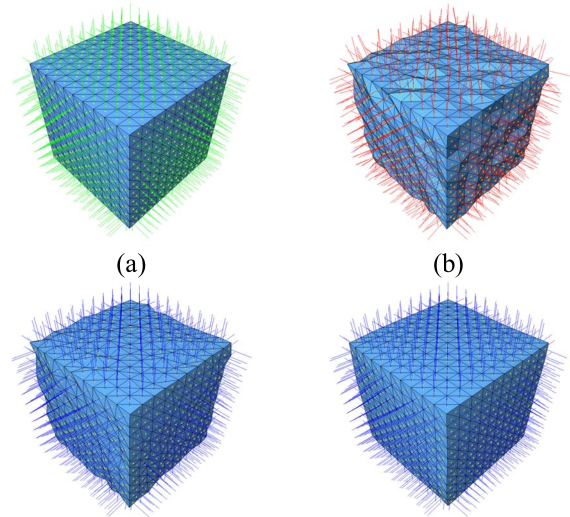
For each face i, we use the bilateral filtering in equation (2) to compute the filtered normal $\bar{\mathbf{n}}_i$, then normalize to $\bar{\mathbf{n}}'_i$ and use it as the smoothed normal. The bilateral filtering and normalization operations can be iterated to achieve a desired level of smoothing. Figure 4 above shows the results from different stages of the smoothing process. Figure 4a is the original cube with surface normals displayed. Each vertex is then displaced along the normal by zero-mean Gaussian noise with $\sigma_{\text{noise}} = 0.1$ of the mean edge length as shown in Figure 4b. In Figure 4c, the noisy surface normals are smoothed with the bilateral filter. Notice that the filtered normals are very close to the original normals. This illustrates that our formulation of "intensity difference" together with the bilateral filtering is effective at removing noise from surface normals while preserving features such as edges and corners. The final smoothed model as shown in Figure 4d is obtained via least squares error (LSE) update of the vertex positions. The LSE vertex positions update is discussed in the next section.

4.2. LSE Vertex Position Update

Since a face normal should be perpendicular to the three edges of a triangular face i, so once the filtered face normal $\bar{\mathbf{n}}'_i$ of the face is obtained, the corresponding triangle vertices $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$ are then updated under the following family of simultaneous linear equations:

$$\begin{cases} \bar{\mathbf{n}}'_i \bullet (\mathbf{v}_j - \mathbf{v}_i) = 0 \\ \bar{\mathbf{n}}'_i \bullet (\mathbf{v}_k - \mathbf{v}_j) = 0 \\ \bar{\mathbf{n}}'_i \bullet (\mathbf{v}_i - \mathbf{v}_k) = 0, \end{cases} \tag{4}$$

The system of equations (4) in reconstructing the vertex positions with respect to a given field of face normals has no solution for general meshes according to the analysis from Taubin in his paper [2]. He proposed to find the least squares solution of equation (4) which is equivalent to minimize the following cost function defined on the mesh



(a) (b) (c) (d)

Figure 4. Bilateral Filtering of Surface Normals

$$\psi(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m) = \sum_f \sum_{\{i,j\} \in \Delta f} \left(\mathbf{n}_f \bullet (\mathbf{v}_i - \mathbf{v}_j) \right)^2, \tag{5}$$

where Δf denotes the set of edges that constitute face f. The gradient of $\psi(\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m)$ with respect to $\mathbf{v}_i$ is

$$\nabla_{\mathbf{v}_i} \psi(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m) = 2 \sum_{j \in i^*} \sum_{f \in F_{ij}} \mathbf{n}_f \mathbf{n}_f^{\mathrm{T}} (\mathbf{v}_i - \mathbf{v}_j), \tag{6}$$

where F_{ij} denotes the set of faces that are adjacent to the edge $\{i, j\}$. Based on this metric and derivatives, the vertex positions can be updated as

$$\mathbf{v}'_i \leftarrow \mathbf{v}_i + \lambda \sum_{j \in i^*} \sum_{f \in F_{ij}} \mathbf{n}_f \mathbf{n}_f^{\mathrm{T}} (\mathbf{v}_j - \mathbf{v}_i),\ i = 1, 2, \ldots, m, \tag{7}$$

where λ is the iteration step size defined in Taubin's paper [2].

Figure 1d shows the smoothed mesh from the vertex positions update according to equation (6) for the given filtered surface normal field in Figure 1c.

We can now state the overall mesh denoising algorithm in the following pseudo-code:

For each face normal filtering iteration N_f:
  For each mesh face i:
    Collect neighbor faces j s.t. $\|\mathbf{c}_j - \mathbf{c}_i\| < \rho = \lceil 2\sigma_c \rceil$

Compute normalized bilateral filtered normal $\overline{\mathbf{n}}_i'$ at face i
For each vertex position update iteration N_V :
For each mesh vertex i :
Compute new vertex position with LSE update

The parameters of the algorithm are: σ_c, σ_s, λ, and the number of iterations N_f and N_v. Appropriate values for the standard deviations σ_s and σ_s are critical in features preservation. We follow the interactive approach proposed by Fleishman in setting the value of σ_c. User selects a face center on the mesh where the surface is expected to be smooth and define a radius of neighborhood from that point. This radius is then assigned to σ_c. For the value of σ_s, we set it to the standard deviation of the normal difference projected on the surface normal at the selected point within the defined radius. In our experiment, we set the values for N_f, N_v and λ to be 3, 10 and 0.01 in order to obtain quality smoothing results.

5. Results

We have implemented the mesh denoising algorithm as described in the previous section. All meshes are rendered with flat shading to show faceting. To demonstrate our feature preserving capability on CAD models, we compare our results to the results of the mesh median filtering from Yagou [11], bilateral mesh filtering from Fleishman [6] and the mean curvature flow (MCF) from Desbrun [3]. We summarize the mesh statistics and the denoising time collected on a 1.5GHz Pentium™ (M) in Table 1. In Figure 5 top row, we can see that our algorithm can deliver quality smoothing in mostly flat region while preserving the sharp edges of the tube model. Unlike the case with Fleishman's algorithm, in which large noises along the edges are mistakenly treated as features and result in corrupted edges. Though median filtering by Yagou [11] can preserve the edges, crispy appearance is unavoidable owing to the nature of order statistic-based filter. Sharp edges are completely smoothed out with MCF as feature-preserving is not considered in the original algorithm design. Similarly in Figure 5 bottom row, we can see that our algorithm performs equally well in preserving corner features of the Fandisk model. Our intensity difference formulation can not only offer a strong edge and corner preserving capability, but also be able to smooth models without losing the fine details. We test our algorithm on the Stanford Bunny model to illustrate this point. In the Figure 6c, it can be seen that the details around the eye region are properly preserved and so are the details near the nose, mouth and the leg. The mean curvature visualizations from Figure 6d to 6c provide a better comparison between the original and the smoothed model.

Table 1. Mesh statistics and denoising times. (a), (b), (c) and (d) are the denoising times collected from Fleishman's, Desbrun's, Yagou's and our algorithm accordingly.

Model	Faces/ Vertices	Denoising Time (seconds)			
		(a)	(b)	(c)	(d)
Tube	16128/8064	4	4	12	3
FanDisk	12946/6475	3	3	10	2

6. Conclusion

Bilateral filtering is proven to be a robust and efficient denoising technique in 2D images and 3D meshes. As surface normals are better in describing the surface variation, our application of bilateral filter on surface normals can best preserve the sharp edges and corners while deliver promising smoothing results. This application is made possible with our formulation of the intensity difference which helps in penalizing averaging of normals across surface features. We have shown that our algorithm is simple, easy to understand and relatively efficient as compared with other recently developed smoothing algorithms.

References

[1] G. Taubin, "A Signal Processing Approach for Fair Surface Design," SIGGRAPH '95 Conf. Proc., pp. 351-358, Aug. 1995.
[2] G. Taubin, "Linear Anisotropic Mesh Filtering," Tech. Rep. IBM Research Report RC2213, Oct. 2001.
[3] M. Desbrun, M. Meyer, P. Schroder, and A.H. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," SIGGRAPH '99 Conf. Proc., pp. 317-324, May. 1999.
[4] I. Guskov, W. Sweldens, and P. Schroder, "Multiresolution Signal Processing for Meshes," SIGGRAPH '99 Conf. Proc., pp. 325-334, Aug. 1999.
[5] T.R. Jones, F. Durand, and M. Desbrun, "Non-iterative, Feature-Preserving Mesh Smoothing," ACM Trans. Graphics, vol. 22, pp. 943-949, July 2003.
[6] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral Mesh Denoising," ACM Trans Graphics, vol. 22, pp. 950-953, July 2003.
[7] P. Persona and J. Malik, " Scale-Space and Edge Detection Using Anisotropic Diffusion," IEEE Trans. Pattern Analysis and Machine Intelligence, pp. 629-639, July 1990.
[8] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric Surface Processing via Normal Maps,"

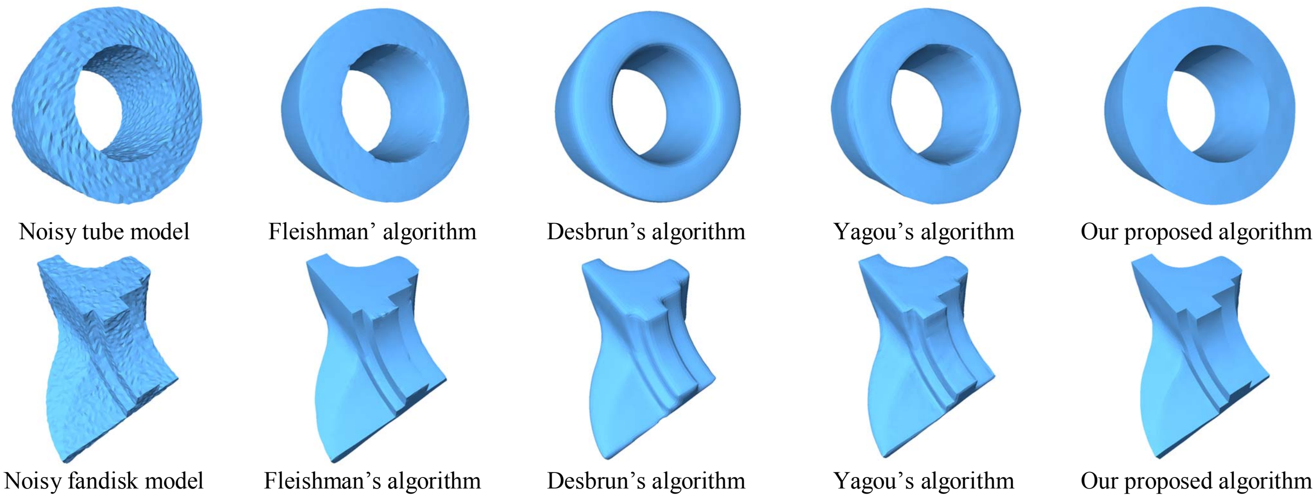


Figure 5. Comparison of our method with other smoothing algorithms. Both noisy tube and fandisk models are prepared with Gaussian noise with σ_{noise}=0.1 of the mean edge length is added. Parameters in each algorithm are chosen to smooth the mostly flat regions effectively. It can be seen that our algorithm can deliver promising smoothing quality with strong edge and corner preserving capability.

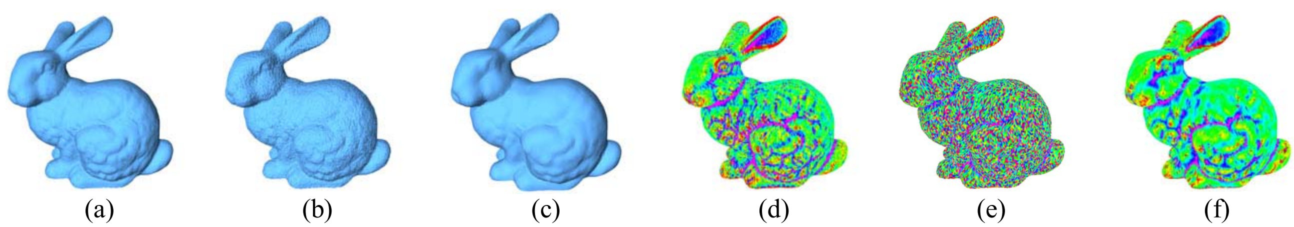


Figure 6. (a) Noise-free Stanford Bunny. (b) Gaussian noise added. (c) Smoothed model with our proposed algorithm. (d),(e),(f) Mean curvature visualization of the corresponding mesh

ACM Trans. Graphics, vol. 22, no. 4, pp. 1012-1033, 2003.

[9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Proc. IEEE Int. Conf. on Computer Vision, pp. 836-846, 1998.

[10] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 6, 2002.

[11] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh Smoothing via Mean and Median Filtering Applied to Face Normals", Geometric Modeling and Processing, pp. 121-131, 2002.

[12] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," SIGGRAPH '00 Conf. Proc., pp. 131-144, July 2000.

[13] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-Time 3D Model Acquisition," ACM Trans. Graphics, vol. 21, pp. 438-446, 2002.

[14] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," SIGGRAPH '87 Conf. Proc., pp. 163-169, 1987.

[15] H. Hoppe, "Progressive Mesh," SIGGRAPH '96 Conf. Proc., pp. 99-108, 1996.

[16] M. Deering, "Geometry Compression," SIGGRAPH '95 Conf. Proc., pp. 13-20, Aug. 1995.

[17] S.M. Smith and J.M. Brady, "SUSAN-A New Approach to Low Level Image Processing," IJCV, vol. 23, pp. 45-78, 1997.

[18] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger, " Robust anisotropic diffusion," IEEE Trans. On Image Processing, vol. 7, no. 3, pp. 421-432, 1998.

[19] K. Hildebrandt, and K. Polthier, "Anisotropic Filtering of Non-Linear Surface Features," Eurographics '04, vol. 23, no. 3, 2004.