Bilateral Filter for Meshes Using New Predictor

Yu-Shen Liu[1], Pi-Qiang Yu[1], Jun-Hai Yong[2], and Hui Zhang[2], and Jia-Guang Sun[1,2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P. R. China
liuyushen00@mails.tsinghua.edu.cn
[2] School of Software, Tsinghua University, Beijing 100084, P. R. China

Abstract. A new predictor of bilateral filter for smoothing meshes is presented. It prevents shrinkage of corners. A major feature of mesh smoothing is to move every vertex along the direction determined by the mean curvature normal with speed defined by the predictor. It prevents unnatural deformation for irregular meshes. In order to remove the normal noise, we use adaptive Gaussian filter to smooth triangle normals.

1 Introduction

Nowadays, mesh smoothing or mesh denoising, whose goal is to adjust vertex positions so that the overall mesh becomes smooth while mesh connectivity is kept, is an important process for many digital geometry applications. Removing noise while preserving important features currently is an active area of research.

Many mesh smoothing algorithms have been developed in the last few years. Taubin [9] pioneered $\lambda|\mu$ algorithm to solve the shrinkage problem caused by Laplacian smoothing. Desbrun et al. [2] extended this approach to irregular meshes using mean curvature flow. However, these techniques are isotropic, and therefore diffuse shape features. Feature-preserving mesh smoothing was recently proposed. Methods presented in [6, 7, 10] achieve this goal by first smoothing the normal field, and then updating vertex positions to match the new normals. The extension from image smoothing to mesh smoothing was explored in [1, 3, 4, 8]. The bilateral filter, which is an alternative edge-preserving image filter [11], has been extended to mesh smoothing in different ways [1, 3, 4]. Since the bilateral filter is simple, fast and well feature-preserving, it is a good choice for smoothing and denoising. However, the bilateral filter is sensitive to the initial normals, and tends to round off corners, which may result in unnatural deformation for irregular meshes.

In this paper, we present a new predictor of bilateral filter which avoids corner shrinkage. This predictor depends on normals of both a vertex and its nearby triangles. We first smooth mesh normals. Then, we move every vertex along the

J. Zhang, J.-H. He, and Y. Fu (Eds.): CIS 2004, LNCS 3314, pp. 1093–1099, 2004.

direction determined by the mean curvature normal with speed defined by the new predictor. The major contributions of our work are as follows.

- The new predictor preserves both sharp edges and corners.
- Combination of the new predictor and the method of normal improvement prevents unnatural deformation for highly irregular meshes.

2 Bilateral Filter for Meshes

A bilateral filter is an edge-preserving filter introduced in image processing for smoothing images [11]. It has been extended to mesh smoothing in different ways [1, 3, 4]. Let $\mathbf{M}$ be a input mesh with some additive noise, and let $\mathbf{s}$ and $\mathbf{p}$ be two points on $\mathbf{M}$. Jones [5] introduces the concept of *predictor*$(\Pi_{\mathbf{p}}(\mathbf{s}))$, which defines the denoised position of $\mathbf{s}$ due to $\mathbf{p}$. The bilateral filter for meshes is defined as

$$E(\mathbf{s}) = \frac{\sum\limits_{\mathbf{p}\in N(\mathbf{s})} f(||\mathbf{p}-\mathbf{s}||)g(||\Pi_{\mathbf{p}}(\mathbf{s})-\mathbf{s}||)\Pi_{\mathbf{p}}(\mathbf{s})}{\sum\limits_{\mathbf{p}\in N(\mathbf{s})} f(||\mathbf{p}-\mathbf{s}||)g(||\Pi_{\mathbf{p}}(\mathbf{s})-\mathbf{s}||)}, \tag{1}$$

where $N(\mathbf{s})$ is a neighborhood of $\mathbf{s}$, and the weight of $\mathbf{p}$ depends on both the spatial distance $||\mathbf{p}-\mathbf{s}||$ and the signal difference $||\Pi_{\mathbf{p}}(\mathbf{s})-\mathbf{s}||$. A *spatial weight* Gaussian f of width σ_f and an *influence weight* Gaussian g of width σ_g are often chosen in practice. Let $\mathbf{n_s}$ be the normal at $\mathbf{s}$, and let $\mathbf{n_p}$ be the normal at $\mathbf{p}$. Here we formally define a displacement signed-distance ds from a current position $\mathbf{s}$ to the predictor $\Pi_{\mathbf{p}}(\mathbf{s})$.

Fleishman et al. [3] have proposed an extension of bilateral filter to meshes. Their predictor can be written by

$$\Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + ((\mathbf{p}-\mathbf{s})\cdot\mathbf{n_s})\mathbf{n_s}, \tag{2}$$

where $\mathbf{p}$ is a vertex in the neighborhood of $\mathbf{s}$. It is illustrated in Fig. 1(a). The predictor does not introduce tangential *vertex drift*. However, it tends to move vertices along the normal direction to round off corners as shown in Fig. 3(b). Considering the case in which the point $\mathbf{s}$ is a corner, Fleishman et al.'s predictor moves great distance from $\mathbf{s}$ to $\Pi_{\mathbf{p}}(\mathbf{s})$ as shown in Fig. 2(a).

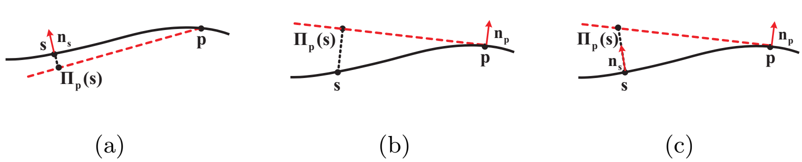


Fig. 1. (a) Fleishman et al.'s predictor. (b) Jones et al.'s predictor. (c) Our predictor

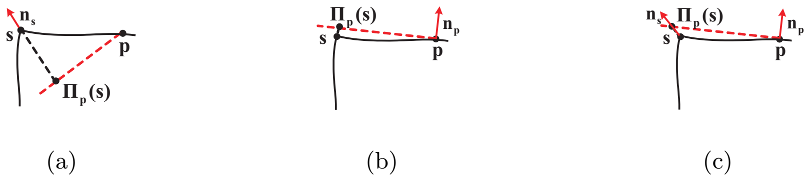


Fig. 2. $\mathbf{s}$ is a corner. (a) Fleishman et al.'s predictor. (b) Jones et al.'s predictor. (c) Our predictor

Independently, Jones et al. [4] present a similar algorithm. Their approach projects the central vertex $\mathbf{s}$ onto the planes of nearby triangles, while that of Fleishman et al. project nearby vertices onto the tangent plane of the central vertex $\mathbf{s}$. The predictor of Jones et al. can be written by

$$\Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + ((\mathbf{p} - \mathbf{s}) \cdot \mathbf{n_p})\mathbf{n_p}, \tag{3}$$

where $\mathbf{p}$ is the centroid of a triangle in the neighborhood of $\mathbf{s}$. It is illustrated in Fig. 1(b). In Fig. 2(b), Jones et al.'s predictor moves a little distance from $\mathbf{s}$ to $\Pi_{\mathbf{p}}(\mathbf{s})$ when $\mathbf{s}$ is a corner. However, since $\mathbf{s}$ does not move along the direction of normal $\mathbf{n_s}$, it may introduce tangential vertex drift. This produces unnatural deformation for irregular meshes as shown in Fig. 4(e).

2.1 New Predictor of Bilateral Filter

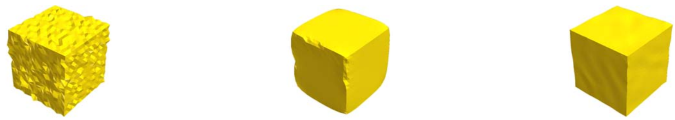
To avoid unnatural deformation arising from the predictors of Fleishman et al. and Jones et al., we present a new predictor that considers both the vertex normal and its nearby triangle normals. Our approach moves the central vertex $\mathbf{s}$ to the tangent planes of nearby triangles along the direction of the normal $\mathbf{n_s}$. The new predictor prevents corner shrinkage and tangential vertex drift. Our predictor satisfies

$$\Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + ds\mathbf{n_s} \quad \text{and} \quad (\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{p})\mathbf{n_p} = 0,$$

where $\mathbf{p}$ is the centroid of a triangle in the neighborhood of $\mathbf{s}$. By solving the above equations, we obtain

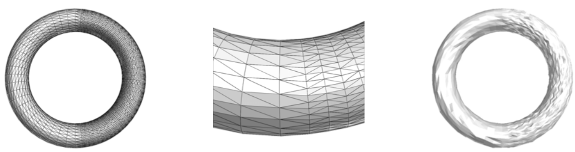
$$\Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + \left(\frac{(\mathbf{p} - \mathbf{s}) \cdot \mathbf{n_p}}{\mathbf{n_s} \cdot \mathbf{n_p}}\right)\mathbf{n_s}. \tag{4}$$

It is illustrated in Fig. 1(c). Since our predictor moves vertices along the normal direction, no vertex drift occurs. Due to the combination with nearby triangle normals, corners can be preserved. We consider the case where the point $\mathbf{s}$ is a corner. Compared with Fleishman et al.'s predictor which tends to round off corners as shown in Fig. 2(a), our predictor is able to preserve corners as

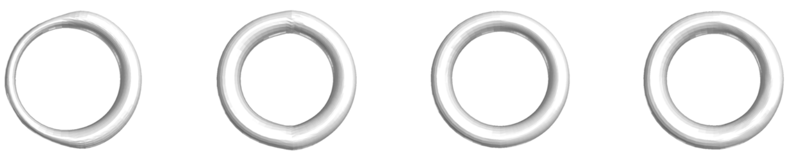


(a) (b) (c)

Fig. 3. Smoothing of CAD-like model with large noise. (a) Input noisy model. (b) Fleishman et al.'s result (5 iterations). (c) Our result (5 iterations)



(a) (b) (c)



(d) (e) (f) (g)

Fig. 4. (a) A torus with different sampling rates. (b) A magnified view of a part of the torus. (c) The torus with additive Gaussian noise in both vertex positions and normals. (d) Fleishman et al.'s method deforms the initial shape. (e) Jones et al.'s method smoothes well but slightly deforms the initial shape. (f) Mean curvature flow smoothes well. (g) Our method smoothes well as (f)

shown in Fig. 2(c). In Fig. 3 we show the smoothing results of a CAD object. In Fig. 3(b), the corners are rounded off by Fleishman et al.'s predictor, while they are preserved by our new predictor as shown in Fig. 3(c). Compared with Jones et al's predictor which introduces tangential vertex drift as shown in Fig. 2(b), our predictor moves vertices along the normal direction as shown in Fig. 2(c). Fig. 4(e) shows the result of vertex drift. Our result achieves better smoothing with respect to the shape as shown in Fig. 4(g).

3 Improving and Smoothing Normals

Fleishman et al. compute the normal at a vertex as the weighted average (by the area of the triangles) of the normals to the triangles in the 1-ring neighborhood of the vertex, where the normal direction depends on the parameterization defined by the areas of the neighborhood triangles. Moving vertex along this direction may result in unnatural deformation for highly irregular meshes (see Fig. 4(d)). To overcome this problem, we use the mean curvature normal. According to [2], a good estimation of the mean curvature normal at vertex $\mathbf{p}$ is given by

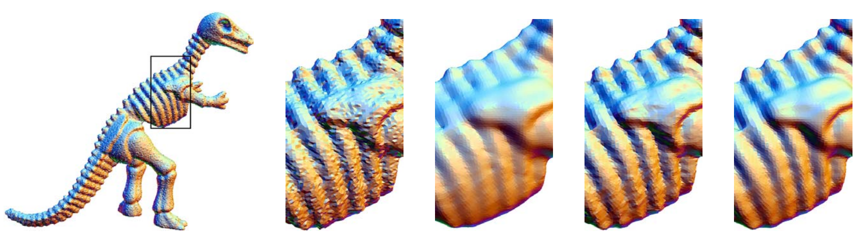
$$\mathbf{Hn}(\mathbf{p}) = \frac{1}{4A} \sum_{i \in V(\mathbf{p})} (\cot \alpha_i + \cot \beta_i)(\mathbf{q}_i - \mathbf{p}), \tag{5}$$

where A is the sum of the areas of the triangles around $\mathbf{p}$, $V(\mathbf{p})$ is the set of adjacent vertex indexes to $\mathbf{p}$, $\mathbf{q}_i$ corresponds to the i^{th} adjacent vertex to $\mathbf{p}$, and α_i and β_i are the two angles opposite to the edge $\mathbf{pq}_i$. In this paper, we use the unit vector $\mathbf{n}(\mathbf{p}) = \frac{\mathbf{Hn}}{||\mathbf{Hn}||}$ as the normal at vertex $\mathbf{p}$ instead of the normal used by Fleishman et al.. Roughly speaking, our smoothing schemes consist of moving every vertex along the direction determined by the mean curvature normal of Equation (5), with speed defined by the new predictor of Equation (4). This prevents unnatural deformation for irregular meshes (see Fig. 4(g)).

Our predictors are also based on the normals of the nearby triangles. Since the normals are sensitive to noise [4], we smooth normals by adaptive Gaussian filter applied to triangle normals [7].

4 Results and Discussion

We demonstrate our results in Figs. 4-5. The execution time is reported on a Pentium IV 1.70GHz processor with 256M RAM excluding that of loading meshes. All meshes are rendered with flat shading. In Table 1, we indicate model sizes, the number of iterations, running time, and the parameters. The σ_f and



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

Fig. 5. Results of smoothing the dinosaur model. (a) Input noisy model. (b) A magnified view of (a). (c) Fleishman et al.'s result. (d) Jones et al.'s result. (e) Our result. Notice that details such as the skeletons are better preserved by our method, while flat regions are equivalently smoothed

Table 1. Comparison of smoothing results

Model	Fig.	Verts.	Iterations	Time	σ_f	σ_g
Dinosaur	5(c)	56K	3	1.2s	Interactive	Interactive
	5(d)		non-iterative	22.5s	4.0	0.2
	5(e)		3	7.8s	4.0	0.2

σ_g are expressed as ratios of the mean edge length used by Jones et al.'s [4]. Fig. 4 shows a comparison for smoothing a irregular mesh with other algorithms. In Fig. 5, we compare our result to the results of other bilateral filter algorithm for the dinosaur model.

We have presented a novel predictor of bilateral filter which prevents shrinkage of corners. Based on this predictor and the mean curvature normal, we introduced a new mesh smoothing method which prevents unnatural deformation for irregular meshes. In the future, we wish to find a way to automatically select parameters used in bilateral filter such that smoothing is adaptively achieved.

Acknowledgements. The research was supported by Chinese 973 Program (2002CB312106). The third author was supported by the project sponsored by SRF for ROCS, SEM (041501004) and FANEDD (200342).

References

1. Choudhury, P., Tumblin, J.: The trilateral tilter for high contrast images and meshes. Proc. of the Eurographics Symposium on Rendering (2003) 186-196
2. Desbrun, M., Meyer, M., Schröder, P., and Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In SIGGRAPH'99 Conference Proceedings (1999) 317-324
3. Fleishman, S., Drori, I., and Cohen-Or, D.: Bilateral mesh denoising. In SIGGRAPH' 03 Conference Proceedings (2003) 950-953
4. Jones, T., Durand, F., and Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. In SIGGRAPH' 03 Conference Proceedings (2003) 943-949
5. Jones, T.: Feature preserving smoothing of 3D surface scans. Master's Thesis, Department of Electrical Engineering and Computer Science, MIT (2003)
6. Ohtake, Y., Belyaev, A., and Bogaevski, I.: Mesh regularization and adaptive smoothing. Computer-Aided Design **33** (2001) 789-800
7. Ohtake, Y., Belyaev, A., and Seidel, H.P.: Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normal. In Vision, modeling and visualization (2002) 203-210
8. Shen, Y., Barner, K.E.: Fuzzy vector median-based surface smoothing. IEEE Transactions on Visualization and Computer Graphics **10** (2004) 252-265

9. Taubin, G.: A signal processing approach to fair surface design. In SIGGRAPH'95 Conference Proceedings (1995) 351-358
10. Taubin, G.: Linear Anisotropic mesh filtering. Tech. Rep. IBM Research Report RC2213 (2001)
11. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In Proc. IEEE Int. Conf. on Computer Vision (1998) 836-846