Visual Comput (2009) 25: 139–148
DOI 10.1007/s00371-008-0210-7

ORIGINAL ARTICLE

Zhong Li
Lizhuang Ma
Xiaogang Jin
Zuoyong Zheng

A new feature-preserving mesh-smoothing algorithm

Published online: 15 February 2008
© Springer-Verlag 2008

Z. Li (✉)
Department of Mathematics and Science, Zhejiang Sci-Tech University, Hangzhou 310018, China
lizhongzju@hotmail.com

Z. Li · L. Ma · Z. Zheng
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, China

X. Jin
State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China

Abstract We present a novel mesh denoising and smoothing method in this paper. Our approach starts by estimating the principal curvatures and mesh saliency value for each vertex. Then, we calculate the uniform principal curvature of each vertex based on the weighted average of local principal curvatures. After that, we use the weighted bi-quadratic Bézier surface to fit the neighborhood of each vertex using the least-square method and obtain the new vertex position by adjusting the parameters of the fitting surface. Experiments show that our smoothing method preserves the geometric feature of the original mesh model efficiently. Our approach also prevents the volume shrinkage of the input mesh and obtains smooth boundaries for non-closed mesh models.

Keywords Mesh smoothing · Principal curvature · Bi-quadratic Bézier surface · Least-square fitting

1 Introduction

With the fast development of geometry scanners such as 3D CT and MRI scanners, robust and efficient geometry processing becomes increasingly desirable. While the acquired 3D models inevitably contain noise for different reasons, it is necessary to remove noise and obtain a smooth mesh model before further mesh processing. Therefore, the mesh denoising and smoothing research has been a subject of intensive interest in computer graphics, computer-aided design (CAD) and other related fields [18].

For the mesh denoising and smoothing, early methods were based on the norm optimization by constructing the energy equation [20]. With the development of image-denoising techniques, these methods were extended to the mesh denoising. Taubin [19] introduced signal processing on surfaces based on the definition of the Laplacian operator on meshes. Desbrun et al. [2] provided the geometric diffusion and anisotropic diffusion algorithms for meshes. Fleishman et al. [3] and Jones et al. [5] presented the bilateral mesh denoising. Another mesh-smoothing strategy is based on the curvature uniformity. It reduces the curvature variance of each vertex to realize the mesh smoothing [6]. As for the mesh fairing, the main difficulties are how to preserve the features and prevent the volume shrinkage when denoising the mesh. For non-closed mesh models, the denoised objects should keep the smooth boundaries. How to obtain a robust, feature- and volume-preserved mesh-smoothing algorithm is still a challenging problem for researchers.

In this paper, we present a new mesh denoising and smoothing method. We estimate the principal curvatures and mesh saliency value of each vertex on the mesh model. The mesh saliency value is taken as a weighting factor to obtain the uniform principal curvature of each vertex. We use the weighted bi-quadratic Bézier surface to fit each vertex and its neighborhood by the least-square method. The smoothing vertex is obtained by adjusting the parameters of the local fitted surface. Compared to other mesh denoising and smoothing algorithms, our method prevents the volume shrinkage and keeps the smooth boundaries because we adopt the uniform principal curvature smoothing strategy. Our method also preserves the geometric features of mesh models because we set the mesh saliency value as the weighting factor. Since we use the popular bi-quadratic Bézier surface to fit the local neighborhood of each vertex, our smoothing algorithm is robust.

2 Related work

Recently, mesh denoising and smoothing research has been a hot topic in the fields of computer graphics, CAD, etc. The early smoothing methods were built up by simplifying the original mesh model or by implicit surface modeling [10, 12]. In these methods, the number of vertices and facets is usually changed and noise sometimes is hard to be eliminated. Other popular smoothing methods are based on the minimum-energy function. Welch and Witkin [20] and Kobbelt [7] provided the curvature estimation for the discrete mesh model, and gave a smooth solution by constructing the energy equation. The problems of these methods are that the computation process is complex and the smoothing speed is slow.

In 1995, Taubin [19] presented the Laplacian operator, a signal-processing approach to fair surface design. The Laplacian smoothing is an iterative process. In each step, every vertex of the mesh is moved to the barycenter of its neighborhood. Since it is a linear operator, the smoothing process is fast. The main problem of Laplacian smoothing is the volume shrinkage. Later, Taubin modified the Laplacian algorithm to solve this problem. Desbrun et al. [2] extended Taubin's work and provided an implicit fairing method using diffusion and curvature flow. This implicit solver gives a better solution to the volume-shrinkage problem. Recently, some researchers also provided other feature-preserving denoising methods based on anisotropic diffusions [16], the mesh fairing based on the bilateral filter [3, 5] and the trilateral filter [1], etc. In the denoising and smoothing process, because features and noise sometimes are treated simultaneously, how to obtain an efficient feature-preserving smoothing algorithm is still a challenging problem.

Another smoothing strategy is based on the uniform principal curvature. As the goal of this method is to reduce the curvature variance among the vertices on the mesh model, the difference between the uniform curvature and the original curvature on each vertex is set as the adjusting reference. The advantage of this method is that it preferably keeps the volume of a mesh and obtains smooth boundaries for non-closed mesh models. Karbacher and Haeusler [6] proposed a uniform principal curvature method for the mesh smoothing by using the arc-fitting process. This method prevents the volume shrinkage, but it may lead to a sphere shape on each vertex's neighborhood [9].

With further researches for 3D mesh models, there exist some novel ideas to process them [8, 17]. We adopt the ideas of these methods in our mesh denoising and smoothing process. We set the mesh saliency value as the weighting factor and use the robust weighted bi-quadratic Bézier surface which adds some factors to fit local areas on the mesh model. Our uniform curvature smoothing method efficiently prevents the volume shrinkage, preserves the geometric features and keeps a smooth boundary for non-closed mesh models.

2.1 Notation

A triangular mesh is denoted by M, which consists of an ordered set of n vertices and a set of triangles made up of these vertices. The triangular mesh is also rewritten as $M = (V, E)$, where V is the set of all vertices and E is the set of all edges. For the vertex q_i, the normal vector, the mean curvature and the Gaussian curvature are expressed by n_i, H and K, respectively. The 1-ring neighborhood of q_i is defined as a set of all the vertices adjacent or connected to q_i by an edge. Similarly, we denote the 2-ring neighborhood of q_i.

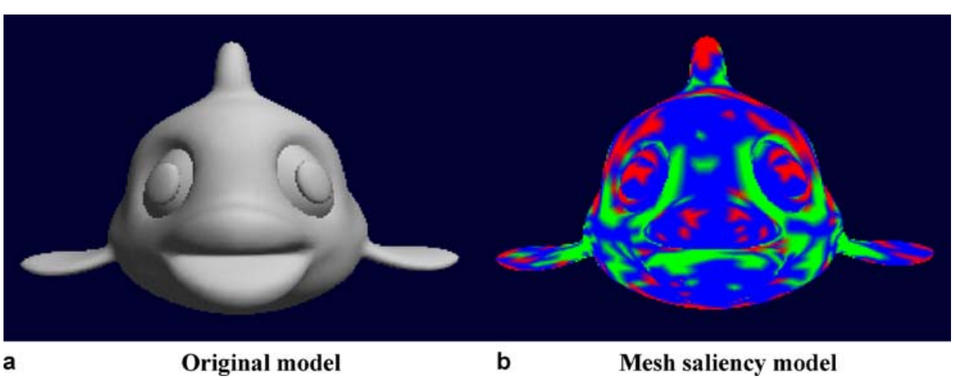


Fig. 1a,b. Mesh saliency of the fish model

2.2 Mesh saliency

Mesh saliency was proposed by Lee et al. in 2005 [8], which combines the perceptual criteria by the low-level human visual system cues and the geometric properties based on discrete differential geometry for 3D meshes. It successfully captures salient regions in meshes, as shown in Fig. 1. In the mesh saliency method, a robust multi-scale measure is defined by the curvatures. Namely, we need to construct the mesh saliency function to capture the salient character of each vertex and unveil the difference between the vertex and its surrounding area. The mesh saliency method has been applied in many fields of mesh processing [8, 11]. Here, we modify the mesh saliency method into our denoising and smoothing algorithm.

For the main operations of mesh saliency, the key step is to choose the curvature of each vertex on the mesh model. Lee et al.'s method [8] defined the mesh saliency using the mean curvature. Later, Mao et al. [11] modified the mesh saliency by the largest absolute value of the principal curvature. From the view of cognition, Hoffman and Singh [4] presented a part saliency theory which sets up the minima rule. They considered that human vision defines part boundaries at the negative minima of curvature on silhouettes, and along negative minima of the principal curvatures on surfaces. Therefore, we set the negative minima of principal curvatures as the curvature reference in our presented approach.

3 Mesh fairing based on uniform principal curvature and mesh saliency

The key idea of our mesh fairing method is that we estimate the principal curvatures and the mesh saliency value for each vertex. Mesh saliency is then adopted as the weighting factor in the following fairing process. We construct the weighted bi-quadratic Bézier surface to fit each vertex and its neighborhood by the least-square method. According to the uniform principal curvature, we set the new parameter (u_i, v_i) on the local fitted surface and compute the smoothing vertex of the triangular mesh.

3.1 The estimation of normal vector and principal curvatures of each vertex

We need to estimate the normal vector of every vertex on the mesh model for the following operations. Here, the normal vector of vertex q_i is set as the weighted average of all normal vectors of neighboring triangles, where the weighting factor is the area of each triangle.

We also need to estimate the principal curvatures of each vertex on the mesh model. There are some estimation algorithms. Here, we use Meyer et al.'s method [13], which comparatively obtains the accurate mean curvature and the Gaussian curvature, as shown in Fig. 2. The mean curvature H and the Gaussian curvature K at q_i are computed as follows:

$$H = \frac{\Delta_M \cdot n}{2}, \quad K = \frac{2\pi - \sum_j \theta_j}{A},$$

where n is the normal vector of each vertex, $\Delta_M = (1/A)\sum_{j\in N(i)} \frac{\cot\alpha_{ij}+\cot\beta_{ij}}{2}(q_j - q_i)$, A is the Voronoi area of 1-ring neighborhood $N(i)$, θ_j denotes the angle composed of q_iq_j and q_iq_{j+1}, $\alpha_{ij} = \angle q_iq_{j-1}q_j$ and $\beta_{ij} = \angle q_iq_{j+1}q_j$.

The principal curvatures are then computed from the mean curvature and the Gaussian curvature.

3.2 Fitting the vertex and its neighborhood by the weighted bi-quadratic Bézier surface

A bi-quadratic Bézier surface is written as

$$B(u, v) = \sum_{i=0}^{2}\sum_{j=0}^{2} B_{i,2}(u)B_{j,2}(v)b_{ij}, \quad u, v \in [0, 1],$$

where $B_{i,2}(u)$, $B_{j,2}(v)$ are Bernstein basis functions; the b_{ij} are the Bézier control points which form the Bézier control net.

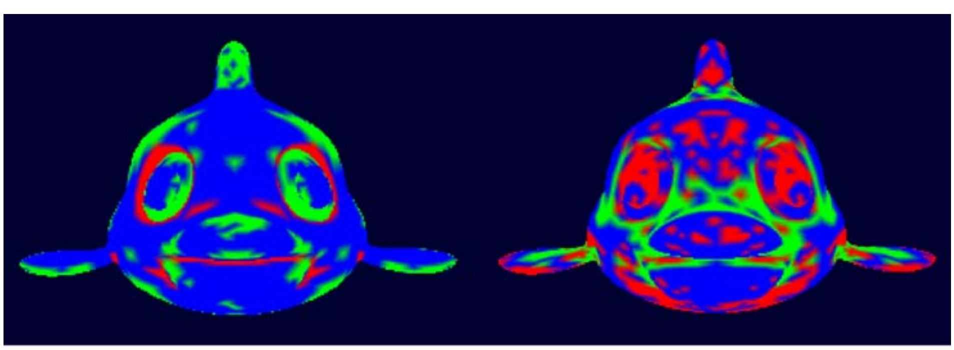


Fig. 2a,b. Curvatures of the fish model

For fitting a vertex and its neighborhood in the mesh surface, popular methods use the quadratic algebraic surface [14, 21] through the least-square method. Here, we use the bi-quadratic Bézier surface to fit the vertex and its local area. The advantages of parametric surface fitting are that it is designed easily and conveniently, and for the mesh surface with noise, we are able to add the adjusting matrix and factor to fit the local neighborhood, which guarantee the more accurate fitting surface [17].

When using the bi-quadratic Bézier surface to fit the vertex and its neighborhood, we normally regard all vertices in the 1-ring neighborhood of vertex q_i as the fitted points. In this paper, in order to make the fitted surface closer to the original mesh surface, we set all vertices in the 2-ring neighborhood of vertex q_i as the fitted points, as shown in Fig. 3.

To fit a given vertex q_i and all vertices in the 2-ring neighborhood by the bi-quadratic Bézier surface, we need to compute the positions of control points b_{ij} according to these given vertices. Supposing that the parameter of each vertex in the fitted surface is (u_i, v_i), we require to obtain all these parameters at first. In this work, these parameters are calculated by projecting vertices onto a tangent plane and scaling them to the [0, 1] range.

Supposing that there are n vertices in the 2-ring neighborhood of vertex q_i, we compute the approximate normal vector N on q_i by the arithmetic average of all normal vectors on given vertices. We build up a tangent plane which is vertical to N and set q_i as the origin of coordinates. Other vertices in the neighborhood are projected in this tangent plane. If some projected points of n vertices in the tangent plane are the same position or create a folding situation, we change the origin of coordinates and recalculate the approximate normal vector to obtain the new tangent plane, or we restrict the vertices in the 1-ring neighborhood as the projected points to avoid those cases. Then, we construct local Cartesian coordinates in this tangent plane. The coordinates of all projected points are enclosed by a min–max box. This min–max box is scaled to the [0, 1] range; the coordinates of projected points in this range are regarded as corresponding parameters of given vertices, as shown in Fig. 4.

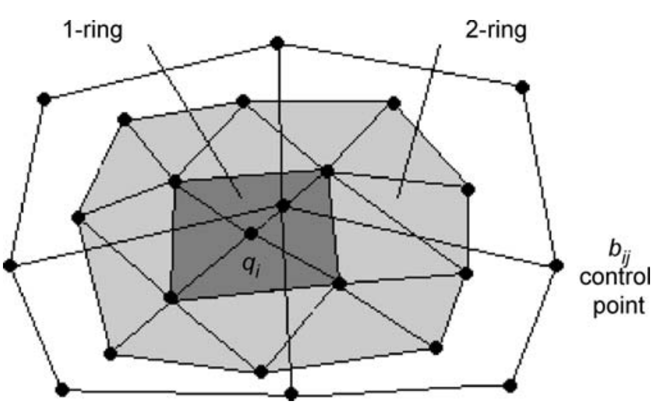


Fig. 3. Fit of the vertex and its 2-ring neighborhood by bi-quadratic Bézier surface

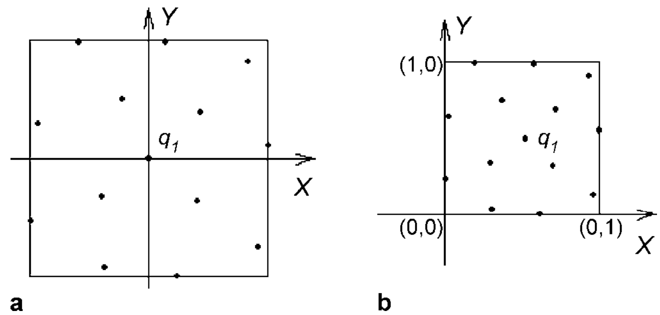


Fig. 4a,b. The parameter acquisition of all fitted vertices

For the vertex q_i and other vertices in its 2-ring neighborhood, after obtaining the corresponding parameter (u_i, v_i) of all these vertices, we obtain a linear equation system $Ax = B$, where

$$A = \begin{bmatrix} B_0^2(u_0)B_0^2(v_0) & B_0^2(u_0)B_1^2(v_0) & \cdots & B_2^2(u_0)B_2^2(v_0) \\ B_0^2(u_1)B_0^2(v_1) & B_0^2(u_1)B_1^2(v_1) & \cdots & B_2^2(u_1)B_2^2(v_1) \\ & & \vdots & \\ B_0^2(u_n)B_0^2(v_n) & B_0^2(u_n)B_1^2(v_n) & \cdots & B_2^2(u_n)B_2^2(v_n) \end{bmatrix},$$

$$x = [b_{0,0} \quad b_{0,1} \quad \cdots \quad b_{2,2}]^{\mathrm{T}},$$

$$B = [q_0 \quad q_1 \quad \cdots \quad q_n]^{\mathrm{T}}.$$

From this equation system, we compute x to obtain the control points b_{ij}, which construct the bi-quadratic Bézier surface.

Because there may be noise in the original mesh model, we add the adjusting matrix and the adjusting factor to modify the fitted surface by the least-square method. The modified equation system is as follows [17]:

$$\begin{bmatrix} \alpha A \\ (1-\alpha)S \end{bmatrix} [x] = \begin{bmatrix} \alpha B \\ 0 \end{bmatrix},$$

where A, x and B are given in the above definitions; the matrix S is added to make each of the four quadrilaterals formed by the $b_{i,j}$ as close to a parallelogram as possible, which is set as

$$S = \begin{bmatrix} 1 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 \end{bmatrix}.$$

The factor α is added to adjust the fitting surface's shape in [0, 1] depending on the intensity of noise in the mesh model. When the noise of a mesh model is heavy, we assign a small value to α. Otherwise we assign a large value to α. Here, α is chosen according to the mesh saliency value. When the difference of mesh saliency values in the neighborhood of vertex q_i is large, there may be heavy noise in the local area, so we set a small α value on q_i. When the difference of mesh saliency values in the neighborhood is small, we set a large α value on q_i. Supposing that the maximum and minimum of mesh saliency in the neighborhood are $\text{Saliency}_{\text{Max}}$ and $\text{Saliency}_{\text{Min}}$, α is chosen as $\text{Saliency}_{\text{Min}}/\text{Saliency}_{\text{Max}}$. When $\text{Saliency}_{\text{Max}}$ is close to zero, we understand that there is little noise in the neighborhood, so we directly set α as 1.

For the above equation system, we use the least-square method to obtain the control points of the weighted bi-quadratic Bézier surface. At this time, the corresponding point $B(u_i, v_i)$ on the fitted surface is identical to vertex q_i.

3.3 The vertex adjustment based on the uniform principal curvature

The motivation of our uniform principal curvature method comes from the Euler formula. We know that if two principal curvatures and two principal directions are the same on two different points in the surface, then the normal section curvatures of the random direction on two points are also the same. Based on the observation, we conclude that if the principal curvatures vary uniformly in the neighborhood of one vertex, then the normal section curvatures vary uniformly; so do the mean curvature and the Gaussian curvature [9]. We also notice that the sphere is a smooth surface and one invariant property is the curvature. So, we use the uniform principal curvature for the vertex adjustment in this paper. Since the principal directions approximated for the mesh vertices are not necessarily consistent in orientation, we define the weighted average of principal curvatures in the neighborhood as the uniform principal curvature on vertex q_i. After obtaining the uniform principal curvatures on vertex q_i, we look for the new point on the weighted bi-quadratic surface whose principal curvatures are equal to the uniform principal curvatures. We regard this new point on the fitting surface as the smoothed vertex, as shown in Fig. 5.

We construct local Cartesian coordinates $Oxyz$ on vertex q_i, whose the z axis is identical to its normal direction and the x and y axes are the same as two principal directions in the tangent plane, as shown in Fig. 6. We use the Euler formula to obtain the uniform principal curvatures.

Supposing that two principal curvatures on vertex q_i are k_1 and k_2, for the neighboring vertex q_j, θ_{1j} and θ_{2j} are two angles composed of the projected direction of its principal direction and the x axis. Two corresponding normal section curvatures k_{1j} and k_{2j} on vertex q_i are

$$k_{1j} = k_1 \cos\theta_{1j}^2 + k_2 \sin\theta_{1j}^2,$$
$$k_{2j} = k_1 \cos\theta_{2j}^2 + k_2 \sin\theta_{2j}^2.$$

The uniform principal curvatures on vertex q_i are computed by the weighed average of all the above normal section curvatures. The weighting factor w_j is set by the mesh saliency value of each vertex q_j. So, the uniform principal curvatures k_1', k_2' on q_i are computed as follows:

$$k_1' = \sum (k_{1j} \cdot w_j) / \sum w_j,$$
$$k_2' = \sum (k_{2j} \cdot w_j) / \sum w_j.$$

For the mesh vertex q_i, namely, the point $B(u_i, v_i)$ on the weighted bi-quadratic Bézier surface, we require to find the new parameter (u_i', v_i') to obtain the new point $B(u_i', v_i')$, whose principal curvatures are equal to the uniform principal curvatures. Here, we provide a solution to obtain the new parameter.

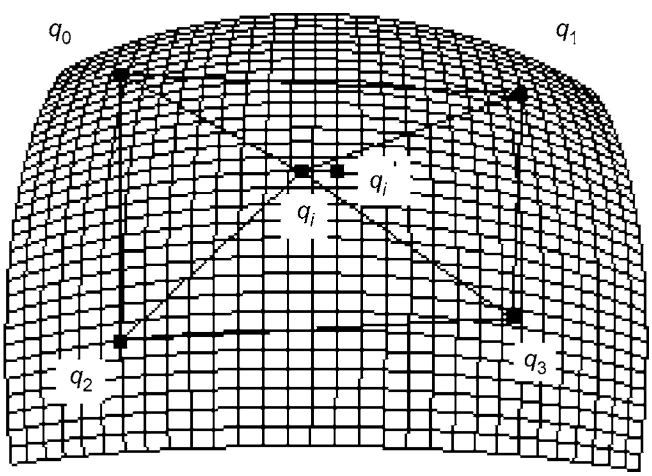


Fig. 5. Smoothing the vertex q_i on the fitting surface

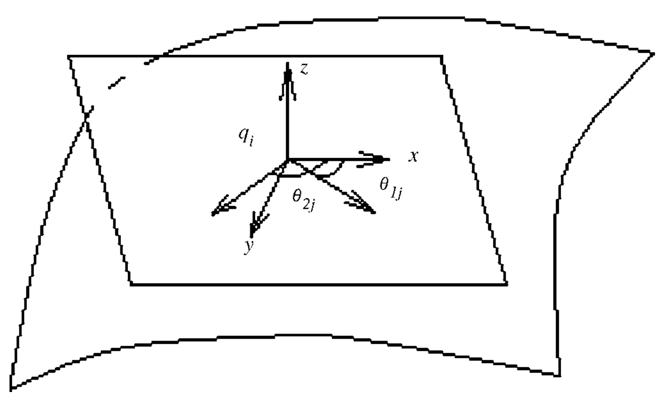
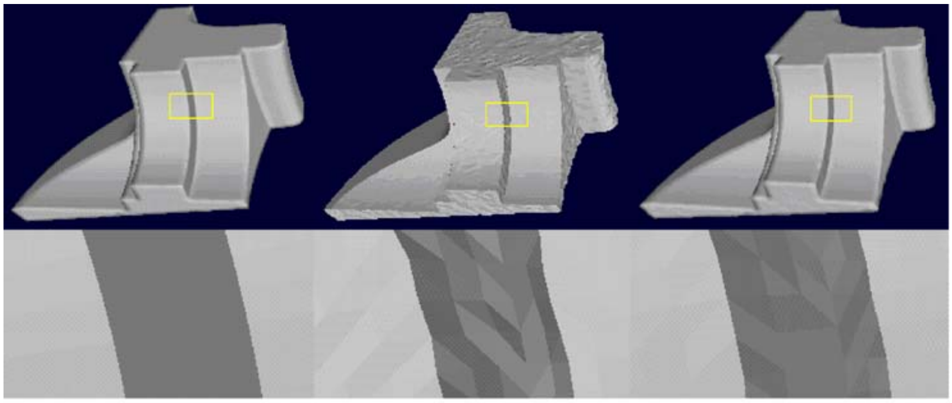
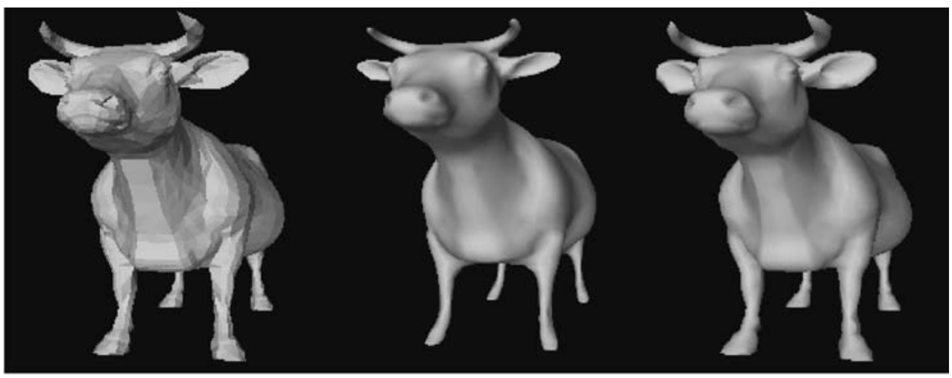


Fig. 6. The local coordinates on vertex q_i



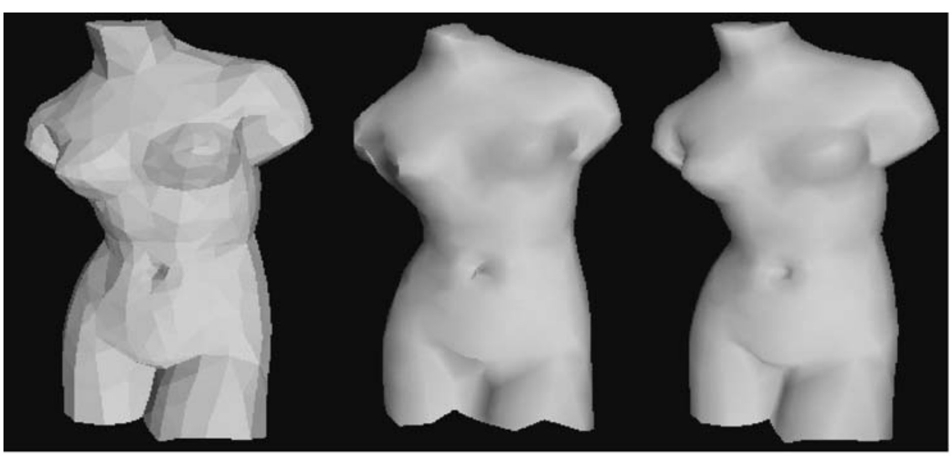
a Original model b Noised model c Our smoothing result

Fig. 7a–c. Fan disk mesh model



a Noised model b Laplacian method's result c Our smoothing result

Fig. 8a–c. Cow mesh model (the volume variance ratio of the Laplacian method is 0.8998, the ratio of our method is 0.9942)



a Noised model b Desbrun's result c Our smoothing result

Fig. 9a–c. Venus mesh model (the belly button part appears with the vertex drifting, the neck and leg edge sections produce a morphing shape in Desbrun et al.'s method)

For the point $B(u_i, v_i)$ on the parametric surface, the unit normal vector is computed as

$$n = \frac{B_u \times B_v}{|B_u \times B_v|}.$$

Supposing that $E = B_u^2$, $F = B_u B_v$, $G = B_v^2$, $L = B_{uu} \cdot n$, $M = B_{uv} \cdot n$ and $N = B_{vv} \cdot n$, according to the differential geometry formula, the Gaussian curvature and the mean curvature on this point are computed as

$$K = \frac{LN - M^2}{EG - F^2}, \quad H = \frac{NE - 2MF + LG}{2(EG - F^2)}.$$

Obviously, for given K and H, we obtain two equations for (u_i', v_i'), which are solved by the following equation system:

$$K\left(u_i', v_i'\right) = k_1' \cdot k_2', \quad H\left(u_i', v_i'\right) = \frac{k_1' + k_2'}{2}.$$

For the above equation system, we use the Newton iteration method to obtain the solution (u_i', v_i'), and the initial iterative values are chosen as (u_i, v_i). Because the Newton iteration method converges fast, we set the iterative number as 2. After the Newton iteration process is finished, if the new parameter is less than 0, we approximately set it as 0; if the parameter is larger than 1, we set it as 1. So, we guarantee that the point $B(u_i', v_i')$ is found on the fitted surface by the new parameter. The point $B(u_i', v_i')$ on the fitting surface is the smoothed vertex of q_i.

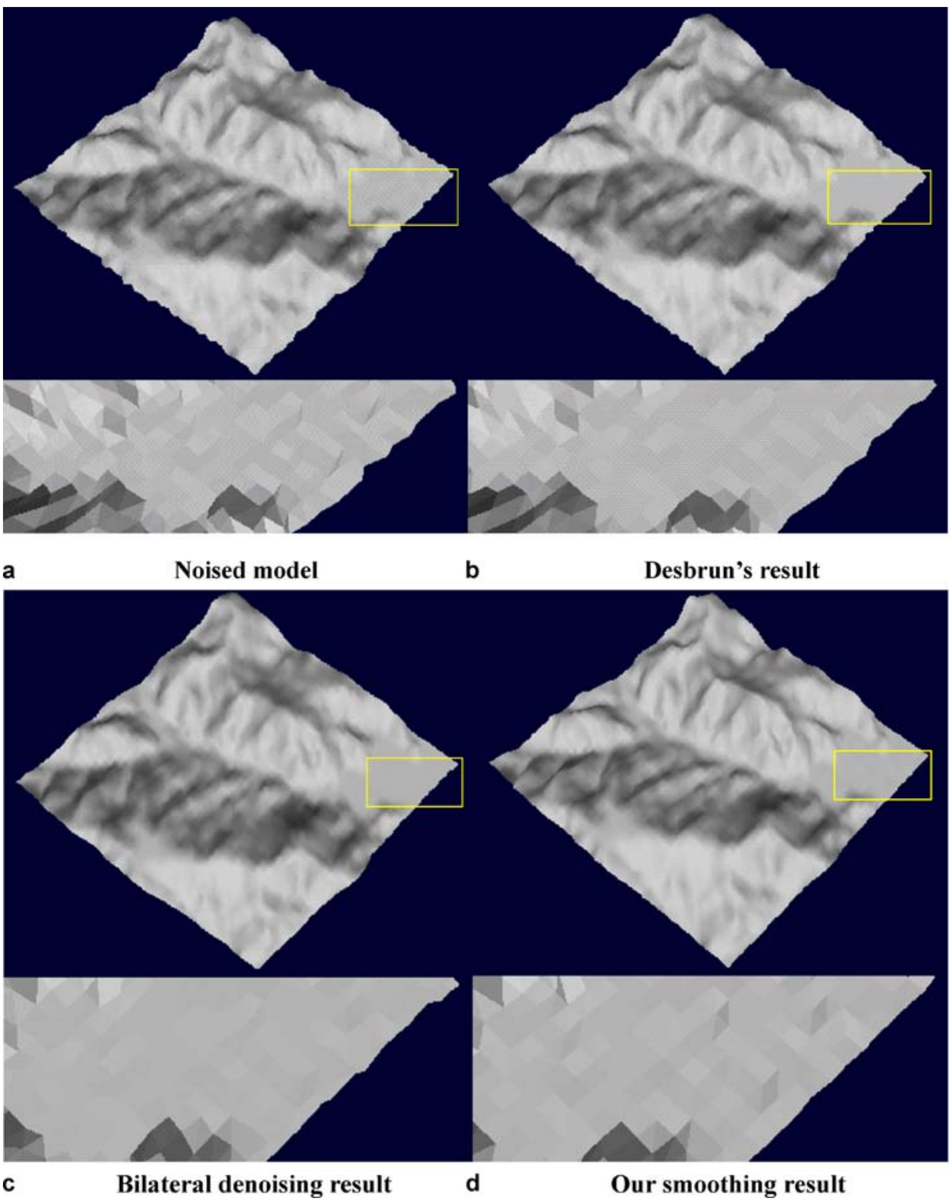


Fig. 10a–d. Mountain mesh model

4 Experimental results

We use VC++6.0 and OpenGL to implement our mesh-smoothing algorithm. Here, the main computer configuration is with a 1.8 GHz P4 CPU and 256 MB RAM. We use different mesh models to test and explore the efficiency and robustness of the proposed algorithm. In this work, we add the Gaussian noise to the mesh model. Figure 7a is the original fan disk mesh model. Figure 7b is the noised mesh model. Figure 7c is our smoothing result for the noised mesh model. The lower parts of Fig. 7 are the close-up figures from the feature section. The experimental results show that our smoothing algorithm preserves the geometrical features of the mesh model.

Figure 8a is a cow mesh model with noise. Figure 8b is the smoothing result using the Laplacian method after five iterations (setting the smoothing parameter $\lambda = 0.5$). We notice that the horns, ears and legs are shortened and the whole volume of the cow model is reduced markedly. Figure 8c is the smoothing result of our algorithm. We use Desbrun et al.'s method [2] to measure the ratio of the original volume and the volume after the model is smoothed; the ratio of the Laplacian method with five iterations is 0.8998, while the ratio of our method is 0.9942. Our approach efficiently prevents the volume shrinkage.

Figure 9a is a Venus mesh model with noise. Figure 9b is the denoising result by Desbrun et al.'s algorithm after three iterations. We find some problems such as the belly button part appearing with the vertex drifting, and the neck and leg edge parts producing a morphing shape. Figure 9c is the experimental result by our smoothing algorithm, which successfully avoids these problems.

Figure 10a is a non-closed mountain mesh model with noise. Figure 10b is the smoothing result by Desbrun et al.'s algorithm. On some boundaries of the mesh model there appear some zigzags. Figure 10c is the result by the bilateral denoising method. Figure 10d is the smoothing result by our algorithm. The lower parts of Fig. 10 are the close-up figures from the boundary section. Compared to other smoothing methods, our algorithm obtains satisfying smoothing boundaries.

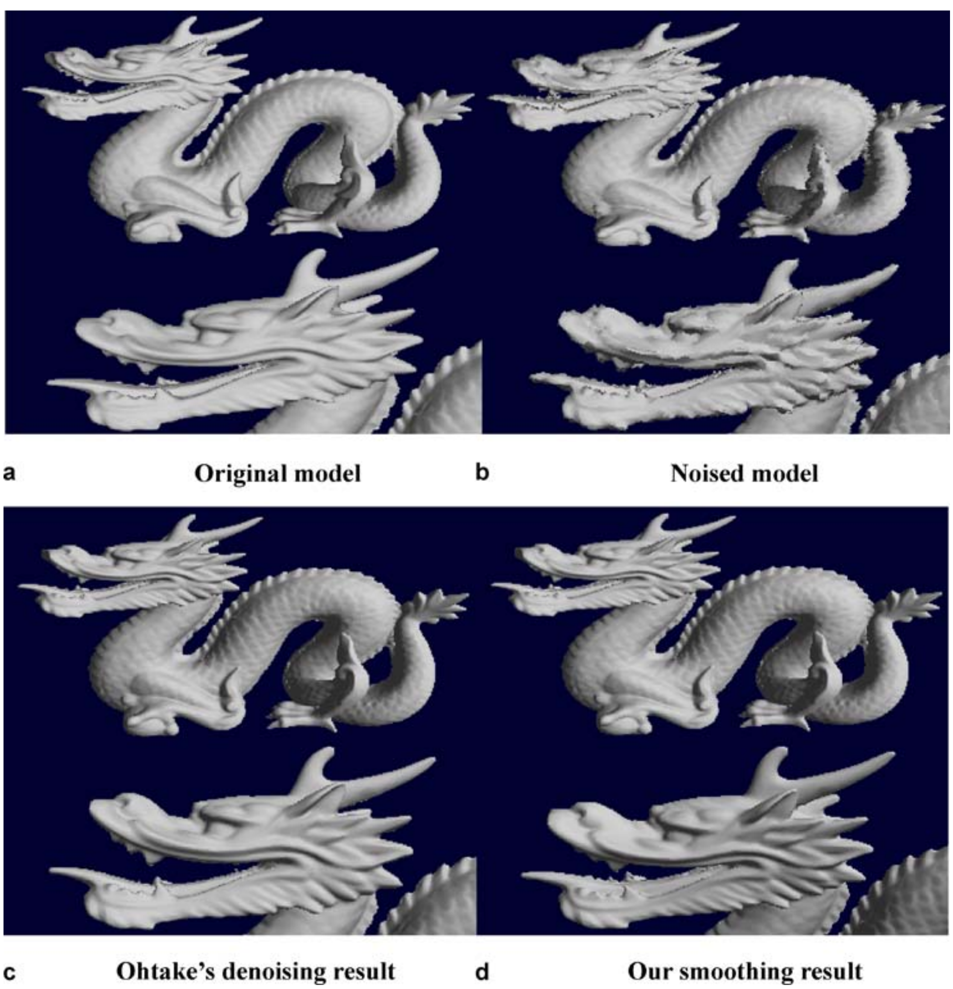


Fig. 11a–d. Dragon mesh model

Table 1. The time comparison of different algorithms for smoothing noised mesh models (s), including the normal vector computation of each vertex

	Vertices	Triangles	Laplacian operator	Desbrun et al.'s method	Bilateral denoising	Our algorithm
Fan disk	6475	12 946	2.72	5.93	6.65	8.29
Cow	2903	5804	1.24	2.69	2.94	3.75
Venus	711	1396	0.29	0.65	0.73	0.91
Mountain	2500	4802	1.07	2.32	2.59	3.19
Dragon	100 000	200 000	50	98	106	131

Figure 11a shows a large dragon mesh model. Figure 11b is the noised mesh model. Figure 11c is the smoothing result by Ohtake et al.'s denoising method [15] since the source code is available online. Figure 11d is the fairing result by our algorithm. The lower parts of Fig. 11 are the close-up figures from the dragon head section. Our algorithm is applicable for the large-mesh model and is feature preserving.

Table 1 lists the time statistics for different denoising algorithms to smooth the above mesh models. Considering the time complexity of the smoothing algorithm, our method is slower than the classical Laplacian algorithm, Desbrun et al.'s algorithm and the bilateral algorithm, but the time difference is not very great, especially for the small-mesh models.

5 Conclusion and future work

In this paper, we adjusted the mesh saliency computation and modified the weighted bi-quadratic Bézier surface to fit the vertex and its neighborhood by the least-square method. We provide a new uniform principal curvature strategy for the mesh smoothing. The smoothed vertex is obtained by adjusting the parameter on the local fitting surface. The algorithm is robust, preserves the features, prevents the volume shrinkage and obtains a smooth boundary for non-closed mesh models.

Currently, our denoising and smoothing algorithm aims at the manifold mesh model including closed and non-closed cases. Smoothing the non-manifold mesh model is one of our future researches. How to improve our algorithm's performance using fast approximation methods, for example to approximately compute the uniform principal curvature or to calculate the new parameter on the fitted surface by other fast numerical methods, also deserves future research.

Acknowledgement The authors are very grateful to the anonymous referees for the useful comments and suggestions which improved this paper considerably. This research was supported by the National Natural Science Foundation of China under Grant No. 60573147 and the Natural Science Foundation of Zhejiang Province of China under Grant No. Y106207. Xiao-gang Jin is supported by the China 973 program under Grant No. 2002CB312101 and the China 863 program under Grant No. 2006AA01Z314.

References

1. Choudhury, P., Tumblin, J.: The trilateral filter for high contrast images and meshes. In: Proceedings of the Eurographics Symposium on Rendering, pp. 186–196. Eurographics Association, Leuven (2004)
2. Desbrun, M., Meyer, M., Schroder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH'99, pp. 317–324. ACM Press, Los Angeles, CA (1999)
3. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. In: Proceedings of SIGGRAPH'03, pp. 950–953. ACM Press, San Diego, CA (2003)
4. Hoffman, D., Singh, M.: Salience of visual parts. Cognition **63**(1), 29–78 (1997)
5. Jones, T., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. In: Proceedings of SIGGRAPH'03, pp. 943–949. ACM Press, San Diego, CA (2003)
6. Karbacher, S., Haeusler, G.: A new approach for modeling and smoothing of scattered 3D data. In: Proceedings of SPIE'03, pp. 168–177. SPIE Press, San Jose, CA (2003)
7. Kobbelt, L.: Discrete fairing. In: Proceedings of the 7th IMA Conference on the Mathematics of Surfaces, pp. 101–131. Springer, Cirencester (1996)
8. Lee, C., Varshney, A., Jacobs, D.: Mesh saliency. In: Proceedings of SIGGRAPH'05, pp. 659–666. ACM Press, Los Angeles, CA (2005)
9. Liu, S., Zhou, R., Nei, J., Zhou, L.: Mesh smoothing using principal curvature flow. Chin. J. Comput. **27**(1), 79–84 (2004) (in Chinese)
10. Loop, C.: A G1 triangular spline surface of arbitrary topological type. Comput. Aided Geom. Des. **11**(3), 303–330 (1994)
11. Mao, Z., Ma, L., Zhao, M., Li, Z.: A modified Laplacian smoothing approach with mesh saliency. In: Proceedings of Smart Graphics'06, pp. 105–113. Springer, Kyoto (2006)
12. Menon, J.: Constructive shell representations for freeform surfaces and solids. IEEE Comput. Graph. Appl. **14**(2), 24–36 (1994)
13. Meyer, M., Desbrun, M., Schroder, P., Barr, A.: Discrete differential geometry operators for triangulated 2-manifolds. In: Proceedings of Visualization and Mathematics, pp. 52–58. Springer, Berlin (2002)
14. Milroy, M., Bradley, C., Vickers, G.: Segmentation of a wrap-around model using an active contour. Comput. Aided Des. **29**(4), 299–320 (1997)
15. Ohtake, Y., Belyaev, A., Alexa, M.: Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In: Eurographics Symposium on Geometry Processing, pp. 149–158. Eurographics Association, Vienna (2005)

16. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
17. Razdan, A., Bae, M.: Curvature estimation scheme for triangle meshes using biquadratic Bézier patches. Comput. Aided Des. **37**(14), 1481–1489 (2005)
18. Rusinkiewicz, S., Hall-Holt, O., Levoy, M.: Real-time 3D model acquisition. ACM Trans. Graph. **21**(3), 438–446 (2002)
19. Taubin, G.: A signal processing approach to fair surface design. In: Proceedings of SIGGRAPH'95, pp. 351–358. ACM Press, Los Angeles, CA (1995)
20. Welch, W., Witkin, A.: Free-form shape design using triangulated surfaces. In: Proceedings of SIGGRAPH'94, pp. 247–256. ACM Press, Orlando, FL (1994)
21. Yokoya, N., Levine, M.: Range image segmentation based on differential geometry: a hybrid approach. IEEE Trans. Pattern Anal. Mach. Intell. **11**(6), 643–649 (1989)

ZHONG LI is an associate professor of mathematics and science at Zhejiang Sci-Tech University. He received the Ph.D. degree in mathematics from Zhejiang University in 2003. He was a post-doctoral fellow of computer science at Shanghai Jiao Tong University from 2004 to 2006. Now he is a visiting scholar of computer science division at University of California at Berkeley, U.S.A. His current research interests are mesh processing and garment CAD.

LIZHUANG MA is a professor of computer science at Shanghai Jiao Tong University. He received the Ph.D. degree in the State Key Laboratory of CAD&CG from Zhejiang University in 1991. His research interests are Computer Graphics and CAGD.

XIAOGANG JIN is a professor of the State Key Lab of CAD&CG, Zhejiang University. He received his BSc degree in computer science in 1989, MSc and PhD degrees in applied mathematics in 1992 and 1995, all from Zhejiang University. His current research interests include implicit surface computing, special effects simulation, mesh fusion, texture synthesis, crowd animation, cloth animation and facial animation.

ZUOYONG ZHENG is a PhD candidate of computer science at Shanghai Jiao Tong University, his research interests are Computer Graphics and Computer Vision.