ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Journal of Computational and Applied Mathematics ▮▮▮ (▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/cam

Triangular surface mesh fairing via Gaussian curvature flow☆

Huanxi Zhao[a, b, *], Guoliang Xu[b]

[a] *Department of Mathematics, Central South University, Changsha 410083, China*
[b] *LSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*

Received 15 August 2004; received in revised form 30 March 2005

Abstract

Surface mesh fairing by the mean curvature flow and its various modifications have become a popular topic. However, very few researches have been attempted on using the Gaussian curvature flow in surface fairing. The aim of this paper is to investigate such a problem. We find that Gaussian curvature flow can only be used to smooth convex meshes. Hence, it cannot be used to smooth noisy surface meshes because a noisy surface mesh is not convex. To overcome this difficulty, we design a new diffusion equation whose evolution direction depends on the mean curvature normal and the magnitude is a properly defined function of the Gaussian curvature. Experimental results show that the designed fairing scheme can effectively remove the noise and simultaneously preserve the sharp features, such as corners and edges.

Keywords: Mesh smoothing; Gaussian curvature flow; Feature preserving

1. Introduction and previous work

Many mesh fairing techniques have been proposed in recent years (see [4] for references), among them the most important one is the Laplace fairing. In the Laplace fairing method, a diffusion-style partial differential equation is used to control the process of the polygon mesh fairing. The diffusion equation

☆ Project Supported in part by NSFC Grants 10371130 and National Key Basic Research Project of China (2004CB318000).
* Corresponding address: LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China.
E-mail addresses: hxzhao@ustc.edu (H. Zhao), xuguo@lsec.cc.ac.cn (G. Xu).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.03.094

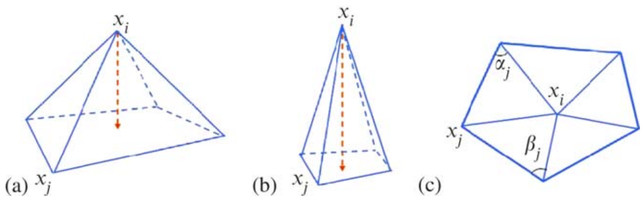

Fig. 1. The umbrella operator depends only on edge lengths. Thus, it does not distinguish cases (a) and (b). Mean curvature operator depends on angles and lengths, and hence the cases (a) and (b) have the different approximated curvature.

can be defined as follows:

$$\frac{\partial S}{\partial t} = \nabla^2 S, \tag{1}$$

where ∇^2 is the Laplacian and S represents a given polygon mesh. For simplicity, Eq. (1) is usually solved by a finite difference approach using the following iterative process:

$$S^{t+1} = S^t + \lambda \nabla^2 S^t, \tag{2}$$

where λ is a user-defined constant. In practice, a so-called umbrella operator is adopted to linearly approximate the Laplacian. The umbrella operator is defined as follows:

$$\mathcal{U}(x_i) = \frac{1}{\sum_j \omega_j} \sum_j \omega_j x_j - x_i,$$

where the weights ω_j can be chosen in many different ways, and x_i is a vertex of the mesh or surface. The simplest choice is to set the weights equal to one. Another choice that leads to ideal results is to set the weights as the inverse distances between x_i and its neighbors x_j, that is $\omega_i = \|x_i - x_j\|^{-1}$ (see Fig. 1).

Laplacian smoothing is simple, and so far it is the most commonly used technique for mesh smoothing. Many modifications and improvements have been proposed (see [10,12,17,18,20,22–24]) for achieving various specific aims. Perhaps the most interesting modifications and improvements presented in this direction is Taubin's filter and the mean curvature flow. Here, we discuss these two approaches towards mesh fairing in detail.

As we know, Laplacian smoothing has shrinkage effect. To avoid the shrinkage, Taubin presented in [20] an approach using two scale factors of opposite signs with the negative factor of larger magnitude in the Laplacian smoothing. The diffusion equation that Taubin used can be defined by the following iterative equation:

$$\widetilde{S}^t = S^t + \lambda \mathcal{U}(S^t),$$
$$S^{t+1} = \widetilde{S}^t + \mu \mathcal{U}(\widetilde{S}^t),$$

where $\lambda > 0$, $\mu < -\lambda$. Combining these two successive steps, we arrive at

$$S^{t+1} = S^t + (\lambda + \mu)\mathcal{U}(S^t) + \lambda\mu\mathcal{U}^2(S^t). \tag{3}$$

Taking $\lambda = -\mu$ in (3), we get a special case of the Taubin method, that is, the so-called bi-Laplacian smoothing

$$S^{t+1} = S^t - \lambda^2 \mathscr{U}^2(S^t).$$

Taubin's filter amplifies low-frequency information in order to balance the shrinking for meshes with arbitrary connectivity. However, the bi-Laplacian flow does not enhance low-frequency surface features.

Desbrun et al. [10] used the following mean curvature flow instead of Laplacian diffusion equation:

$$\frac{\partial x_i}{\partial t} = -H_i \vec{n_i}, \tag{4}$$

where H_i and $\vec{n_i}$ are the mean curvature and the outer unit normal vector at x_i, respectively. Mean curvature flow smoothes the surface by moving the surface along the normal direction with a speed proportional to curvatures, and achieves the best smoothing result with respect to the shape.

Desbrun et al. used the following definition of mean curvature normal to build the operator for meshes with arbitrary connectivity:

$$\lim_{\text{diam } A \to 0} \frac{\nabla A}{2A} = H_i \vec{n_i},$$

where A is the area of a small region surrounding x_i, and ∇A is the gradient of A with respect to the coordinates of x_i. Therefore, the discretized mean curvature normal can be derived as

$$H_i \vec{n_i} = \frac{1}{4A} \sum_{j \in N_1(i)} (\cot(\alpha_j) + \cot(\beta_j))(x_j - x_i), \tag{5}$$

where α_j and β_j represent the angles opposite to the edge $x_i x_j$, and $N_1(i)$ is the index set of the neighbors (or 1-ring neighbors) of x_i (see Fig. 1(c)).

Since the shape of an object is independent of the choice of an external coordinate system, the evolution should rely only on the surface itself. Therefore, the evolution should depend on intrinsic properties of surface or mesh. From Fig. 1, we know that umbrella operator depends only on the edge lengths, and does not distinguish cases (a) and (b), and assumes regular parametrization. Unlike the umbrella operator, the mean curvature flow is based on the geometric information instead of the topological information, and is formulated so that the changes in the shape of the triangles are avoided. Because of the advantage of the mean curvature flow mentioned above, smoothing by the mean curvature flow and its various modifications have become popular in geometric image processing (see, for instance, [19] for references) and surface mesh fairing. Now, similar to using mean curvature flow, it is natural to think about the use of Gaussian curvature flow in computer vision and mesh processing. Some authors had studied the Gaussian curvature flow in surface evolution [2,3,8,9,13,14] and in image processing [1,6,15]. However, to the best of our knowledge, Gaussian curvature flow has not been introduced into mesh processing. A question one may ask is: can this kind of curvature flow be effective in surface mesh processing? In this paper, we will answer this question.

The rest of this paper is organized as follows. In Section 2, we introduce some results about Gaussian curvature flow and Gaussian curvature-based surface evolution. These results are the basis of surface smoothing by the diffusion process. In Section 3, we explain why Gaussian curvature flow cannot be used to smooth noisy meshes, and derive a Gaussian curvature-dependent smoothing scheme. In Section 4, we discuss the numerical simulations of the modified Gaussian curvature flow. In Section 5, we present the experimental results and demonstrate the effectiveness of the new fairing approach. Section 6 concludes the paper and points out the directions for future research.

2. Some results on Gaussian curvature flow

In this paper, S or $S(.,t)$ denotes a mesh or a compact surface which is regular, orientable without boundary in the Euclidean space $\mathbb{R}^3$, x_i a vertex of this mesh, and e_{ij} the edge (if existing) connecting x_i to x_j. We use $N_1(i)$ to denote the index set of the neighbors (or 1-ring neighbors) of x_i.

As we know, the Gauss curvature flow evolves a closed surface $S(.,t)$ as follows:

$$\frac{\partial S}{\partial t} = K\vec{n}, \tag{6}$$

where K is the Gaussian curvature of the surface $S(.,t)$. It is easy to prove that the enclosed volume V evolves according to

$$\frac{\partial V}{\partial t} = -\oint K \, \mathrm{d}\sigma,$$

where $\mathrm{d}\sigma$ is the element of area.

The main result on the total Gaussian curvature is the Gaussian–Bonnet theorem (see [5]). Namely

Lemma 2.1 (*Gaussian–Bonnet theorem*). *Let $S(.,t)$ be an orientable compact surface, then*

$$\oint_{S(,.t)} K \, \mathrm{d}\sigma = 2\pi\chi(S(.,t)), \tag{7}$$

where $\chi(S(.,t))$ is the Euler characteristic of $S(.,t)$.

In general, the Euler characteristic is a topological invariant. For a closed, orientable surface of genus g, we have $\chi(S(.,t)) = 2(1-g)$. Now, it is easy to obtain the following lemma.

Lemma 2.2. *Let $S(.,t)$ be an orientable compact surface, and it evolves with the flow $\partial S/\partial t = K\vec{n}$, then the enclosed volume V evolves according to*

$$\frac{\partial V}{\partial t} = -4\pi(1-g). \tag{8}$$

Therefore, depending on the genus of the surface, the evolution process will have the shrinking/expanding effects or will be exact volume-preserving. The Gaussian curvature flow was introduced by Firey [11], who showed that it shrinks smooth, compact, strictly convex, and centrally symmetric hypersurfaces in $\mathbb{R}^3$ to round points, Chou [8] showed that if the initial surface is a compact, smooth, strictly convex surface, then there exists a unique, smooth solution of the Gaussian curvature flow, and the diffusion surface is strictly convex, and converges to a point $q \in \mathbb{R}^3$. Chow [9] proved that, under certain restrictions on the second fundamental form of the initial surface, the Gaussian curvature flow shrinks smooth compact

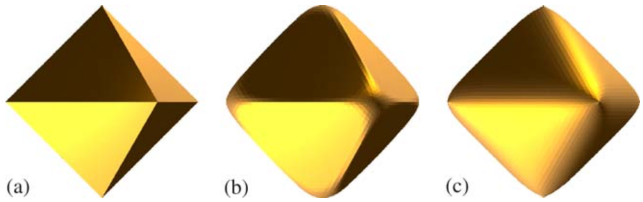

Fig. 2. The difference of Gaussian curvature flow and mean curvature flow in mesh processing: (a) Original octahedron; (b) the result of Gaussian curvature flow evolution; (c) the result of mean curvature flow evolution.

strictly convex hypersurfaces to round points. In [2,3], Andrews showed that the Gaussian curvature flow shrinks compact convex hypersurfaces to round points.

In the last few years, the theory of the Gaussian curvature flow has been generalized to a class of nonconvex surfaces. For example, in [13], Ishii et al. studied the existence and the uniqueness of a viscosity solution to the PDE that describes the time evolution of a nonconvex graph by a convexified Gaussian curvature. In [14], Ishii et al. discussed the existence and the uniqueness of the motion (or time evolution) of a nonconvex compact set which evolves by a convexified Gaussian curvature in $\mathbb{R}^n$ $(n \geqslant 2)$, by the level set approach in the theory of viscosity solutions. In this paper, the above-mentioned work is the basis of our approach to the mesh (or surface) fairing.

3. Fairing mesh using Gaussian curvature flow

In this section, we will discuss mesh smoothing using the Gaussian curvature flow.

3.1. Convex mesh fairing

For a convex surface, the Gaussian curvature is nonnegative, the discrete Gaussian curvature flow moves every vertex in the inner normal direction with speed equal to a discrete approximation of the Gaussian curvature at the vertex. The figures in the following (see Fig. 2(b)) demonstrate that a flat region remains flat for finite nonzero time before the entire octahedron becomes strictly convex under the Gaussian curvature flow evolution. This is in contrast to mean curvature flow where the octahedron becomes strictly convex quickly (see Fig. 2(c)).

For the Gaussian curvature motion (6), the evolved convex surface shrinks to the zero according to Eq. (8). In fact, consider the volume $V(t)$ enclosed by the surface $S(., t)$, it is easy to see

$$V(t) = V(0) - 4\pi t.$$

3.2. Nonconvex mesh fairing

Fig. 3 shows that the Gaussian curvature flow leads to instabilities. The figure also demonstrates that the Gaussian curvature flow cannot be applied to arbitrary meshes for smoothing. The main reason is that the Gaussian curvature flow makes the concave part of the mesh sharper when it smoothes the convex part. Observe that the points on the convex part and concave part should move in the opposite directions. We will establish a new flow equation based on the Gauss curvature such that convex points move inward,

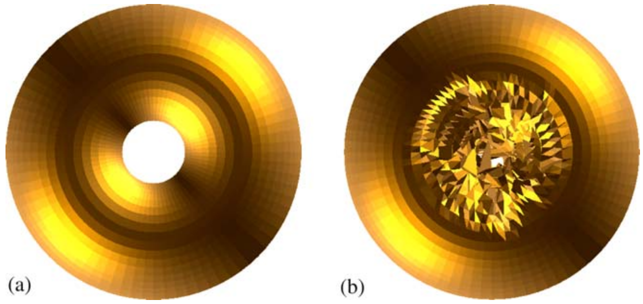

Fig. 3. Gaussian curvature flow leads to instabilities: (a) Original ring; (b) evolving result by Gaussian curvature flow.

Table 1
Gaussian curvature flow diffusion direction of velocity of a point on a surface with respect to point type

Classification	Evolution direction
Elliptic convex: $K > 0$, $H > 0$	Inward: $-\vec{n}$
Elliptic concave: $K > 0$, $H < 0$	Outward: $\vec{n}$
Hyperbolic: $K < 0$, $H \neq 0$	Inward:(or outward:) $-\vec{n}$(or $\vec{n}$)
Parabolic: $K = 0$	Velocity is zero
$H = 0$, $K \neq 0$	Velocity is zero

and concave points move outward. We now consider constraints on elliptic, hyperbolic, and parabolic points to determine the appropriate direction of the velocity vector for each depending only on curvatures (k_1, k_2), where k_1 and k_2 denote the higher and the lower magnitude principal curvatures, respectively. A point p on a regular surface S is classified based on the sign of its Gaussian curvature $K(p)$ and mean curvature $H(p)$ as given in the following Table 1. We establish a direction for the movement of the point p as follows:

- The elliptic convex points should move inward, elliptic concave points should move outward. Our aim is to make $K(p)$ lower with surface diffusion.
- In the hyperbolic points case, if $k_1(p) = -k_2(p)$ $(H = 0)$, we let the saddle points not move. Now, we consider an arbitrary saddle point p with $|k_1(p)| > |k_2(p)|$. The velocity at p can be in the direction of the higher-magnitude curvature $k_1(p)\vec{n}$, or the lower-magnitude curvature $k_2(p)\vec{n}$, and can be zero. First, suppose that p moves along $k_1(p)\vec{n}$, we conclude that $|k_1|$ must become lower with surface diffusion, and $|k_2|$ must become higher with surface diffusion. The process will be repeated until $k_1 = -k_2$, that is $H = 0$. Second, suppose that p moves along $-k_2(p)\vec{n}$ with surface diffusion, we conclude that $|k_2|$ must become lower with surface diffusion. The process will be repeated until $K = 0$. If the velocity at p is zero, then the saddle point p does not move at all.
- For parabolic points, since the Gaussian curvature $K = 0$, the parabolic point should have zero velocity.

Now, we can give a summary (see Table 1) for the direction of the Gaussian curvature flow during surface diffusion. Roughly speaking, our main idea of mesh smoothing is that we change the direction of diffusion when convex surface converts into concave surface. To achieve this goal, we establish the following flow:

$$\frac{\partial S}{\partial t} = \begin{cases} \operatorname{sign}(H)K\vec{n}, & K > 0, \ H \neq 0, \\ \alpha K\vec{n}, & K < 0, \\ 0, & K = 0 \text{ or } H = 0, \end{cases} \tag{9}$$

where $|\alpha| < 1$. According to our experiments, a better choice is $|\alpha| = 0$–0.0005.

As we know, in the discrete case, H may be zero even if $H \neq 0$ in the continuous case. On the contrary, when $H = 0$ in the continuous case, H may be greater or less than zero in the discrete case. Therefore, for the sake of stability of algorithm, our fairing model is changed to

$$\frac{\partial S}{\partial t} = \begin{cases} H\vec{n}, & |H| < \varepsilon, \\ \operatorname{sign}(H)K\vec{n}, & K > 0, \ |H| \geqslant \varepsilon, \\ \alpha K\vec{n}, & K < 0, \ |H| \geqslant \varepsilon, \end{cases} \tag{10}$$

here ε is a small positive parameter. According to our numerical simulations, the better choice is $\varepsilon = 0.01$–0.5.

3.3. Adaptive anisotropic fairing

The Laplace diffusion, Taubin diffusion, mean curvature flow diffusion, or their combinations mentioned above are classified as isotropic diffusion-type fairing. The factor λ and μ in Taubin's approach remains constant in an isotropic diffusion operation, regardless of the diffusion direction. An isotropic diffusion operation can eliminate noise very effectively but also smooth out important features. An adaptive anisotropic fairing uses a function of two principal curvatures, k_1 and k_2, as the weights for each diffusion direction. The adaptive anisotropic fairing schemes of (2) and (4) are as follows:

$$S^{t+1} = S^t + \chi(k_1, k_2)\mathscr{U}(S^t), \tag{11}$$

or

$$\frac{\partial x_i}{\partial t} = -\chi(k_1, k_2) H_i \vec{n}_i, \tag{12}$$

where $\chi(k_1, k_2)$ is an anisotropic diffusion weighting function. With a proper design of $\chi(k_1, k_2)$ in accordance with diffusion directions, the anisotropic fairing scheme can effectively remove the noise and at the same time preserve the shape of corners and edges.

Meyer et al. in [16] used the following rules to determine the value of $\chi(k_1, k_2)$:

$$\chi(k_1, k_2) = \begin{cases} 1 & \text{if } |k_1| \leqslant T \text{ and } |k_2| \leqslant T, \\ 0 & \text{if } |k_1| > T \text{ and } |k_2| > T \text{ and } K > 0, \\ k_1/H & \text{if } k_1 = \min(|k_1|, |k_2|, |H|), \\ k_2/H & \text{if } k_2 = \min(|k_1|, |k_2|, |H|), \\ 1 & \text{if } |H| = \min(|k_1|, |k_2|, |H|), \end{cases} \tag{13}$$

where T is a user-defined constant. In [7], Chen et al. gave the following weighting function $\chi(k_1, k_2)$ of a bi-directional curvature mapping:

$$\chi(k_1,k_2)=\begin{cases}1 & \text{if } |k_1|\leqslant T \text{ and } |k_2|\leqslant T,\\ 0 & \text{if } |k_1|\geqslant |k_2|\geqslant T,\\ \mathrm{e}^{-(k_2/k_1-1)^2}-\mathrm{e}^{-1} & \text{if } |k_1|\leqslant T,\ |k_2|>T,\\ \mathrm{e}^{-((2T-k_2)/k_1-1)^2}-\mathrm{e}^{-1} & \text{if } |k_1|>T,\ \text{and } T<|k_2|<2T.\end{cases} \tag{14}$$

In this paper, we simply determine the value of $\chi(k_1, k_2)$ by

$$\chi(k_1,k_2)=\frac{1}{1+|K|^{\beta}}, \tag{15}$$

where $1 \leqslant \beta < \infty$. According to our experiment results, the smoothing scheme with $\beta = 1$–3 produces desirable results. Then, our final algorithm is the following anisotropic smoothing scheme:

$$\frac{\partial S}{\partial t}=\begin{cases}\dfrac{1}{1+K^{\beta}}H\vec{n}, & |H|<\varepsilon,\\ \dfrac{1}{1+K^{\beta}}\operatorname{sign}(H)K\vec{n}, & K>0,\ |H|\geqslant\varepsilon,\\ \dfrac{\alpha}{1+K^{\beta}}K\vec{n}, & K<0,\ |H|\geqslant\varepsilon.\end{cases} \tag{16}$$

4. Implementation

4.1. Normal computation

In this paper, we compute vertex normal using the so-called "mean weighted by angle" algorithm proposed in [21] in 1998:

$$\vec{n}_i=\sum_{j\in N_1(i)}\alpha_j\frac{(x_j-x_i)\times(x_{j+1}-x_j)}{|(x_j-x_i)\times(x_{j+1}-x_j)|}, \tag{17}$$

where α_j is the angle between the two edge vector e_{ij} and $e_{i(j+1)}$ of the jth facet sharing the vertex x_i with $e_{i(n+1)} = e_{i1}$. The α_j can be quickly computed from the cross product of the edge vectors by the formula

$$\alpha_j=\arcsin\frac{|(x_j-x_i)\times(x_{j+1}-x_i)|}{|x_j-x_i||x_{j+1}-x_i|}. \tag{18}$$

4.2. Gaussian curvature computation

A discretization of the Gaussian K_i at a vertex x_i depends on the vertices in a local neighborhood. A discretization algorithm that seems ideal for our needs was described in [17]. We can express the Gaussian curvature at x_i as

$$K_i = \frac{3}{\sum_{j\in N_1(i)} \sigma_j}\left(2\pi - \sum_{j\in N_1(i)} \varphi_j\right), \tag{19}$$

where φ_i is the angle between $x_i x_j$ and $x_i x_{j+1}$, σ_j is the area of the triangle $x_j x_i x_{j+1}$.

There are other point-wise definitions for the discrete Gaussian curvature. For instance, Meyer et al. [16] gave a formula similar to (19). They divided $2\pi - \sum_{j\in N_1(i)} \varphi_j$ by an area which is different from $3/\sum_{j\in N_1(i)} \sigma_j$ and obtain

$$K_i = \frac{1}{\mathscr{A}_{\text{Mixed}}}\left(2\pi - \sum_{j\in N_1(i)} \varphi_j\right), \tag{20}$$

where $\mathscr{A}_{\text{Mixed}}$ is the area of Voronoi region (see [16] for detail).

4.3. Explicit fairing

We finally present the numerical algorithm. Substituting (5), (17) and (19) (or (20)) into (16), we get the following discretized smoothing scheme:

$$S_{\text{new}} = S_{\text{old}} + \rho * \begin{cases} \dfrac{1}{1+K^{\beta}(S_{\text{old}})} H(S_{\text{old}})\vec{n}(S_{\text{old}}), & |H| < \varepsilon, \\ \dfrac{1}{1+K^{\beta}(S_{\text{old}})} \text{sign}(H(S_{\text{old}}))K(S_{\text{old}})\vec{n}(S_{\text{old}}), & K > 0,\ |H| \geqslant \varepsilon, \\ \dfrac{\alpha}{1+K^{\beta}(S_{\text{old}})} K(S_{\text{old}})\vec{n}(S_{\text{old}}), & K < 0,\ |H| \geqslant \varepsilon, \end{cases} \tag{21}$$

where ρ is a user-defined time step length.

5. Experiment results

Important shape features such as ridges and corners often exist in mesh models with random noise. A good fairing scheme should effectively remove the noise and preserve geometric feature of the mesh. This section illustrates the numerical simulations of our approach for a class of typical models. Fig. 4 demonstrates that the anisotropic Gaussian curvature flow preserves and enhances sharp features like edges or corners of a surface. In Figs. 5 and 6, we show how well our approach is able to smooth a mesh and preserve the nonlinear feature of the mesh, while Taubin's (λ, μ) filtering, the bi-Laplacian smoothing and the mean curvature flow all blur badly the ridges and corners.

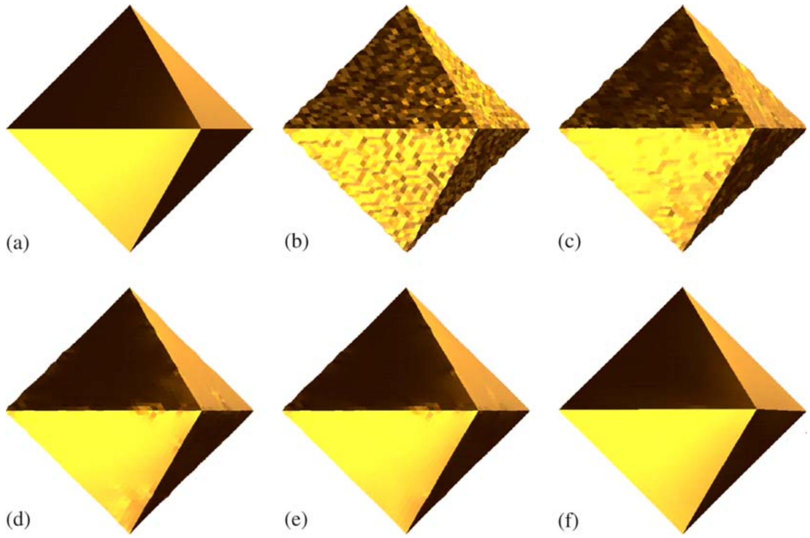

Fig. 4. (a) Original octahedron; (b) with a uniform noise added; (c) after 40 iterations; (d) after 80 iterations; (e) after 100 iterations; (f) after 120 iterations. All these results are obtained with time step length $\rho = 0.001$, $\beta = 2$, $\varepsilon = 0.001$, $\alpha = 0.0005$.

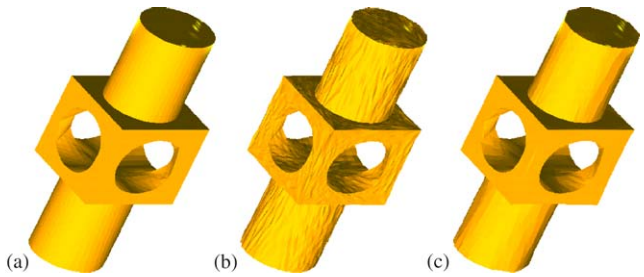

Fig. 5. (a) Original polygon mesh; (b) model with a uniform noise added; and (c) the result of our fairing approach. The result is obtained with time step length $\rho = 0.001$, $\beta = 2$, $\varepsilon = 0.001$, $\alpha = 0.0005$.

6. Conclusion

We have investigated mesh fairing using the modified Gaussian curvature flow. We find that if the evolution direction of the modified Gaussian curvature flow is dictated by the mean curvature normal, and while its magnitude is a properly defined function of the Gaussian curvature, then the modified Gaussian curvature flow is a good mesh smoothing scheme. We incorporate anisotropy into the Gaussian curvature flow to denoise and sharpen nonlinear features like the curved edges which typically appear in CAD models. Experimental results show that our anisotropic Gaussian curvature flow reproduces the sharp features with very high quality comparing with the previous approaches. Our method is easy to implement and very efficient. The weakness of the approach is that the temporal direction discretization currently used is explicit, the inherent problem with this approach is that explicit methods behave poorly

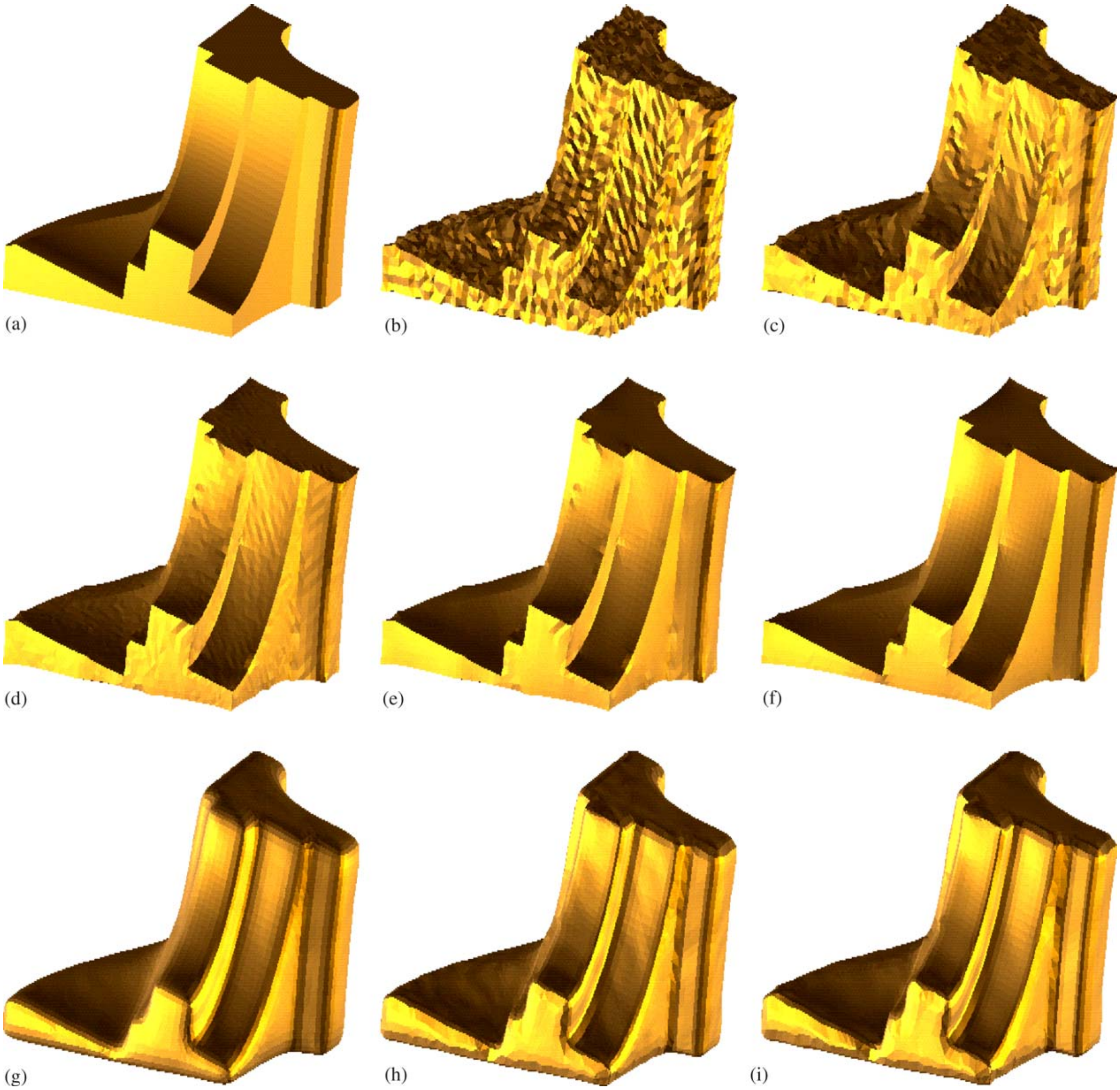

Fig. 6. (a) Original fandisk; (b) with a uniform noise added; (c)–(f) the results after 10, 20, 30, 40 iterations of our fairing approach, respectively; (g) the result after four iterations of mean curvature flow; (h) the result after 30 iterations of Taubin's (λ, μ) filtering; (i) result after 60 iterations of bi-Laplacian smoothing.1 Results (c)–(f) are obtained with time step length $\rho = 0.001$, $\beta = 2$, $\varepsilon = 0.001$, $\alpha = 0.0005$, the others are obtained with time step length 0.001.

if the system is stiff, and in order to converge to the correct solution it is necessary to use small time steps. Therefore, the future research is to design an implicit integration scheme to avoid the time step limitation.

References

[1] L. Alvarez, P.-L. Lions, J.-M. Morel, Image selective smoothing and edge detection by nonlinear diffusion: II, SIAM J. Numer. Anal. 29 (3) (1992) 845–866.
[2] B. Andrews, Contraction of convex hypersurfaces in Euclidean space, Calc. Var. Partial Differential Equations 2 (1994) 151–171.
[3] B. Andrews, Motion of hypersurfaces by Gauss curvature, Pacific J. Math. 195 (2000) 1–34.
[4] C.B. Bajaj, G. Xu, Anisotropic diffusion of surface and functions on surfaces, ACM Trans. Graphics 22 (1) (2003) 4–32.
[5] M.P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, Englewood Cliffs, NJ, 1976.
[6] F. Catte, P.-L. Lions, J.-M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal. 29 (1) (1992) 182–193.
[7] C.Y. Chen, K.Y. Cheng, H.Y. Liao, Fairing of polygon meshes via Bayesian discriminant analysis, J. WSCG 12 (1–3) (2004).
[8] K.S. Chou, Deforming a hypersurface by its Gauss–Kronecker curvature, Comm. Pure Math. Appl. 38 (1985) 867–882.
[9] B. Chow, Deforming convex hypersurfaces by the nth root of the Gaussian curvature, J. Differential Geom. 22 (1985) 117–138.
[10] M. Desbrun, M. Meyer, P. Schroder, A.H. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, ACM SIGGRAPH ’99, pp. 317–324.
[11] W.J. Firey, On the shapes of worn stones, Mathematika 21 (1974) 1–11.
[12] I. Guskov, W. Sweldens, P. Schroder, Multiresolution signal processing for meshes, ACM SIGGRAPH’99, pp. 325–334.
[13] H. Ishii, T. Mikami, Convexified Gauss curvature flow of sets: a stochastic approximation, SIAM J. Math. Anal. 36 (2) (2002) 552–579.
[14] H. Ishii, T. Mikami, A level set approach to the wearing process of a nonconvex stone, Calc. Var. Partial Differential Equations 19 (1) (2004) 53–93.
[15] B.B. Kimia, K. Siddiqi, Geometric heat equation and nonlinear diffusion of shapes and images, Comput. Vision Image Understanding 64 (1996) 305–322.
[16] M. Meyer, M. Desbrun, P. Schrder, A.H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: VisMath, 2002.
[17] J.M. Morvan, B. Thibert, Smooth surface and triangular mesh: comparison of the area, the normals and the unfolding, in: Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications, ACM Press, New York, 2002, pp. 147–158.
[18] Y. Ohtake, A.G. Belyaev, I.A. Bogaevski, Polyhedral surface smoothing with simultaneous mesh regularization, Proceedings of the Geometric Modeling and Processing 2000, GMP2000, pp. 229–237.
[19] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, 1999.
[20] G. Taubin, A signal processing approach to fair surface design, in: R. Cook (Ed.), SIGGRPH 95 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH’95, pp. 351–358.
[21] G. Thurmer, C. Wuthrich, Computing vertex normals from polygonal facets, J. Graphics Tools 3 (1) (1998) 43–46.
[22] J. Vollmer, R. Mencl, H. Müller, Geometric modeling: improved Laplacian smoothing of noisy surface meshes, Comput. Graphics Forum 18 (3) (1999) 131–138.
[23] G. Xu, Surface fairing and featuring by mean curvature motions, J. Comput. Appl. Math. 163 (1) (2004) 295–309.
[24] S. Yoshizawa, A.G. Belyaev, Fair triangle mesh generation with discrete elastica, GMP’02, pp. 119–123.