ELSEVIER

Computer-Aided Design 33 (2001) 789–800

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Mesh regularization and adaptive smoothing

Yutaka Ohtake[a], Alexander Belyaev[b,*], Ilia Bogaevski[c]

[a]Graduate School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan
[b]Department of Computer Software, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan
[c]Independent University of Moscow, Bolshoi Vlasevsky Pereulok 11, Moscow, Russia

Accepted 15 January 2001

Abstract

The paper presents a set of mesh smoothing tools developed to increase mesh regularity, reduce oversmoothing, and enhance crease lines. Mesh smoothing with simultaneous increasing mesh regularity and reducing oversmoothing is achieved by combining together the Laplacian flow and a mesh evolution by a function of the mean curvature. To enhance salient ridge and ravine structures we use a coupled nonlinear diffusion of the mesh normals and vertices. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Mesh smoothing; Mesh regularization; Nonlinear diffusion; Crease enhancement

1. Introduction

A computer graphics object reconstructed from real-world data contains undesirable noise and small-scale oscillation. An important problem is how to remove the noise and oscillations while preserving desirable geometric features of the object.

1.1. Related work

Two of the most popular approaches for smoothing and denoising polygonal surfaces are minimizing energy functionals associated with differential-geometric surface characteristics and Laplacian smoothing. Minimizing an energy functional is usually a computationally expensive task. Moreover, energy minimization smoothing lacks local shape control. Laplacian smoothing is simple, fast, and so far the most common technique for mesh smoothing. The Laplacian smoothing flow, in its simplest form, repeatedly and simultaneously moves each mesh vertex by a displacement equal to a positive scale factor times the difference between the average of the neighboring vertices and the vertex itself. Actually, the Laplacian smoothing flow can be considered as the gradient descent flow for a simple quadratic energy functional. However, due to its simplicity the Laplacian flow offers many opportunities for modifications and improvements.

Taubin [21,22] proposed to alternate two scale factors of opposite signs with the negative factor of larger magnitude in a weighted Laplacian smoothing flow. Such smoothing does not produce shrinkage and suppresses high frequencies of a discrete Laplacian operator defined in the mesh, while enhancing low frequencies slightly.

When the scale factors are equal in magnitude, the Taubin smoothing scheme reduces to the bilaplacian smoothing flow [11], which can be considered a discrete approximation of the gradient descent flow for the thin-plate energy functional.

Another non-shrinking modification of the Laplacian smoothing flow was very recently proposed [24].

A further development of the Laplacian smoothing method, mesh smoothing by the mean curvature flow [6,19] (see also references therein), came from mathematics and material science. The mean curvature flow is the gradient descent flow for the area functional. This explains the strong regularization effect of the flow. The discrete mean curvature flow moves every vertex in the normal direction with speed equal to a discrete approximation of the mean curvature at the vertex. Smoothing by the mean curvature flow and its various modifications have become extremely popular in geometric image processing (see, for instance, [19] for references). Accurate and robust discrete approximation of the mean curvature vector and corresponding discrete implementation of the mean curvature flow were recently proposed [6]. They demonstrate high reliability even on irregular meshes.

Laplacian smoothing, Taubin smoothing, and the discrete

* Corresponding author. Tel.: +81-242-372-771; fax: +81-242-372-728.
E-mail addresses: d8011101@u-aizu.ac.jp (Y. Ohtake), belyaev@u-aizu.ac.jp (A. Belyaev), bogaevsk@mccme.ru (I. Bogaevski).

PII: S0010-4485(01)00095-1

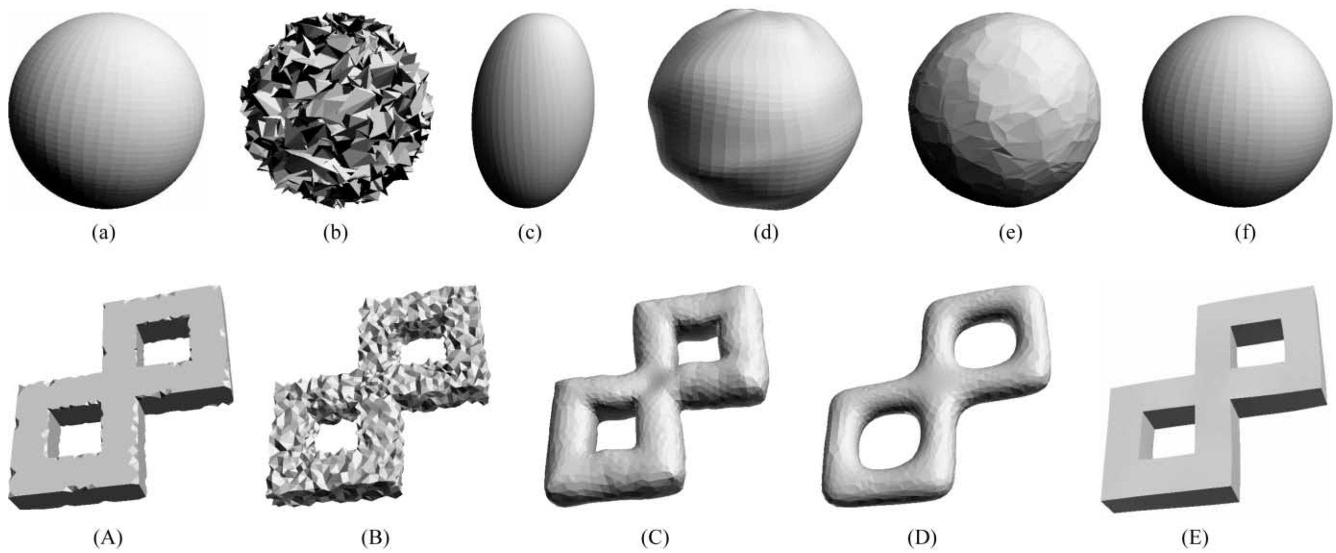

Fig. 1. Top row: (a) a polygonal sphere whose triangulation is denser near the north and south poles; (b) sphere with uniform noise added along the normal direction; (c) Laplacian smoothing develops unnatural deformations; (d) smoothing by the Taubin method converts high-frequency surface oscillations into low-frequency waves; (e) smoothing by mean curvature flow increases mesh irregularity; and (f) smoothing according to a method proposed in this paper, see Section 3. Bottom row: (A) a polygonal two-holed torus; (B) the two-holed torus with a noise added; (C) Taubin smoothing reduces high-frequency surface oscillations but enhances low frequency oscillations; (D) the mean curvature flow suppresses the noise but rounds off the sharp creases (the Laplacian flow and smoothing by our method developed in Section 3 produce similar results); and (E) smoothing by a coupled nonlinear diffusion of the mesh normals and vertices: the noise is removed, main sharp creases are completely restored (see Section 5 for details).

mean curvature flow contain a number of drawbacks. The Laplacian smoothing flow increases mesh regularity but develops unnatural deformations when applied to a highly irregular mesh. Smoothing by the discrete mean curvature flow is relatively independent of the mesh density but increases mesh irregularity. Both the Laplacian and mean curvature flows do not decelerate the smoothing process and may lead to oversmoothing and loss of desirable geometric features. The Taubin smoothing scheme lacks local shape control and enhances low frequency surface features (the latter drawback was fixed in Ref. [23]).

In Ref. [15] we proposed to combine together the Laplacian flow and a mesh evolution by a function of the mean curvature. It allows us to achieve mesh smoothing with simultaneous increasing mesh regularity and to reduce oversmoothing.

Computer graphics objects reconstructed from real-world data may also contain creases and corners. Such shape features are usually diffused and lost when isotropic smoothing schemes such as mean curvature flow or Laplacian filtering are applied. To preserve or even enhance shape creases and corners, anisotropic diffusion schemes have been recently proposed [3–5,7,13,22]. The main idea of anisotropic filtering consists of non-uniform smoothing in different directions. Taubin [22] achieves anisotropic filtering by using Laplacians with non-symmetric neighborhoods and properly prescribed weights. Methods presented in Refs. [7,14] extend PDE-based image processing techniques [16,20,25] for surface processing, while the method developed in Ref. [5] is based on the evolution of the mesh vertices in the normal directions by a properly defined function of the principal curvatures. In Refs. [3,13], inspired by diffusion-based processing techniques [16,18], we proposed to use a nonlinear diffusion of the mesh normals for enhancing and detecting mesh creases and corners. According to our experiments, our approach outperforms other crease enhancement methods.

1.2. Contributions

The paper presents new methods for polygonal surface smoothing with simultaneous increasing mesh regularity [2,15], slowing down the smoothing process adaptively in order to reduce possible oversmoothing[15], and preserving and enhancing salient ridge and ravine structures [3,13].

In Section 2, we analyze Laplacian smoothing, Taubin smoothing, and the discrete mean curvature flow and reveal their strengths and weaknesses.

Our smoothing scheme developed in Section 3 combines together the best properties of the Laplacian and mean curvature flows. Roughly speaking, our method consists of moving every vertex of a noisy mesh in the direction defined by the Laplacian flow with speed equal to a properly chosen function of the mean curvature at the vertex. The top row of Fig. 1 demonstrates some of our results.

In Section 4, we propose a simple modification of the method in order to avoid oversmoothing.

In Section 5, we present a diffusion scheme for preserving and enhancing salient ridge and ravine structures [3,13]. The scheme is based on a coupled nonlinear diffusion of the

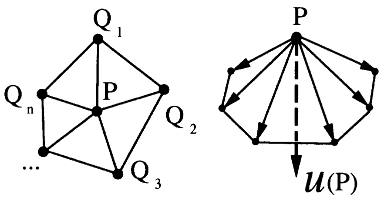

Fig. 2. Umbrella-operator vector associated with a vertex is defined as the difference between a weighted average of the neighboring vertices and the vertex itself; see (1).

mesh normals and vertices. The bottom row of Fig. 1 demonstrates advantages of our approach compared with the Taubin smoothing scheme and the mean curvature flow. Not only does our diffusion process remove the noise perfectly, but it also repairs small defects of the original model.

We conclude and sketch directions for future research in Section 6.

2. Laplacian flow, Taubin method, bilaplacian flow, mean curvature flow

In this section we analyze four methods for polygonal surface smoothing: Laplacian smoothing, Taubin smoothing [21,22], the bilaplacian flow [11], and the mean curvature flow [6], and reveal their strengths and weaknesses.

2.1. Laplacian smoothing

Let us consider a triangulated surface and, for any vertex P, let us define the so-called 'umbrella-operator' [11]

$$\mathcal{U}(P) = \frac{1}{\sum_i w_i} \sum_i w_i Q_i - P \tag{1}$$

where summation is taken over all neighbors of P, w_i are positive weights. The geometric idea behind the umbrella-operator is illustrated by Fig. 2. The local update rule

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda \mathcal{U}(P_{\text{old}}) \tag{2}$$

applied typically to every inner point of the triangulated surface is called Laplacian smoothing of the surface. Here λ is a small positive number and the process (2) is executed repeatedly.

The Laplacian smoothing algorithm reduces the high frequency surface information and tends to flatten the surface.

The weights can be chosen in many different ways. The simplest choice is to define the weights equal to each other: $w_i = 1$,

$$\mathcal{U}_0(P) = \frac{1}{n} \sum_i Q_i - P, \tag{3}$$

where n is the number of neighbors. Another choice that produces good results [22] is to set the weights to the inverse distances between P and its neighbors:

$$\mathcal{U}_1(P) = \frac{1}{\sum_i w_i} \sum_i w_i Q_i - P, \qquad w_i = \|P - Q_i\|^{-1}. \tag{4}$$

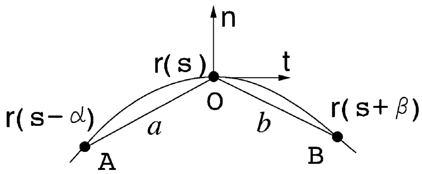

Fig. 3. Smooth curve $\mathbf{r}(s)$ and its local approximation by polyline AOB.

Note that smoothing with $\mathcal{U}_0$ homogenizes the mesh density, whereas smoothing with $\mathcal{U}_1$ worsens the mesh. To demonstrate this, let us consider a plane curve $\mathbf{r}(s)$ parameterized by its arclength s. Consider three points on the curve

$$A = \mathbf{r}(s - \alpha), \qquad O = \mathbf{r}(s), \qquad B = \mathbf{r}(s + \beta)$$

with distances $a = |OA|$ and $b = |OB|$ between them. Let $\mathbf{t} = \mathrm{d}\mathbf{r}/\mathrm{d}s$ and $\mathbf{n} = \mathbf{t}^{\top}$ compose the Frenet frame at O (see Fig. 3). According to the Frenet formulas, the derivatives of the tangent $\mathbf{t}$ and normal $\mathbf{n}$ with respect to arclength s are given by $\mathbf{t}' = k\mathbf{n}$ and $\mathbf{n}' = -k\mathbf{t}$, where $k(s)$ is the curvature. Simple manipulations with Taylor series expansions and the Frenet formulas show that

$$\frac{2}{a+b}\left[\frac{\overrightarrow{OA}}{a} + \frac{\overrightarrow{OB}}{b}\right] = \mathbf{n}\left(k + \frac{b-a}{3}k' + O(a,b)^2\right) + \mathbf{t}\left(\frac{a-b}{4}k^2 + O(a,b)^2\right), \tag{5}$$

$$\frac{2}{a^2+b^2}\left[\overrightarrow{OA} + \overrightarrow{OB}\right] = \mathbf{n}\left(k + \frac{b^3-a^3}{3(a^2+b^2)}k' + O(a,b)^2\right) + \mathbf{t}\left(\frac{2(b+a)}{a^2-b^2} + \frac{a^3-b^3}{4(a^2+b^2)}k^2 + O(a,b)^2\right). \tag{6}$$

Expansions (5) and (6) reveal asymptotic properties of $\mathcal{U}_0$ and $\mathcal{U}_1$, respectively. If, for example, point O is located closer to A than to B, $a < b$, then, due to the tangent components in the above expansions, one step of Laplacian smoothing with $\mathcal{U}_0$ shifts O closer to B and one step of the Laplacian smoothing with $\mathcal{U}_1$ shifts O closer to A.

However, Laplacian smoothing flow with $\mathcal{U}_0$ develops unnatural deformations while acting on non-homogeneous meshes. For example, Laplacian smoothing a noisy sphere whose triangulation is denser near the north and south poles deforms the sphere (see Fig. 1c). A similar effect for a torus is demonstrated by the images of the top row of Fig. 4.

2.2. Taubin smoothing

Taubin [21,22] proposed to alternate two scale factors of opposite signs with the negative factor of larger magnitude in the Laplacian smoothing flow. Such smoothing suppresses high frequencies of the umbrella operator (1), while preserving and enhancing its low frequencies [21,22]. Combining the two successive steps of the Taubin method in one local update rule we arrive at

$$P_{\text{new}} \leftarrow (1-\mu\mathcal{U})(1+\lambda\mathcal{U})P_{\text{old}}$$

$$= P_{\text{old}} - (\mu-\lambda)\mathcal{U}(P_{\text{old}}) - \mu\lambda\mathcal{U}^2(P_{\text{old}}), \tag{7}$$

where $\mu > \lambda > 0$, $\mathcal{U}^2$ is the squared umbrella operator

$$\mathcal{U}^2(P) = \frac{1}{\sum_i w_i}\sum_i w_i\mathcal{U}(Q_i) - \mathcal{U}(P).$$

According to Ref. [22], the best smoothing with (7) is obtained when $\mathcal{U} = \mathcal{U}_0$ or $\mathcal{U} = \mathcal{U}_1$. In our experiments, we use Taubin smoothing with $\mathcal{U} = \mathcal{U}_0$. According to our experience, it often produces poor results on non-homogeneous meshes. See, for example, Fig. 1d, where the Taubin filtering flow with $\lambda = 0.5$ and $1/\lambda - 1/\mu = 0.1$ deforms a noisy sphere whose triangulation is denser near the north and south poles. In Fig. 1c, the Taubin filtering scheme demonstrates good performance. Nevertheless, one can note enhancement of low-frequency surface wrinkles. Taubin smoothing of a noisy torus from Fig. 4c is shown in Fig. 4e.

2.3. Bilaplacian flow

The explicit bilaplacian flow

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda\mathcal{U}^2(P_{\text{old}})$$

being a discrete analog of the steepest descent flow for the thin-plate energy functional has nice smoothing properties [11]. It can be obtained from the Taubin smoothing scheme if the positive and negative scale factors are equal in magnitude. The bilaplacian flow does not enhance low-frequency surface features. In our implementation of the bilaplacian flow we use the plain umbrella operator (3). Smoothing of the noisy torus from Fig. 4c by the bilaplacian flow is shown in Fig. 4f.

2.4. Mean curvature flow

The explicit approximation of the mean curvature flow

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda H(P_{\text{old}})\mathbf{n}(P_{\text{old}}), \tag{8}$$

where H is a discrete version of the mean curvature and $\mathbf{n}$ is the unit normal vector, can be considered as a generalization of Laplacian smoothing (2).

Desbrun et al. [6] proposed an accurate and robust discrete approximation of the mean curvature vector at a mesh vertex P:

$$H\mathbf{n} = -\frac{\nabla A}{2A},$$

where $A = \sum A_i$is the sum of the areas of the triangles surrounding P.

Calculations [6] show that

$$H\mathbf{n}(P) = \frac{1}{4A}\sum_i(\cot\alpha_i + \cot\beta_i)(Q_i - P), \tag{9}$$

where α_i and β_i are the two angles opposite to the edge Q_iP (as in Fig. 5). Discrete approximation (5) has been proven

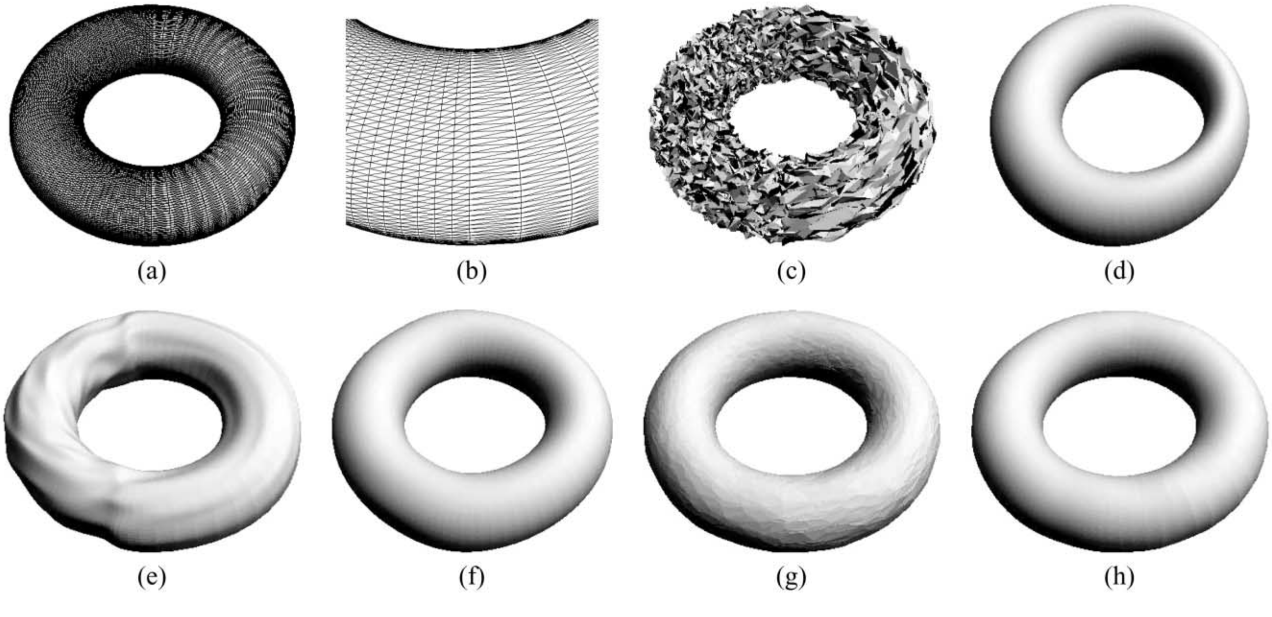

Fig. 4. (a) A torus consisting of two parts with different mesh densities; (b) a magnified view of a part of the torus; (c) the torus with a uniform noise added; (d) Laplacian smoothing deforms the initial shape; (e) smoothing by the Taubin method reduces high-frequency surface oscillations but develops low-frequency surface waves; (f) the bilaplacian flow smoothes well but slightly deforms the initial shape; (g) the mean curvature smoothes well but produces irregular mesh; and (h) smoothing according to (13).

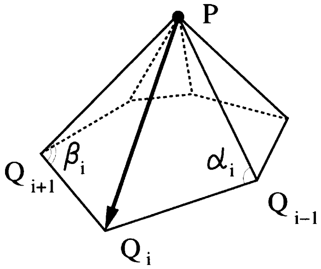

Fig. 5. α_i and β_i are the two angles opposite to the edge Q_iP.

accurate, robust, and reliable even on irregular meshes [5–7].

The two-dimensional analog of (5) is given by the left-hand-side of (5), since

$$\nabla(a+b) = \nabla a + \nabla B = -\left[\frac{\overrightarrow{OA}}{a} + \frac{\overrightarrow{OB}}{b}\right]$$

Thus, similar to $\mathscr{U}_1$, smoothing by discrete curvature flow (8), (9) worsens the mesh quality. See Fig. 1e where the mean curvature flow performs well in smoothing a noisy sphere but produces uneven distribution of vertices. See also Fig. 4d where the noisy torus from Fig. 4c is smoothed by the mean curvature flow.

3. Modified mean curvature flow

Let us consider a family of smooth surfaces $S(u, v, t)$, where (u, v) parameterize the surface and t parameterizes the family. We suppose (u, v) to be independent of t. Let us assume that this family evolves according to the evolution equation

$$\frac{\partial S(u,v,t)}{\partial t} = F\mathbf{n}, \qquad S(u,v,0) = S^{(0)}(u,v), \tag{10}$$

where $\mathbf{n}(u, v, t)$ is the unit normal vector for $S(u, v, t), F$ is a speed function, and $S^{(0)}(u, v)$ is an initial surface. The family parameter t can be considered as the time duration of the evolution. Eq. (10) means that the surface $S(u, v, t)$ moves along its normals with speed equal to F. Consider the flow

$$\frac{\partial S(u,v,t)}{\partial t} = F\mathbf{n} + G\mathbf{t}, \qquad S(u,v,0) = S^{(0)}(u,v), \tag{11}$$

where $\mathbf{t}$ is a vector tangent to the surface and G is a given function. Note that the tangent speed component does not affect the geometry of the evolving surface and changes only surface parameterization.

Solving (10) for a polygonal surface by an explicit Euler scheme is more stable for regular meshes. Therefore,

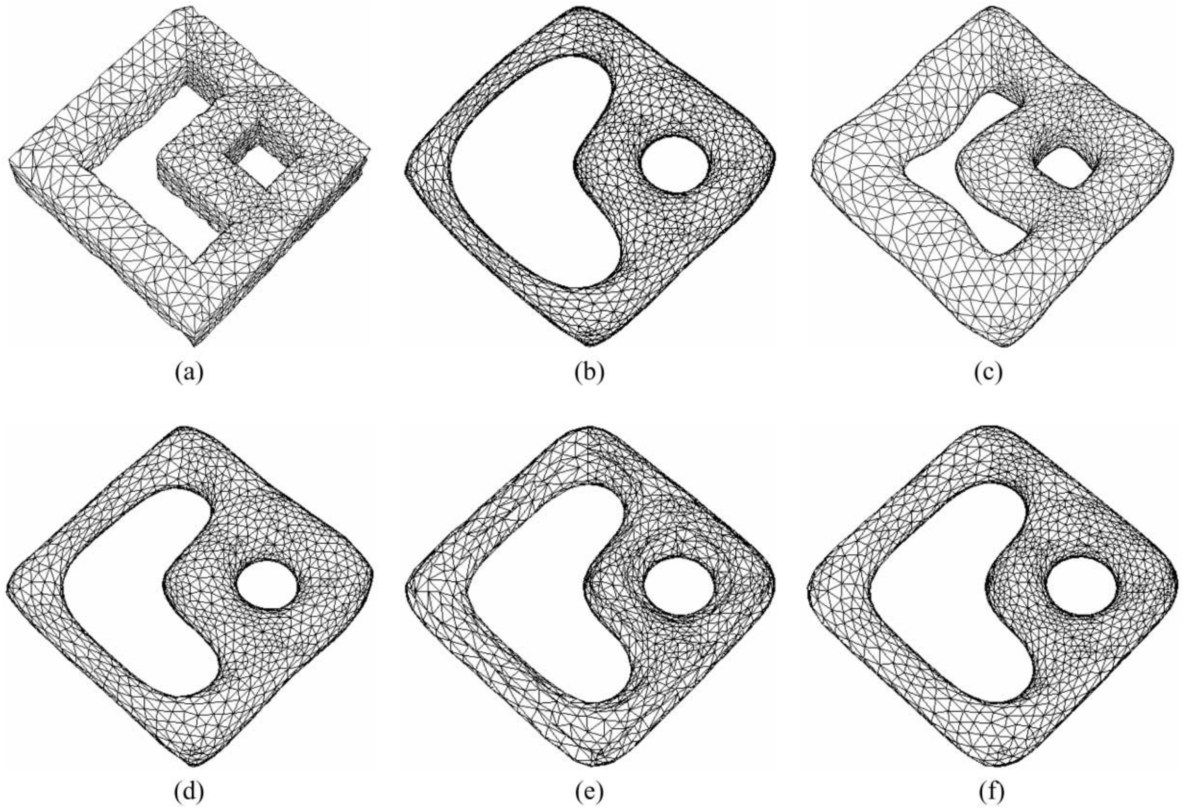

Fig. 6. (a) A two-holed torus given as a mesh of irregular connectivity and consisting of parts with different densities, and small random noise is added in the normal direction; (b) Laplacian smoothing homogenizes the mesh density, but deforms the initial shape irregularly; (c) smoothing by the Taubin method blurs shape creases and corners less than other flows but it also deforms the initial irregularly; (d) the bilaplacian flow deforms the initial shape in an irregular manner; (e) the mean curvature flow produces irregular mesh; and (f) smoothing according to (14), (15) produces a regular meshing surface, which shape is close to the shape produced by the mean curvature flow.

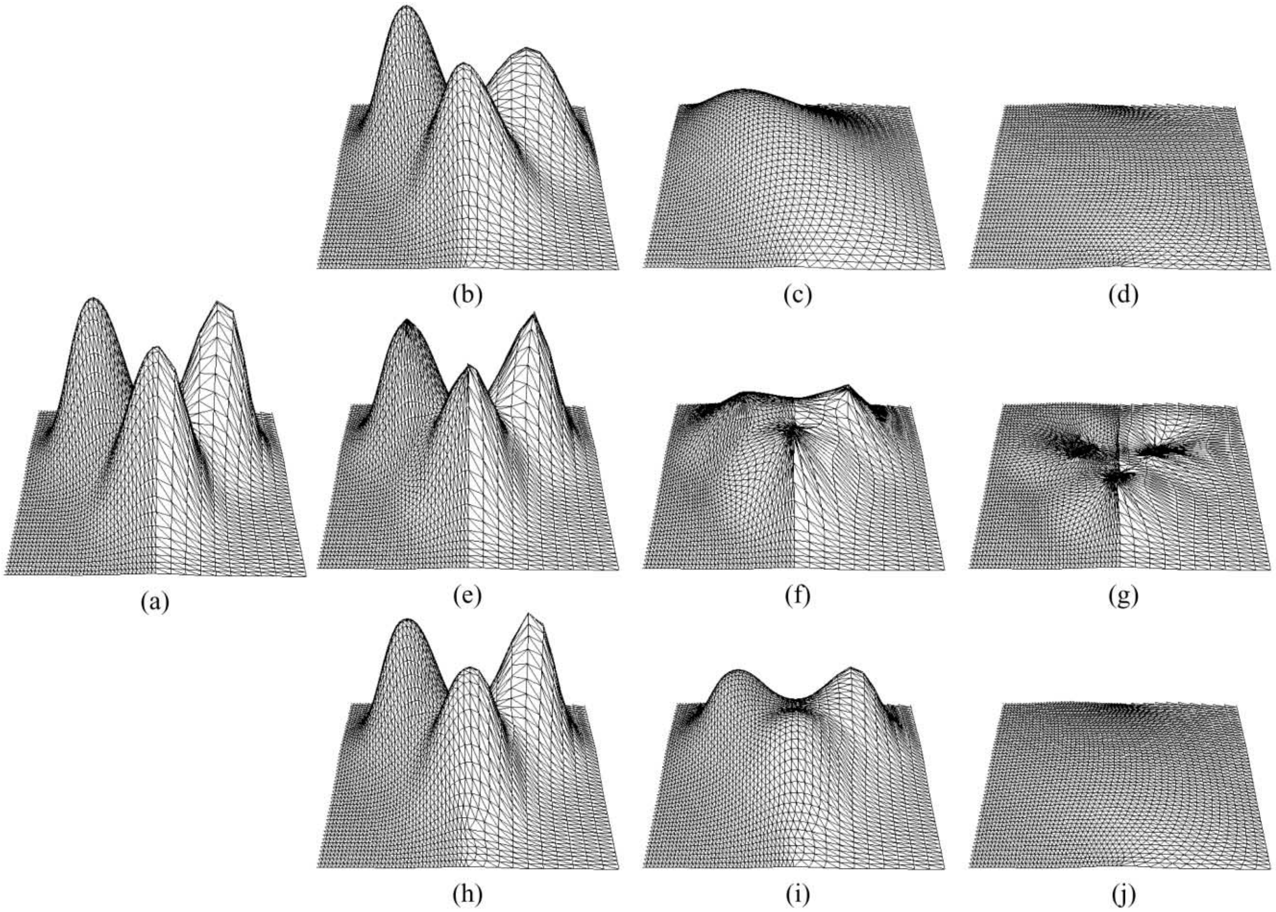

Fig. 7. (a) A polygonal surface consisting of two parts with different mesh densities; (b–d) deformation of (a) into a flat patch by the Laplacian flow; (e–g) deformation of (a) into a flat patch by the mean curvature flow; and (h–j) deformation of (a) into a flat patch by our method (14), (15).

it allows the use of larger time steps to achieve faster smoothing.

Note that a simple backward scheme proposed and used for smoothing in Ref. [6], in spite of its absolute stability, is not superior to Euler's scheme. For larger time steps where Euler's scheme is unstable, the backward scheme is inaccurate [12]. Another limitation of the backward scheme is explained in Ref. [25]. To take full advantage of absolute stability, it would be better to use the Crank–Nicholson method [17], which is second-order accurate in time, or adapt the AOS scheme proposed in Ref. [25].

Roughly speaking, our main idea of simultaneous mesh smoothing and regularization consists of using the normal speed component $F\mathbf{n}$ for polygonal surface smoothing and the tangent speed component for homogenizing the mesh density.

The idea of employing a tangential speed for a suitable shape parameterization is not new, of course. It was utilized in Ref. [8] for a suitable change of variables in a differential equation defining curve evolutions. It was used in Ref. [10] for selective image smoothing by the mean curvature and Beltrami flows and, in a similar way, in [7] for feature-preserving denoising of height data. A tangential speed component was also used in Ref. [1] in order to keep uniform distribution of the vertices of an evolving polyline. However, versions of (11) used in Refs. [1,7,10] fail to work properly for mesh regularization purposes. So, in Refs. [14,15] we proposed two flows combining the mesh evolution by the mean curvature with tangential regularization. In the rest of this section we describe the flows.

Let us use the discrete mean curvature flow (8), (9) for smoothing and the Laplacian flow (2), (3) for mesh regularization. One possible implementation is to take

$$F\mathbf{n} = H\mathbf{n} \text{ and } G\mathbf{t} = C[\mathcal{U}_0 - (\mathcal{U}_0\cdot\mathbf{n})\mathbf{n}], \tag{12}$$

where $\mathcal{U}_0$ is the umbrella vector (3), C is a positive constant, and the dot operator is the scalar product. One can see that $\mathcal{U}_0 - (\mathcal{U}_0\cdot\mathbf{n})\mathbf{n}$ is the projection of the umbrella vector onto the tangent plane. Flow (11), (12) was considered in Refs. [15,26]. Its explicit approximation leads to the local update rule

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda\{H(P_{\text{old}})\mathbf{n}(P_{\text{old}}) + C[\mathcal{U}_0(P_{\text{old}}) - (\mathcal{U}_0(P_{\text{old}})\cdot\mathbf{n}(P_{\text{old}}))\mathbf{n}(P_{\text{old}})]\} \tag{13}$$

Smoothing by (13) produces shapes of the same quality as the mean curvature flow but with uniform distribution of vertices, as seen in Fig. 4.

The mesh evolution (11), (12) was also used in Ref. [26] for mesh quality improvement.

One can also define the parameter C as a function of surface curvatures to achieve a higher mesh density in the curved surface regions.

However, according to our experience, a smoothing scheme described below produces better results.

Let $\mathbf{m} = \mathcal{U}_0/\|\mathcal{U}_0\|$ and θ be the angle between the mean curvature vector $H\mathbf{n}$ and $\mathbf{m}$: $\cos\theta = \mathbf{m}\cdot H\mathbf{n}/|H|$. Vector $\mathbf{m}$ defines a 3D analog of 2D median direction. Our basic idea is to move the vertices in the median direction such that the normal speed component is equal to the mean curvature. However, since for saddle vertices the median direction vector $\mathbf{m}$ and the mean curvature vector $H\mathbf{n}$ may have opposite normal components (i.e. $\theta > \pi/2$), as seen in Fig. 8b, we use the following flow

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda\mathcal{F}(P_{\text{old}}), \tag{14}$$

where

$$\mathcal{F} = \begin{cases} \dfrac{|H|\mathbf{m}}{\cos\theta} & \text{if } \cos\theta > \epsilon \\ 2H\mathbf{n} - \dfrac{|H|\mathbf{m}}{\cos\theta} & \text{if } \cos\theta < -\epsilon \\ 0 & \text{if } |\cos\theta| \le \epsilon \end{cases} \tag{15}$$

Here, ϵ is a small positive parameter. Geometric ideas behind (14) are explained in Fig. 8.

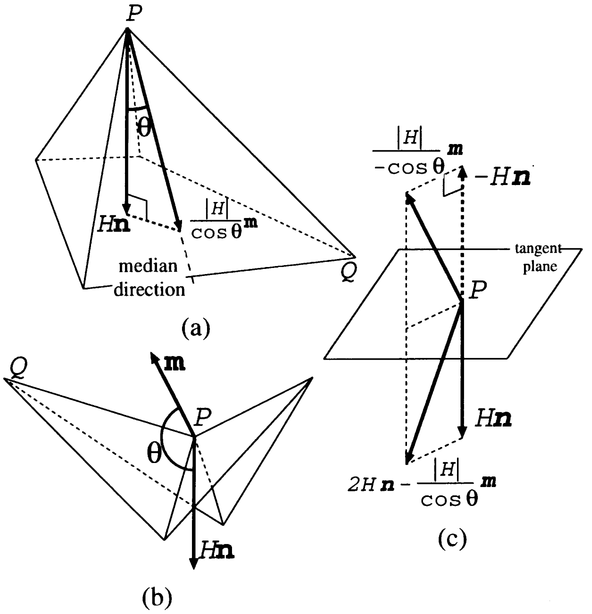
P
θ
Hn
$\frac{|H|}{\cos\theta}\mathbf{m}$
median direction
Q
(a)
Q
m
P
θ
Hn
(b)
$\frac{|H|}{-\cos\theta}\mathbf{m}$
$-H\mathbf{n}$
tangent plane
P
Hn
$2H\mathbf{n} - \frac{|H|}{\cos\theta}\mathbf{m}$
(c)

Fig. 8. (a) The case when the median and normal vectors lie on the same side of the tangent plane: moving in the median direction with normal speed component equal to the mean curvature; (b) the case when the median and the normal vectors lie on opposite sides of the tangent plane may happen for saddle vertices; and (c) computation of the speed vector in (b).

If the normal and median vectors are almost orthogonal to each other at a vertex ($|\cos\theta| \le \epsilon$), we do not move the vertex at all.

According to our experience, choosing $\epsilon = 0.1$ produces good results independently of the mesh density.

The speed function $\mathcal{F}$ defined in (15) is discontinuous and the flow may potentially develop undesirable shape deformations. However, we have not encountered such problems in our numerical experiments.

The smoothing scheme (14), (15) demonstrates better results than the Laplacian flow, the bilaplacian flow, the mean curvature flow, and the Taubin smoothing scheme, as seen in Figs. 1 and 6. See also Fig. 7 where a polygonal surface consisting of two parts with different mesh densities is smoothed by the Laplacian flow, the mean curvature flow, and our scheme (14) and (15).

4. How to avoid oversmoothing

Laplacian smoothing, the mean curvature flow, and our new technique presented in the previous section smooth polygonal surfaces by suppressing high frequency surface oscillations. Lacking local surface control, they may oversmooth and, therefore, destroy desirable surface features. To incorporate local control for smoothing and to reduce possible oversmoothing due to excessive iterations, let us consider a simple but very useful modification allowing the smoothing process to slow down adaptively:

$$P_{\text{new}} \leftarrow P_{\text{old}} + \lambda q\mathcal{F}(P_{\text{old}}) \tag{16}$$

with

$$q\mathcal{F} = \begin{cases} (\|\mathcal{F}(P)\| - T)\dfrac{\mathcal{F}(P)}{\|\mathcal{F}(P)\|} & \text{if } \|\mathcal{F}(P)\| > T \\ 0 & \text{if } \|\mathcal{F}(P)\| \le T \end{cases} \tag{17}$$

where T is a positive threshold specified by a user. According to this modification, the smoothing is performed only on those mesh vertices P where $\|\mathcal{F}(P)\| > T$. See Fig. 9 to compare performance of smoothing schemes (14)–(17).

Of course, the above simple modification can be applied to any of the previously considered smoothing flows.

In our experiments, typified in Fig. 9, the parameter T in (16) is the same for all vertices. It is natural to allow T to be dependent on shape characteristics at the vertices: $T = T(P)$. In our experiments, we define the threshold $T(P)$ at the vertex P as the arithmetic mean of the mean curvatures computed at the first ring of neighbors of P or at the first and second rings of neighbors of P. Fig. 10 demonstrates our experiments with smoothing a Noh mask triangulated model reconstructed from a cloud of points.

According to our experiments, the best smoothing strategy consists of periodically recomputing the threshold $T(P)$ for each vertex P after a fixed number of smoothing iterations. Fig. 10c (the bottom row) shows advantages of such a threshold recomputation scheme.

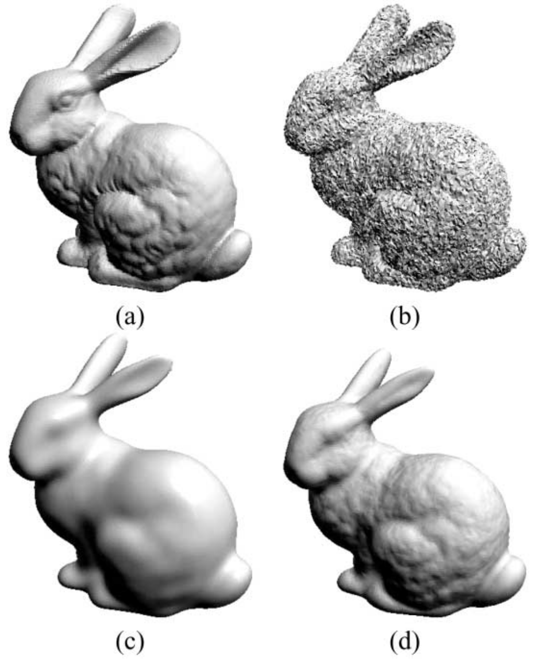

Fig. 9. (a) The original Stanford bunny; (b) uniform noise added; (c) smoothing by (14) (the Laplacian and mean curvature flows produce similar results); and (d) smoothing by (16) with the same number of iterations as in (c).

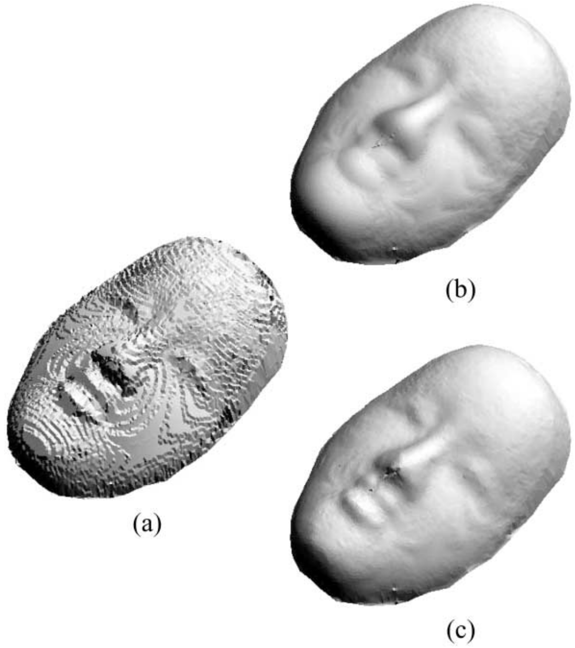

Fig. 10. (a) A triangulated Noh mask model reconstructed from a scattered data generated by a laser scanner system; (b) smoothing according to (16) with constant T chosen manually; and (c) smoothing according to (16) with $T(P)$ equal to the arithmetic mean of the mean curvatures of the first and second rings of neighbors of P.

5. Crease-enhancing diffusion

Conventional smoothing schemes flatten high curvature surface regions first and, therefore, destroy surface crease structures. In this section, we propose an adaptive smoothing method which preserves sharp variation points of the surface normals and, hence, is good for stable extraction of perceptually salient surface creases.

Our idea is based on the Saint-Marc–Medioni adaptive image smoothing method [18], which can be considered a simplification of the Perona–Malik nonlinear diffusion approach [16]. The method is based on iterative use of a 3×3 nonlinear averaging scheme. Let a gray-scale image be given by its intensity function $z = I(x, y)$. The scheme is applied iteratively to the image intensity function

$$I(x, y, k+1) = \frac{1}{\sum w_{ij}^k} \sum_{i,j=-1}^{1} w_{ij}^k I(x+i, y+j, k),$$

where $I(x, y, 0) = I(x, y)$, the coefficients w_{ij}^k are values of a monotonically decreasing function applied to the absolute values of the image intensity gradient computed at each neighbor's position, and k enumerates the iterations. For example, a convenient choice of the weights w_{ij}^k is given by

$$w_{ij}^k = \exp(-c|\nabla((x, y, k)|^2),$$

where c is a positive constant.

Since the image intensity gradient is large at edge pixels, the method smoothes an image while preserving and enhancing image intensity discontinuities (edges).

To extract gradient discontinuities (image creases), Saint-Marc and Medioni apply their nonlinear averaging method to the image intensity gradient image [18].

5.1. *Diffusion of normals*

Our basic idea is to apply a similar nonlinear diffusion process to the surface normals. Consider an oriented triangular mesh. Let R be a mesh triangle and $\mathbf{n}(R)$ be the unit normal of R. Denote by $\mathcal{N}(R)$ the set of all mesh triangles that have a common edge or vertex with R. Consider the following discrete diffusion process acting on vectors associated with the mesh triangles

$$\mathbf{m}(R) = \frac{1}{\sum w_k(R, S)} \sum_{S \in \mathcal{N}(R)} w_k(R, S)\mathbf{m}(S, k) \qquad (18)$$

$$\mathbf{m}(R, k+1) = \mathbf{m}(R)/\|\mathbf{m}(R)\| \qquad (19)$$

$$\mathbf{m}(R, 0) = \mathbf{n}(R),$$

with weights $w_k(R, S)$ given by

$$w_k(R, S) = A(S)\exp(-cK^2), \qquad K = \varphi d,$$

where $A(S)$ denotes the area of S, c is a positive constant, $\varphi = \angle(\mathbf{m}(R, k), \mathbf{m}(S, k))$ is the angle between the vectors $\mathbf{m}(R, k)$ and $\mathbf{m}(S, k)$, and d is the distance between the centroids of triangles R and S. The geometric meaning of the constant K is explained in the right side of Fig. 11: if $\mathbf{m}(R, k)$ and $\mathbf{m}(S, k)$ were the usual normals for triangles R

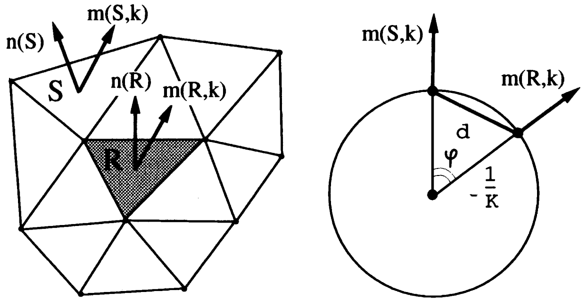

Fig. 11. Left: mesh triangles R and S, their normals $\mathbf{n}(R)$ and $\mathbf{n}(S)$, 'diffused normals' $\mathbf{m}(R, k)$ and $\mathbf{m}(S, k)$ (see (18), (19)). Right: directional curvature estimation.

and S, then K would approximate the directional curvature at the centroid of R in the direction defined by the centroid of S. A similar approximation of the directional curvature was used in Ref. [9].

The diffusion process (18), (19) allows us to detect salient mesh creases as the edges whose adjacent triangles, say R and S, have the angle $\vartheta = \angle(\mathbf{m}(R),\mathbf{m}(S))$ close to a straight angle. See the top row of Fig. 12 where the Venus model (left) and its rendering according to the vector field $\{\mathbf{m}\}$obtained after a number of averaging iterations (18), (19) (right) are presented.

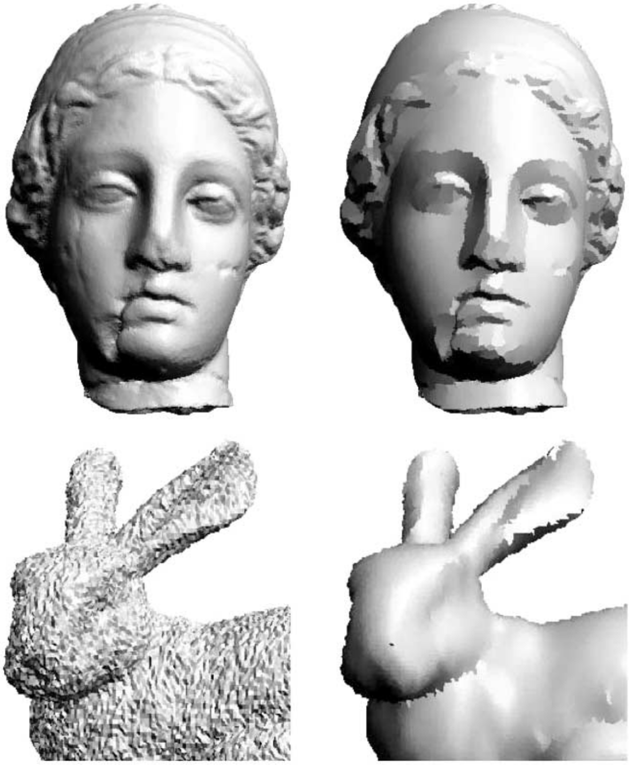

Fig. 12. Top left: the Venus model (courtesy Cyberware). Top right: the model is shaded according to a vector field obtained after a number of averaging iterations (18), (19). Bottom left: a noisy bunny model. Bottom right: the model is shaded according to a vector field generated by (18), (19).

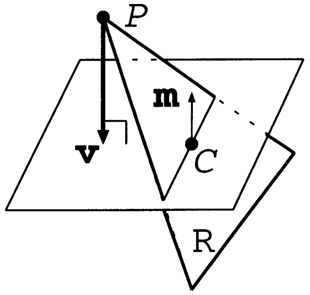

Fig. 13. Updating vertex position, see (20).

The diffusion process (18), (19) can also be used for detection of salient creases on noisy models. See the bottom row of Fig. 12 for our experiments with a noisy bunny model.

5.2. Coupled diffusion of normals and vertices

The integrability of the family of planes defined by the normals $\mathbf{m}(R, k)$ will be lost after even one pass of (18). To enforce integrability, we update the vertex positions after every step of the diffusion process (18) as follows. Let P be a mesh vertex, R be a mesh triangle adjacent to P, $C = C(R)$ be the centroid of R, and $\mathbf{m}(R)$ be the vector computed according to (18) and associated with R. The new position for vertex P is computed by

$$P_{\text{new}} \leftarrow P_{\text{old}} + \frac{1}{\sum A(R)} \sum A(R)\mathbf{v}(R), \qquad (20)$$

where the sums are taken over all triangles R adjacent to P, $\mathbf{v}(R) = [\overrightarrow{PC}\cdot\mathbf{m}(R)]\mathbf{m}(R)$ is the projection of the vector $\overrightarrow{PC}$ onto the $\mathbf{m}(R)$ direction (see Fig. 13), and $A(R)$ denotes the area of R.

Now the complete mesh diffusion process consists of repeating the following two successive steps applied to the mesh triangles and vertices.

Step 1. For each mesh triangle R, compute the triangle normal $\mathbf{n}(R)$ and perform the following weighted averaging of normals:

$$\mathbf{m}(R) = \frac{1}{\sum w(R,S)} \sum_{S\in\mathcal{N}(R)} w(R,S)\mathbf{n}(S) \qquad (21)$$

with

$$w(R,S) = A(S)\exp(-cK^2), \qquad K = \varphi/d. \qquad (22)$$

Here, c is a positive constant, $\vartheta = \angle(\mathbf{n}(R),\mathbf{n}(S))$ is the angle between the vectors $\mathbf{n}(R)$ and $\mathbf{n}(S)$, and d is the distance between the centroids of triangles R and S (see the left side of Fig. 11).

Step 2. For each mesh vertex P, perform the vertex updating procedure (20).

This coupled diffusion of normals and vertices removes

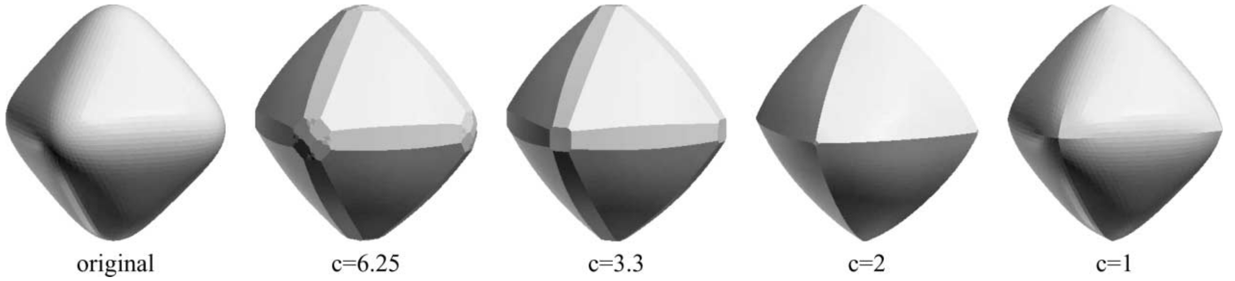

Fig. 14. A rounded octahedron (left) taken as the initial condition in (20)–(22) and its steady-states obtained for different values of the exponential coefficient c in (22) (500 iterations were used). Smoothing with large c leads to enhancing even subtle crease structures while smoothing with small c enhances only the main creases and corners.

noise and small-scale oscillations while preserving and sharpening salient shape creases. The bottom row of Fig. 1 demonstrates advantages of our approach in comparison with Taubin's smoothing scheme and the mean curvature flow. Not only does our coupled diffusion of normals and vertices remove the noise perfectly, but it also repairs small defects of the original model.

Fig. 14 demonstrates dependence of the smoothing process on the exponential coefficient c in (22). Smoothing with large c leads to enhancing even subtle crease structures, while smoothing with small c enhances only the main creases and corners.

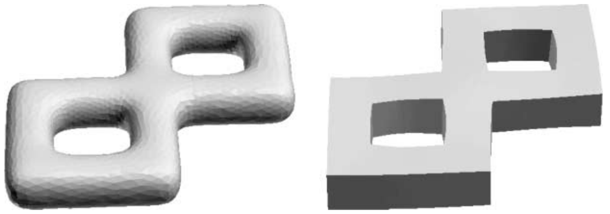

Fig. 15. Left: a polygonal two-holed torus model. Right: after smoothing by the coupled diffusion of normals and vertices, the sharpened creases provide natural decomposition of the model.

Fig. 15 shows how the proposed coupled diffusion of normals and vertices sharpens ridge structures on a polygonal two-holed torus model. The sharpened ridges provide a perceptually natural decomposition of the model.

Fig. 16 exposes restoration of a piecewise linear polygonal surface (octahedral) corrupted by noise. Our nonlinear diffusion of normals and vertices did the restoration job better than the method developed in Ref. [4]where a similar octahedron model was used for testing.

For dense triangular meshes approximating smooth shapes accurately, the dihedral angles between triangles sharing a common edge are close to a straight angle. Let us call an edge sharp if its dihedral angle is below a certain threshold, for example 160°. We define the ridge and ravine edges as the convex and concave sharp edges, respectively. Fig. 17 demonstrates results of detection of the ridge (black) and ravine (white) edges on the Stanford bunny model for different values of the exponential coefficient c in (22). The sharp edges

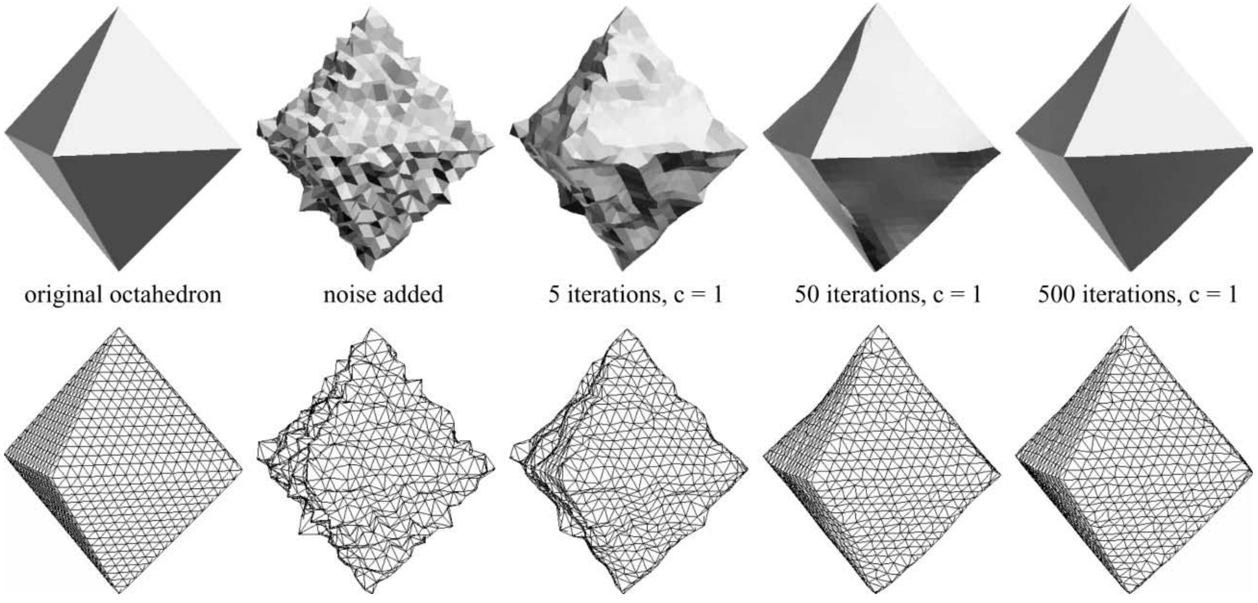

Fig. 16. Smoothing a noisy octahedron by the proposed coupled diffusion of normals and vertices.

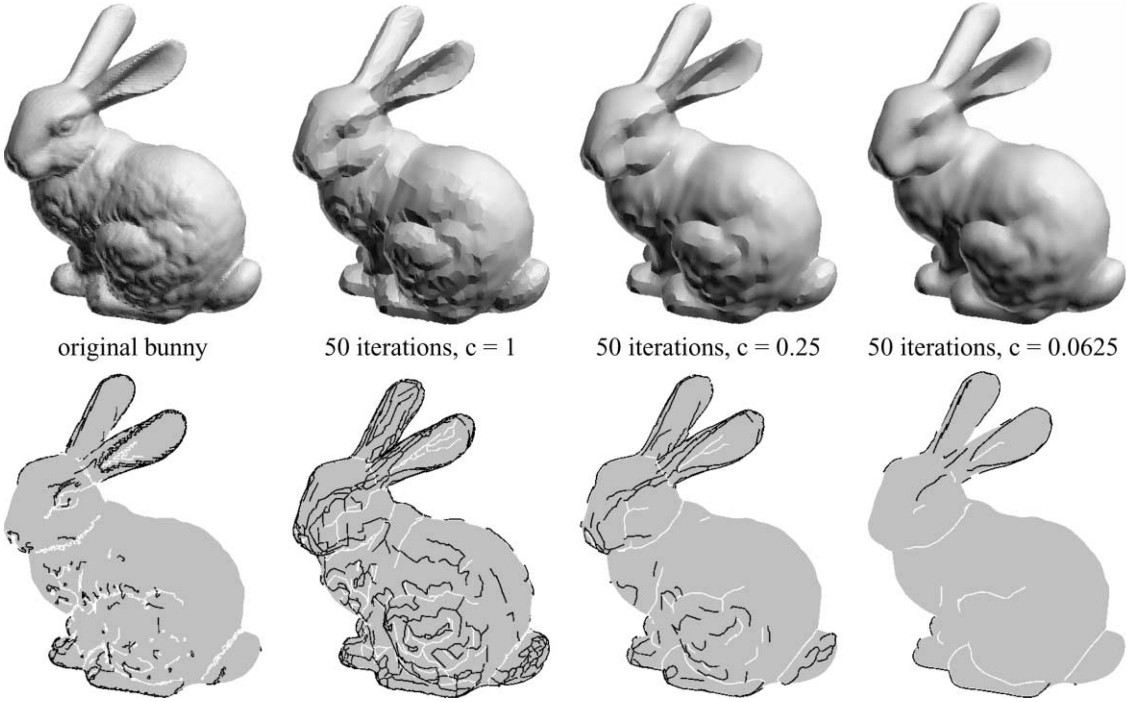

Fig. 17. Top row: smoothing the Stanford bunny model, with different values of the exponential coefficient c in (22). Bottom row: ridges (convex sharp edges) and ravines (concave sharp edges) colored in black and white, respectively, are detected on the top row models.

detected on the smoothed models form a pattern of ridges and ravines which is in good agreement with our perception of ridges and ravines on the original model. The exponential coefficient c can be used for multiscale shape analysis.

6. Conclusion

We have presented new methods for smoothing polygonal surfaces with simultaneous mesh regularization (Section 3), feature preserving (Section 4), and crease enhancement (Section 5).

Our crease-enhancing coupled diffusion of normals and vertices can be used for stable extraction of salient ridge and ravine structures and for perceptually natural shape decomposition.

The current implementation of the coupled diffusion of normals and vertices is sensitive to the choice of exponential coefficient c in (22). In future, we plan to choose c automatically and change it during the smoothing process according to statistical properties of an evolving polygonal surface.

Another interesting direction for future research is shape enhancement, the opposite operation to smoothing. Simple inversion of a smoothing flow is usually unstable. However, stability can be achieved by incorporating a curvature-based local control similar to that which we developed in Section 4.

Acknowledgements

We would like to thank the anonymous reviewers of this paper for their valuable and constructive comments, from which the revision of this paper has benefited significantly. Ilia Bogaevski is grateful to the University of Aizu where he spent one year as a visiting researcher 1998–1999 when this research was partially conducted.

References

[1] Belyaev AG, Anoshkina EV, Yoshizawa S, Yano M. Polygonal curve evolutions for planar shape modeling and analysis. International Journal of Shape Modeling 1999;5(2):195–217.

[2] Belyaev AG, Ohtake Y. An image processing approach to detection of ridges and ravines on polygonal surfaces. In: de Sousa A, Torres JC, editors. Eurographics 2000, Short Presentations, August, 2000. p. 19–28.

[3] Belyaev AG, Ohtake Y, Abe K. Detection of ridges and ravines on range images and triangular meshes. In: Vision Geometry IX, Proc. SPIE 4117, July–August, San Diego, 2000. p. 146–54.

[4] Clarenz U, Diewald U, Rumpf M. Anisotropic geometric diffusion in surface processing. In: IEEE Visualization 2000, October, 2000. p. 397–405.

[5] Desbrun M, Meyer M, Schröder P, Barr AH. Discrete differential-geometry operations in *n*D. Available on WWW at http://www.multires.caltech.edu/pubs/pubs.htm

[6] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. Computer Graphics SIGGRAPH 1999;99:317–24.

[7] Desbrun M, Meyer M, Schröder P, Barr AH. Anisotropic feature-preserving denoising of height fields and bivariate data. In: Graphics Interface 2000, May, 2000. p. 145–52.

[8] Gage M. On an area-preserving evolution equation for plane curves. Contemporary Mathematics 1986;51:51–62.

[9] Hausler G, Karbacher S. Reconstruction of smoothed polyhedral surfaces from multiple range images. In: Girod B, Niemann H, Seidel H-P, editors. 3D Image Analysis and Synthesis '97, Sankt Augustin, Germany, Infix Verlag, 1997. p. 191–8.
[10] Kimmel R, Malladi R, Sochen N. Images as embedding maps and minimal surfaces: movies, colors, and volumetric medical images. In: IEEE CVPR '97, June, Puerto Rico, 1997. p. 350–5.
[11] Kobbelt L, Campagna S, Vorsatz J, Seidel H-P. Interactive multi-resolution modeling on arbitrary meshes. In: Computer Graphics SIGGRAPH 98 Proceedings, 1998. p. 105–14.
[12] Mohtar R, Segerlind L. Accuracy-based time step criteria for solving parabolic equations. In: Modeling, mesh generation, and adaptive numerical methods for partial differential equations, IMA volume 75. New York, Springer, 1995. p. 153–63.
[13] Ohtake Y, Belyaev AG. Nonlinear diffusion of normals for stable detection of ridges and ravines on range images and polygonal moels. In: MVA 2000, IAPR Workshop on Machine Vision Applications, November, Tokyo, 2000. p. 497–500.
[14] Ohtake Y, Belyaev AG, Bogaevski IA. Polyhedral surface smoothing with modified Laplacian and curvature flows. Journal of Three Dimensional Images 1999;13(3):19–24.
[15] Ohtake Y, Belyaev AG, Bogaevski IA. Polyhedral surface smoothing with simultaneous mesh regularization. In: Martin R, Wang W, editors. Geometric Modeling and Processing 2000, April, Hong Kong, 2000. p. 229–37.
[16] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12(7):629–38.
[17] Press W, Flannery B, Teukolsky S, Vetterling W. Numerical recipes in C: the art of scientific computing. New York, Cambridge University Press, 1992.
[18] Saint-Marc P, Medioni G. Adaptive smoothing for feature extraction. In: Proc. of DARPA Image Understanding Workshop, April, Cambridge, MA, 1988. p. 1100–13.
[19] Sethian JA. Level set methods and fast marching methods. New York, Cambridge University Press, 1999.
[20] Solé AF, López A, Sapiro G. Crease enhancement diffusion. IMA Preprint Series #1707, Institute for Mathematics and Its Applications, April 2000.
[21] Taubin G. Curve and surface smoothing without shrinkage. In: International Conference on Computer Vision (ICCV'95), 1995. p. 852–7.
[22] Taubin G. A signal processing approach to fair surface design. In: Computer Graphics SIGGRAPH 95 Proceedings, 1995. p. 351–8.
[23] Taubin G, Zhang T, Golub G. Optimal surface smoothing as filter design. In: Fourth European Conference on Computer Vision (ECCV'96), 1996.
[24] Vollmer J, Mencl R, Muller H. Improved Laplacian smoothing of noisy surface meshes. Computer Graphics Forum (Proc. Eurographics 1999) 1999;18(3):131–8.
[25] Weickert J. Anisotropic diffusion in image processing. Stuttgart: Teubner, 1998.
[26] Wood Z, Desbrun M, Schröder P, Breen D. Semi-regular mesh extraction from volumes. In: IEEE Visualization 2000, October, Salt Lake City, 2000.

Yutaka Ohtake is currently a doctoral course student at the Graduate School of Computer Science and Engineering of the University of Aizu, Japan. He received his BS and MS degrees in Computer Science and Engineering from the University of Aizu in 1997 and 1999, respectively. His research interests include geometric modeling and processing, computational geometry and topology.

Alexander Belyaev is currently an Associate Professor at the Department of Computer Software of the University of Aizu, Japan. His research interests include geometric modeling, image processing, classical differential geometry, and partial differential equations. Belyaev received MS and PhD degrees in Mathematics from Lomonosov Moscow State University in 1986 and 1990, respectively. See http://www.u-aizu.jp/~belyaev/ for more information.

Ilia Bogaevski is currently an Associate Professor at the Independent University of Moscow, Russia. He belongs to the famous Russian school of singularity theorists under the leadership of V.I. Arnold. Bogaevski received MS and PhD degrees in Mathematics from Lomonosov Moscow State University in 1986 and 1990, respectively.