Polyhedral Surface Smoothing with Simultaneous Mesh Regularization

Yutaka Ohtake
The University of Aizu
Aizu-Wakamatsu City
Fukushima 965-8580 Japan
d8011101@u-aizu.ac.jp

Alexander G. Belyaev
The University of Aizu
Aizu-Wakamatsu City
Fukushima 965-8580 Japan
belyaev@u-aizu.ac.jp

Ilia A. Bogaevski
Independent University of Moscow
Bolshoi Vlasevsky Pereulok 11
Moscow, Russia
bogaevsk@msfu.msk.ru

Abstract

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

This paper develops methods for polyhedral surface smoothing and denoising with simultaneous increasing mesh regularity.

We also propose an adaptive smoothing method allowing to reduce possible oversmoothing.

Roughly speaking, our smoothing schemes consist of moving every vertex in the direction defined by the Laplacian flow with speed equal to a properly chosen function of the mean curvature at the vertex.

Keywords: mesh smoothing and denoising, mesh regularity, Laplacian smoothing, mean curvature flow

1 Introduction

Recent advances in 3D data-acquisition hardware call for development of methods to rapidly remove undesirable noise and small scale oscillations from an irregularly triangulated data.

Two the most popular approaches for smoothing and denoising of polyhedral 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, it lacks a 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, moves recursively each vertex of the mesh by a displacement equal to a positive scale factor times the average of the neighboring vertices. 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 opens many ways for modifications and improvements.

Taubin in [8] 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 produces shrinkage and suppresses high frequencies of a discrete Laplacian operator defined on the mesh, while enhancing low frequencies.

When the scale factors are equal in magnitude, the Taubin smoothing scheme turns to the bilaplacian smoothing flow [4] which can be considered as 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 in [9].

A significant development of the Laplacian smoothing method, mesh smoothing by the mean curvature flow [3, 6] (see also references therein), came from mathematics and material science. 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 became extremely popular in geometric image processing (see, for instance, [6] for references).

Laplacian smoothing, Taubin smoothing, and the discrete mean curvature flow contain a number of drawbacks. The Laplacian smoothing flow increases the mesh regularity but develops unnatural deformations while being applied to a highly irregular mesh. Smoothing by the discrete mean curvature flow is relatively independent of the mesh sampling rate but increases the mesh irregularity. Both the Laplacian and mean curvature flows do not decelerate the smoothing process and may lead to oversmoothing and loosing desirable geometric features. The Taubin smoothing scheme lacks a local shape control and enhances low frequency surface features.

In this paper we propose simple and effective polyhedral

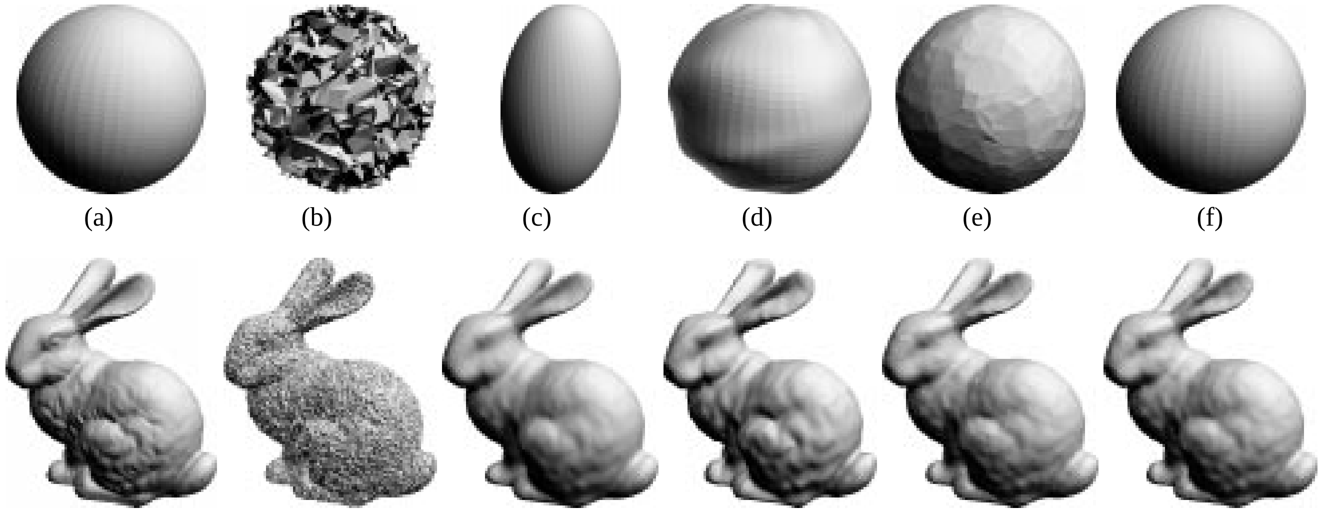

Figure 1. The top row: (a) a polyhedral sphere; (b) the sphere with a uniform noise added; (c) Laplacian smoothing develops unnatural deformations; (d) smoothing by the Taubin method converts high-frequency surface oscillations into low-frequency waves; (e) smoothing by the mean curvature flow increases the mesh irregularity; (f) smoothing according to a method proposed in this paper, see Section 3. The bottom row: (a) the Stanford bunny; (b) the bunny with a uniform noise added; (c) Laplacian smoothing with a number of iterations chosen to achieve 'a good looking' result: extra iterations will lead to oversmoothing; (d) Taubin smoothing reduces high-frequency surface oscillations but enhances low-frequency oscillations: extra iterations will lead to enhancing of surface wrinkles; (e) smoothing by the mean curvature flow with a number of iterations chosen to achieve 'a good looking' result: extra iterations will lead to oversmoothing; (f) smoothing according to a method proposed in this paper: the smoothing process slows down automatically and extra iterations will produce almost the same appearance, see Section 4.

surface smoothing schemes which combine the best properties of the Laplacian smoothing flow and discrete mean curvature flow, often outperform best existing smoothing methods, and, in addition, reduce possible oversmoothing.

Fig. 1 demonstrates some of our results.

The paper is organized as follows. In Section 2, we introduce the Laplacian flow, the Taubin smoothing method, and the discrete mean curvature flow. In Section 3, our new technique which combines the best properties of the Laplacian and discrete mean curvature flow is explained. In Section 4, we describe our technique to reduce possible oversmoothing. We discuss application of smoothing methods for stable detection of ridges and ravines on a polyhedral surfaces in Section 5. 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 will introduce and analyze four methods for polyhedral surface smoothing: Laplacian smoothing, Taubin smoothing [8], the bilaplacian flow [4], and the mean curvature flow [3].

Laplacian Smoothing. Let us consider a triangulated surface and, for any vertex P, let us define the so-called umbrella-operator [4]

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

where summation is taken over all neighbors of P, w_i are positive weights. See Fig. 2.

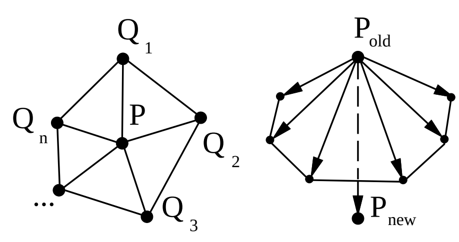

Figure 2.

The local update rule

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

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 set 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 [8] is set the weights as the inverse distances between P and its neighbors:

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

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

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

with distances $a = |OA|$ and $b = |OB|$ between them. Let $d\mathbf{r}/ds = \mathbf{t}$ and $\mathbf{n} = \mathbf{t}^{\perp}$ compose the Frenet frame at O, see Fig. 3. Simple manipulations with Taylor series expansions

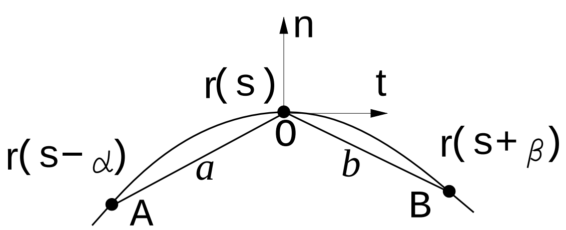

Figure 3.

and Frenet formulas show that

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

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

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 with $\mathcal{U}_0$ develops unnatural deformations, see Fig. 1c (the top row).

Taubin Smoothing. Taubin in [8] 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 [8]. Combining the two successive steps of the Taubin method in one local update rule we arrive at

$$P_{\text{new}} \longleftarrow (1-\mu\mathcal{U})(1+\lambda\mathcal{U})P_{\text{old}} = \\ = P_{\text{old}} - (\mu-\lambda)\mathcal{U}(P_{\text{old}}) - \mu\lambda\mathcal{U}^2(P_{\text{old}}), \tag{6}$$

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 [8], the best smoothing with (6) is obtained when $\mathcal{U} = \mathcal{U}_0$ or $\mathcal{U} = \mathcal{U}_1$. In our experiments, Taubin smoothing with $\mathcal{U} = \mathcal{U}_0$ works much better than with $\mathcal{U} = \mathcal{U}_1$. However, even the first Taubin smoothing scheme often produces poor results, see for example Fig. 1d (the top row) where we used $\lambda = 0.3$ and $1/\lambda - 1/\mu = 0.1$. In Fig. 1d (the bottom row) the Taubin filtering scheme demonstrates a good performance. Nevertheless, one can note enhancing of low-frequency surface wrinkles.

Bilaplacian flow. The bilaplacian flow

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

is a discrete analog of the steepest descent flow for the the thin-plate energy functional [4]. 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).

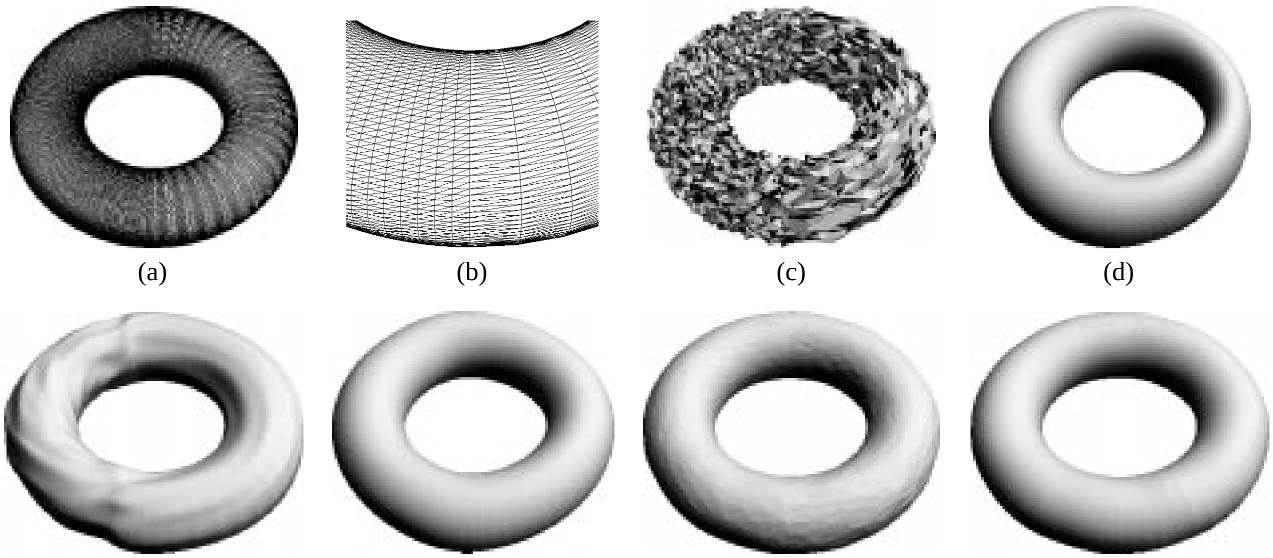

Figure 4. (a) A torus with consisting of two parts with different sampling rates; (b) a magnified view of a part of the torus; (c) the torus with a uniform noise added; (d) Laplacian smoothing deforms the initial shape; (e) smoothing by the Taubin method reduces high-frequency surface oscillations but develops low-frequency surface waves; (f) the bilaplacian flow smoothes well but slightly deforms the initial shape; (g) the mean curvature flow smoothes well but produces irregular mesh; (h) smoothing according to (11).

Mean curvature flow. Recently it was reported [3] that the smoothing procedure

$$P_{\text{new}} \longleftarrow P_{\text{old}} + \lambda\, H(P_{\text{old}})\, \mathbf{n}(P_{\text{old}}), \tag{7}$$

where H is a discrete version of the mean curvature and $\mathbf{n}$ is the unit normal vector, produces better results than Laplacian smoothing (2).

According to [3], a good estimation of the mean curvature vector at a vertex P is given by

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

where $A = \sum A_i$ is the sum of the areas of the triangles surrounding P. Calculations [3] show that

$$H\mathbf{n}(P) = \frac{1}{4A} \sum_i (\cot\alpha_i + \cot\beta_i)(Q_i - P), \tag{8}$$

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

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

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

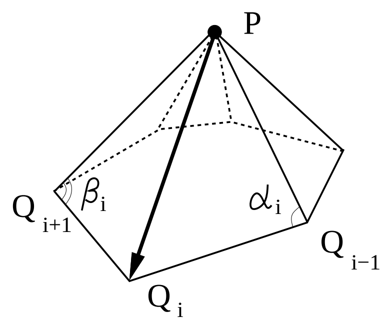

Figure 5.

Thus, similar to $\mathcal{U}_1$, smoothing by discrete curvature flow (7), (8) worsens the mesh sampling rate. See Fig. 1e (the top row) where the mean curvature flow demonstrates a good performance in smoothing a noisy sphere but produces uneven distribution of vertices.

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

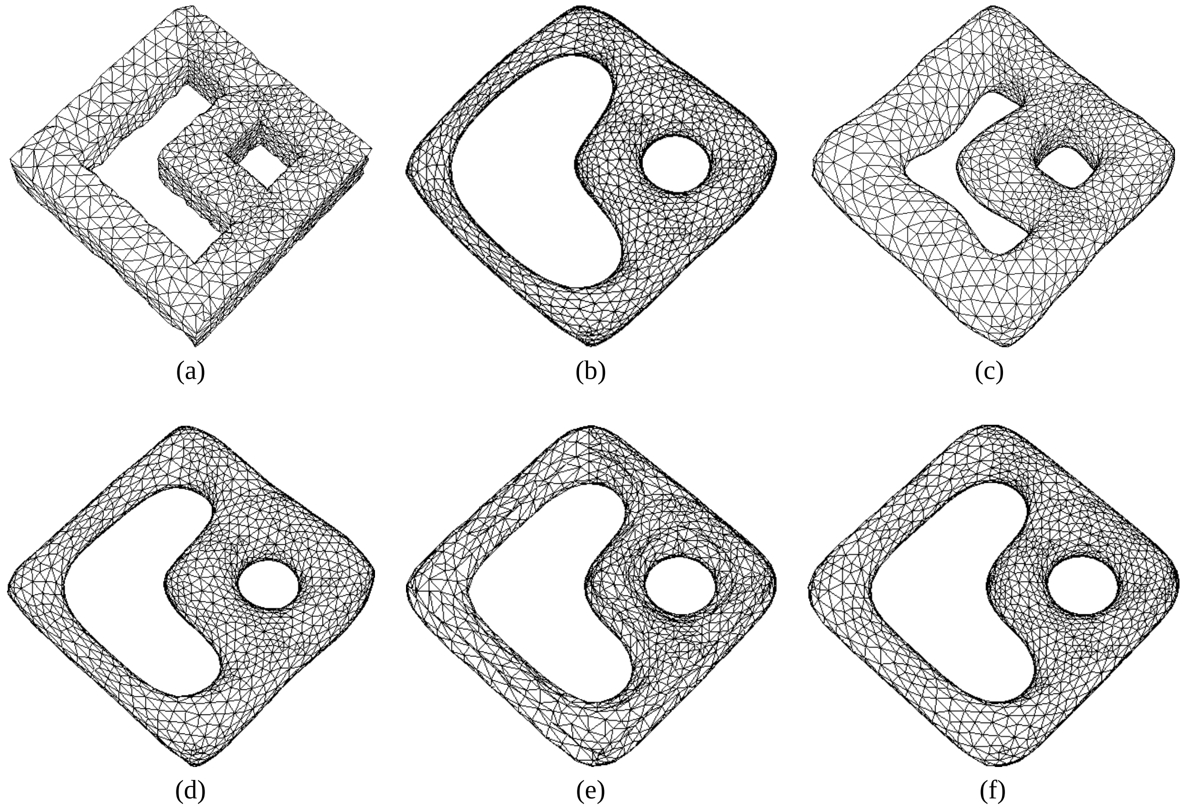

Figure 6. (a) A polyhedral two-holed torus given as a mesh having irregular connectivity and consisting of parts with different sampling rates, a small noise is also added; (b) Laplacian smoothing improves the mesh sampling rate but deforms the initial shape; (c) smoothing by the Taubin method substantially deforms the initial shape in unnatural way; (d) the bilaplacian flow also deforms the initial shape in an irregular manner. (e) the mean curvature flow produces irregular mesh; (f) smoothing according to (12), (13) produces a regular meshing surface which shape is close to the shape produced by the mean curvature flow.

equation

$$\frac{\partial S(u,v,t)}{\partial t} = F\,\mathbf{n}, \qquad S(u,v,0) = S^{(0)}(u,v), \qquad (9)$$

where $\mathbf{n}(u,v,t)$ is the unit normal vector for $S(u,v,t)$, F is a speed function, $S^{(0)}(u,v)$ is an initial surface. The family parameter t can be considered as the time duration of the evolution. Equation (9) means that the surface $S(u,v,t)$ moves along its normals with speed equal to F. Consider now the flow

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

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 (9) for a polyhedral surface by an explicit Euler scheme is more stable for polyhedral surfaces with uniform mesh sampling rates and, therefore, allows to use larger time steps to achieve faster smoothing. Note that a backward scheme (it was proposed and used for smoothing in [3]) is no better than Euler's scheme. For larger time steps where Euler's scheme is unstable the backward scheme is inaccurate [5].

Roughly speaking, *our main idea of simultaneous mesh smoothing and regularization consists of using the normal speed component* $F\,\mathbf{n}$ *for polyhedral surface smoothing and the tangent speed component to improve the mesh sampling rate*. For polygonal curve evolutions this idea was proposed and used in [1].

Let us use the discrete mean curvature flow (7), (8) for smoothing and the Laplacian flow (2), (3) for improving the mesh sampling rate. One possible implementation is to take

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

where $\mathcal{U}_0$ is the umbrella vector (3), C is a positive constant,

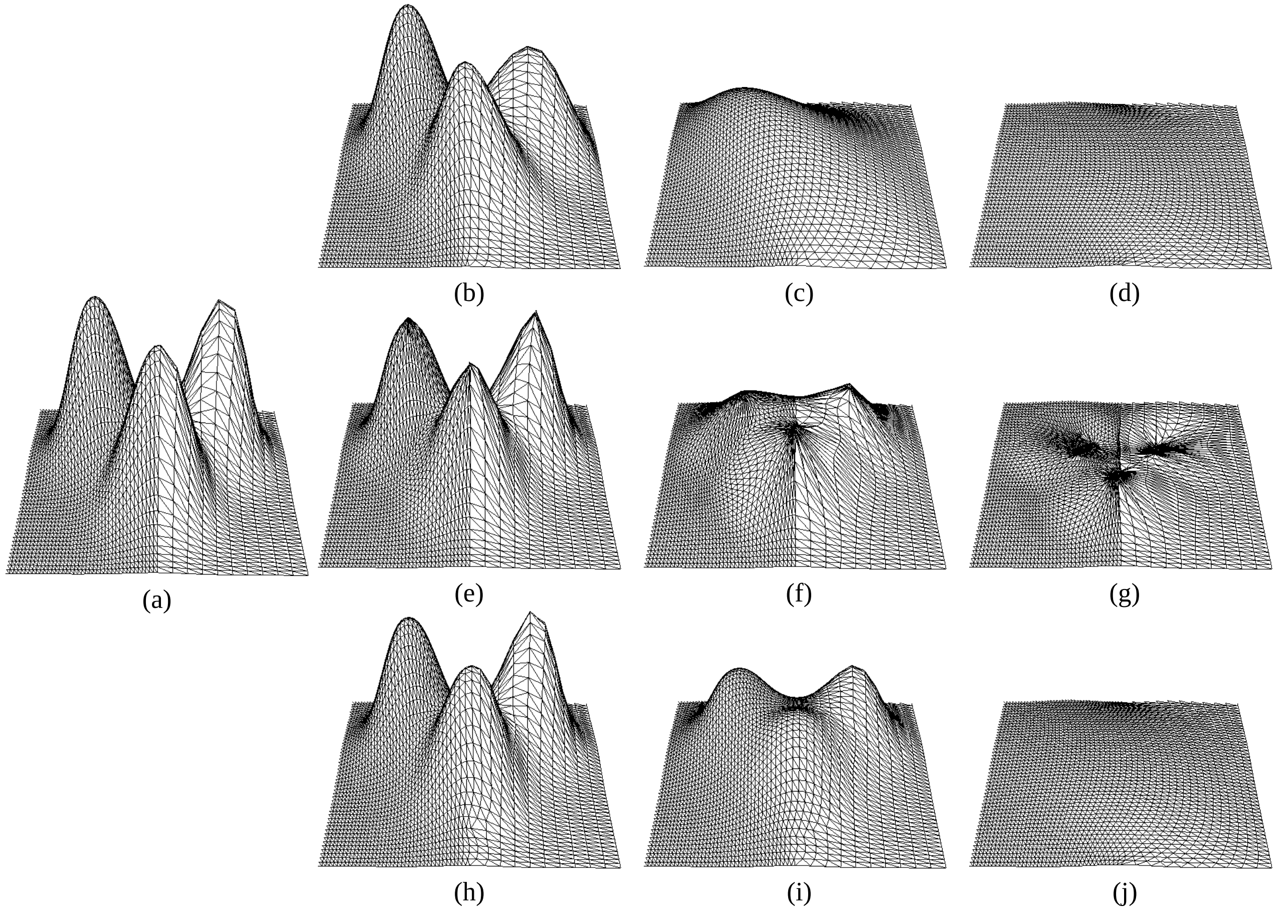

Figure 7. (a) A polyhedral surface consisting of two parts with different sampling rates. (b)–(d) Deformation of (a) into a flat patch by the Laplacian flow. (e)–(g) Deformation of (a) into a flat patch by the mean curvature flow. (h)–(j) Deformation of (a) into a flat patch by our method (12), (13).

and the dot stands for the scalar product. One can see that $\mathcal{U}_0 - (\mathcal{U}_0 \cdot \mathbf{n})\,\mathcal{U}_0$ is the projection of the umbrella vector onto the tangent plane. It leads to the local update rule

$$P_{\text{new}} \longleftarrow P_{\text{old}} + \lambda \Big\{ H(P_{\text{old}})\mathbf{n}(P_{\text{old}}) + \qquad (11)$$
$$+\, C\big[\mathcal{U}_0(P_{\text{old}}) - (\mathcal{U}_0(P_{\text{old}}) \cdot \mathbf{n}(P_{\text{old}}))\,\mathcal{U}_0(P_{\text{old}})\big] \Big\}$$

Smoothing by (11) produces shapes of the same quality as the mean curvature flow but with uniform distribution of vertices. See Fig. 4.

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

However, according to our experiments, a similar smoothing scheme produces better results.

Let $\mathbf{m} = \mathcal{U}_0/||\mathcal{U}_0||$ and θ is 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., $\theta > \pi/2$), see Fig. 8b, we use the following flow

$$P_{\text{new}} \longleftarrow P_{\text{old}} + \lambda\,\mathcal{F}(P_{\text{old}}), \qquad (12)$$

where

$$\mathcal{F} = \begin{cases} \dfrac{|H|\mathbf{m}}{\cos\theta} & \text{if} \quad \cos\theta > \varepsilon \\ 2\,H\,\mathbf{n} - \dfrac{|H|\,\mathbf{m}}{\cos\theta} & \text{if} \quad \cos\theta < -\varepsilon \\ 0 & \text{if} \quad |\cos\theta| \le \varepsilon \end{cases} \qquad (13)$$

Here ε is a small positive parameter. Geometric ideas behind (12) are explained in Fig. 8.

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

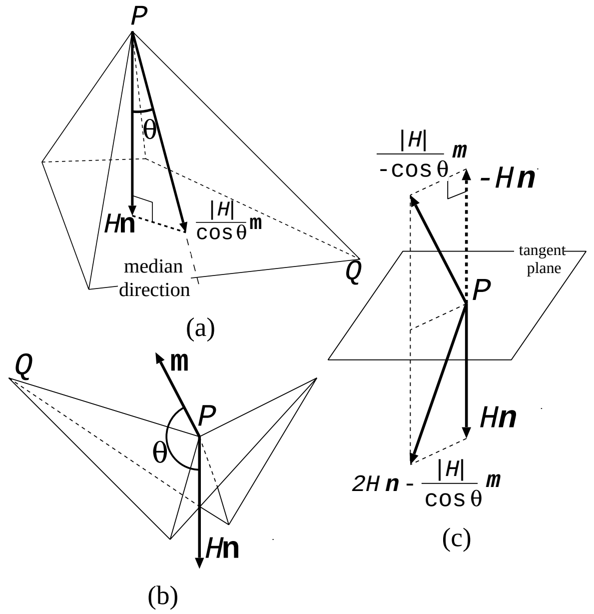

Figure 8. (a) The case when the median and normal vectors lie on the same side from the tangent plane: moving in the median direction with normal speed component equal to the mean curvature. (b) The case the median and normal vectors lie on the opposite sides from the tangent plane may happen for saddle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

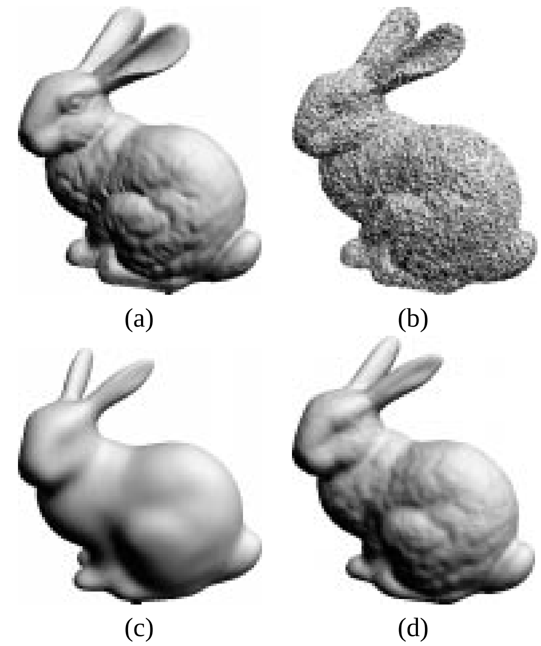

Figure 9. (a) The original Stanford bunny; (b) the bunny with a uniform noise added; (c) smoothing by (12) (Laplacian and mean curvature flows produce almost the same results); (d) smoothing by (14) with the same number of iterations as in (c).

According to our experiments, choosing $\varepsilon = 0.1$ produces good results independently of the mesh sampling rate.

Our smoothing scheme (12) demonstrates better results than the Laplacian flow, the mean curvature flow, and the Taubin smoothing scheme, see Fig. 1. See also Fig. 7 where a polyhedral surface consisting of two parts with different sampling rates is smoothed by the Laplacian flow, the mean curvature flow, and our scheme (12) and (13).

4 How to Avoid Oversmoothing

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

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

with

$$\widetilde{\mathcal{F}}(P) = \begin{cases} \left(\|\mathcal{F}(P)\| - T\right) \dfrac{\mathcal{F}(P)}{\|\mathcal{F}(P)\|} & \text{if } \|\mathcal{F}(P)\| > T \\ 0 & \text{if } \|\mathcal{F}(P)\| \le T \end{cases}$$

where T is a positive threshold selected 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 (12), (13) and (14) with $\mathcal{F}(P)$ defined in (13).

Of course the above simple modification can be applied to the any smoothing flow considered before.

In our experiments exposed in Fig. 9 the parameter T in (14) is the same for all vertices. It is natural to allow T 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

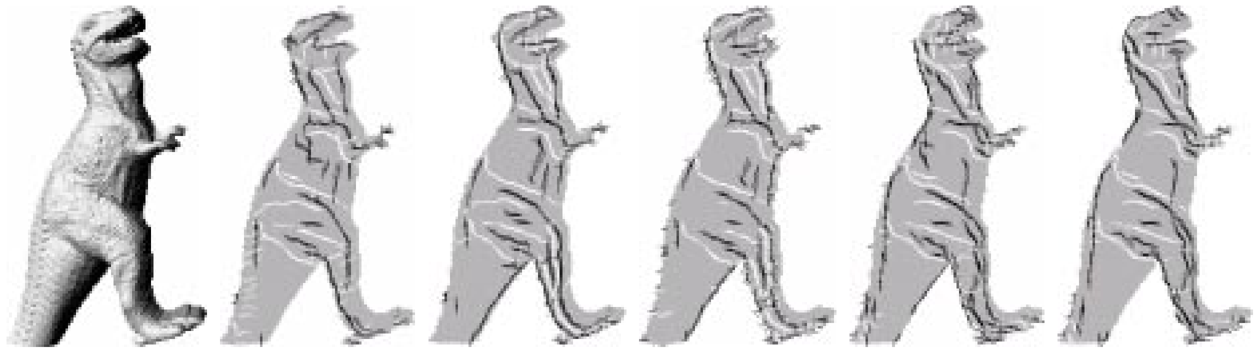
(a) (b) (c) (d) (e) (f)

Figure 11. (a) A Godzilla model. (b) - (f) The ridges (black) and ravines (white) detected on the Godzilla model smoothed by various smoothing schemes. (b) Smoothing by the Laplacian flow. (c) Smoothing by the Taubin method. (d) Smoothing by the bilaplacian flow. (e) Smoothing by the mean curvature flow. (f) Smoothing by our method (12), (13).

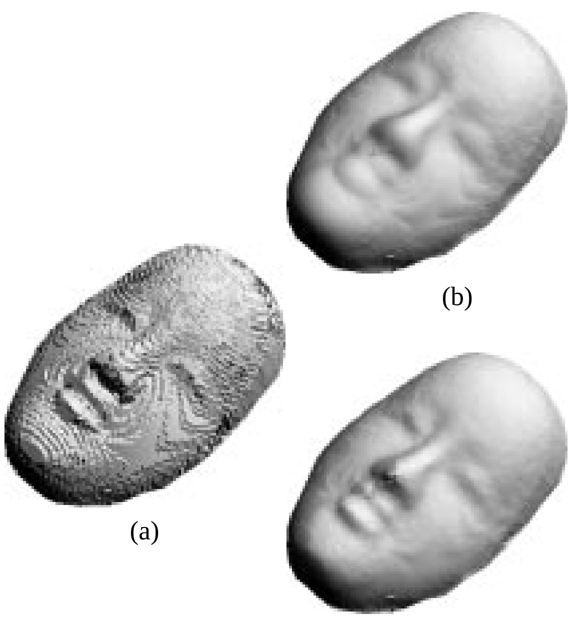
(a) (b) (c)

Figure 10. (a) A triangulated Noh mask model reconstructed from a scattered data generated by a laser scanner system; (b) smoothing according to (14) with constant T chosen manually; (c) smoothing according to (14) with $T(P)$ equal to the arithmetic mean of the mean curvatures of at the first and second rings of neighbors of P.

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 recomputing periodically the threshold $T(P)$ for each vertex P after a fixed number of smoothing iterations. Fig. 1f (the bottom row) shows advantages of such threshold recomputing scheme.

5 Applications

We apply our smoothing technique (12), (13) to detect ridges and ravines on a smooth surface approximated by a triangular mesh.

Let us define the *ridges* as the locus of points where the maximal principal curvature attains a positive maximum along its curvature line and the *ravines* as the locus of points where the minimal principal curvature attains a negative minimum along its curvature line [2].

Practical detection of the ridges and ravines on a surface involves estimation of high-order surface derivatives and, therefore, requires a careful surface smoothing before detection. It seems natural to choose a smoothing procedure minimizing vertices drift over the surface. On the other hand, a smoothing scheme increasing the mesh regularity improves curvature estimation.

For practical detection of the ridge and ravine vertices on the mesh, we first smooth the mesh and then estimate the normal, the principal curvatures, and the principal directions at the mesh vertices. We use the method proposed in [7]. To check whether a given vertex P is a ridge vertex, we find the intersection between the polygon composed the ring of the first neighbors of P and the normal plane generated by the normal vector and maximal principal direction at P. We estimate the maximal principal curvature at the intersection points by linear interpolation. Finally we compare the the values of the maximal principal curvature at P and the intersection points.

Some results of our experiments with various smoothing schemes are shown in Fig. 11 where the ridges are colored in black and ravines are colored in white. We think that our smoothing scheme produces one of the best results.

6 Conclusion

In this paper we propose simple and effective polyhedral surface smoothing schemes which combine the best properties of the Laplacian smoothing flow and discrete mean curvature flow, are good for highly irregular meshes, outperform existing methods, and reduce possible oversmoothing.

In future we are planning to extend ideas presented in the paper to various discrete implementations of the Laplacian and mean curvature flows. Also we want to examine how the developed methods can be enhanced by bilaplacian flows and surface diffusion processes. We also hope to make our smoothing methods faster and more accurate by using the implicit Crank-Nicolson method.

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

Acknowledgments

We thank the anonymous reviewers for their careful reading and helpful suggestions. Ilia Bogaevski is grateful to the University of Aizu where he spent one year as a visiting researcher in 1998/99 when this research was mainly conducted. Thanks to Shin Yoshizawa for the Godzilla 3D data set.

References

[1] A. G. Belyaev, E. V. Anoshkina, S. Yoshizawa, and M. Yano. Polygonal curve evolutions for planar shape modeling and analysis. *International Journal of Shape Modeling*, to appear.

[2] A. G. Belyaev and Y. Ohtake. An image processing approach to detection of ridges and ravines on polyhedral surfaces. *Submitted for publication*, 2000.

[3] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics (SIGGRAPH 99 Proceedings)*, pages 317–324, 1999.

[4] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Computer Graphics (SIGGRAPH 98 Proceedings)*, pages 105–114, 1998.

[5] R. Mohtar and L. Segerlind. Accuracy-based time step criteria for solving parabolic equations. In *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations, IMA Volume 75*, pages 153–163. Springer, 1995.

[6] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge Univ. Press, 1999.

[7] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. ICCV'95*, pages 852–857, 1985.

[8] G. Taubin. A signal processing approach to fair surface design. In *Computer Graphics (SIGGRAPH 95 Proceedings)*, pages 351–358, 1985.

[9] J. Vollmer, R. Mencl, and H. Muller. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum (Proc. Eurographics 1999)*, 18(3):131–138, 1999.