A comparison of mesh smoothing methods

Article · January 2003
Source: OAI

2 authors:

Alexander Belyaev
Heriot-Watt University

Yutaka Ohtake
The University of Tokyo

A Comparison of Mesh Smoothing Methods

Alexander Belyaev * Yutaka Ohtake
Computer Graphics Group, Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany
Phone: [+49](681)9325-408 Fax: [+49](681)9325-499
{belyaev,ohtake}@mpi-sb.mpg.de

Abstract

We consider several conventional mesh smoothing approaches such as the Laplacian and bilaplacian smoothing methods, mean curvature flow (implicit implementation), and the Taubin $\lambda|\mu$ scheme and give a quantitative evaluation of them. To compare two close meshes having the same connectivity we consider vertex-based and normal-based L^2 error metrics. We also give a visual comparison of the methods.

We also compare the above mesh smoothing schemes with our method which is based on a linear diffusion of mesh normals. The method demonstrates the best performance in denoising meshes with sharp features and high resistance to oversmoothing.

Keywords: triangle mesh, mesh smoothing, error metrics

Conference topic: shape recovery

1 Introduction

Computer graphics objects reconstructed from real-world data usually contain undesirable noise. An important problem consists of developing robust methods for removing noise with minimal damage caused to geometric features of the object.

In this paper, we deal with computer graphics object represented by dense triangle meshes and give a quantitative evaluation of such conventional mesh smoothing procedures as the Laplacian filtering [12, 6], bilaplacian smoothing flow [6], the mean curvature flow [3], and the Taubin $\lambda|\mu$ approach. We consider two L^2 error metrics measuring deviations of the vertices and normals of two close meshes and use the metrics to compare the smoothing methods. Our normal-based L^2 error metric mimics L^2 norm for function gradients. Thus the sum of the error metrics imitates H^1 norm used widely in the theory of partial differential equations.

We also introduce an iterative mesh smoothing procedure each iteration of which consists of two steps: mean filtering on mesh normals and updating mesh vertex positions according to the modified normals.

Since mesh smoothing is one of the most important mesh processing operations, many interesting mesh smoothing techniques were invented. However, in this paper, we compare our iterative mean filtering scheme with the four basic methods because, in our opinion, they provide a user with the best combinations of simplicity, efficiency, and speed and, as a consequence, are widely used in various geometric modeling applications. So for our comparison we have chosen the following methods: Laplacian smoothing, the Taubin $\lambda|\mu$ smoothing scheme [12], the bilaplacian flow [6], and the mean curvature flow [3]. We consider three variations of the Taubin method: with equal weights [12], inverse distance weights [12], and cotangent weights [3]. For each test, we choose the weights producing the best result.

The paper is organized as follows. In Section 2, we introduce our iterative mean filter smoothing a mesh via weighted averaging the mesh normals. Vertex-based and normal-based L^2 error metrics are described in Section 3. We compare the considered smoothing methods in Section 4 and conclude in Section 5.

2 Iterative Mean Filtering on Mesh Normals

The mesh smoothing method described in this section explores a general "processing mesh normals" scheme developed in [1, 8, 9] (see also [4, 5, 13, 11] for similar ideas). One iteration of the scheme consists of the following three steps:

1. given a triangle mesh, consider the triangle normals as a vector-valued image defined over the mesh;
2. adapt image processing tools for denoising the mesh normals and produce smoothed normals;
3. reconstruct a smoothed mesh from the smoothed normals.

Below we consider simple mean filtering on mesh normals. More sophisticated filtering schemes are studied in [9, 14].

* On a leave from University of Aizu, Japan.

In order to fit the mesh to the set of modified normals we use a method developed in [9].

Consider an oriented triangle mesh. Let T be a mesh triangle, $\boldsymbol{n}(T)$ be the unit normal of T, and $A(T)$ be the area of T. Denote by $\mathcal{N}(T)$ the set of all mesh triangles that have a common edge or vertex with T. One iteration of the iterative mesh mean filtering scheme consists of the following two successive steps.

Step 1: Averaging Normals. For each mesh triangle T, let us compute the triangle normal $\boldsymbol{n}(T)$ and apply the following weighted averaging procedure to the field of normals:

$$\boldsymbol{m}(T) = \sum_{S \in \mathcal{N}(T)} A(S)\boldsymbol{n}(S). \tag{1}$$

Then we normalize the field of averaged normals $\boldsymbol{m}(T)$:

$$\boldsymbol{m}(T) \longleftarrow \frac{\boldsymbol{m}(T)}{\|\boldsymbol{m}(T)\|}.$$

Thus every mesh triangle T is equipped with the modified (averaged) unit normal $\boldsymbol{m}(T)$.

Step 2: Fitting Mesh to Modified Normals. We modify the mesh vertex positions in order to fit the mesh to the set of modified normals $\{\boldsymbol{m}(T)\}$.

Let us introduce an error function measuring how good a modified mesh fits the field of modified normals $\{\boldsymbol{m}(T)\}$. For each mesh vertex P, let us define

$$E_{\text{fit}}(P) = \sum A(T)\big|\boldsymbol{n}(T) - \boldsymbol{m}(T)\big|^2, \tag{2}$$

where the sum is taken over all triangles adjacent to P. Now a new position for P is found by minimizing $E_{\text{fit}}(P)$:

$$P_{\text{new}} = \operatorname*{argmin}_P E_{\text{fit}}(P) \tag{3}$$

We use a simple conjugate gradient method [10] to approach a minimum of $E_{\text{fit}}(P)$. Since

$$A(T)\big|\boldsymbol{n}(T) - \boldsymbol{m}(T)\big|^2 = 2A(T)(1 - \boldsymbol{n}\cdot\boldsymbol{m}),$$

deriving the gradient $\nabla_P E_{\text{fit}}(P)$ is reduced to computing the gradient of the area $A(T)$ and its projection onto the plane orthogonal to $\boldsymbol{m}$. A formula for the area gradient can be found in [3] (see also [7] for a simple derivation of the formula).

Mesh smoothing. Now the complete smoothing procedure consists of applying $\mathbf{Step\,1} + \mathbf{Step\,2}$ a sufficient number of times. It turns out that the mesh evolution process $(\mathbf{Step\,1} + \mathbf{Step\,2})^n$ converges quickly as $n \to \infty$ and in practice 30-50 iterations are enough to achieve a steady-state.

3 Vertex-based and Normal-Based L^2 Error Metrics

In order to give a qualitative comparison of the described mesh smoothing schemes we introduce two error metrics. The first metric resembles an L^2 version of the *Metro* mean distance [2] while the second one measures deviations between the corresponding normals of two given surfaces. As demonstrated in Fig. 1, minimizing deviations between the corresponding normals of two meshes / surfaces improves drastically an approximation quality.

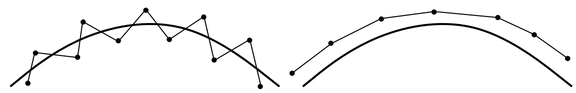

Figure 1. A curve approximated by two polygonal lines. Left: fitting the vertex positions only does not assure a good approximation. Right: fitting the vertex positions and normals simultaneously improves the approximation drastically.

Consider a reference mesh $\mathcal{M}$ (we assume that $\mathcal{M}$ is dense enough) and a mesh $\mathcal{M}'$ obtained from $\mathcal{M}$ by adding noise and applying several iterations of a smoothing process. Consider a vertex P' of the smoothed mesh $\mathcal{M}'$. Let us set $\mathrm{dist}(P', \mathcal{M})$ equal to the distance between P' and a triangle of the reference mesh $\mathcal{M}$ which is closest to P'. Our L^2 vertex-based mesh-to-mesh error metric is then given by

$$E_v = \sqrt{\frac{1}{3A(\mathcal{M}')} \sum_{P' \in \mathcal{M}'} A(P')\,\mathrm{dist}(P', \mathcal{M})^2} \tag{4}$$

where $A(P')$ is the sum of areas of all triangles of $\mathcal{M}'$ incident with P' and $A(M')$ is the total area of $\mathcal{M}'$.

Our L^2 normal-based mesh-to-mesh error metric measuring deviations between the corresponding normals of two meshes $\mathcal{M}$ and $\mathcal{M}'$ is defined in a similar way. Consider a triangle T' of the mesh $\mathcal{M}'$ and let us find a triangle T of $\mathcal{M}$ closest to T'. Let $\boldsymbol{n}(T)$ and $\boldsymbol{n}(T')$ be the orientation unit normals of T and T', respectively. The metric is defined by

$$E_n = \sqrt{\frac{1}{A(\mathcal{M}')} \sum_{T' \in \mathcal{M}'} A(T')|\boldsymbol{n}(T) - \boldsymbol{n}(T')|^2}, \tag{5}$$

where $A(T')$ denotes the area of T'.

Note that E_n is scale-independent.

The error metric E_n is sensitive to mesh degradation at sharp features and highly curved regions. Fig. 2 illustrates this property of E_n and gives a visual comparison of E_n with E_v.

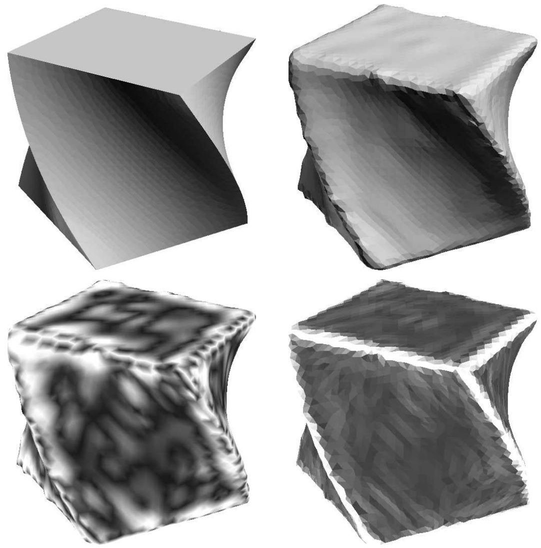

Figure 2. Top left: A mesh with sharp features. Top right: the same mesh after adding noise and smoothing (the Taubin $\lambda|\mu$ method with equal weights was used). Bottom left: the smoothed mesh is colored according to E_v-distance from the original mesh (dark-grey corresponds to regions with small error and light-grey indicates large-error regions). Bottom right: the smoothed mesh is colored according to E_n-distance from the original mesh, notice how well it penalizes the error at the sharp features of the original mesh.

4 Comparison Results

In this section we compare the Laplacian, bilaplacian, mean curvature flows, the Taubin smoothing approach, and our iterative mean filtering scheme using the introduced error metrics E_v and E_n.

Comparing the meshes $\mathcal{M}$ and $\mathcal{M}'$ we do not optimize their orientations and centroid positions. According to our experiments, when mesh smoothing is applied, the mesh rotation and centroid shifting effects are negligible to compare with shape deformations caused by smoothing.

Let a mesh $\mathcal{M}_t$ be obtained from a reference (ideal) mesh $\mathcal{M}$ by adding noise and applying t iterations of a smoothing process. We plot and analyze the graphs of $E_v(\mathcal{M}, \mathcal{M}_t)$ and $E_n(\mathcal{M}, \mathcal{M}_t)$ as functions of the iteration number t. We use the following marks to plot the graphs E_v and E_n for the considered mesh smoothing methods:

- $\circ$ Laplacian smoothing flow;
- $\times$ implicit mean curvature flow;
- $\square$ Taubin $\lambda|\mu$ method;
- Δ bilaplacian smoothing flow;
- $+$ mean filtering on mesh normals.

We start out comparison by testing the smoothing methods on a simple geometric object, a sphere, see Fig. 3.

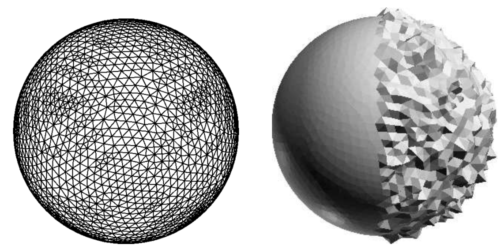

Figure 3. Left: a polygonal sphere. Right: a sphere is partially corrupted by random noise.

For the sphere, we use the Taubin $\lambda|\mu$ scheme with equal weights.

According to Fig. 4, the mean curvature flow demonstrates the best performance since a closed surface evolved by the mean curvature flow often converges to a sphere.

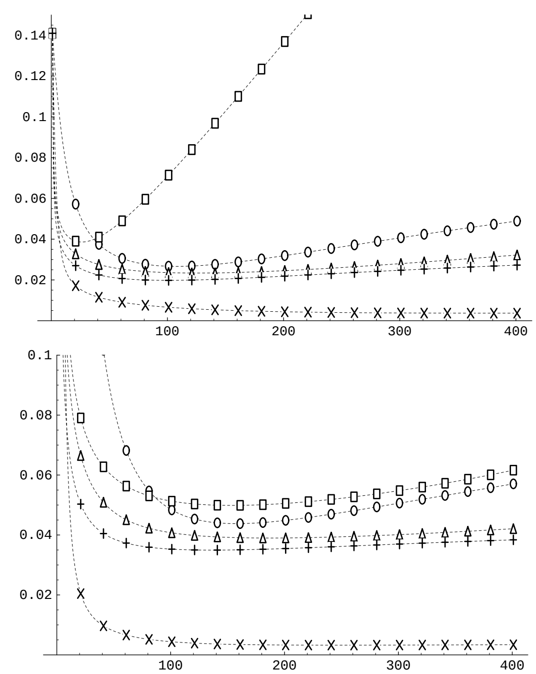

Figure 4. Graphs of E_v (top) and E_n (bottom) for smoothing the noisy sphere from right image of Fig. 3.

Note that each smoothing method considered in the paper has its own speed which in its turn depends on the step-size parameter of the method. To achieve a fair comparison of the methods we use the following strategy. Given a reference mesh $\mathcal{M}$ and a mesh obtained from $\mathcal{M}$ by adding noise and applying several smoothing iterations, for each method we define the optimal number of iterations τ is defined by

$$\tau = \frac{1}{2}\Big(\operatorname*{argmin}_t E_v(M, M_t) + \operatorname*{argmin}_t E_n(M, M_t)\Big), \quad (6)$$

where M_t is the noisy mesh smoothed by t iterations of the method. We use τ as the x-axis unit (its own for each considered smoothing method) to measure the number of iterations.

For the next tests we use the geometric models shown in the left images of Fig. 5 and Fig. 6.

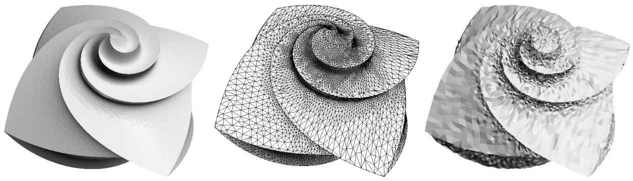

Figure 5. Left: a complex geometric model having sharp edges and represented by an uneven mesh. Middle: the wireframe image of the model. Right: random noise is added.

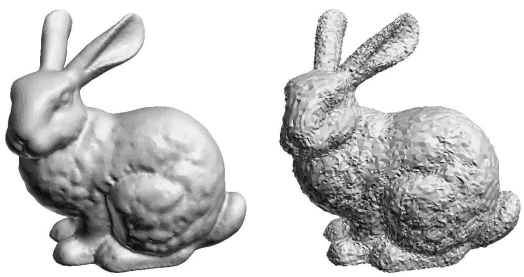

Figure 6. Left: a remeshed Stanford bunny model (highly irregular meshing is used). Right: random noise is added.

For the noisy model shown in Fig. 5, Fig. 7 presents the graphs of the vertex-based and normal based error metrics as functions of $k\tau$, $k = 0, 1, 2, \ldots$, where τ is defined by (6). According to the graphs, the bilaplacian flow, Taubin $\lambda|\mu$ method (we use it with the inverse distance weights), and our approach demonstrate the best performance. The top row of Fig. 8 shows smoothing results after τ iterations. The bottom row of Fig. 8 delivers a visual comparison of the methods according to the resistance to oversmoothing, 5τ iterations are used. Notice that our simple averaging scheme produces the best visual result.

Fig. 9 presents similar results for the noisy bunny model. The Taubin $\lambda|\mu$ scheme demonstrates the best performance.

5 Conclusion

In this paper, we have proposed a procedure for qualitative comparison of mesh smoothing methods. We have also introduced iterative mean filtering on mesh normals.

The Laplacian smoothing flow and implicit integration method for the mean curvature flow [3] are the fastest mesh smoothing if their time step-size parameters are chosen properly. Although both the Taubin $\lambda|\mu$ method and bilaplacian smoothing flow are slower, they do a substantially better job in noise removing while preserving salient shape features. In addition, the Taubin method does not require a user to choose a time step-size parameter. Mean filtering on mesh normals is the slowest method. However, according to our experiments, it outperforms the other considered methods in denoising shapes with sharp features.

An interesting direction for future research consists of developing error metrics corresponding to human visual perception of 3D shapes. An automatic selection of an optimal stopping time for a mesh smoothing process constitutes another promising theme for future work.

Acknowledgments. We are grateful to Hiromasa Suzuki for drawing our attention to the problem of a quantitative comparison of mesh smoothing methods.

References

[1] A. G. Belyaev, Yu. Ohtake, and K. Abe. Detection of ridges and ravines on range images and triangular meshes. In *Vision Geometry IX, Proc. SPIE 4117*, pages 146–154, San Diego, July-August 2000.

[2] P. Cignoni, C. C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

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

[4] S. Karbacher and G. Häusler. New approach for modeling and smoothing of scattered 3D data. In *Three-Dimensional Image Capture and Applications, Proc. SPIE 3313*, pages 115–125, San Jose, California, January 1998.

[5] S. Karbacher, X. Laboureux, and G. Häusler. Curvature weighted smoothing of triangle meshes. In *Lehrstuhl für Optik, Annual Report 2000*, Physikalisches Institut, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2001.

[6] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH 98*, pages 105–114, 1998.

[7] Yu. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Polyhedral surface smoothing with modified Laplacian and curvature flows. *The Journal of Three Dimensional Images*, 13(3):19–24, 1999.

[8] Yu. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Mesh regularization and adaptive smoothing. *Computer-Aided Design*, 33(4):789–800, 2001.

[9] Yu. Ohtake, A. G. Belyaev, and H.-P. Seidel. Mesh smoothing by adaptive and anisotropic Gaussian filter. In *Vision, Modeling, and Visualization 2002*, pages 203–210, Erlangen, Germany, November 2002.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C: The Art of Scientific Computing*. Cambridge University Press, 1993.

[11] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of IEEE Visualization 2002*, pages 125–132, November 2002.

[12] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of ACM SIGGRAPH 95*, pages 351–358, 1995.

[13] G. Taubin. Linear anisotropic mesh filtering. IBM Research Report RC22213 (W0110-051), IBM, October 2001.

[14] H. Yagou, Yu. Ohtake, and A. G. Belyaev. Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. Submitted for publication.

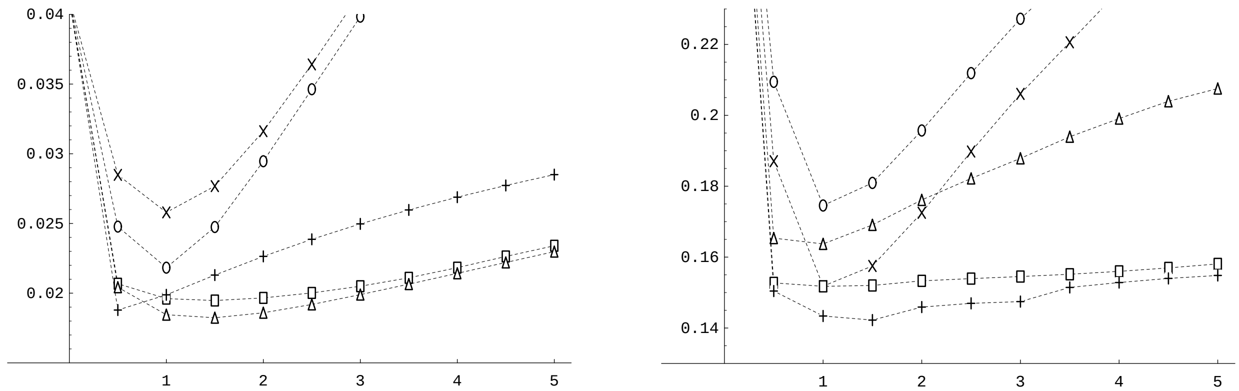

Figure 7. Graphs of E_v (left) and E_n (right) for smoothing the model shown in the bottom left image of Fig. 5.

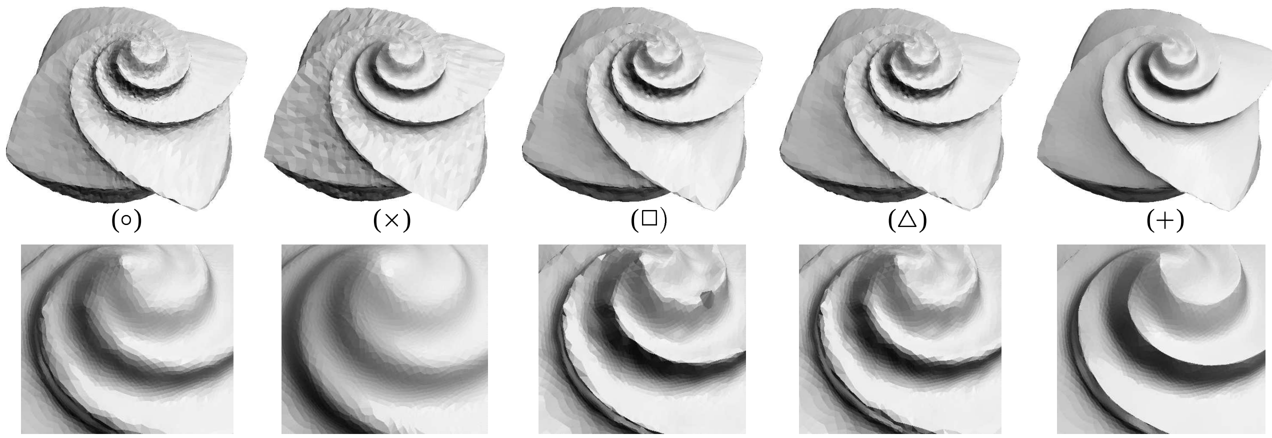

Figure 8. Top: optimal number of iterations τ (its own for each method) is used for ($\circ$) Laplacian smoothing flow, ($\times$) mean curvature flow, ($\square$) Taubin scheme with inverse distance weights, ($\triangle$) bilaplacian flow, ($+$) our method. Bottom: 5τ iterations are selected for each method.

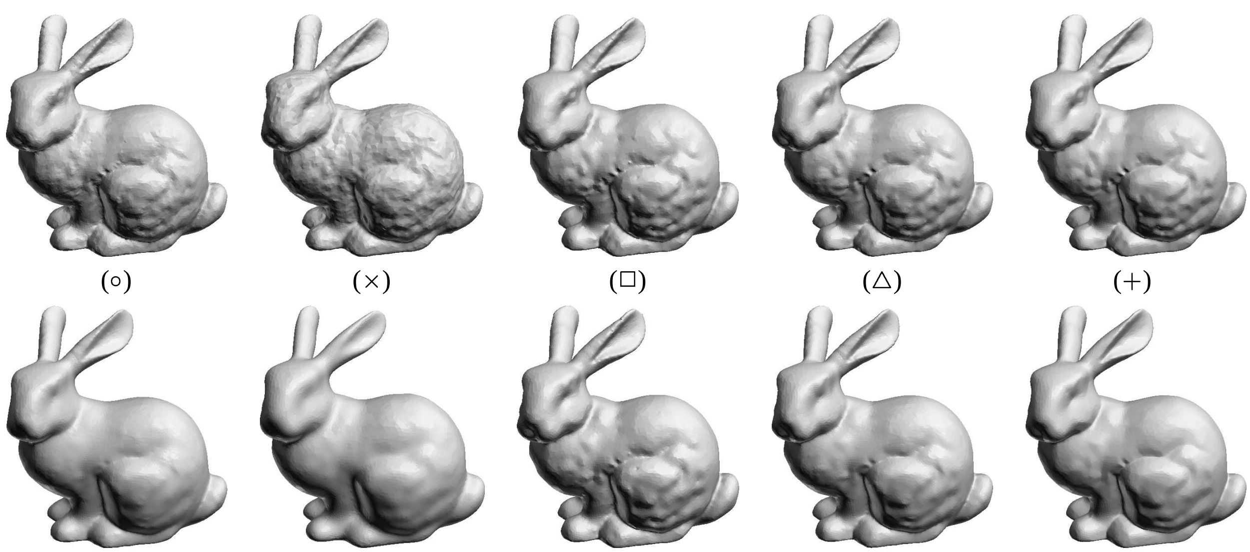

Figure 9. Top: optimal number of iterations τ is chosen for ($\circ$) Laplacian smoothing flow, ($\times$) mean curvature flow, ($\square$) Taubin scheme with cotangent weights, ($\triangle$) bilaplacian flow, ($+$) our method. Bottom: 5τ iterations are applied.