Reconstruction of Shapes Based on Normals Analysis

Irina Semenova [1], Vladimir Savchenko [2], Ichiro Hagiwara [3]

[1] Dept. of Mech. Sciences and Eng., Tokyo Institute of Technology, Japan,
semenova@stu.mech.titech.ac.jp

[2] Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan,
vsavchen@k.hosei.ac.jp

[3] Dept. of Mech. Sciences and Eng., Tokyo Institute of Technology, Japan,
hagiwara@mech.titech.ac.jp

Abstract

Most mesh processing filters (including remeshing, simplification, and subdivision) affect vertices of the mesh. Vertices coordinates are modified, new vertices are added and some original ones are removed, with the result that the shape of the original surface is changed. While a great deal of research is concentrated on preservation of surface shape during some mesh processing, there is no general tool that can be used for surface reconstruction at post processing stage. To the best of our knowledge, this paper is the first one to present a restoring algorithm that allows to "repair" output of various mesh processing filters. The proposed scheme is straightforward way to put "off-surface" vertices of the deformed mesh back to the original smooth shape. It does not require any surface parameterization and is based on normal analysis. The procedure is demonstrated by using it as post processing tool after applying local node movement and simplification algorithms. However, the technique is versatile enough to be used in a large variety of mesh optimization algorithms including remeshing and subdivision schemes.

Keywords: *Mesh Processing, Shape Reconstruction, Normal Analysis.*

1. INTRODUCTION

Polygonal meshes provide simple and effective representation of complex geometric models.

Unfortunately, most surface meshes can hardly be called satisfactory in terms of their size, element shapes and vertex sampling that makes them unsuitable for engineering and computer graphics applications. Depending on the problem and application, optimization of a given mesh may imply simplification, remeshing, subdivision or smoothing (by smoothing we do not understand denoising scheme but node movement technique for improving geometric mesh quality that does not change mesh connectivity).

Any mesh processing filter affects vertices of the mesh, with the result that the shape of the original model is changed. The important problem is how to reconstruct the original shape without loosing benefits obtained with the applied filter. This means that the restoration procedure should not change the number of mesh elements and mesh connectivity and should not cause damage to mesh element quality.

In this paper, we describe a new approach to solve this problem if the original mesh approximates a smooth C^1 continuous surface. This means that the mesh should be dense enough to represent C^1-shape correctly (variation of curvatures between adjacent vertices should not be considerable).

1.1 Related works

To the best of our knowledge, there is no general method that can be used as a restoring technique after applying various mesh processing filters. Most research addresses this problem only for particular algorithms. For instance, in several works devoted to subdivision (of 3D surfaces [1-2] or 2D curves [3]) there have been proposed interesting schemes for generating smooth curves and surfaces with minimum curvature variations. The newly introduced vertices are moved in order to interpolate the smooth shape. A great deal of research has focused on preserving shape of the original model for node movement techniques improving geometric mesh quality [4-6]. Recently, several remeshing algorithms allowing to keep the vertices on the approximated smooth surface have been discussed in [7-10].

However, as is mentioned, we have not found in the literature a general approach that can be used for surface reconstruction at post processing stage if applied filter caused some damage to the original shape.

1.2 Contribution

In this paper, we present a novel technique allowing to put "off-surface" points of the deformed mesh back to the original surface. By original surface we do not mean the original mesh but smooth shape approximated by the mesh.

The basic idea of the proposed scheme is as follows.

At each vertex of the deformed mesh we find an "optimal" direction defined by the normals of the nearest triangle in the original mesh. We then seek for the new position of the considered vertex along this direction. The distance of movement is computed from the intersection of the "gliding" normals of the closest triangle and the direction of movement.

The idea of moving points along some directions (usually normals directions) to fit the smooth surface is not new. It has been successfully used for many problems including subdivision [1-2], displaced subdivision surfaces [11], and normal meshes [12].

One of our contributions is to adjust this idea to the problem of repairing the deformed meshes. Most approaches allow to move points to the approximated surface from the triangles of the original mesh (or control mesh). But what should we do if "off-surface" point of the deformed mesh does not belong to the original mesh? We propose solution to this problem. Also, instead of fitting triangles of the original mesh with a smooth function and trying to project the vertices of the deformed mesh onto this function, we combain these two steps into a single simple procedure based on normal analysis. The method does not require any parametric representation of the surface that allows to save computational time without sacrificing any quality in results. Moreover, we will demonstrate that the proposed technique does not cause considerable damage to the mesh element quality. Thus the procedure may be successfully used as post-processing for smoothing and remeshing filters.

Remarks (what we mean by "reconstruction") Let us note that we do not state the problem to guarantee geometric fidelity of the deformed mesh to the original one. We assume that the deformed shape is geometrically faithful representation of the initial surface. If there are no vertices in the mesh, which can represent some features of the original model, these features cannot be reconstructed with our technique since the algorithm does not offer any mechanism to insert additional points. Fig. 1(a) illustrates 2D case. Our objective is just to move the vertices of the deformed mesh as close as possible to the original smooth surface. However, the proposed technique is powerful tool allowing to make precise reconstruction of the original shape if the resulting mesh contains enough points (see for illustration of 2D case Fig. 1(b)).

Figure 1: (a) Our restoring technique cannot reconstruct salient features (shown by red color) of the smooth curve since the polyline does not contain enough points that can be put to the curve; (b) Polyline curve (shown by dotted line) represents the shape of the smooth curve geometrically correctly. However, the accuracy of approximation can be improved by moving vertices of the polyline to the curve with our restoring technique.

2. PRELIMINARIES

The input to our scheme is two oriented triangular meshes M and M' of arbitrary genus. First mesh is a piecewise linear approximation of some smooth surface S, which is C^1-continuous. M' is the mesh obtained from M with some mesh processing algorithm (smoothing, remeshing, simplification or subdivision scheme). Further we will refer M as an ***original*** mesh and M' as a ***resulting*** mesh.

The vertices of the resulting mesh M' are ***"off-surface" points*** if they do not belong to the surface S approximated by the original mesh M.

Normal gliding is movement of normals of the original mesh along circular arcs defined by the vertices of the original and resulting meshes.

Our objective is to move "off-surface" points of the resulting mesh to the original smooth surface S.

2.1 Normal estimation

To apply our scheme, we first need to estimate normals to the smooth surface approximated by the original mesh. Strictly such normals should be computed from a continuous analytic surface representation. However, it is computationally inefficiently. Therefore usually, the normal at a mesh node is computed by averaging the normals of the incident triangles with some weights. Since according to analysis in [13] "weighting-by-inverse-areas" scheme asymptotically better than other popular "one-ring-neighborhood" methods, we use this scheme to compute the normals at the vertices of the original mesh.

3. RESTORING TECHNIQUE

As is mentioned, the core of our technique is finding for each vertex of the resulting mesh the "optimal" direction and moving the vertex along this direction. To define the distance of movement, we use the "gliding" normals of the closest triangle in the original mesh along circular arcs towards the "optimal" direction.

The scheme is based on the following fact. If the sampling density of the original mesh is sufficiently high to allow the variations of surface curvatures between adjacent vertices to be neglected the underlying smooth surface can be locally approximated by circular arcs (Fig. 2). The same idea has been successfully applied to denoising and subdivision of triangular meshes in [1-2,14].

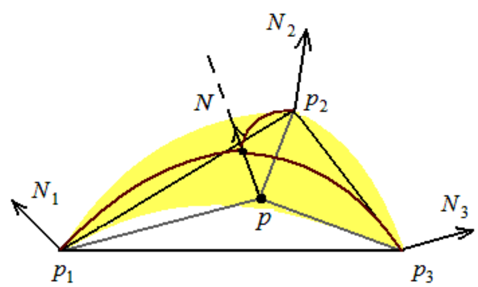

Figure 2: Interpolation of a smooth surface over the flat triangle by blending circular arcs

3.1 "Optimal" direction

Let us consider some vertex $p_i' \in M'$ in the resulting mesh and closest triangle in the original mesh $\Delta T_i = \Delta p_{i1} p_{i2} p_{i3} \in M$. We will seek a new position of p_i' along the "optimal" direction at p_i'. To simplify the procedure, let the "optimal" direction be defined by the normal unit vector N_i^{opt} at some point p_{0i} inside the triangle ΔT_i.

N_i^{opt} should satisfy the following conditions:

1) N_i^{opt} should be a linear interpolation of the normals at the vertices of the triangle ΔT_i. Namely, $N_i^{opt} = \sum_{j=1}^{3} b_j N_j \Big/ \left\| \sum_{j=1}^{3} b_j N_j \right\|$, where b_j, $j = 1,2,3$, are

barycentric coordinates of the origin $p_{i0} \in \Delta T_i$ of the vector N_i^{opt} ; N_j are the normals at the vertices $p_{ij} \in \Delta T_i$, $j = 1,2,3$.

2.) N_i^{opt} should define a line l_i^{opt} ("optimal" direction) passing through the considered vertex p_i' . See for illustration Fig. 3.

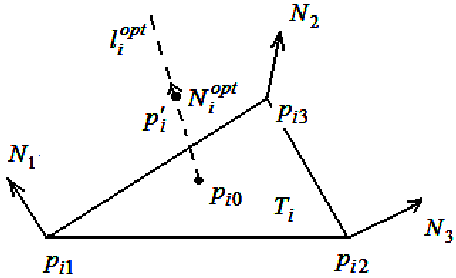

Figure 3: Finding the optimal direction N_i^{opt} at the vertex p_i' .

These conditions define a system of three quadratic equations with unknown coordinates of the point $p_{i0} \in \Delta T_i$. The system can be solved numerically with Newton method. However, it is easy to get exact solution reducing the system to one cubic equation.

3.2 Algorithm

We define the new position of the vertex $p_i' \in M'$ by the following formula:

$$p_i^{new} = \frac{1}{3}\sum_{j=1}^{3} p_{ij}^{new} = p_i' + \frac{1}{3}\sum_{j=1}^{3} \delta_j \cdot \overrightarrow{N_i^{opt}} , \qquad (1)$$

where δ_j , $j = 1,2,3$, are the values of displacement obtained by the "gliding" normals at the vertices of the triangle ΔT_i .

Let l_j , $j = 1,2,3$, l_i^{opt} be the lines passing through the vectors N_j , $j = 1,2,3$, N_i^{opt} respectively. Let N_j glide along a circular arc defined by l_j and l_i^{opt} that originates from p_{ij} . Then the value of displacement of the point p_i' along N_i^{opt} is defined as:

$$\delta_j = \left\| \overrightarrow{p_i' p_{ij}} \right\| \cdot \left(\cos\beta_j + \sqrt{\frac{\left(1-\cos^2\beta_j\right)\left(1-\cos\alpha_j\right)}{1+\cos\alpha_j}} \right), \qquad (2)$$

where $\cos\alpha_j = \left(\overrightarrow{N_i^{opt}}, \overrightarrow{N_j} \right)$, $\cos\beta_j = \dfrac{\left(\overrightarrow{N_i^{opt}}, \overrightarrow{p_i' p_{ij}} \right)}{\left\| \overrightarrow{p_i' p_{ij}} \right\|}$,

$j = 1,2,3$ (Fig. 4(a)).

Two cases (max and min-algorithms)depending on the directions of N_j and N_i^{opt} must be considered. First case corresponds to the local surface maximum at p_{ij}^{new} in the direction $\overrightarrow{p_{ij}^{new} p_{ij}}$ (see Fig. 4 for illustration). To process second case (local minimum at p_{ij}^{new} in the direction $\overrightarrow{p_{ij}^{new} p_{ij}}$) we need to replace in formulas (1-2) $\overrightarrow{N_j}$, $\overrightarrow{N_i^{opt}}$ to $-\overrightarrow{N_j}$, $-\overrightarrow{N_i^{opt}}$ respectively.

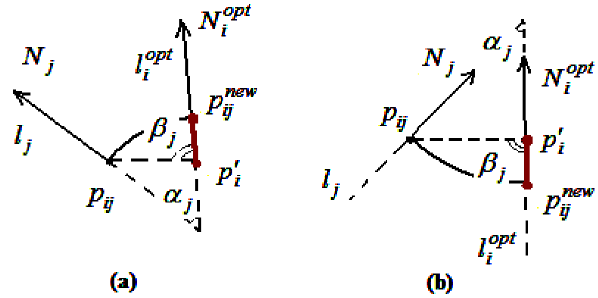

Figure 4: (a) Finding the new position p_{ij}^{new} for the vertex p_i' with max-algorithm; the distance of movement for the vertex p_i' is marked by red color; (b) Finding the new position p_{ij}^{new} for the vertex p_i' with min-algorithm; the distance of movement for the vertex p_i' is marked by red color.

3.3 Implementation notes

The restoring algorithm we described so far demonstrates perfect results for uniform meshes with the elements close to equilateral ones. To improve the results for non-uniform meshes with highly distorted elements we make the following changes in the scheme. We average obtained displacements δ_j , $j = 1,2,3$, with a coefficient $k = 1/6$ (that is $p_i^{new} = p_i' + 1/6 \left(\sum_{j=1}^{3} \delta_j \right) \cdot N_i^{opt}$).

After that the procedure is repeated several times with the updated position of the considered vertex p_i' and the vector N_i^{opt} . Our experiments show that 5 iterations is always enough to achieve good results.

In the case of highly nonuniform meshes there can be also some difficulties in defining whether max or min-algorithms should be implemented. One possible way to make the scheme more robust is to estimate the curvatures at the vertices of the original mesh M . Let p_i' be a vertex in the resulting mesh M' whose new position is to be found. Consider the vertex $p_{ij} \in \Delta T_i \in M$, where ΔT_i is the closest triangle to p_i' . If p_{ij} is a local surface maximum (that is maximum ($\lambda_{\max}$) and minimum ($\lambda_{\min}$) curvatures at p_{ij} are greater than 0) max-algorithm should be used. In the case of a local minimum ($\lambda_{\max} < 0$, $\lambda_{\min} < 0$) min-algorithm is implemented. If we deal with a saddle point ($\lambda_{\max} \cdot \lambda_{\min} < 0$) the choice of the algorithm depends on the sign of the normal curvature (λ_n) in the direction corresponding to the vector $\overrightarrow{p_{ij} p_i'}$. If $\lambda_n > 0$ we use max-algorithm; otherwise min-algorithm is applied.

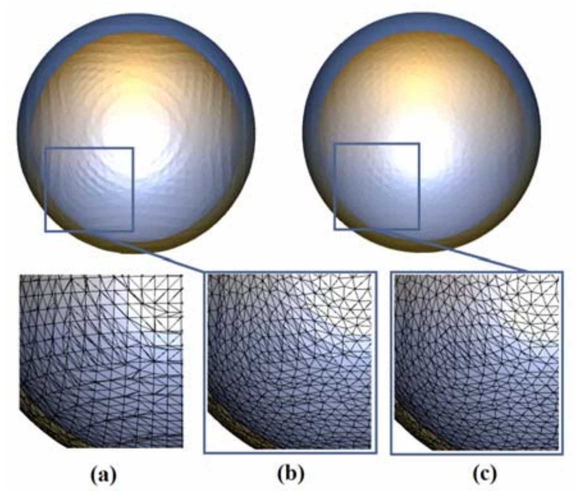

Figure 5: (a) Fragment of the original mesh of the sphere; (b) Model processed with Laplacian smoothing and fragment of the corresponding mesh; (c) The model (b) optimized with our restoring technique and fragment of the corresponding mesh.

Table 1: Deviation of the meshes processed with various techniques from the original analytical surfaces.

	$E_{\max}$ sphere	E_{aver} sphere	$E_{\max}$ cub. surf.	E_{aver} cub. surf.
M'	0.00239	0.0014	0.007234	0.001524
M^{opt}	0.00017	0.00002	0.000842	0.000163
M^{b}	0.00216	0.0019	0.007591	0.002753
M^{e}	0.00194	0.0013	0.007153	0.001438
$M^{\cot}$	0.00192	0.0012	0.007154	0.001527

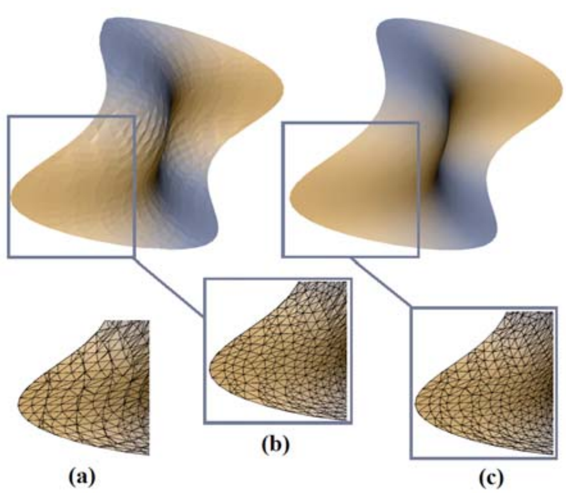

Figure 6: (a) Fragment of the original mesh of the cubic surface; (b) Model processed with Laplacian smoothing and fragment of the corresponding mesh; (c) The model (b) optimized with our restoring technique and fragment of the corresponding mesh.

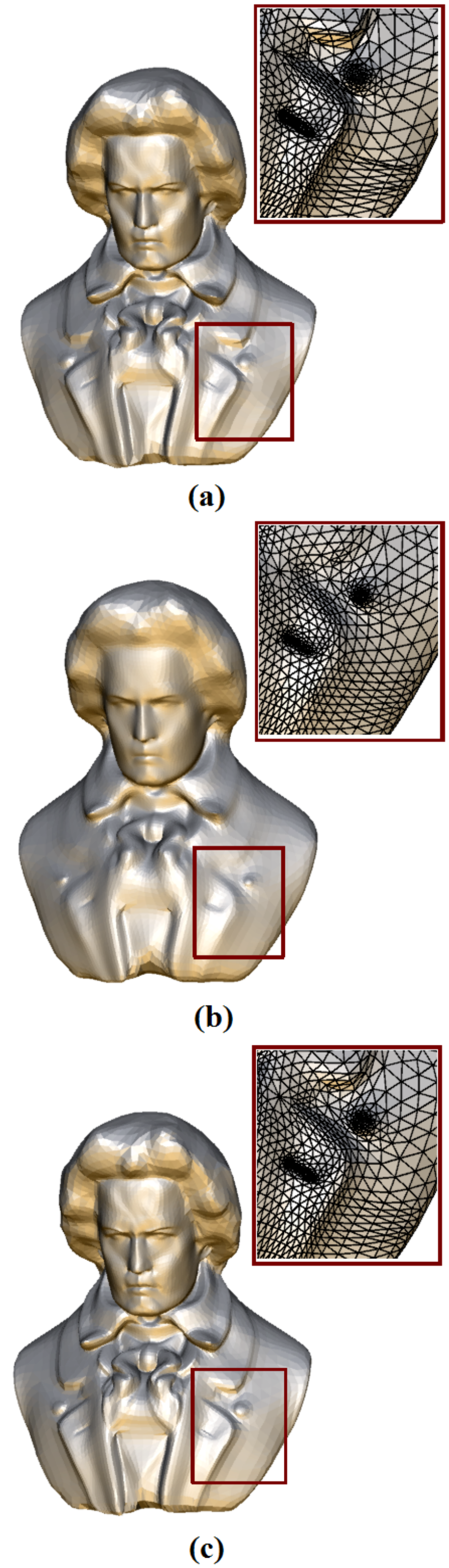

Figure 7: (a) Original model of Beethoven and fragment of the corresponding mesh; (b) The model processed with Laplacian smoothing during 2 iterations and fragment of the corresponding mesh; (c) The model (b) optimized with our restoring technique and fragment of the corresponding mesh.

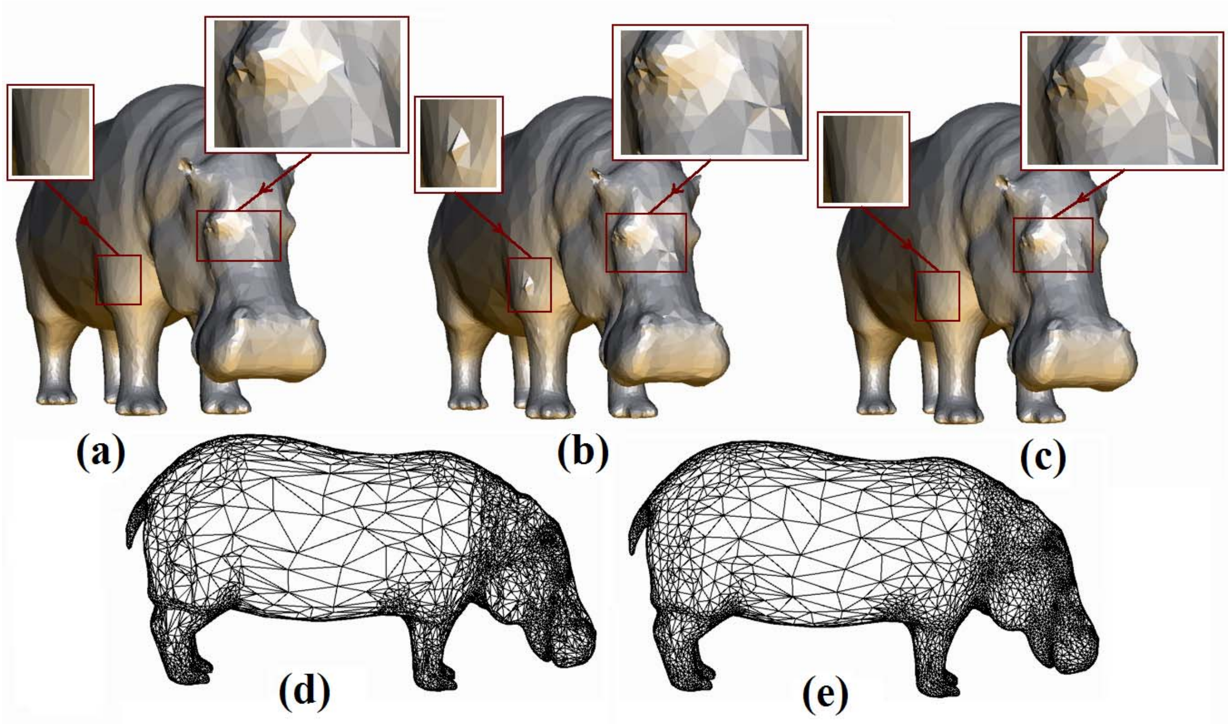

Figure 8: (a) Original Hippopotamus model; (b) The model processed with BDS approach; (c) The model (b) optimized with our restoring technique; (d) Mesh of the original Hippopotamus model; (e) Mesh of the (c) model; Note that our restoring algorithm allows to preserve mesh element quality improved with BDS approach.

4. APPLICATIONS

4.1 Smoothing

Our original motivation of this research started with the studying problem of improving element quality for surface meshes [5-6]. We concentrated on the local node movement techniques, which do not change mesh connectivity and commonly called in engineering community mesh smoothing.

There are three main possibilities for node movement. We can flatten the original surface by repositioning the node to the geometric center of its neighboring nodes, like is done in Laplacian smoothing; we can keep the node on the original discrete surface [4] (let us call this approach "Back to the Discrete Surface" (BDS)); or we can try to put the node on the approximated smooth surface [5-6]. While the latter approach seems to be the most promising, the techniques proposed in [5-6] have some limitations.

Let us demonstrate how our restoring procedure can optimize the results of Laplacian smoothing and BDS approach.

We first processed with Laplacian smoothing very irregular meshes of a sphere and a cubic surface (Fig. 5(a), 6(a)). From Fig. 5(b), 6(b) it can be seen that though geometric mesh quality has been improved the obtained surfaces are far from the original smooth models. Results of applying our restoring technique to these surfaces are demonstrated in Fig. 5(c), 6(c). Note that resulting models are very smooth while the mesh quality improved with Laplacian smoothing has been hardly changed.

One can ask why we need the restoring technique if the meshes in above examples can be simply processed with some smoothing filter [15-17]. Our answer is the following.

- Although smoothing filters produce very smooth shapes, they do not allow to move mesh vertices on the original surface. On the contrary, our algorithm allows to put vertices back to the original shape (Fig. 9). This property of the algorithm is very attractive especially for engineering applications where accuracy of numerical simulations heavily depends on accuracy of surface approximation. The data from Table 1 verifies that maximum ($E_{\max}$) and average (E_{aver}) deviations from the original analytical surfaces for the meshes processed with our restoring technique (M^{opt}) much lesser then those for the meshes processed with Laplacian smoothing (M'), bilapacian flow [17] (M^b), Taubin smoothing scheme with equal weights [15] (M^e) and cotangent weights [16] ($M^{\cot}$).
- The restoring algorithm works not only as a smoother. Our technique provides much more possibilities. Given the original mesh M and the resulting mesh M' we can reconstruct lost features of the original smooth shape. Let us give an example.

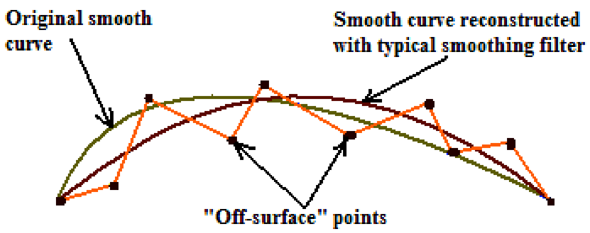

Figure 9: Typical smoothing filter produces the smooth surface but fails to reconstruct the original shape.

Fig. 7(a) demonstrates the original model of Beethoven and the same model processed with Laplacian smoothing during 2 iterations (Fig. 7(b)). We can see that almost all essential features of the model have been diffused. Our restoring technique allows to reconstruct the original shape (Fig. 7 (c)). And again the technique does not cause considerable damage to the mesh element quality improved with Laplacian smoothing.

Another example is connected with above-mentioned BDS approach. The simplest BDS scheme is to find an optimal node position in a locally derived tangent plane and project it back to the original discrete surface. The difficulty is to define correctly the tangent plane. Fig. 8(a) shows an example when incorrectly estimated normals and tangent planes result in a rather bad non-smooth surface with distortion of some features of the original model. Our restoring technique can efficiently eliminate this drawback while preserving the geometric mesh quality improved with the technique (Fig. 8(ce)).

4.2 Simplification

Finally, let us demonstrate how our algorithm can be applied to simplification procedure. When the mesh is simplified be edge collapse, it is possible for the two endpoints to be replaced by a new point at the midpoint of the collapsed edge (such procedure is applied in many engineering applications [18-19]) or at another point of the edge (like is done in Quadric Error Metric algorithm [20]). However, such procedure may cause the same problems as BDS approach since new points always stay on the original mesh but not on the surface approximated by this mesh.

Fig. 10(b) demonstrates the simplified model of Buddha obtained by replacing collapsed edges by the midpoints. We can clearly see "oversmoothing" effect: many features of the original model were diffused. The restoring technique allows to reconstruct these features as shown in Fig. 10(c). Fig. 10(def) demonstrates the similar results for the Dragon model.

Our algorithm can be used not only as post-processing optimization but also as an intermediate step in the described simplification procedure that may improve essentially the output model.

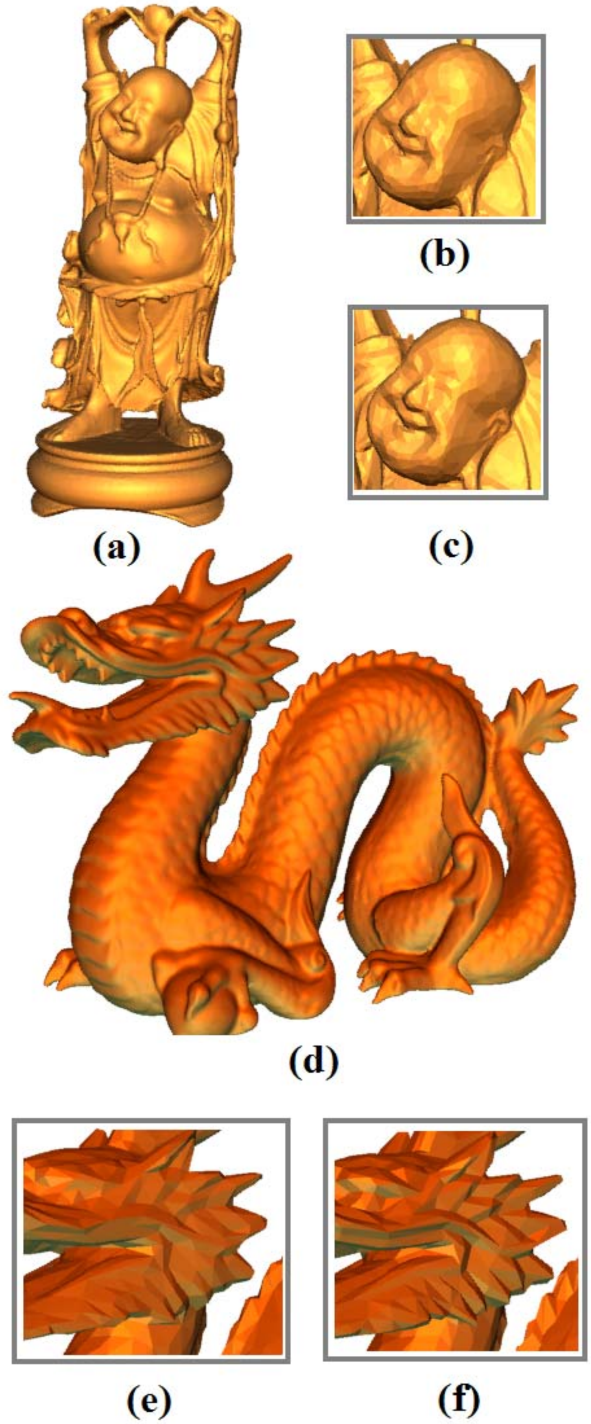

Figure 10. (a) The original Buddha model (144647 points); (b) Fragment of the Buddha model simplified by replacing collapsed edges by the midpoints (14625 points); (c) Fragment of the model (b) optimized with our restoring technique (14625 points); (d) The original Dragon model (437645 points); (e) Fragment of the Dragon model simplified by replacing collapsed edges by the midpoints (4603 points); (f) Fragment of the model (e) optimized with our restoring algorithm (4603 points)

5. COMPARISON OF THE ALGORITHM WITH THE SCHEME OF KARBACHER ET. AL.

As is mentioned, our scheme is close in spirit to the one of Karbacher et. al. [1-2] that has been used for non-linear subdivision of the triangular meshes. Several important observations can be made:

- While the scheme of Karbacher et. al. allows to move points from the triangles of the original mesh to the approximated smooth surface, we propose the simple procedure to put any "off-surface" point to the approximated smooth shape.
- In addition to the max-algorithm used in works of Karbacher et. al. we have introduced the min-algorithm that allows to improve significantly "restoring" properties of the technique.
- Subdivision scheme based on circular arcs movement requires sometimes additional smoothing of the final surface [2]. We eliminate this drawback by using "weighting-by-inverse-areas" scheme for normal estimation and iterative procedure described in Section 3.3.

6. SUMMARY AND FUTURE WORKS

We have presented the novel approach for reconstruction of shapes from output of various mesh processing filters.

The proposed technique has been applied to optimize output of local node movement and simplification algorithms. We plan to apply also the algorithm as an optimization step for remeshing process.

We clearly realize limitation of our technique. As is mentioned in Section 1.2, the algorithm is able to "reconstruct" the original shape only if the resulting mesh is geometrically faithful representation of the initial model. In our future research we will explore ways to extend capacities of the technique.

Another problem arises when applied mesh processing results in un-aligned features of the two meshes. In that case, using a simple Euclidean distances for the correspondence between the vertices of the deformed mesh and the elements of the original mesh may result in erroneous restoration. One way to fix it is to associate features in the original and resulting meshes by using more sophisticated metric. Another possible solution is to combine the applied filter and our technique. Whenever vertex of the original mesh is moved, we need to put it to the original approximated smooth shape with our algorithm. We are going to address this problem in our future research.

7. REFERENCES

[1] S. Karbacher, S. Seeger, G. Hausler, A non-linear subdivision scheme for triangle meshes, In *Vision, Modeling and Visualization 2000*, pp. 163-170, Saarbrucken, Germany, (2000)

[2] S. Karbacher, S. Seeger, G. Hausler, Refining triangle meshes by non-linear subdivision, In *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, pp. 270-277, IEEE Computer Society, (2001)

[3] Y. Ohtake, A. Belyaev, H.-P. Seidel, Interpolatory Subdivision via Diffusion of Normals, *Computer Graphics International 2003*, Tokyo Institute of Technology (2003)

[4] R. V. Garimella, M. J. Shashkov, P. M. Knupp, Optimization of Surface Mesh Quality Using Local Parameterization, *Proceedings of the 11th International Meshing Roundtable*, pp. 41-52, (2002)

[5] I. Semenova, V. Savchenko, I. Hagiwara, Two Techniques to Improve Mesh Quality and Preserve Surface Characteristics, *Proceedings in 13th International Meshing Roundtable*, pp. 277-288, (2004)

[6] I. Semenova, V. Savchenko, I. Hagiwara, Two Techniques to Improve Mesh Quality and Preserve Surface Characteristics, *Proceedings in 13th International Meshing Roundtable*, pp. 277-288, (2004)

[7] V. Surazhsky, C. Gotsman, Explicit Surface Remeshing, in *Proceedings of Eurographics Symposium on Geometry Processing*, pp. 17-28, (2003)

[8] P. Frey, H. Borouchaki, Geometric Surface Mesh Optimization, *Computer and Visualization in Science*, pp. 113-121, (1998)

[9] G. Turk, Re-Tiling Polygonal Surfaces, In *ACM SIGGRAPH Proceedings*, pp. 55-64, (1992)

[10] L. Kobbelt, J. Vorsatz, U. Labsik, H.-P. Seidel, A Shrink Wrapping Approach to Remeshing Polygonal Surfaces, *Eurographics Proceedings (CGF)18*, pp. 119-130 (1999)

[11] A. Lee, H. Moreton, H. Hoppe, Displaced Subdivision Surfaces, *Proceedings of ACM SIGGRAPH 2000*, pp.85-94 (2000)

[12] I. Guskov, K. Vidimce, W. Sweldens, P. Shroder, Normal Meshes, *Proceedings of ACM SIGGRAPH 2000*, pp.95-102 (2000)

[13] E. V. Anoshkina, A. G. Belyaev, H.-P. Seidel, Asymptotic Analysis of Three-Point Approximations of Vertex Normals and Curvatures, *Vision, Modeling, and Visualization 2002*, pp. 211-216, (2002)

[14] S. Karbacher, G. Hausler, New Approach for Modeling and Smoothing of Scattered 3D Data, In *Three-Dimensional Image Capture and Applications*, Proc. SPIE 3313, pp. 115-125, (1998)

[15] G. Taubin, Linear Anisotropic Mesh Filtering, *IBM Research Report RC22213 (W0110-051)*, (2001)

[16] M. Desbrun, M. Meyer, P. Schroder, A. H. Barr, Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow, *Proceedings of ACM SIGGRAPH'99*, pp. 317-324, (1999)

[17] L. Kobbelt, S. Campagna, J. Vorsatz, H.-P. Seidel, Interactive Muliresolution Modeling on Arbitrary Meshes, In *Proceedings of ACM SIGGRAPH'98*, pp. 105-114, (1998)

[18] T. J. Baker, Mesh Movement and Metamorphosis, *Proceedings in 10th International Meshing Roundtable*, Sandia National Laboratories, pp.387-396 (2001)

[19] C. Ollivier-Gooch, Coarsening Unstructured Meshes by Edge Contraction, *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, Ltd., Vol 57, Num 3, pp.391 – 414 (2003)

[20] M. Garland, P. S. Heckbert, Surface Simplification Using Quadric Error Metrics, In *Proceedings of ACM SIGGRAPH'97*