Explicit Method to Optimize Surface Mesh Quality

IRINA B. SEMENOVA [1], ICHIRO HAGIWARA [2]
Dept. of Mech. Sciences and Eng., Tokyo Institute of Technology, JAPAN,

Abstract: - In this paper, we introduce a novel technique to improve surface mesh quality while maintaining all essential surface characteristics. In contrast to previous methods we do not constrain node movement to the underlying discrete surface. Instead our approach allows keeping resultant mesh very close to the surface approximated by the original mesh. Proposed method is explicit, that is we do not need any parameterization of the original mesh. All operations are performed directly on the surface. As a result our technique is robust and runs at interactive speeds. Several quantitative measures and error-metrics are presented to prove the effectiveness of proposed technique and compare it with the previous approaches.

Key-Words: - Surface mesh quality, surface characteristics, error-metrics

1 Introduction

Improvement of surface mesh quality is important problem for numerical simulations, solid mesh generation and computer graphics applications.

There are two main ways to achieve high quality mesh: modifications of mesh topology by inserting/deleting mesh nodes or edge flipping [1][2] and node movement methods commonly called mesh smoothing. This paper focuses on the latter way.

To improve mesh quality in the plane a number of smoothing techniques have been developed ranging from simple Laplacian smoothing (see, for instance, [3] and references therein) to more sophisticated algorithms. Among them there are physically based methods [4][5] where nodes are moved under the influence of some forces so that the shape of incident elements is improved. Instead of local mesh optimization by moving each node on the basis some geometric characteristics (as is done in Laplacian smoothing, angle-based [6] and physically based methods) the optimization-based techniques allow improving all original mesh. In these techniques so called cost function [7] is optimized. As such function aspect ratio [8] or distortion metrics [9] [10] can be used.

It is necessary to note that the good shape of mesh elements is not only the criteria for mesh quality when surface meshes are considered. It is also essential to minimize changes in the surface characteristics like normals and curvatures. As it has been pointed out in [11] preservation of such characteristics is important for preventing drastic changes in the volume enclosed by the surfaces and in forces like surface tension that depends on surface properties.

1.1 Previous works

Several techniques to improve surface mesh quality have been developed over the last decade. Most of them are based on the idea to constrain node movement to the underlying discrete surface. A simple way is to reposition each node in a locally derived tangent plane and project it back to the surface [12][13]. More robust algorithms use global parameterization of the original mesh, and then improvement in the parameter domain [14][15][16]. The main drawback of these methods is high computational cost since they involve the solution of a large set of equations. Moreover, global parameterization may distort the complicated 3D structure. The alternative to global parameterization has been proposed in [11]. The nodes of the mesh are moved in a series of local parametric spaces derived from individual mesh elements.

Let us note, however, that all these methods allow keeping new nodes on the original mesh but not on the surface approximated by this mesh. As a simple example consider a sphere and a mesh with the nodes situated on this sphere. Applying algorithms described above we will obtain new nodes situated on the original mesh but not on the original sphere. Therefore, unlike initial mesh the new mesh will not be discrete approximation of the original sphere. Furthermore, several

iterations will cause considerable shrinking initial surface.
In [17] we have presented a novel technique called Curvature Based Mesh Improvement, which effectively solves foregoing problems. To find new location of each node we used value of maximum curvature defined at this node. Proposed method gives good results in the sense of preserving surface characteristics. But it can be applied only for meshes representing smooth surfaces without sharp edges and corners. Furthermore this method has rather high computational cost since it involves curvature estimation.

1.2 Contribution and overview

In this paper we present a simple approach to overcome drawbacks of the previous algorithms. We do not pose a problem to preserve new nodes on the original mesh that is on the discrete surface. Instead we propose method to keep resultant mesh very close to the surface approximated by the initial mesh. This method can be called explicit because all operations are performed directly on the surface. That is the reason why our technique is robust and fast. But for all that we do not sacrifice any quality in results. Moreover the method can be applied iteratively.
The presented scheme can be improved by introducing some constraints. It allows significantly reducing damage to sharp edges of the surface.
Thus our main contribution in this paper is a robust, iterative technique, which optimizes surface mesh quality and does not cause considerable damage to the important surface characteristics.
Another contribution is a set of several error-metrics to quantify the deviation of the resultant mesh from the original one.
The rest of this paper is organized as follows. In Section 2 we give a detailed description of our algorithm and its weighted version. Section 3 describes some quantitative measures and error-metrics to examine mesh quality and deviation of a resultant surface from the original one. Examples of applying our algorithm to various meshes are presented in Section 4. In this Section we also compare our approach with the previous ones. We close by offering some concluding remarks in Section 5.

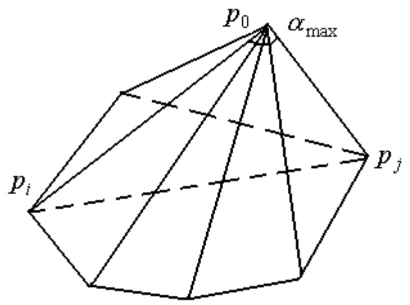


Fig. 1: Search for the node p_j such as vectors $\overrightarrow{p_0 p_i}$ and $\overrightarrow{p_0 p_j}$ compose a maximal angle.

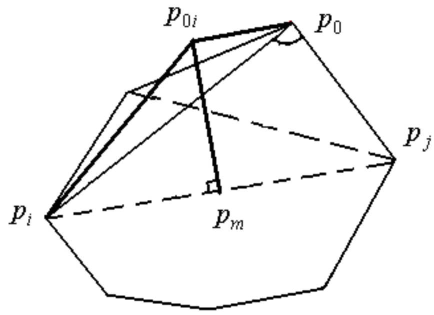


Fig. 2: Search for the new location of the node p_0 with regard to the node p_i.

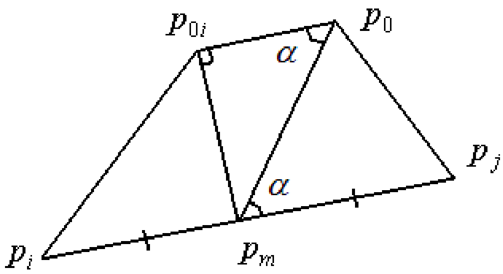


Fig. 3: Search for the coordinates of p_{0i}.

2 Algorithm

Let us consider some node p_0 of the original mesh and all nodes $P = \{p_i\}, i = 1,...k$ associated with this node. For each node p_i the new position of the node p_0 is obtained according to the following procedure.
At first, we find the node $p_j \in P$ such as vectors $\overrightarrow{p_0 p_i}$ and $\overrightarrow{p_0 p_j}$ compose a maximal angle as shown in Fig. 1. The new position p_{0i} of the node p_0 with regard to p_i will be a vertex of a trapezium such as a triangle consisting of the points p_i, p_{oi}, p_j is isosceles (Fig. 2). Let us emphasize two aspects of the algorithm. $\overrightarrow{p_0 p_{0i}} \parallel \overrightarrow{p_j p_i}$, therefore new node will be very close to the original surface. $\Delta p_i p_{0i} p_j$ is isosceles

that provides improvement of correspondent mesh elements.

Let us denote by p_m the midpoint of the segment $p_i p_j$. The new coordinates of p_{0i} may be found using following formulas:

$$p_0 -> p_0 + \left\| \overrightarrow{p_0 p_{0i}} \right\| \cdot \frac{\overrightarrow{p_j p_i}}{\left\| \overrightarrow{p_j p_i} \right\|} = p_0 + \cos\alpha \cdot \left\| \overrightarrow{p_m p_0} \right\| \cdot \frac{\overrightarrow{p_j p_i}}{\left\| \overrightarrow{p_j p_i} \right\|} = p_0 + \frac{\left(\overrightarrow{p_m p_0}, \overrightarrow{p_m p_j} \right)}{\left\| \overrightarrow{p_m p_j} \right\|} \cdot \frac{\overrightarrow{p_j p_i}}{\left\| \overrightarrow{p_j p_i} \right\|}$$

(see Fig. 3).

After all p_{0i} have been found the new position of the node p_0 is obtained by averaging coordinates of p_{0i}, $i = 1,...k$.

2.1 Quality control

Using any optimization technique we must be sure that there will be no invalid or badly shaped elements in the resultant mesh. The simplest way is to check whether the quality of mesh elements improved after applying technique or not. It is common to use for that minimal angle like it is done in smart Laplacian smoothing [9]. However let us note that such procedure reduces the risk of obtaining inverted elements but still cannot guarantee validity of the new mesh. To solve this problem we propose to use signed aspect ratio $k = 4\sqrt{3} \cdot A \big/ \left(l_1^2 + l_2^2 + l_3^2\right)$, where l_i, $i = 1,2,3$ are the lengths of triangle sides and A is the signed area of triangle (i.e., with respect to counter-clockwise orientation). For oriented mesh all valid triangles have positive areas and all inverted triangles have negative areas. It is obvious that for any triangle $-1 \le k \le 1$. A value 1 corresponds to an equilateral triangle, while –1 indicates an inverted equilateral triangle. When all the three points of the triangle are co-linear, the triangle is degenerate that yields a value of 0.

After the new position of the node has been found we need to calculate minimal aspect ratio k_{new} for the triangles adjacent to this node and compare it with the minimal aspect ratio k_{old} with regard to the old position of the node. If $k_{new} > k_{old}$ we move the node to the new position. Otherwise we keep the node at its initial position.

Let us note that computational cost of calculating signed aspect ratio is the same as computational cost of calculating the minimal angle. But unlike minimal angle signed aspect ratio guarantees the validity of the resultant mesh.

(a) (b) (c)

Fig. 4: (a) Fragment of Cylinder model; (b) Fragment of Cylinder model processed with proposed algorithm; (c) Fragment of Cylinder model processed with constrained algorithm.

2.2 Constrained algorithm

As it will be shown described algorithm demonstrates very good results in the sense of preserving main surface characteristics such as curvatures and normals. But it still cannot guarantee conservation sharp edges of the surface (Fig. 4(a,b)).

We propose a very simple improvement to the original scheme, which significantly reduces the damage to sharp edges.

Let us consider some node p_0 of the original mesh and all nodes $P = \{p_i\}$, $i = 1,...k$ associated with this node. For each edge $p_0 p_i$, $i = 1,...k$ we

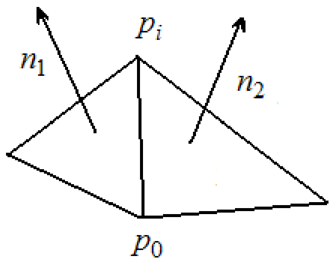


Fig.5: Calculation of the angle between two adjacent normals.

calculate angle α_i between normals to the adjacent faces (Fig. 5). Then we find two maximal angles α_i, α_j. Denote by α_{ij} an angle between $p_0 p_i$ and $p_0 p_j$. If $\left|\alpha_i - \alpha_j\right| < \varepsilon_1$, $\alpha_i, \alpha_j > \varepsilon_2$, and $\alpha_{ij} > \varepsilon_3$ where $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are some threshold values, we believe that the nodes p_i, p_0, p_j belong to the sharp edge of the surface and apply procedure described in Section 2 only for nodes p_i and p_j.

In other words we construct trapezium for the nodes p_i, p_0, p_j. Obtained fourth vertex of this trapezium is new position of the node p_0. Usually we define threshold values to be $\varepsilon_1 = 0{,}2$, $\varepsilon_2 = 0{,}7$, and $\varepsilon_3 = 2{,}6$.

The robustness of our improvement is demonstrated in Fig. 4(c).

3 Quantitative measures

The problem of surface mesh improvement includes two main aspects: improvement of mesh element shapes and preserving surface characteristics and features as much as possible. Thus a tool for evaluation of optimized mesh should include geometric quality measures and so-called error-metrics to estimate the deviation between original and resultant meshes. In this Section we introduce such tool.

3.1 Signed aspect ratio

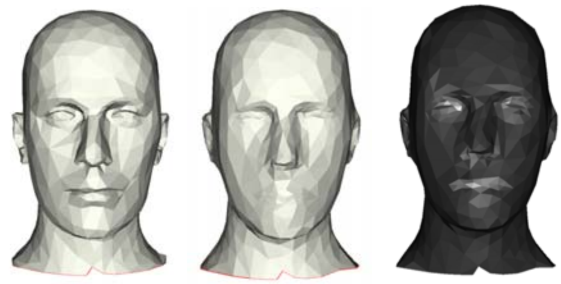
To measure the geometric properties of the obtained mesh we use signed aspect ratio described in Sections 2.1. The quantity of triangles with aspect ratio $k < 0$ (inverted elements), $0 < k < 0.2$ and $0.2 < k < 0.4$ are computed for original and resultant meshes to demonstrate the improvement of mesh quality.

3.2 Normal-based metric

To quantify the deviation resultant mesh from the original one we propose L^1 normal-based metric. Let M and M' be original and resultant mesh respectively. Consider a triangle T' of the mesh M' and find a triangle T of M closest to T'. Denote by $n(T)$ and $n(T')$ the oriented unit normals to T and T'. Let $\alpha(T,T')$ be an angle between $n(T)$ and $n(T')$. Then, the metric is defined by $E_n = \sum_{T' \in M'} \alpha(T,T') \big/ \pi \cdot m$, where m is the number of mesh triangles.

Since for any triangle T' $\alpha(T,T') \le \pi$, it is clear that $E_n \in [0,1]$ and $E_n = 0$ in case that the resultant mesh coincides with the original one.

Let us note that this metric is sensitive to degradation of sharp features and highly curved regions. To illustrate this property of the metric we use model of Mannequin and the same model after applying simple Laplacian smoothing. The latter is colored according to E_n-distance from the original model. Dark-gray color corresponds to the small-error regions and light-gray color marks out the regions with considerable distortion of the original surface (Fig. 6).



(a) (b) (c)

Fig. 6: (a) Model of Mannequin; (b) An example after processing by Laplacain smoothing; (c) Smoothed model colored according to E_n-distance from the original model.

3.3 Curvature-based metric

The essential geometric features of any surface are its curvatures. Roughly speaking, the degree of surface “bending” in space is defined by mean curvatures. Therefore we use the notion of mean curvature to construct our curvature-based metric. It is easy to see that polyhedral surface “bends” along edges.

Denote by $\beta(e_i)$ the dihedral angle between two faces adjacent to the edge e_i of the mesh. We calculate this angle in such a way that $\beta(e_i) > 0$ if e_i is convex edge, $\beta(e_i) < 0$ if e_i is concave edge, and $\beta(e_i) = 0$ if e_i is plane edge.

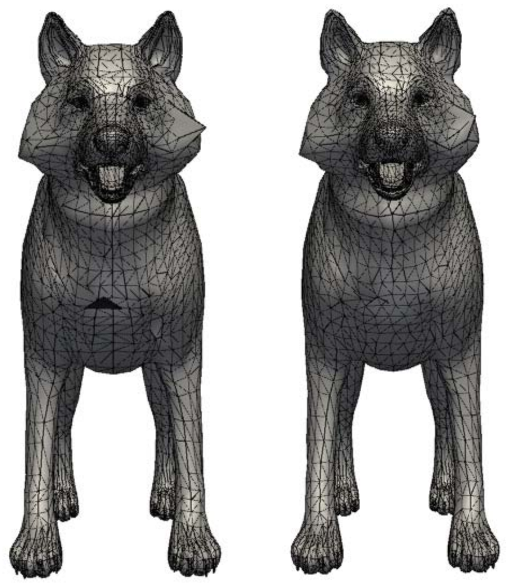
Then our L^1 curvature-based metric is defined by $E_c = \sum_{e_i' \in M'} \left| \beta(e_i) - \beta(e_i') \right| \big/ 2\pi m$, where m is the number of mesh edges.

This metric has the same properties as our normal-based metric. It penalizes the error at sharp edges of the mesh and its value varies from zero to unity.

4 Results

Firstly let us demonstrate the visual effect of applying described algorithm. From the results shown in Fig. 7, we can see that even after fifth iteration the model of Wolf preserves all its characteristic features. Fig. 7 demonstrates also that our algorithm does not destroy special features of the model such as anisotropy and local refinement. We can see that local refinement near wolf’s mouse and eyes is preserved even after fifth iteration. It is useful to remark that proposed technique can handle inverted elements. The invalid triangles on the original model of Wolf are marked by black patches. From Fig. 7(b) it is easily seen that algorithm efficiently eliminates inverted elements. Statistics from Tables 1 and 2 prove that the deviation of the resultant mesh from the original surface is small while geometric quality of the mesh is improved greatly.

We have also implemented our algorithm with weights to the original Stoned model shown in Fig. 8(a). Fig. 8(c) illustrates that introduced weights allow preserving sharp edges of the model even after several iterations. Tables 4 and 5 show the statistics for optimized model.



(a) (b)

Fig. 7: (a) Model of Wolf (number of polygons is 113992); (b) The optimized model of Wolf after fifth iteration.

	Original mesh	1st iteration	5th iteration
$k \le 0$	4	1	0
$0 < k \le 0.2$	182	67	4
$0.2 < k \le 0.4$	1057	634	289

Table 1: Histograms of signed aspect ratio for the mesh of Wolf.

	1st iteration	5th iteration
E_n	0.017770	0.0369311
E_c	0.019596	0.0348723

Table 2: Error-metrics on the mesh of Wolf.

4.1 Comparison of our algorithm and previous approaches

To compare our approach with the previous ones we chose simple method illustrated the main idea of the preceding works. At each node of the original mesh we define local tangent plane, project neighboring nodes to it and find new location of the concerned node using smart Laplacian smoothing. After that we project obtained node back to the discrete surface. To be precise, let us call this approach “Back to the Discrete Surface” (BDS).

We applied usual Laplacian smoothing, BDS and our algorithms to the triangular mesh of the cubic surface $z = 4x^3 + 6y^3$. We also implied our method Curvature Based Mesh Improvement (CBMI) described in [17].

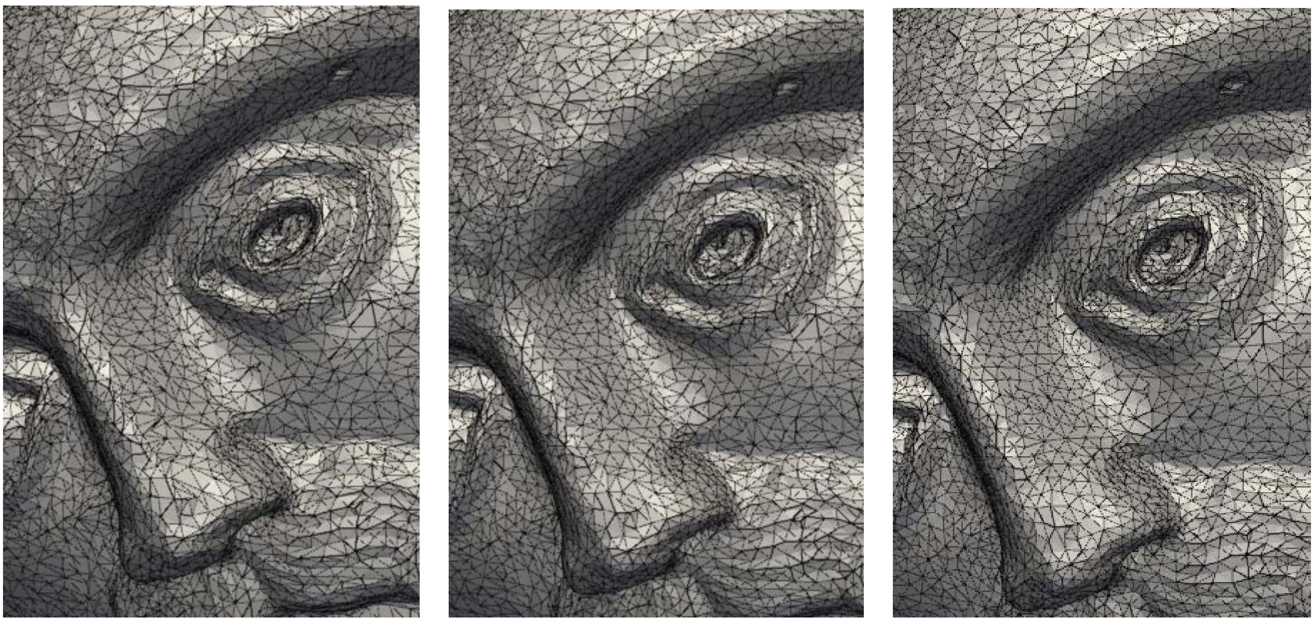
Different quantities measuring the change in the obtained meshes and the original surface can be easily computed since we have mesh on the analytical surface. We calculated maximum and average changes in Gauss and Mean curvatures ($\Delta_{\max}\lambda_{Gauss}, \Delta_{aver}\lambda_{Gauss}, \Delta_{\max}\lambda_{mean}, \Delta_{aver}\lambda_{mean}$), maximum and average changes in normals ($\Delta_{\max}\alpha_n, \Delta_{aver}\alpha_n$), maximum and average deviations from the original surface ($E_{\max}, E_{aver}$).

The statistics from the Table 3 confirm that CBMI algorithm gives the best results in the sense of preserving surface characteristics. The note must be made that our new technique causes little large changes in surface curvatures and normals than CBMI algorithm. Nevertheless it also keeps resultant mesh very close to the original surface. And let us remind that in contrast to CBMI algorithm we do not need to estimate surface curvatures and normals. Therefore we take a great advantage in speed. Moreover our new technique can be applied not only to the smooth surfaces but also to the surfaces with sharp corners and edges. Thus we believe that our algorithm is an ideal trade-off among mesh quality, preservation of surface characteristics and computational cost.

	Lapl. smooth.	BDS	CBMI	New method
$\Delta_{\max}\lambda_{Gauss}$	8.488	4.213	2.951	3.
$\Delta_{aver}\lambda_{Gauss}$	0.566	0.504	0.352	0.395
$\Delta_{\max}\lambda_{mean}$	2.007	1.434	1.438	1.075
$\Delta_{aver}\lambda_{mean}$	0.144	0.124	0.087	0.095
$\Delta_{\max}\alpha_n$	4.89^0	4.73^0	3.46^0	3.46^0
$\Delta_{aver}\alpha_n$	1.05^0	0.8^0	0.56^0	0.65^0
$E_{\max}$	0.007	0.003	0.0011	0.0018
E_{aver}	0.001	0.0004	0.0002	0.0003

Table 3: Changes in surface characteristics for the mesh of cubic surface optimized with Laplacian smoothing, BDS, CBMI and our new algorithm.

(a) (b) (c)



(d) (e) (f)

Fig. 8: (a) The original Stoned model courtesy of R. Scopigno and M. Callieri of Institute CNUCE (number of polygons is 173902); (b) Model optimized with weighted algorithm after first iteration; (c) Model optimized with weighted algorithm after fifth iteration; (d) Fragment of original mesh; (e) Fragment of mesh after first iteration; (f) Fragment of mesh after fifth iteration.

	Original mesh	1st iteration	5th iteration
$0 < k \leq 0.2$	1186	606	407
$0.2 < k \leq 0.4$	7314	3260	1688

Table 4: Histograms of signed aspect ratio for the mesh of the Stoned model.

	1st iteration	5th iteration
E_n	0.0185485	0.0328791
E_c	0.0204367	0.0340501

Table 5: Error-metrics on the mesh of Stoned model

5 Conclusion

In this paper, we introduced novel approach to improve surface mesh quality while maintaining the essential surface characteristics. In contrast to the existing methods we did not tend to keep new vertices on the original discrete surface. All operations are performed directly on the surface. It allows us taking an advantage in speed without sacrificing any quality in results. Moreover as it has been shown the results are even superior to the previous ones in the sense of preserving surface curvatures and normals.

Introducing angle-based weights into original scheme allows considerably reducing damage to the sharp edges of the surface.

The procedure has been successfully tested on a number of complex triangular meshes. Different quantitative measures and error-metrics proved that proposed technique do not cause considerable changes in surface characteristics while improving mesh quality. The algorithm can be applied iteratively that allows the user attaining resultant mesh more suitable for his application.

Let us note that although we consider only triangular meshes our algorithm can be applied to quadrilateral meshes in the same way. While for triangular meshes our method uses the neighboring nodes connected with the central node, to apply these algorithm to quadrilateral meshes we simply need to consider all surrounding nodes.

References:

[1] H. Hoppe, T. Duchamp, J. McDonald, W. Stuetzle, Mesh Optimization, *Computer Graphics (SIGGRAPH' 93)*, 1993, pp. 19-26.

[2] H. L. de Cougny, Refinement and Coarsening of Surface Meshes, *Engineering with Computers*, Vol. 14(3):214, 1998.

[3] D. A. Field, Laplacian Smoothing and Delauney Triangulation, *J. Communications in Applied Numerical Methods,* Vol. 4, 1998, pp. 709-712.

[4] I. Babushka, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliviera (eds), Accuracy Estimates and Adaptive Refinement in Finite Element Computations, *John Wiley &Sons, Chichester,* 1986, pp. 281-297.

[5] K. Shimada, D. C. Gossard, Bubbled Mesh: Automated Triangular Meshing of Non-manifold Geometry by Sphere Packing, *Proceedings of the ACM Third Symposium on Solid Modeling and Applications,* 1995, pp. 409-419.

[6] T. Zhou, K. Shimada, An Angle-Based Approach to Two-Dimensional Mesh Smoothing, *Proceedings of the 9th International Meshing Roundtable,* 2000, pp. 373-384.

[7] L. A. Freitag, On Combining Laplacian and Optimization Approach to Finite Element Mesh Smoothing Techniques, *AMD Trends in Unstructured Mesh Generation,* Vol. 220, 1997, pp. 37-43.

[8] V. Parthasarathy, S. Kodiyalam, A Constrained Optimization Approach to Finite Element Mesh Smoothing, *J. Finite Element Analysis and Design,* Vol. 9, 1991, pp. 309-320.

[9] S. A. Canann, J. R. Tristano, M. L. Staten, An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, an Quad-Dominant Meshes, *Proceedings of the 7th International Meshing Roundtable,* 1998, pp. 479-494.

[10] O. P. Jacquotte, G. Goussement, Structured Mesh Adaptation: Space Accuracy and Interpolation Methods, *Computer Methods in Applied Mechanics and Engineering,* Vol. 101, 1992, pp. 397-432.

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

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

[13] P. M. Knupp, Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities, *International Journal for Numerical Methods in Engineering,* Vol. 48, 2000, pp. 401-420.

[14] M. S. Floater, Parameterization and Smooth Approximation of Surface Triangulation, *Computer Aided Geometric Design,* Vol. 14, 1997, pp. 231-250.

[15] K. Hormann, U. Labsik, G. Greiner, Remeshing Triangulated Surfaces with Optimal Parameterizations, *Computer Aided Geometric Design,* Vol. 33, 2001, pp. 779-788.

[16] M. Desbrun, M. Meyer, P. Alliez, Intrinsic Parameterizations of Surface Meshes, *Proceedings of Eurographics Conference,* 2002.

[17] I. Semenova, V. Savchenko, I. Hagiwara, Curvature Based Mesh Improvement, *ASCOMS*, 2004 (accepted).