Improvement of Triangular and Quadrilateral Surface Meshes

Irina B. Semenova[1], Vladimir V. 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

In this paper, we present a new technique called Trapezium Drawing to improve surface mesh quality while maintaining the essential surface characteristics. In contrast to previous methods we do not tend to preserve new mesh vertices on the original discrete surface. Instead our approach allows keeping new mesh close to the surface approximated by the initial mesh. All operations are performed directly on the surface. As a result our technique is robust and runs at interactive speeds. It can be applied to triangular and quadrilateral meshes iteratively. Various quantitative measures are presented to demonstrate the effectiveness of proposed technique.

Keywords: *Surface Mesh Quality, Surface Characteristics, Triangles, Quadrilaterals*

.

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 for mesh optimization: modifications of mesh topology (remeshing) by inserting/deleting mesh nodes or edge flipping [1][2][3][4] and node movement methods commonly called mesh smoothing. Let us note that changing the mesh topology may cause some problems in simulations requiring solution transfer from the original mesh to the resultant one. Moreover many remeshing algorithms can be applied only to triangular meshes. Therefore in this paper we focuses on node movement technique for surface mesh improvement.

To improve mesh quality in the plane a number of smoothing techniques have been developed ranging from simple Laplacian smoothing (see, for instance, [5] and references therein) to more sophisticated algorithms. Among them there are physically based methods [6][7] 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 [8] and physically based methods) the optimization-based techniques allow improving all original mesh. In these techniques so called cost function [9] is optimized. As such function aspect ratio [10] or distortion metrics [11] [12] 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 surface characteristics like normals and curvatures. As it has been pointed out in [13] preservation of such characteristics is important for preventing drastic changes in the volume enclosed by the surfaces and in forces like surface tension that depend 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 an 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 [14][15]. More robust algorithms use global parameterization of the original mesh, and then improvement in the parameter domain [16][17][18]. 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 [13]. The nodes of the mesh are moved in a series of local parametric spaces derived from individual mesh elements.

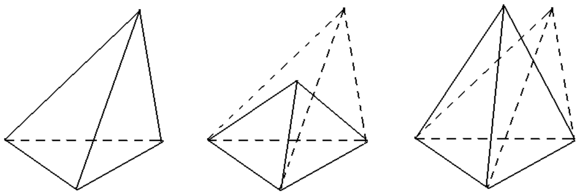
Let us note, however, that all these methods allow keeping new nodes on the original mesh but not on the surface approximated by this mesh. As a simple example consider a sphere and a mesh with the nodes situated on this sphere. Applying algorithms described above we will obtain new nodes situated on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

1.2 Contribution and overview

We looked at the problem of surface mesh improvement from another point of view. We do not pose a problem to preserve new nodes on the original mesh that is on the discrete surface. Instead we propose method called Trapezium Drawing (TD) 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. Even after several passes there is no degenerating mesh quality and all main features of the surface are preserved.

Figure 1 demonstrates the main difference between our approach to the problem and previous ones.

The rest of the paper is organized as follows. In Sections 2 and 3 we give a detailed description of the TD algorithm for triangular and quadrilateral meshes respectively. Section 4 describes some quantitative measures to examine mesh quality and deviation of a resultant surface from the original one. Examples of applying TD algorithm to various meshes are presented in Section 5. We close by offering some concluding remarks in Section 6.



(a) (b) (c)

Figure1: (a) Original pyramid; (b) Optimized pyramid with the apex vertex constrained to the original surface; (c) Pyramid optimized with TD algorithm.

2. ALGORITHM FOR TRIANLGE MESHES

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 Figure 2. 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 (Figure 3). Let us emphasize two aspect of the algorithm. $\overrightarrow{p_0 p_{0i}} \parallel \overrightarrow{p_j p_i}$, therefore new node will be close to the original not discrete but approximated smooth 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 Figure 4). After all p_{0i} have been found the new position of the node p_0 is obtained by averaging coordinates of $p_{0i}, i = 1,...k$.

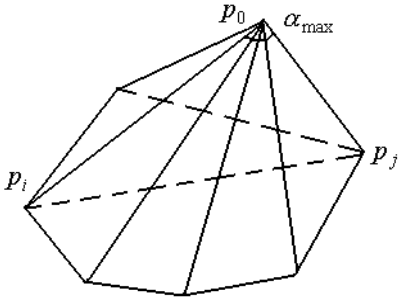


Figure 2: Search for the node p_j such as vectors $\overrightarrow{p_0 p_i}$ and $\overrightarrow{p_0 p_j}$ compose a maximal angle.

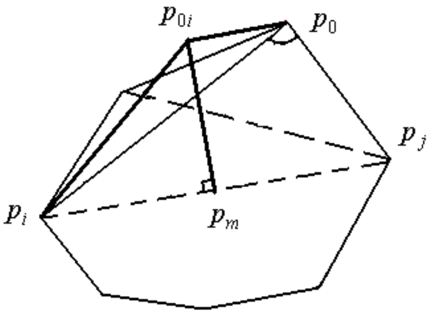


Figure 3: Search for the new location of the node p_0 with regard to the node p_i .

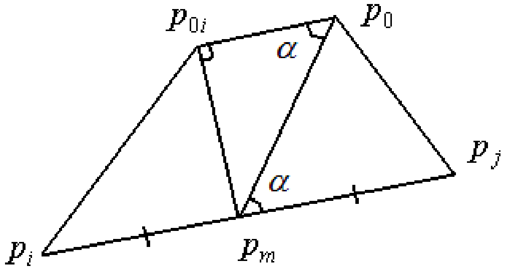


Figure 4: Search for the coordinates of p_{0i} .

2.1 Quality control

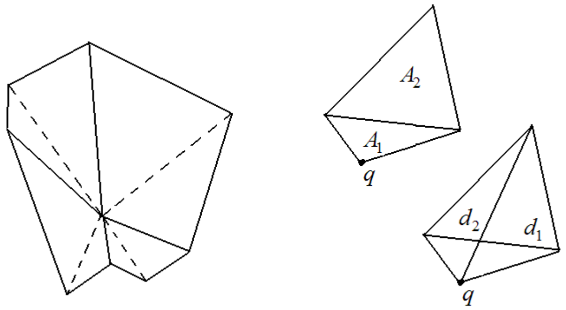
Using any optimization technique we must be sure that there will be no invalid or badly shaped elements in the resultant mesh. The simplest way is to check whether the quality of mesh elements improved after applying technique or not. It is common to use for that minimal angle like it is done in smart Laplacian smoothing [11]. 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 $k = 4\sqrt{3} \cdot \frac{A}{l_1^2 + l_2^2 + l_3^2}$, where l_1, l_2, l_3 are 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 three vertexes 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 validity of the resultant mesh.

3. ALGORITHM FOR QUADRILATERAL MESHES

As it has been described above for triangular meshes TD algorithm uses the neighboring nodes connected with the central node. To apply this algorithm to quadrilateral meshes we simply need to consider all surrounding nodes (Figure 5a).



(a) (b)

Figure 5: (a) TD algorithm for quadrilateral meshes; (b) Quadrangle adjacent to the node q; A_1, A_2 are the areas of the triangles; d_1, d_2 are the lengths of quadrangle diagonals.

3.1 Quality control

Just as in the case of triangular meshes we need some measure to guarantee validity of the resultant mesh. It seems good idea to decompose quadrangle in the pair of triangles and estimate the quality of these triangles. Frey in [19] proposed to use following quality measure: $k = \sqrt{l_1^2 + l_2^2 + l_3^2 + l_4^2} \cdot h_{\max} \Big/ \min_i A_i$, where $l_i, i = 1,...,4$ are the lengths of quadrangle sides, $h_{\max} = \max\{l_i, i = 1,...,4, d_1, d_2\}$ and A_i, $i = 1,2$, are the areas of triangles (Figure 5b). In [20] it has been proved that this measure has all desired properties for a quadrangle quality measure: extremal and asymptotic. But let us remark that we need some addition checks to detect invalid, e.g. self-intersected, elements (see [19] for details). Therefore we propose slight modification of this measure that allows effectively discovering inverted elements.

$$k = 4\sqrt{2} \cdot \min_i A_i^+ \Big/ \sqrt{l_1^2 + l_2^2 + l_3^2 + l_4^2} \cdot h_{\max},$$

where A_i^+, $i = 1,2$, are the signed areas of triangles. It can easily be checked that $-1 \le k \le 1$ for any quadrangle, $k = 1$ in the case of square, $k \to 0$ in the case of degenerate element, and $-1 \le k < 0$ in the case of inverted element.

Similarly to the case of triangle meshes after the new location of the node q is found we calculate minimal aspect ratio for the quadrangles adjacent to this node and compare it with the minimal aspect ratio for the old node position. If $k_{new} > k_{old}$ we move the node to the new position. Otherwise we keep the node at its initial position.

4. QUANTITATIVE MEASURES

We use various quantitative measures to demonstrate that after applying TD algorithm mesh quality is improved and the deviation of the resultant mesh from the original surface is small.

4.1 Signed aspect ratio

To measure the geometric properties of the obtained mesh we use signed aspect ratios described in Sections 2.1 and 3.1. Let us remind that the closer the value of k to 1, the closer element to equilateral one. $k < 0$ corresponds inverted elements while $k = 0$ indicates degenerate elements. The histograms of aspect ratio for original and resultant meshes are presented to demonstrate the improvement of mesh quality.

4.2 Change of the volume

Having a triangular or quadrilateral mesh it is easy to calculate the interior volume. This can be done by summing the volumes of all oriented pyramids centered at a point in space (the origin, for example) and with a triangle or quadrangle of the mesh as a base. We computed the volumes enclosed by the original and resultant surfaces and calculated the difference between them.

4.3 Changes in surface normals

For numerical simulation it is very important to minimize changes in the surface normals while improving mesh quality. We calculated differences between normals at the nodes of the resultant meshes and corresponding nodes of the original meshes to show that these changes are small.

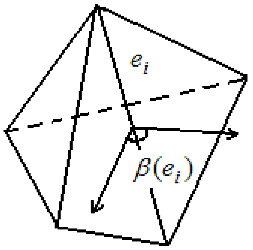


Figure 6: Dihedral angle used for approximation of mean curvature.

4.4 Changes in surface curvatures

It is also important to keep changes in surface curvatures small. Roughly speaking, the degree of surface "bending" in space is defined by mean curvatures. Therefore we use notion of mean curvature to prove that there is no large deviation between original and resultant surfaces. It is easy to see that polyhedral surface "bends" along its edges.

Denote by $\beta(e_i)$ the dihedral angle between two faces adjacent to the edge e_i of the mesh (Figure 6). 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.

The difference in dihedral angles of the original and resulting mesh is calculated to demonstrate that the change of the surface shape is small.

5. RESULTS

In order to show the effectiveness of the proposed TD algorithm let us consider two aspects of the problem of surface mesh improvement. Firstly let us demonstrate the visual effect of applying TD algorithm. From the results shown in Figure 7, we can see that even after fifth iteration the model of Mannequin preserves all its characteristic features. The statistics of the mesh of Mannequin after first and fifth passes are presented in Tables 1 and 2. We calculated maximum and average changes in dihedral angles ($\Delta_{\max}\lambda_{mean}$, $\Delta_{aver}\lambda_{mean}$ (degree)), maximum and average changes in normals ($\Delta_{\max}\alpha_n$, $\Delta_{aver}\alpha_n$ (degree)), and difference of the volume ($\Delta Vol(\%)$). We can see that average changes in normals and dihedral angles are small. Let us note also that without any special constraints [21] TD algorithm gives very good results in the sense of volume preserving.

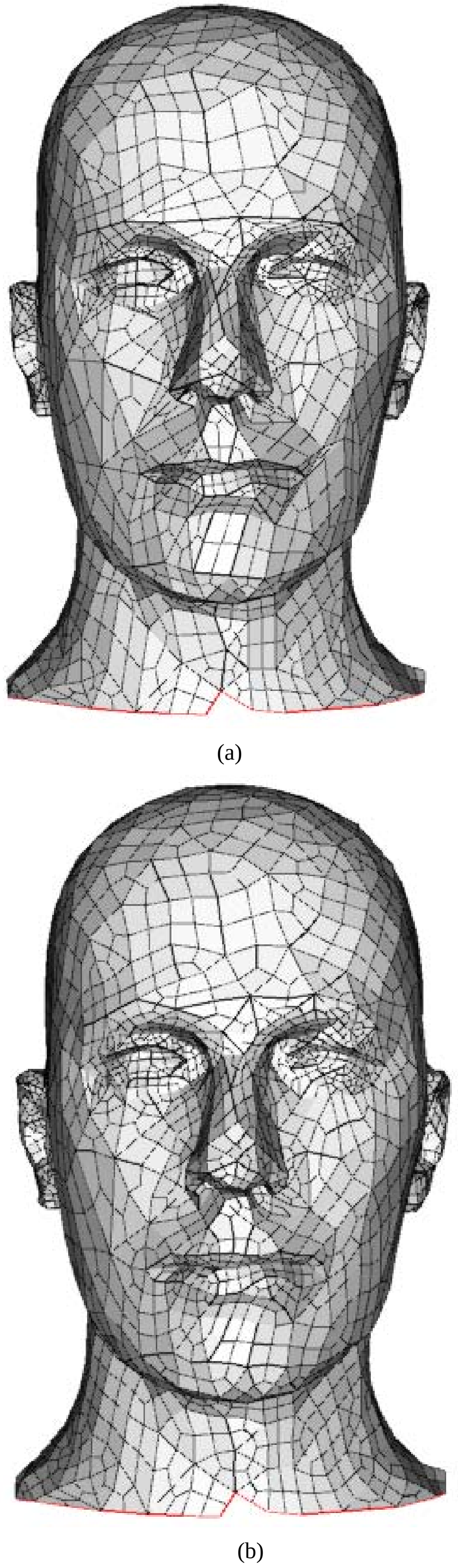
	Mannequin (1st pass)	Mannequin (5th pass)
$\Delta_{\max}\lambda_{mean}$	56.81^0	82.93^0
$\Delta_{aver}\lambda_{mean}$	0.652^0	1.75^0
$\Delta_{\max}\alpha_n$	33.918^0	63.077^0
$\Delta_{aver}\alpha_n$	1.566^0	3.418^0
$\Delta Vol(\%)$	0.017%	0.069%

Table 1: Statistics on the model of Mannequin.

	Initial	1st pass	5th pass
$0 < k < 0.2$	7,09%	4.37%	2.4%
$0.2 \le k < 0.4$	36.26%	32.59%	25.05%
$0.4 \le k < 0.6$	52.13%	41.28%	27.36%
$0.6 \le k < 0.8$	3.13%	18.61%	24.17%
$0.8 \le k \le 1$	1.21%	3.15%	21.02%

Table 2: Histograms of signed aspect ratio for mesh of Mannequin (percentage).

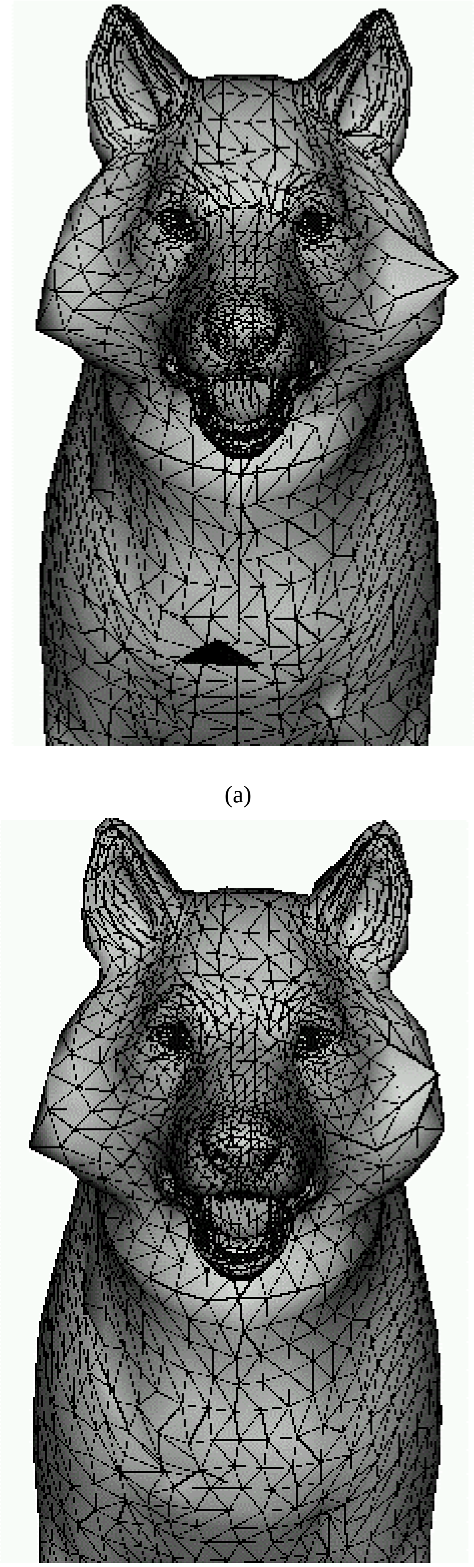
Figure 8 demonstrates that TD algorithm does not destroy special features of the model such as anisotropy and local refinement. We can see that local refinement near wolf's mouse and eyes is preserved even after fifth iteration. It is useful to remark that TD algorithm can handle inverted elements. Black patches mark the invalid triangles on the original model of Wolf. From Figure 8 it is easily seen that TD algorithm efficiently eliminates inverted elements. All statistics for the mesh of Wolf can be found in Tables 3 and 4.



(a)

(b)

Figure 7: (a) Mesh of Mannequin; (b) Mesh optimized with TD algorithm after fifth iteration.



(a)

(b)

Figure 8: (a) Mesh of Wolf; (b) Mesh optimized with TD algorithm after fifth iteration.

	Wolf (1st pass)	Wolf (5th pass)
$\Delta_{\max}\lambda_{mean}$	21.68^0	64.87^0
$\Delta_{aver}\lambda_{mean}$	1.39^0	3.13^0
$\Delta_{\max}\alpha_n$	27.959^0	54.833^0
$\Delta_{aver}\alpha_n$	2.171^0	4.712^0
$\Delta Vol(\%)$	0.027%	0.094%

Table 3: Statistics on the model of Wolf.

	Initial	1st pass	5th pass
$k < 0$	0.1%	0.04%	0%
$0 < k < 0.2$	4.79%	2.24%	0.7%
$0.2 \le k < 0.4$	29.87%	23.97%	15.4%
$0.4 \le k < 0.6$	49.32%	37.21%	18.92%
$0.6 \le k < 0.8$	13.51%	26.05%	31.67%
$0.8 \le k \le 1$	2.41%	10.49%	33.31%

Table 4: Histograms of signed aspect ratio for mesh of Wolf.

Now let us demonstrate more precisely how TD algorithm influences on the shape and size of the mesh elements.

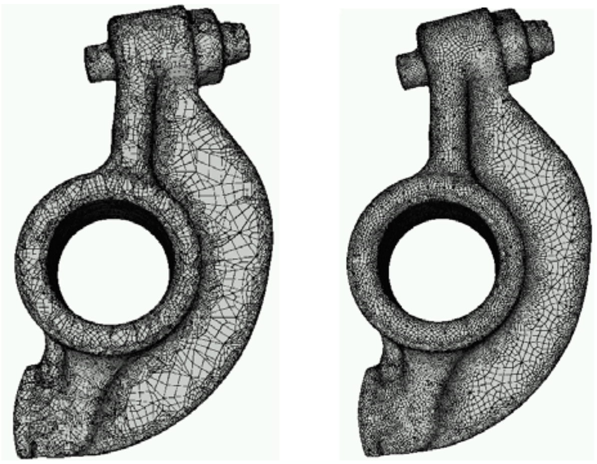
One important factor affecting the accuracy and efficiency of the finite element analysis is size quality. There are two cases that must be considered: uniform and graded meshes. For uniform meshes it is necessary to keep difference among sizes of the elements as small as possible. Graded meshes contain several layers that should not be diffused after applying some technique and size change at the transitional zone should be smooth. In Figure 9 we can see that even after several iterations there is no “diffusion” of layers, size change is very smooth and uniform zones remain uniform.

As is well known Laplacian smoothing may degrade mesh quality if the algorithm iterates more than a few times. On the contrary, as it can be seen from Table 5 TD algorithm tends to improve the aspect ratio as more passes are performed.

The rest of the statistics for quadrilateral mesh of Rocker Arm is presented in Tables 6.

	Initial	1st pass	5th pass
$0 < k < 0.2$	4.33%	2.54%	2.19%
$0.2 \le k < 0.4$	52.2%	44.4%	32.47%
$0.4 \le k < 0.6$	36.62%	38.21%	34.18%
$0.6 \le k < 0.8$	6.23%	12.07%	21.37%
$0.8 \le k \le 1$	0.62%	2.78%	9.79%

Table 5: Histograms of signed aspect ratio for quadrilateral mesh of Rocker Arm.



(a) (b)

Figure 9: (a) Quadrilateral mesh of Rocker Arm; (b) Mesh optimized with TD algorithm after fifth iteration.

	Rocker Arm (1^{st} pass)	Rocker Arm (5^{th} pass)
$\Delta_{\max}\lambda_{mean}$	$18,74^0$	64.12^0
$\Delta_{aver}\lambda_{mean}$	0.128^0	0.296^0
$\Delta_{\max}\alpha_n$	39.28^0	59.68^0
$\Delta_{aver}\alpha_n$	0.718^0	1.492^0
$\Delta Vol(\%)$	0.003%	0.0034%

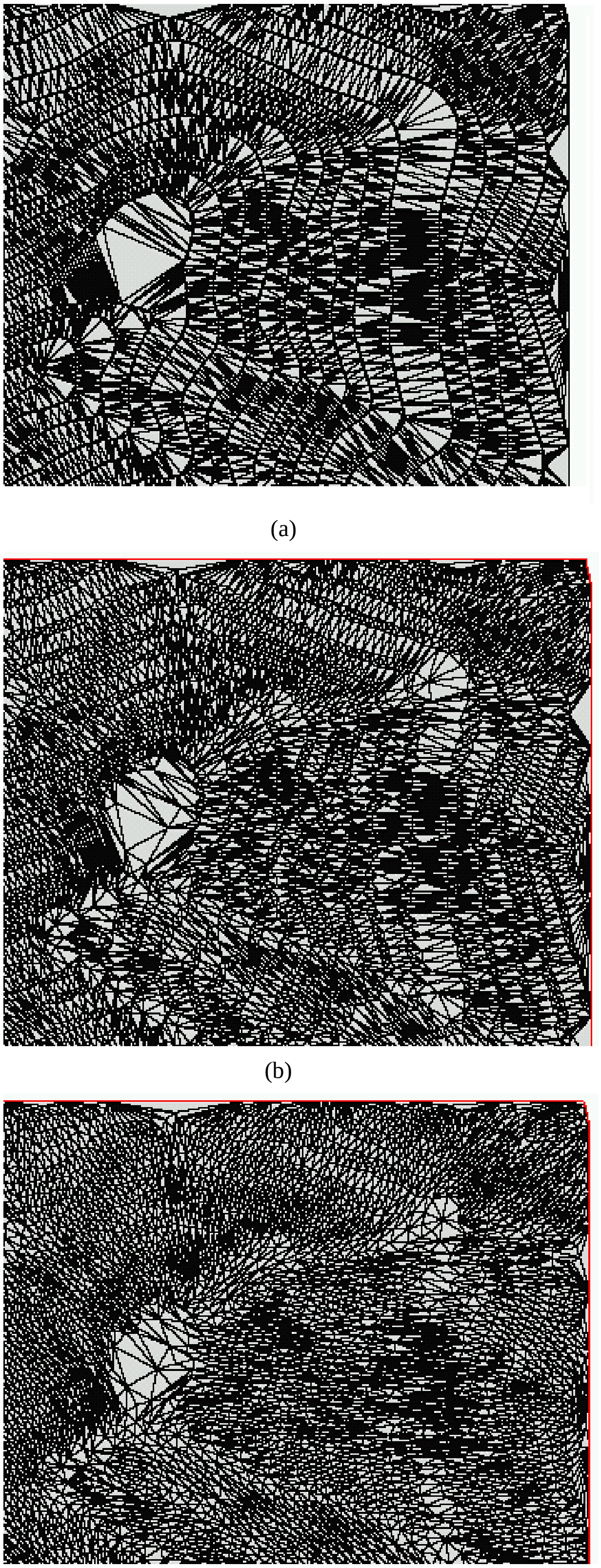
Table 6: Statistics on the quadrilateral model of Rocker Arm.

5.1 About applying TD to plane meshes

Although TD algorithm is meant for working with surface meshes it also gives good results for plane meshes. Actually in many cases, the results are even superior to the mostly used techniques.

As is well known Laplacian and smart Laplacian smoothing are the most commonly used techniques to improve the quality of plane meshes because of its low computational cost. But both methods may produce badly shaped or even inverted elements. In [8] there has been proposed effective method called angle-based approach with low computational cost. Quality of a mesh optimized with this method is much better than after Laplacian and smart Laplacian smoothing and chance to obtain inverted elements is reduced. But in [22] it has been pointed out that this is true mostly for the meshes with regular connectivity. When mesh contains very distorted elements angle-based approach also may fail. To solve this problem Surazsky and Gostman in [22] proposed to use weights. This scheme reduces the risk of generating inverted elements but still cannot guarantee that the obtained mesh will be valid. In fact, both angle-based and weighted angle-based approaches may produce the meshes with even worse quality than Laplacian smoothing.

Now let us give some interesting example of applying TD algorithm. As is well known, data can be obtained from different sources of information, in particular from existing contour maps; an example is given in Figure 10a. Despite a flurry of activity in the generation and visualization of terrain data from scattered data



(a)

(b)

(c)

Figure 10: (a) Fragment of the initial mesh of Bandai Mountain; (b) Mesh optimized with TD algorithm after 1^{st} iteration; (c) Mesh optimized with TD algorithm after 10^{th} iteration.

points this matter remains a difficult and computationally expensive problem. Traditionally, visualization of elevation data consists of three major steps: tessellation, modeling and rendering. The most commonly used tessellation is Delaunay triangulation. Unfortunately, this method has some serious drawbacks resulting in a confusing image because clearly visible "traces" of triangulation can be observed as it can be seen in Figure 10a. To attenuate this effect, we process the mesh of Bandai Mountain, Japan, with all foregoing techniques and TD algorithm. The results are following: Laplacian smoothing produced 211 inverted triangles; smart Laplacian – 179; angle-based approach – 3673; weighted angle-based approach – 1449; TD algorithm – no inverted triangles during all performed iterations. The fragments of the original mesh and mesh optimized with TD algorithm after first and tenth iterations are presented in Figure 10.

Let us stress that because of introducing into the scheme signed aspect ratio TD algorithm ensures that the resultant mesh will be valid not only after first iteration but also after several iterations.

5.2 Comparison of TD algorithm and previous approach to the problem of surface mesh improvement

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 TD algorithms to the triangular mesh of the cubic surface $z = 4x^3 + 6y^3$. Since we have analytic surface curvatures and normals can be easily calculated at each point of the surface. Curvatures for resultant meshes can be computed numerically [23]. We calculated maximum and average changes in Gaussian 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 ($\Delta_{\max}\alpha_n$, $\Delta_{aver}\alpha_n$ (degree)), maximum and average deviations from the original surface ($E_{\max}$, E_{aver}). To measure the geometric quality of the resultant meshes we again used signed aspect ratio described in Section 2.1.

The data from the Table 7 verify that TD algorithm gives the best results in the sense of preserving surface characteristics.

6. 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.

	Laplacian smoothing	BDS	TD
$\Delta_{\max}\lambda_{Gauss}$	8.48812	4.2132	3.00218
$\Delta_{aver}\lambda_{Gauss}$	0.566505	0.504979	0.395472
$\Delta_{\max}\lambda_{mean}$	2.007724	1.434133	1.075598
$\Delta_{aver}\lambda_{mean}$	0.144465	0.124508	0.095078
$\Delta_{\max}\alpha_n$	4.89858	4.73702	3.46709
$\Delta_{aver}\alpha_n$	1.051849	0.809531	0.656461
$E_{\max}$	0.007234	0.00361	0.00181
E_{aver}	0.001524	0.000491	0.000326

Table 7: Statistics for the mesh of cubic surface processed with Laplacian smoothing, BDS approach and TD algorithm.

	Initial	Laplacian smoothing	BDS	TD
$0 < k < 0.2$	20,05%	4.43%	4.34%	4.83%
$0.2 \le k < 0.4$	24.79%	17.75%	18.83%	19.78%
$0.4 \le k < 0.6$	37.18%	19.24%	20.1%	20.73%
$0.6 \le k < 0.8$	6.72%	20.4%	22,87%	23.1%
$0.8 \le k \le 1$	11.26%	38,18%	33.86%	32.56%

Table 8: Histograms of signed aspect ratio for mesh of cubic surface processed with Laplacian smoothing, BDS approach and TD algorithm..

The procedure has been successfully tested on a number of complex triangular and quadrilateral surface and plane meshes. Different quantitative measures 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.

TD algorithm can be considered as universal approach since it can be applied for any surface or plane triangular or quadrilateral meshes.

An interesting and important problem is to define the number of iterations. The geometric mesh quality is improved after each iteration. On the other hand after each iteration deviation from the original surface is increased. Therefore we need to find some compromise between geometric quality improvement and degree of deviation from the original surface. It seems an interesting idea to construct some error-metrics (for instance, normal-based and curvature-based) for measuring the deviation between original and resultant meshes. We can define some threshold values for these metric and stop iterations when these values are reached. We leave careful study of this problem to future work.

7. ACKNOLEDGMENTS

Authors would like to thank Professor A. G. Kushnirenko (Scientific Institute of System Analysis of the Russian Academy of Sciences, Russia) for his support and encouragement, which were very important during our work at this paper.

8. REFERENCES

[1] H. Hoppe, T. Duchamp, J. McDonald, W. Stuetzle, Mesh Optimization, *Computer Graphics (SIGGRAPH 93)*, pp. 19-26 (1993)

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

[3] L. Alboul, R. van Damme, Polyhedral metrics in surface reconstruction: tight triangulations, *The Mathematics of Surfaces VII*, T. Goodman and R.Martin (eds.), Clarendon Press, Oxford, pp. 309-336 (1997)

[4] N. Dyn, K. Hormann, S.-J. Kim, D. Levin, Optimizing 3D Triangulations Using Discrete Curvature Analysis, *Mathematical Methods for Curves and Surfaces,* Vanderbilt Univ. Press Innovations in Applied Mathematics Series, pp. 135-146 (2001)

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

[6] 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*, pp. 281-297 (1986)

[7] 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,* pp. 409-419 (1995)

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

[9] L. A. Freitag, On Combining Laplacian and Optimization-Based Mesh Smoothing Techniques, *AMD Trends in Unstructured Mesh Generation,* Vol. 220 pp. 37-43 (1997)

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

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

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

[13] 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)

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

[15] 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 pp. 401-420 (2000)

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

[17] K. Hormann, U. Labsik, G. Greiner, Remeshing Triangulated Surfaces with Optimal Parameterizations, *Computer Aided Design,* 33(11):779-788 (2001)

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

[19] P. J. Frey, P. L. George, Mesh Generation, *Hermes Science Publishing,* Oxford & Paris (2000)

[20] P. P. Pebay, Planar Quadrangle Quality Measures: Is There Really a Choice, *Proceedings of the 11th International Meshing Roundtable,* pp. 53-62 (2002)

[21] A. Kuprat, A. Khamayseh, D. Goerge, L. Larkey, Volume Conserving Smoothing for Piecewise Linear Curves, Surfaces, and Triple Lines, *Journal of Computational Physics,* 172, pp. 99-118 (2001)

[22] V. Surazhsky, C. Gostman, High Quality Compatible Triangulations, *Proceedings of the 11th International Meshing Roundtable,* pp. 183-192 (2002)

[23] G. Farin, Curves and Surfaces for Computer Aided Design, 4th edition, *Academic Press,* Boston (1997)