Adaptation of Edges in a Triangular Mesh

Miroslav Marić, Sana Stojanović

School of Mathematics, University of Belgrade, Serbia
E-mail: maricm@matf.bg.ac.yu, sana@matf.bg.ac.yu

Abstract: Reconstruction of 3D surface that is given by a set of points that lie near that surface can be performed by construction of a triangular mesh. Accuracy of a mesh obtained in this way can be rather imprecise. That gives us a motivation for transformation of the mesh.

1 Introduction

Goal of this paper is to describe one method for optimization of an initial triangular mesh that approximates surface in 3D space. Surface is given by an unorganized set of points. Initial mesh is formed by approximation of a signed distance function and applying 'marching cubes' algorithm which is described in [5]. The mesh obtained in this way can be imprecise. After adaptation of triangle edges of the mesh we can get more precise approximation of the initial surface. In this paper we will describe optimization of the mesh through adaptation of edges.

2 Problem Formulation

Mesh obtained in an initial approximation consists of a dense grid of triangles that approximates the surface. Triangles obtained by the 'marching cubes' algorithm are approximately the same size, because they were formed within cubes of the same edge (that equals parameter value K). This gave us a restriction regarding number of triangles and their sizes. Optimization will give us an opportunity to modify these values. Our goal is that, in the collection of all meshes that approximate the surface and are the same topological type as the initial mesh, we choose the one that has an ideal balance between number of triangles and a quality of the approximation.

3 Energy of the Mesh

Optimization of the mesh is performed by minimization of 'energy' function that reflects characteristics of the mesh. By characteristics we mean size and quality of the mesh i.e. degree of deviation from the points to the initial set.

Energy of the mesh is defined as a sum:

$$E = E_{dist} + E_{rep} + E_{scope}$$

where E_{dist} represents 'Distance energy' and measures preciseness of an approximation, i.e. deviation of points from the initial set to the mesh. Distance energy is defined as a sum:

$$E_{dist} = \sum_{i=1}^{n} d^2(x_i, U)$$

'Representation energy' E_{rep} measures size of the mesh. It is defined as a multiplication of numbers of triangles - m and a variable T that allows us to define size of the mesh that is required, i.e. to define ratio of preciseness and size of the mesh during optimization:

$$E_{rep} = T \cdot m$$

As the value of T rises, it will increase the 'cost' of triangles and during minimization we will aspire to minimize number of triangles and vice versa, if value of T reduces, during minimization of energy function stress will be on better approximation, i.e. minimization of distance energy E_{dist}.

Although, during formulation of the approximation function, it appeared that these two energies will be sufficient for optimization, experience has showed us that it is not. During minimization of sum $E_{dist} + E_{rep}$ unexpected results aroused, triangles appeared in places where there are not any points, which is the result of the fact that this kind of function perhaps does not have a minimum. Adding the third member to the sum, 'Scope energy' E_{scope}, guaranties existence of the minimum [6].

Scope energy is defined as:

$$E_{scope} = \sum_{(i,j)\in Edges} O \cdot \left\| t_i - t_j \right\|^2$$

where t_i represents vertices of triangles of the mesh, and *Edges* represents set of pairs of vertices that form an edge.

Participation of this energy in the minimizing function is regulated by a parameter *O* which, like in previous case, effects triangle sizes.

4 Minimization of Energy Function

Optimization of an initial mesh is reduced to the minimization of the energy function. Minimization is performed by applying two algorithms alternately:

- Optimization of the vertices positions
- Optimization of the triangle edges

These algorithms are performed until energy function reaches its convergence.

5 Optimization of the Vertices Position

Primary goal of this step is to find, for fixed triangles and mesh edges, optimal positions of vertices so that energy function reaches its minimum. As the number of triangles in the mesh remains the same, representation energy will not change so optimization is reduced to minimizing sum: $E_{dis} + E_{scope}$. In order to do that, we need to find projections of the points to the mesh so that we could calculate distance energy. First, we need to find projections of a point to all planes that contain triangles and then choose point x_i' which is neerest to the initial point and belongs to a triangle. Projections are performed only on triangles that belong to the surrounding of the projecting point.

When we find projections x_i', next step is to find barricentric coordinants for this point in terms of triangle which it belongs to, and then, for fixed barricentric coordinants, find optimal vertices positions. This is accomplished by minimizing a linear system [4].

6 Optimization of the Triangle Edges

Optimization of the triangle edges is used to reduce number of triangles in the mesh, improve approximation of the mesh and reconstruct parts of the surface that are not covered by the initial mesh. This is performed by one of the following operations:

- Edge elimination
- Edge exchanging
- Edge splitting

These steps will be illustrated in the following example. Part of the mesh containing edge AB is shown in Figure 1.

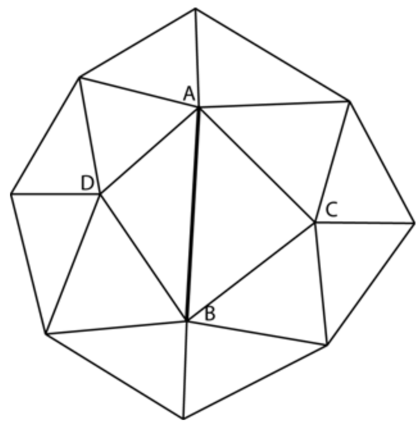

Figure 1
Selected edge of the mesh

‘Elimination of the edge AB’ represents gathering edge into a point, which is shown in Figure 2, and its goal is mesh reduction. Position in which the edge is gathered is usually point A or point B, or middle of edge AB, depending on which point produces maximum energy reduction.

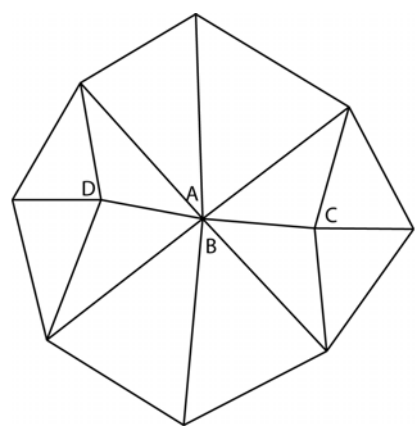

Figure 2
Elimination of the edge AB

‘Edge exchanging’ means substitution of edge vertices with vertices that belongs to the neighborhood triangles as shown in Figure 4. This operation does not change number of triangles, but it can reduce total sum of energies.

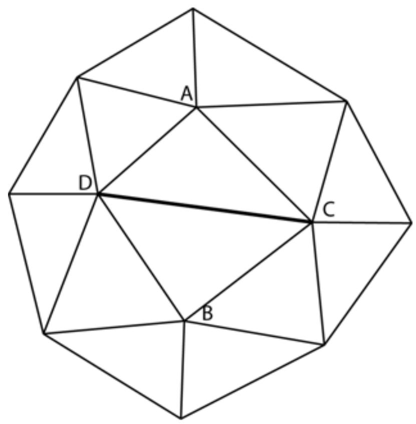

Figure 4
Edge exchanging

‘Edge splitting’ is performed by adding one vertex in a middle of the edge and two new edges in neighborhood triangles as shown in Figure 3. This transformation my lead to more complex mesh and rose of the representation energy, but as a result it my have reduction of the other two energies.

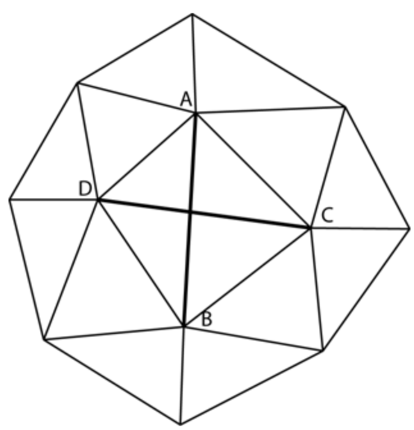

Figure 3
Edge splitting

In order to achieve better result whith optimization of the triangle edges and to speed up algorithm, after each operation local optimization of vertices position is performed.

During optimization of the triangle edges we must not change topological type of a mesh. Before performing each operation we must verify that operation is regular, i.e. it does not change type of a mesh. Therefore we have some limitations to these operations:

- Edge splitting is always regular operation and it can be used without any verification
- Edge exchanging is regular operation if and only if an edge that is added does not already belong to the mesh
- Elimination of the edge is regular operation if eliminated edge satisfies these conditions:
  - If vertices of the edge are boundary than edge itself is boundary too
  - All vertices connected to the edge, i.e. its vertices, with the edge form triangles that belong to the mesh.

In addition, edge is boundary if she belongs to just one triangle and vertex is boundary if there exists boundary edge that contains it.

7 Adaptation of Triangle Edges

Adaptation of triangle edges is performed step by step. First we choose edge AB and we try to apply edge elimination. If it is not a regular operation or it does not reduce energy function, we will try to apply edge exchanging. If this operation neither reduces energy function then we will try to apply edge splitting, which is an operation with least contribution to the minimization of energy function. If no operation contributes optimization, no change is applied and we move on to the next edge.

Edges are chosen from *the set of candidates*. First edge is picked by chance and if any operation can be applied than neighborhood edges are added to the set of candidates. Next edge is picked randomly from the set of candidates. After each applied operation neighborhood edges of the picked edge are added to the set. Process ends when there are no more edges in the set.

During attempts to change the mesh we need to check whether operation contributes reduction of the energy function or not. In order to do that we need to observe that when operation is performed on one specific edge all we need to do is to measure energy in the neighborhood of that edge because rest of the mesh is unchanged and therefore its energy is unchanged. So, when we measure change in energy, we use space division which enables us to find region in which the change occurred. Before applying any operation on an edge first we need to extract neighborhood cubes of space. Then we simulate applying an operation and calculate energy of the mesh in a separated surrounding of the edge. If calculated energy is less than an old one (before the change) operation is applied. Also, after each operation, we need to perform local optimization of vertices position in a surrounding of the edge which quickens an algorithm significantly.

Conclusion

From the initial meshes shown in the left column of the Figures 5-7. This method produces optimized meshes shown in the right column. As we can see, mesh obtained in this way is smaller, more precise and it represents initial surface with more plausibility.

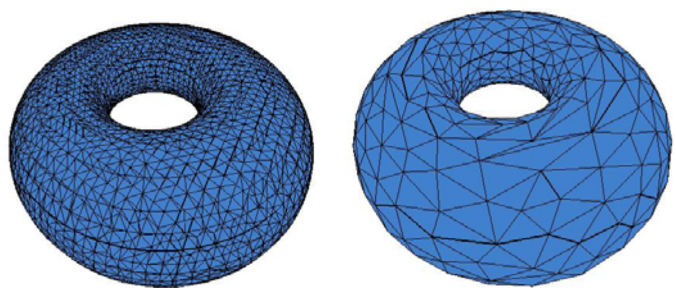

Figure 5
Torus

Inital mesh consists of 13960 vertices, 41880 edges and 27920 triangles

Optimized mesh consists of 479 vertices, 1437 edges and 958 triangles

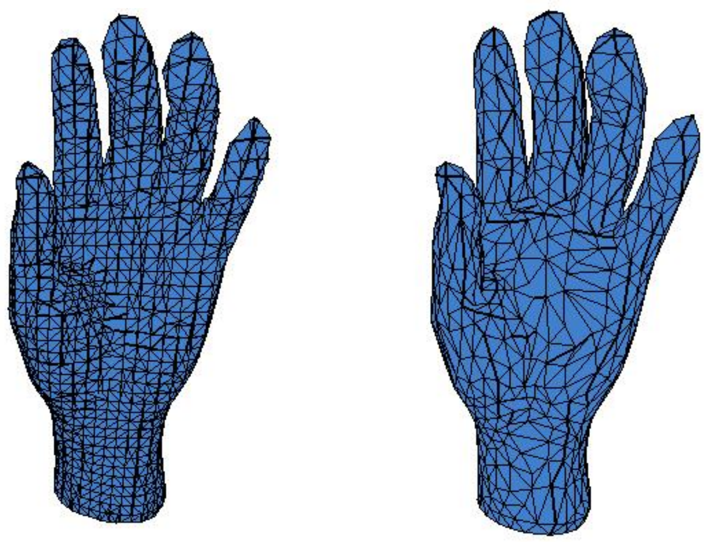

Figure 6
Hand

Inital mesh consists of 4408 vertices, 13218 edges and 8812 triangles

Optimized mesh consists of 812 vertices, 2430 edges and 1620 triangles

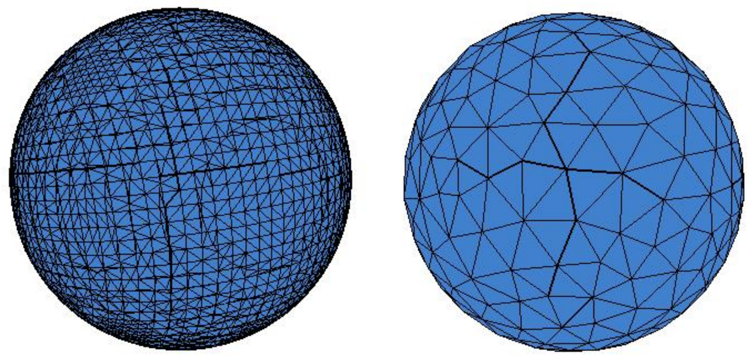

Figure 7
Sphere
Inital mesh consists of 7938 vertices, 23808 edges and 15872 triangles — Optimized mesh consists of 377 vertices, 1125 edges and 750 triangles

Bibliography

[1] Ž. Mijajlović, B. Jovanovoić, F. Marić, M. Marić: MATLAB Toolbox for Analysis of 3D Images, Digitization Preservation of Cultural Heritage (DPCH) Minisymposium, MASSEE 2003

[2] M. Marić, F. Marić, Ž. Mijajlović, B. Jovanović: Automatic Construction of Surface Model, 3rd Serbian-Hungarian Joint Symposium on Intelligent Systems, 2005

[3] M. Marić, M. Tuba: Initial Surface Reconstruction of 3D Scanned Data, PAMM Conference PC-151 at Balatonalmádi, 2006

[4] Miroslav Marić: Optimization of triangulated surface, ETRAN 2006

[5] William E. Lorensen and Harvey E. Cline: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics (Proceedings of SIGGRAPH '87), Vol. 21, No. 4, pp. 163-169

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle: Mesh optimization, Computer Graphics (SIGGRAPH '93 Proceedings), pp 19-26, August 1993