volume data ,triangle mesh generation, mesh smoothing, marching tetrahedra

Rafał STĘGIERSKI*, Paweł MIKOŁAJCZAK*

LAPLACIAN MESH SMOOTHING FOR TETRAHEDRA BASED VOLUME VISUALIZATION

In this paper an improved method for three-dimensional (3-D) surface reconstruction from computed tomography slices is presented. Marching cubes (MC) is a popular method for extracting iso-surfaces from discrete (3-D) data. We implemented a simplification of marching cubes, commonly known as marching tetrahedrons (MT).

1. INTRODUCTION

Displaying scanned medical data poses an interesting problem in computer graphics [4,7]. A volume dataset consists of point in E^3 space, with one or more scalar or vector sample values associated with each point. Volume data are available from many kinds of sources, for example, scanned by MRI (Magnetic Resonance Imaging) or CT (Computed Tomography). Visualization is a powerful technique to enable us to view (3-D) structure from these images. There are two main approaches in visualization: surface and volume rendering. Direct volume rendering (DVR) algorithms map elements directly to the image space. Surface-fitting (SF) algorithms use surface primitives such as polygons (typically triangles) or patches to represent the iso-surface. Iso-surface extraction is a fundamental technique of scientific visualization and one of the most useful tools for visualizing volume data. The predominant algorithm for iso-surface extraction, marching cubes (MC), computes a local triangulation within each voxel of the volume containing the surface. MC takes as input a (3-D) array of density data values, thus representing the scanned object. At each point (x,y,z) in the data, the density values represents the density of the object at the corresponding location in (3-D) space. The algorithm divides the data set into unit cubes with density values at each vertex, and considers each such cube in the data set, thus “marching” through the data. MC method is based on voxel vertices classification. It is known that there are 256 possible cases but they can be reduced to 15 fundamental. Marching cubes is a simple method however, it does not guarantee the surface to be topologically consistent with the data, and creates triangulations which contain many triangles of poor aspect ratio. Marching tetrahedra [1,5,8] is a variation of marching cubes, which overcomes this topological problem. Improvement in triangle aspect ratio has

* Laboratory of Information Technology, UMCS, pl. Marii Curie-Skłodowskiej 1, 20-031 Lublin, Poland

generally been achieved by mesh simplification, a group of algorithms designed to reduce the large number of triangles.

2. MESH GENERATION

Marching tetrahedra (MT) method is based on a decomposition of the voxel space to tetrahedronal space. It gives generally only two fundamental cases as the tetrahedron can be intersected by an iso-surface in three or four points. Tetrahedra are the simplest volume cells possible and their use leads to a great amount of flexibility. For an introduction and survey of tetrahedrization tools we refer the reader to [2,3,6]. For cubic lattices three fundamental cases can be distinguished: cubic primitive, body-centred cubic and face-centred cubic. There are two fundamental decomposition of cubes: five tetrahedral scheme and six tetrahedral scheme. Decomposing a cube into five or six tetrahedral is shown on Fig.1

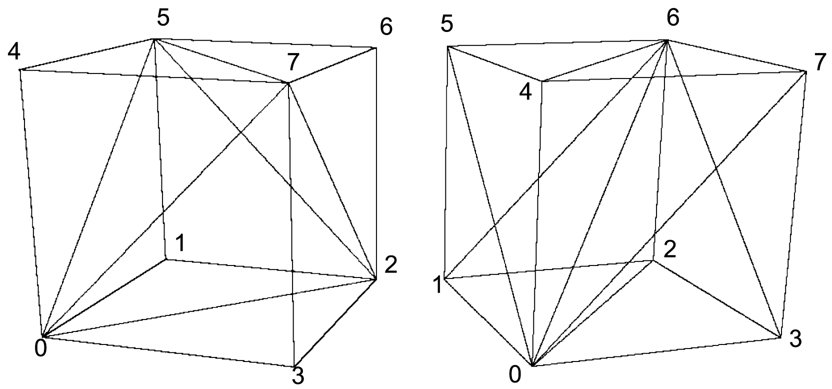

Fig.1. Decomposing a cube into five or six tetrahedral

All possible tetrahedra are listed in Table 1.

Tetrahedra number	6 tetrahedra	5 tetrahedra
1	0,2,3,6	0,2,3,7
2	0,1,2,6	0,1,2,5
3	0,1,5,6	0,4,5,7
4	0,3,6,7	2,5,6,7
5	0,4,6,7	0,2,5,7
6	0,4,5,6	

Tab.1. Possible decomposition to 5 and 6 tetrahedra

The effect of decomposition method on algorithm efficiency was tested. Both implementation of a marching tetrehedra algorithms are very fast. An important aspect when comparing two methods is the time for generating the iso-surface and number of generated triangles. Table 2 presents some experimental results.

Volume	Splitting method	Generation time	Number of vertices
128x128x128	5 tetrahedra	0.97 s	1631460
128x128x128	6 tetrahedra	1.25 s	2027184

Tab.2. Mesh generation time

Given a volume dataset described by a tetrahedra, the isosurface passing through the points of the volume dataset having value *i* (iso-level density) can be reconstructed by using a per cell approach similar to marching cubes algorithm. We therefore adopted the program by considering the intersection of the surface with tetrahedra instead of cubes. We now use the surface intersection to determine the polygons that lie on the surface. To obtain the coordinates of these polygons we linearly interpolate the coordinates at the vertices of the tetrahedron based on the density values. To find a position of vertices we tested two methods.

The first method is based on following formula:

$$X = X_1 + \frac{i - v_1}{v_2 - v_1}\left(X_2 - X_1\right)$$

where X=(x,y,z) is a new vortex position, X_1 and X_2 are old tetrahedra vertices, v_1 and v_2 are voxel intensities and *i* is a iso-level intensity. The value of scaling parameter f:

$$f = \frac{i - v_1}{v_2 - v_1}$$

changes linearly from 0 to 1. With f = 0.5 we have another approximation, which give us the second method :

$$X = X_1 + f\left(X_2 - X_1\right)$$

Both methods produces a set of triangles that approximate our surface which we than use to display the surface. Results of surface rendering is shown on Fig. 2

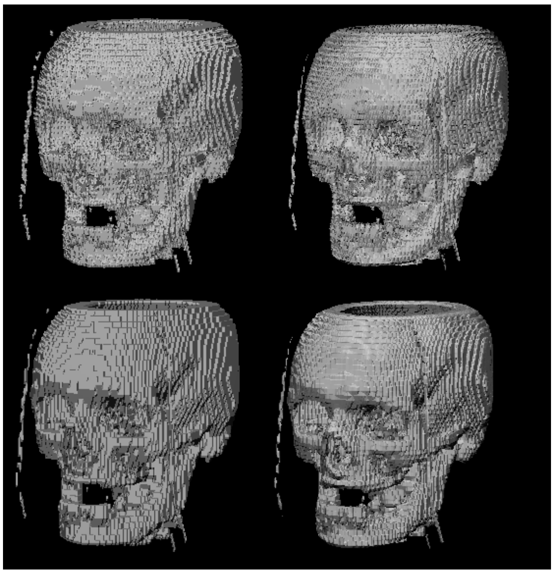

Fig.2. Reconstruction results: 5 tetrahedra with linear f, 5 tetrahedra with f= 05,
6 tetrahedra with linear f, 6 tetrahedra with f=0.5

3. MESH SMOOTHING

Local mesh smoothing algorithms [2,9] have been shown to be effective in repairing distorted elements in automatically generated meshes. The simplest such algorithm is Laplacian smoothing, which moves grid points to the geometric center of incident vertices. To smooth the generated mesh we can use the relaxation method as for example in order to reduce the tension in net of springs. The goal, through unrealistic, of Laplacian smoothing is to relocate the internal nodes of the mesh so that the elements are equilateral. The boundary nodes are fixed due to problem constrains. However, the internal nodes can be relocated.

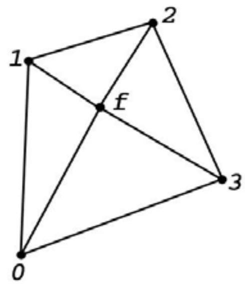

Fig.3. Selected and topologically connected vertices

For each internal node, the node is relocated to the geometric center (Fig.3), or centroid, of the polygon comprised of elements containing the internal node. Typically, two passes through the internal nodes are needed to significantly improve the mesh. The new location x_{fnew} of the free vortex is calculated as:

$$x_{fnew} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where x_i are positions of adjacent vertices. Practically, to generate visually acceptable mesh, five to ten passes through the internal nodes are needed.

4. IMPROVEMENTS AND ACCELERATIONS

Because of increased computational speed, typically generated meshes contain millions of elements. Mesh smoothing procedure in its straightforward implementation is obviously very time consuming and highly inefficient. Additional improvements have to be performed to accelerate the whole smoothing process. A single mesh triangle is represented by three vertices, but most of them are common for more than one triangle.

Base volume	Number of vertices	Number of unique vertices	Ratio
32x32x32	44880	7415	0.17
64x64x64	215616	35888	0.17
128x128x128	1013592	171819	0.17

Tab.3. Number of vertices and unique vertices in generated mesh

Calculation of relaxed mesh is all done on unique vertices (common vertices). Each vortex is indexed. The finding process of the unique vertices is not linear. In order to speed up classification process, we create small, dynamically allocated tables with sorted vertices. Unique vertices are sorted with respect to x-coordinate. We find that the optimal number of such sub-tables is from 150 to 250, see fig.4. Finally we have the table of triangles with vertices represented by index. During the smoothing operation the algorithm uses information about all topologically connected vertices, and finds it searching list of all triangles.

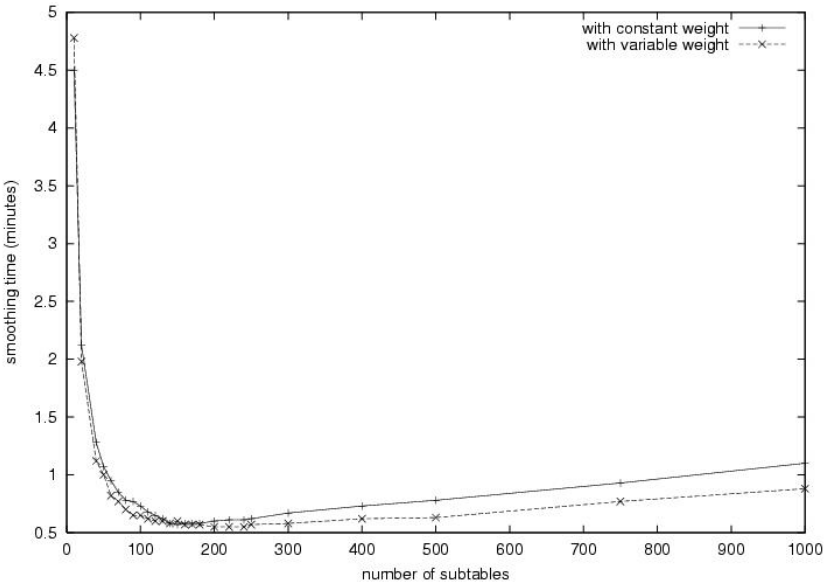

Fig.4. Number of subtables and time of mesh smoothing
(five passes, 337863 triangles)

5. EXPERIMENTAL RESULTS

The whole code has been written in C++ language. The GUI consists of a collection of Qt-3.0.x library widgets. Thanks to this it is a cross-platform software. Currently it is running on a PC (Pentium III, 600 MHz, 256 MB RAM, SO - RedHat Linux 7.x) and a SGI O2 workstation. The proposed algorithm was used to display complex surfaces after triangular mesh smoothing. The results have proved that the implemented smoothing algorithm is applicable, see fig. 5 and fig.6.

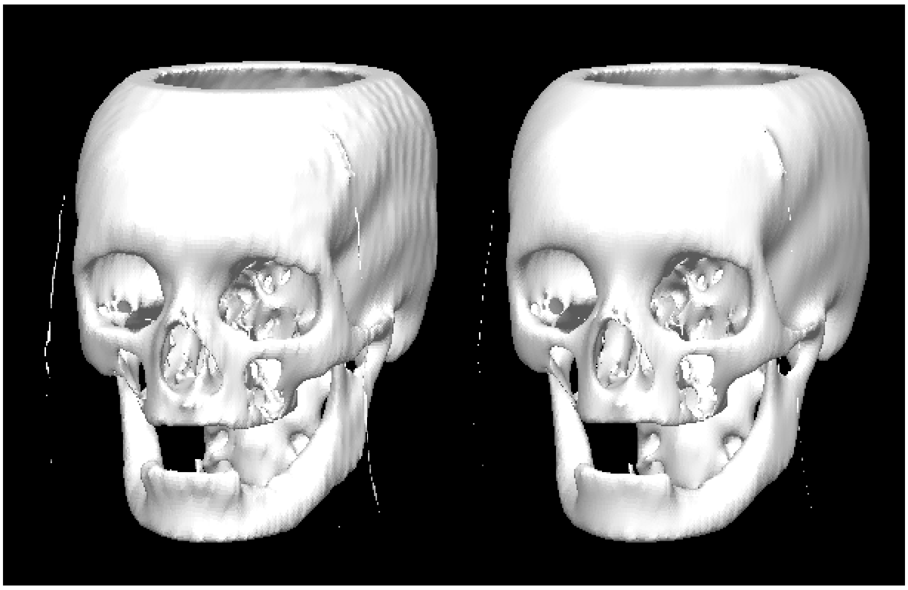

Fig.5. Result of smoothing: 5 and 10 passes, 337 863 triangles, 250
subtables, generation and smoothing time: 37.6s and 39.85s

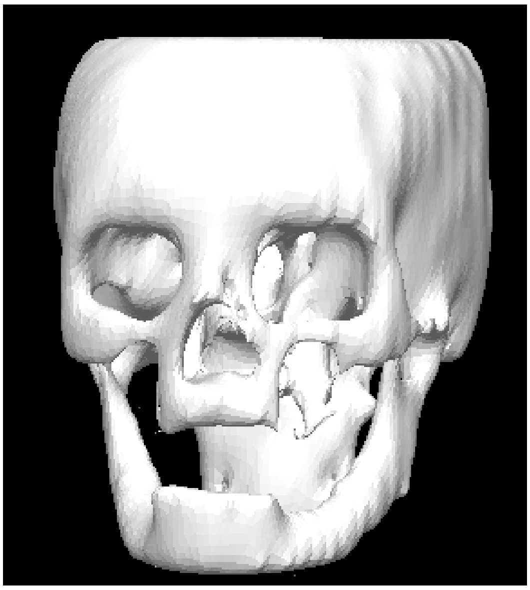

Fig.6. Result of smoothing: 5 passes, 71795 triangles, 250 subtables
generation and smoothing time: 4.58s

6. CONCLUSIONS

A comparison of the marching tetrahedra (MT) algorithm with the decomposition into six and fife tetrahedra has been tested. When is the time of generation or number of triangles measure of quality, the decomposition to 6 tetrahedra gives better result. Results shown that the modified marching tetrahedra algorithm significant reduces the number of polygons generated and thus increase the computation efficiency for surface generation and rendering.

BIBLIOGRAPHY

[1] CIGONI P., MONTANI C., SCOPIGNO R., Tetrahedra Based Volume Visualization, Mathematical Visualization - Algorithms, Applications, and Numerics, H.-C. Hege and K. Polthier (Eds.), Springer, Verlag, ISBN 3-540-63991-8, pp. 3-18,1998

[2] FREITAG L.A., On Combining Laplacian and Optimization-based Mesh Smoothing Techniques, Trends in Unstructured Mesh Generation, ASME Applied Mechanics Division, Volume AMD-Vol 220, pages 37-44, 1997

[3] FREITAG L.A., Local Method for Simplicial Mesh Smoothing and Untangling, ANL Technical Report, Number ANL/MCS-TM-239, Argonne National Laboratory, Argonne, Illinois, March 1999

[4] LOPES A.M., Accuracy in Scientific Visualization, PhD thesis, School of Computer Studies, University of Leeds, 1999

[5] NEUGEBAUER P.J., KLEIN K., Adaptive Triangulation of Object Reconstructed from Multiple Range Images, In IEEE Visualization '97, Late Breaking Hot Topics, Phoenix, Arizona, October 20-24, 1997

[6] NIELSON G.M., Tools for triangulations and tetrahedrizations and constructing function defined over them, in: Scientific Visualization: Overviews, Methodologies, and Techniques, G.M. Nielson, H. Mueller, and Hagen, eds., IEEE Computer Society Press, 1997, pp. 429-525

[7] QUERESHI F.Z., Constructing Anatomically Accurate Face Models using Computed Tomography and Cyberware data, Master's Thesis, Department of Computer Science, University of Toronto, Toronto, Canada, January, 2000

[8] TREECE G.M., PRAGER R.W, GEE A.H., Regularized marching tetrahedra: improved iso-surface extraction, CUED/F-INFENG/TR 333, 1998
[9] ZHOU T., SHIMADA K., An Angle Based Approach to Two-dimensional Mesh Smoothing, Proceedings, 9th International Meshing Roundtable, Sandia National Laboratories, pp.373-384, October 2000

This study has been partially financed by the Polish State Committee for Scientific Research (research project no. 7T11E01921)