A Novel Biomedical Meshing Algorithm and Evaluation based on Revised Delaunay and Space Disassembling

Xiao Yu, Lixu Gu*, Sizhe lv, Jie Liu, Pengfei Huang and Xianming Kong*

***Abstract*—The tetrahedral mesh generation part in Finite Element Method (FEM) of soft tissue simulation is difficult to be realized by traditional mesh algorithms because of the requirements of boundary preservation and quality of all tetrahedra. Aiming to meet the real-time requirement of FEM, we propose a revised Delaunay algorithm with many improved methods, including background gird, random point disarrangement, radial method and visibility check.**

In this paper, two tetrahedral mesh generation algorithms including Space-Disassembling and the revised Delaunay algorithm, are presented based on different mesh requirements. And a comparison of Space-Disassembling Mesh Algorithm, traditional Delaunay algorithm and the revised Delaunay algorithm is processed based on some pivotal criteria.

I. Introduction

Tetrahedral mesh generation algorithm is an important prerequisite of many soft tissue simulation methods, including Mass-Spring, Center-line and Finite Element Method (FEM). Nowadays, aiming to achieve virtual reality, the real-time requirements of simulation have been considered as key parts [1]. FEM which based on a complete mathematics theory can simulate the deformation of soft tissue more accurate, but is also more difficult to meet the real-time requirement because of the time complexity of matrix computation, which is the pivotal part of this method.

In order to improve the computation, the quality of tetrahedral mesh of the soft tissue becomes more important. After research and comparison of several mesh algorithms including Space-Disassembling, Delaunay algorithms and Advancing Front Technology [2], an Space-Disassembling Algorithm and a revised Delaunay algorithm are chosen to realize the discretization of FEM because the first one is very efficient and leads to good mesh inside the soft tissue, while the second one makes a good balance of boundary preservation, quality of tetrahedra and time complexity. In order to make the Delaunay algorithm qualified for the FEM requirements, many improved methods including point random disarrangement, radial method and visibility check, are designed and implemented in the revised Delaunay algorithm to improve its performance.

Manuscript received April 16, 2007. This work was supported in part by National natural science foundation of China, grant No. 70581171, and the Shanghai Municipal Research Fund, grant No. 06dz15010.

Xiao Yu, Lixu Gu (e-mail: gu-lx@cs.sjtu.edu.cn), Sizhe Lv, Jie Liu & Pengfei Huang are with the Laboratory of Image Guided Surgery and Therapy (IGST), Shanghai Jiao Tong University (SJTU), China.

Xianming Kong is with Renji Hospital of Medical School, SJTU. China.

*Lixu Gu & Xianming Kong are Corresponding Authors.

II. Space-Disassembling Mesh Algorithm

Space-Disassembling Algorithm is an intuitive mesh algorithm with a low time complexity [3]. The input of this algorithm includes boundary points and the boundary triangle facets of the object. The output includes all the points and the tetrahedra in the mesh. The algorithm could generate very nice mesh inside the object which are mainly regular or equicrural tetrahedra, but doesn't perform very well on the surface. And another problem is the algorithm can not preserve all the boundary information including points and triangle facets. It could only preserve the topology of the original object in rough. We used this algorithm at the beginning of the FEM research, and it worked very well when the boundary preservation was not a pivotal requirement.

The first step of this algorithm is cutting the bound box of the original object into small cubes, the number of which could be described as the granularity of the mesh. All of the small cubic elements could be divided into three categories: cubes outside the object, cubes inside and cubes on the surface. Exterior cubes should be abandoned and the remain

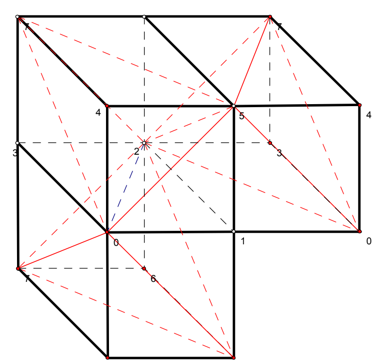


Fig. 1. Cut methods of two adjacent cubes should be symmetrical, in order to eliminate the creation of stationary points.

cubes should be cut into five tetrahedra in the similar way (Fig. 1) which leads to nice mesh inside the object, and the cut methods of two adjacent cubes should be symmetrical, in order to eliminate the creation of stationary points.

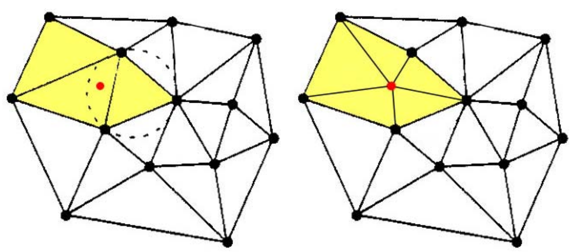
After re-mesh of the boundary tetrahedra, a simple mesh of the original object is generated by the Space-Disassembling Mesh Algorithm.

III. Delaunay Algorithm

Space-disassembling algorithm worked well in the beginning of the FEM research. However, after the boundary preservation and good quality of all tetrahedra become more and more important, Delaunay algorithm turns out more suitable and effective.

Delaunay algorithm is the general name of all algorithms whose mesh results accord with the Delaunay criterion raised by B. Delaunay in 1934, which is based on Voronoi diagrams (also known as Dirichlet tessellations) [4]. This criterion states that a circum-sphere of each simplex in a triangulation contains only the n+1 defining points of the simplex (n represents the number of dimension of the input data).

There is a basic concept in all Delaunay algorithms called Delaunay core of point P, which represents a set of tetrahedra in the mesh whose circum-spheres contain point P (Fig.2.a). According to the Delaunay theory, point P and the Delaunay core of this point are the part of the mesh which does not meet the Delaunay criterion. In order to eliminate this inconsistency, a reconstruction of the Delaunay core and the point is necessary, for which we employed the point-insertion method, one of the most efficient approaches in Delaunay algorithms, to break up all the tetrahedra inside the Delaunay core, and join the new point and the surface of the Delaunay core together to generate a new mesh (Fig.2.b). After this reconstruction, the new point has been inserted to the original mesh successfully, and the mesh still meets the Delaunay criterion.



(a) (b)

Fig. 2.a. Example of Delaunay core of point P represented in 2 dimensions

Fig. 2.b. New mesh could be constructed with the surface of Delaunay core and point P.

IV. Revised Delaunay Algorithm

In order to generate better tetrahedral mesh in FEM, we propose a novel Delaunay algorithm with many improved steps and methods, which optimize the mesh result prominently on the boundary preservation and quality of all tetrahedra. The entire process of seven steps of this algorithm leads to better mesh compared with Space-Disassembling Algorithm and traditional Delaunay algorithms separately.

A. Initial tetrahedral mesh construction

The pivotal part of the revised Delaunay algorithm is iteratively inserting new point into current mesh. So an initial tetrahedral mesh which contains the input object should be constructed first. Considering the optimization of the following steps, we choose an approach as follow.

First, the circum-sphere of the object's bound box, which is a cuboid, should be calculated. Then, the bound box of the circum-sphere could be calculated, which should be a cube. After that, we mesh the bound box of the circum-sphere into five tetrahedra as in the Space-Disassembling Algorithm. Then the initial tetrahedral mesh which contains the original object is constructed completely.

B. Presetting interior points generation

In this step, possible points which could be inserted into the initial mesh would be prepared. The boundary points in the input data should be contained in the mesh considering the boundary preservation requirement, but only inserting boundary points into the initial mesh is far from enough to generate nice mesh. In order to make the mesh more regular, points inside the original object should be generated as well, the method of which is to cut the bound box of the circum-sphere which is calculated in the first step into smaller cubic elements, as we did in the Space-Disassembling Algorithm. Then all vertices of the cubic elements would be the presetting interior points which could be inserted into the mesh in the following steps (Fig. 3).

C. Presetting interior points random disarrangement

The presetting interior points generated in the second step are the vertices of the cubic elements which are all on special positions. According to Cavalcanti and Mello [5], points on special positions, like more than four points co-sphercity and more than three points co-planarity, could cause the failure of the algorithm easily. A research about points on special positions was processed after we implemented this revised Delaunay algorithm. We chose a kidney data which contains 458 boundary points as an input. First, we turned off this random disarrangement step and 5551 tetrahedra were generated. Then, we turned on this step, and more than 6590 tetrahedra were generated based on the same input, which means more points could be inserted into the mesh to form more tetrahedra after random disarrangement.

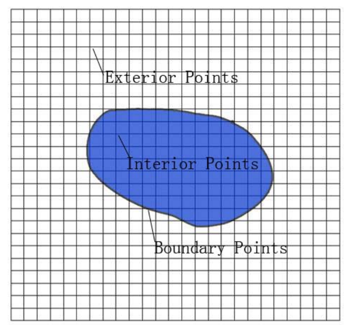


Fig. 3. Cut the bound box into cubic elements to generate the presetting interior points. All vertices of the small cubes should be stored and the real interior points could be separated in the following steps by radial method.

The method to implement this step is very intuitive. A random vector with the value about 10^{-4} based on the input data would be added on all presetting interior points, which solves the problem of points on special positions effectively.

D. Interior points generated based on radial method

In this step, real interior points would be separated from the output of the third step, which contains both points inside and outside the source object. The method we employed to separate them is called radial method which is a classical method to solve this kind of problem in two dimensions.

The pivotal part of this method is quite straightforward. If a radial goes through an object, several points of intersection would be generated (Fig. 4). An assistant variable denoted as Counter with an initial value 0 could help to record the relative position of each section of the radial. When the radial goes into the object from the outside, the Counter variable would be increased by 1. When the radial goes from the inside of the object to the outside, the Counter variable would be decreased by 1. After this process, the Counter variable could help us separate all points on the radial. In this way, we could separate the points from the output of the third step with a radial going across it, and then all points outside the object should be deleted. All points which would be inserted into the initial mesh, including boundary points and interior points have been prepared.

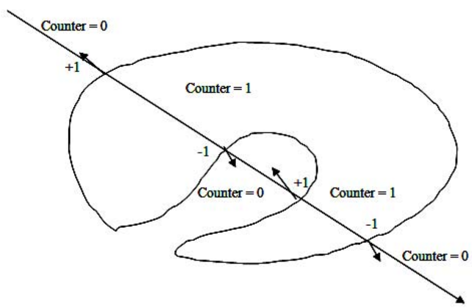


Fig. 4. Radial method in 2 dimensions could record the relative position of each section of the radial, which works well in 3 dimensions too.

E. Delaunay mesh construction

In this step, both boundary points and interior points would be inserted into the initial mesh iteratively as stated above. In this revised Delaunay algorithm, we raise some methods to optimize the traditional point-insertion process, which could reduce the time complexity and eliminate the possibility of interactive tetrahedra generation effectively.

For each point to be inserted, the Delaunay core should be found first. An intuitive way to get the Delaunay core for point P is to go through all tetrahedra in the mesh and check whether it meets the definition of Delaunay core of point P. This process could be very slow, and could cause serious tetrahedra overlap problem (Fig. 5.a). The method to generate the Delaunay core in the revised algorithm is to find the tetrahedron T0 which contains point P first, and then recursively check every tetrahedron which is adjacent to T0 by triangle faces whether the circum-sphere of it contains point P or not until no more tetrahedra could be added into the Delaunay core. The pseudocode of this process could be stated as follow:

```
DelaunayCore dc(Point P);
Tetrahedron T0 = findTetraContainPoint(Point P);
dc.addTetra(T0);
constructDelaunayCore(dc, T0);
//Definition of the function constructDelaunayCore
void constructDelaunayCore(DelaunayCore dc, Tetrahedron T0)
{
    For each triangle face triFace of T0
    {
        Tetrahedron T = findAnotherTetra(triFace, T0);
        If (the circum-sphere of T contains point P)
        {
            dc.addTetra(T);
            constructDelaunayCore(dc, T);
        }
    }
}
```

The background grid technology [6] could help find a set of tetrahedra which may contain point P efficiently. For each tetrahedron in this set, we could construct four tetrahedra with the point P and the four triangle faces of it. Then a comparison of volume of the tetrahedron in the set with the sum of volumes of the four tetrahedra could help find the first tetrahedron in the Delaunay core of point P.

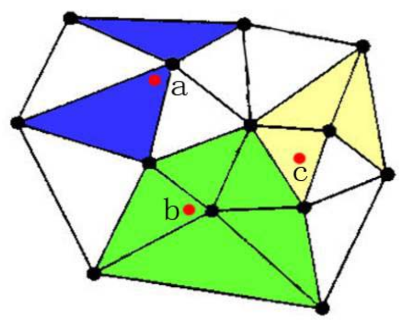


Fig. 5. a. (Blue part) Possible result of intuitive Delaunay core generation could lead to tetrahedra overlap.
Fig.5.b. (Yellow part) Possible result of revised Delaunay core generation could lead to tetrahedra overlap.

Actually, this method of generating Delaunay core does not totally meet the Delaunay theory. By weakening Delaunay theory, this method helps the elimination of time complexity and the possibility of tetrahedra overlap distinctively. The topology of the Delaunay core of point P should be examined before breaking all tetrahedra in it, because bad topology of Delaunay core could lead to tetrahedra overlap as well (Fig. 5.b) [7]. In order to solve this problem, the normal vector of each boundary face of the Delaunay core should be calculated. The direction of the vector would be defined as positive if it points into the core; otherwise, it would be defined as negative. If the point P is on the positive side of all boundary faces of the Delaunay core, breaking tetrahedra in the core would be safe, which means no tetrahedra overlap would be caused. If point P is on the negative side of some boundary faces, tetrahedra containing this face in the Delaunay core should be deleted. Processing this method recursively for each boundary face of the core would make all tetrahedra in the mesh valid without any element overlap, and also could find and eliminate those tetrahedra whose volume equal to zero. After this method, reconstruction of tetrahedra in the core, which is the last thing to do in this step, could be processed as stated above

F. Boundary preservation

In the fifth step of the algorithm, some points may be deleted when the tetrahedra of the Delaunay core are broken

(Fig. 5.c). If the points are boundary points from the input data, it should be recorded and re-inserted into the mesh.

In the point-insertion process, neighbor points would be more easily to be joined together, which may lead to surface topology mistakes (Fig. 6). To solve this problem, we need to

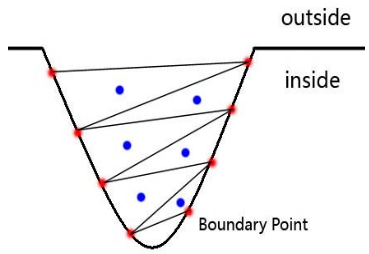


Fig. 6. neighbor-point joining leads to surface topology mistakes of the object which could be solved by remove the tetrahedra whose center points are outside of the object.

process the radial method stated above for the geometric center points of all tetrahedra to separate them into two categories: tetrahedra inside the object and tetrahedra outside of the object. All tetrahedra with exterior centers should be deleted to preserve the basic surface topology of the object.

G. Deal with sliver tetrahedra

Sliver tetrahedra, which would cause failure of the FEM computation easily, could be formed in the last three steps, especially on the surface of the object. Number of sliver tetrahedra is an important part of the tetrahedral mesh benchmark, which makes the elimination of sliver tetrahedra very necessary.

The definition of sliver tetrahedra in this algorithm is based on the standard deviation (SD) of sides in the tetrahedron. First we define a maximum value of sides SD as A0, and then calculate the sides SD for each tetrahedron. If the SD is greater than A0, the tetrahedron should be combined with its neighbor and a re-mesh should be generated until all tetrahedra in the mesh are not sliver.

Table. 1 Comparison of space-disassembling algorithm, traditional Delaunay algorithm and revised Delaunay algorithm

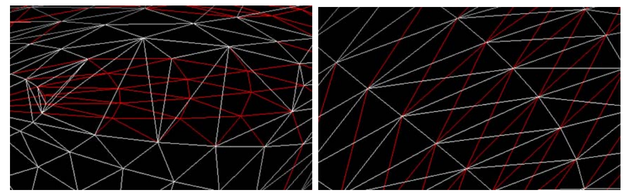
Mesh Criteria	Space disassembling		Traditional Delaunay		Revised Delaunay	
	kidney	breast	kidney	breast	kidney	breast
Output points	431	2065	--	--	538	1629
Output tetrahedra	1303	7052	2151	7480	2100	7753
Time complexity	Good		Medium		Medium	
Sliver tetrahedra	Medium		Medium		Good	
Interior tetrahedra	Good		Good		Good	
Boundary preservation	Bad		Bad		Medium	

The first input object is a kidney surface data containing 458 boundary points and 960 boundary faces. The second input object is a breast surface data containing 994 boundary points and 2816 boundary face.

V. Experiments

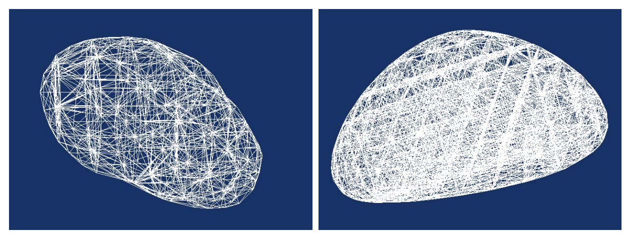
We compared three tetrahedral mesh algorithms, including space-disassembling algorithm, traditional Delaunay algorithm and the revised Delaunay algorithm, based on some pivotal criteria (Table. 1). The algorithm chosen to represent traditional Delaunay algorithm is vtkDelaunay3D in VTK (Visualization Toolkit) [8].

In Fig. 7, a comparison of boundary preservation is processed between vtkDelaunay3D and the revised Delaunay algorithm. Fig.8 is the mesh results generated by the revised Delaunay algorithm of the kidney and breast data.



(a) (b)

Fig. 7.a The white part represents the boundary triangles generated by vtkDelaunay3D. The red part represents the original boundary triangles.
Fig. 7.b The white part represents the boundary triangles generated by the revised Delaunay algorithm. The red part represents the original boundary triangles.



(a) (b)

Fig. 8.a The mesh result of the kidney data by the revised Delaunay algorithm, the granularity of which is 10.
Fig. 8.b The mesh result of the breast data by the revised Delaunay algorithm, the granularity of which is 20.

References

[1] Pengfei Huang, "Real-Time Simulation for Global Deformation of Soft Tissue Using Deformable Centerline and Medial Representation", LNCS 4072, Switzerland, July 2006.

[2] Esdras Medeiros, "A Topological Framework for Advancing Front Triangulation", SIBGRAPI'03, 2003.

[3] NIE Ying, "Virtual Human 3-D Medical Image Processing Based on Finite Elements", Beijing Biomedical Engineering, September 2003.

[4] Will Schroeder, *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*, 3rd Edition, Chapter 9.

[5] Cavalcanti P R, Mellow U T. “Three-dimensional constrained Delaunay triangulation: A minimalist approach”, Proceedings 7th International Meshing Roundtable, Michigan, 1998.

[6] Bonet J, Peraire J. “An alternating digital tree algorithm for geometric searching and intersection problem”, Int. J. Number Methods Eng, 91.

[7] Wang Jianhua, “Delaunay Algorithm and Related Procedures to Generate the Tetrahedron Mesh for an Object with Arbitrary Boundary”, Chinese Jounal of Rock Mechanics and Engineering, May, 2003.

[8] Visualization Toolkit 5.0, Kitware, Inc. http://www.vtk.org