# Novel biomedical tetrahedral mesh methods: algorithms and applications

Xiao Yu*[a], Yanfeng Jin[a], Weitao Chen[a], Pengfei Huang[a], Lixu Gu[a]
[a]Image Guided Surgery and Therapy Lab, School of Software, Shanghai Jiao Tong University, Shanghai, China, 200240

## ABSTRACT

Tetrahedral mesh generation algorithm, as a prerequisite of many soft tissue simulation methods, becomes very important in the virtual surgery programs because of the real-time requirement. Aiming to speed up the computation in the simulation, we propose a revised Delaunay algorithm which makes a good balance of quality of tetrahedra, boundary preservation and time complexity, with many improved methods. Another mesh algorithm named Space-Disassembling is also presented in this paper, and a comparison of Space-Disassembling, traditional Delaunay algorithm and the revised Delaunay algorithm is processed based on clinical soft-tissue simulation projects, including craniofacial plastic surgery and breast reconstruction plastic surgery.

**Keywords:** Tetrahedral Mesh, Soft tissue simulation, Delaunay, Space-Disassembling, FEM, Center line, Mass-spring

## 1. INTRODUCTION

Soft-tissue simulation methods, including Center line [1], Mass-spring [2] and FEM (Finite Element Method) [3], become very popular in virtual surgery programs of biomedical field. Methods like FEM could generate results with high accuracy, but is more difficult to meet the real-time requirement in virtual reality of medical process, because of the time complexity of matrix computation in this method.

After research and analysis, we found the quality of tetrahedral mesh of the soft tissue is one of the bottlenecks in improving the computation speed. In order to generate tetrahedral mesh with good quality, we compared several mesh algorithms including Space-Disassembling [4], Delaunay algorithms and Advancing Front Technology [5], and then propose a Space-Disassembling algorithm and a revised Delaunay algorithm to realize the discretization of the soft tissue methods. The first algorithm 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. And 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.

## 2. SPACE-DISASSEMBLING MESH ALGORITHM

Space-Disassembling Algorithm is an intuitive mesh algorithm with a low time complexity, which could generate very nice mesh inside the object, but doesn't perform well on the surface.

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 (Fig.1). Exterior cubes should be abandoned and the remain cubes should be cut into five tetrahedra in the similar way 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 (Fig.2). After re-mesh of the boundary tetrahedra, a simple mesh of the original object is generated by the Space-Disassembling Mesh Algorithm.

One problem of this algorithm is it cannot 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 Soft-tissue simulation research, and it worked very well when the boundary preservation was not a pivotal requirement. An

MIPPR 2007: Medical Imaging, Parallel Processing of Images, and Optimization Techniques,
edited by Jianguo Liu, Kunio Doi, Patrick S. P. Wang, Qiang Li, Proc. of SPIE Vol. 6789,
67891T, (2007) · 1605-7422/07/$18 · doi: 10.1117/12.749779

experiment which is used in the craniofacial plastic surgery could be found in the Experiments part of this paper. However, after the boundary preservation and good quality of all tetrahedra become more and more important, Delaunay algorithm turns out more suitable and effective.

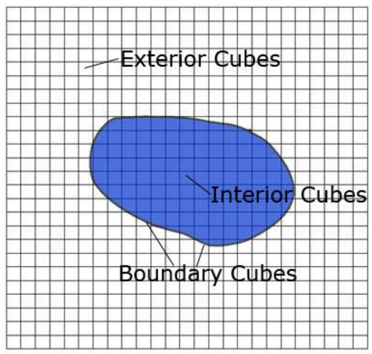



Fig. 1. Three categories of cubic elements after bound box cutting.

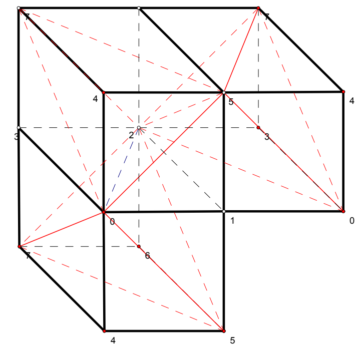


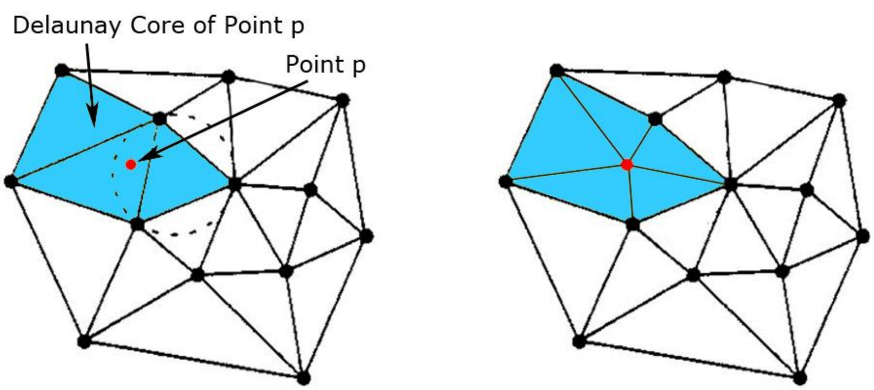
Fig. 2. Cut methods of two adjacent cubes should be symmetrical, in order to eliminate the creation of stationary points.

## 3. REVISED DELAUNAY ALGORITHM

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) [6]. This criterion states that if the 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), the mesh constructed by these simplexes is Delaunay mesh.

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.3.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.3.b). After this reconstruction, the new point has been inserted to the original mesh successfully, and the mesh still meets the Delaunay criterion.

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 nine steps of this algorithm leads to better mesh compared with Space-Disassembling Algorithm and traditional Delaunay algorithms separately.




(a) (b)

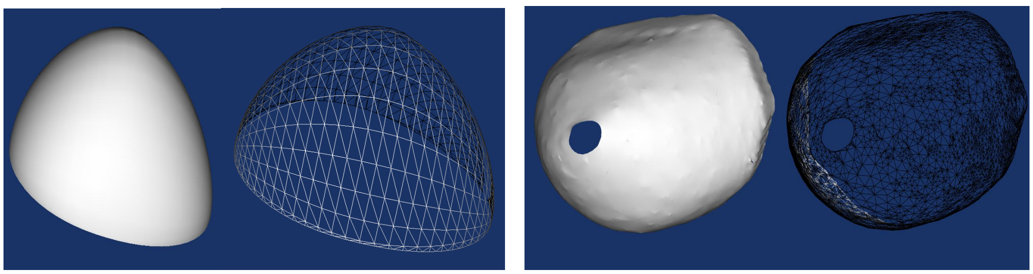
Fig. 3.a. Example of Delaunay core of point P represented in 2 dimensions

Fig. 3.b. New mesh could be constructed with the surface of Delaunay core and point P.

### 3.1 Segmentation and 3D Surface Modeling

The original pixel data from clinical medical images could not be applied into deformation simulation directly. The organs should be separated from other tissues and modeled as 3D Triangle Surface first.

We use the Balloon segmentation algorithm [7] and Butterfly subdivision algorithm [8] to generate the surface data. Balloon segmentation (Fig.4.a), compared with Marching-cube meshing algorithm [9] (Fig.4.b), is a dynamical volumetric segmentation algorithm by approximating a sphere using polygons. The basic idea of balloon algorithm is to add image forces on an initial spherical mesh data, making it expand or shrink towards the surface of soft tissue. The mesh will adjust its shape to conform to the boundary of region of interest as closely as possible after iterating the calculation for specified times, just like a balloon. After this step, a smooth triangle surface with boundary information of organs in the medical images could be generated, based on which, we could check the boundary preservation criteria of all these tetrahedral mesh algorithms.



(a) (b)

Fig. 4.a. Surface generated by Balloon segmentation algorithm and smoothed by Butterfly subdivision algorithm

Fig. 4.b. Surface generated by Marching-cube meshing algorithm and smoothed by Butterfly subdivision algorithm

### 3.2 Surface Triangle Mesh Optimization

If the boundary tetrahedra density and the interior tetrahedra density must match each other, or the boundary tetrahedra density must be larger than that of the interior tetrahedra, considering the deformation of the surface should be more obvious, a surface triangle mesh optimization should be processed before the Delaunay algorithm [10].

The optimization is based on the granularity of the Delaunay mesh. If the surface triangle mesh is sparse while the granularity of the expected mesh is large, respectively. A surface triangle subdivision should be performed before the Delaunay mesh construction. On the other hand, if the surface triangle mesh is dense while the granularity of the expected mesh is small, a surface triangle simplification should be performed before the mesh construction. After this step, the final mesh could be well-proportioned on the boundary and inside the source object.

### 3.3 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.

### 3.4 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.

### 3.5 Presetting Interior Points Random Disarrangement

The presetting interior points generated in the forth step are the vertices of the cubic elements which are all on special positions. According to Cavalcanti and Mello [11], 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.

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.

### 3.6 Interior Points Generated Based on Radial Method

In this step, real interior points would be separated from the output of the fifth 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. 5). 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.

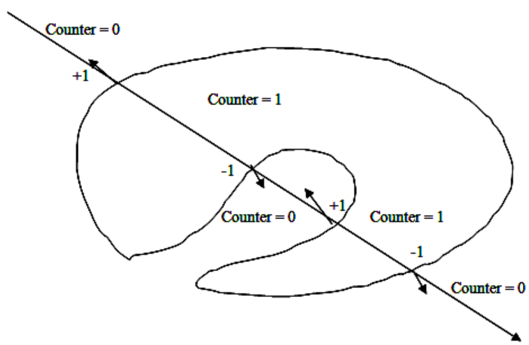



Fig. 5. Radial method in 2 dimensions could record the relative position of each section of the radial, which works well in 3 dimensions as well.

### 3.7 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. 6.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 background grid technology [12] could help find a set of tetrahedra which may contain point P efficiently. For each tetrahedron in this set, we could construct four tetrahedra temporarily 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.

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. 6.b) [13]. 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. And the problems about the Delaunay Cores in Fig.6 could be eliminated, and the result could be found in Fig.7 respectively.

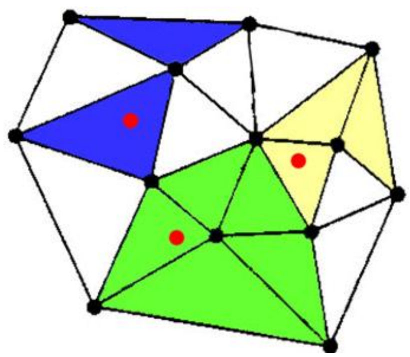


Fig.6.a. (Left-up) Possible result of intuitive Delaunay core generation could lead to tetrahedra overlap.

Fig.6.b. (Right-up) Possible result of revised Delaunay core generation could lead to tetrahedra overlap.

Fig.6.c (Bottom) Possible result of revised Delaunay core generation could lead to boundary point lost.

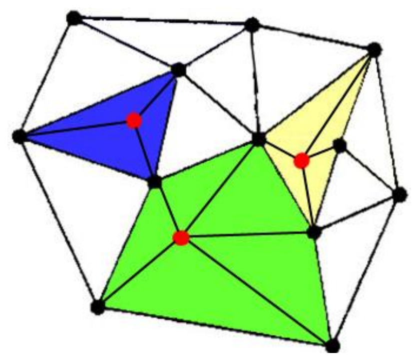


Fig.7.a. (Left-up) Reconstruction result of the intuitive generation.

Fig.7.b. (Right-up) Reconstruction result after vision check.

Fig.7.c (Bottom) Reconstruction result which could lead to boundary point lost.

### 3.8 Boundary Preservation

In the eighth step of the algorithm, some points may be deleted when the tetrahedra of the Delaunay core are broken (Fig. 6.c, Fig.7.c). If the points are boundary 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.8). To solve this problem, we need to 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.

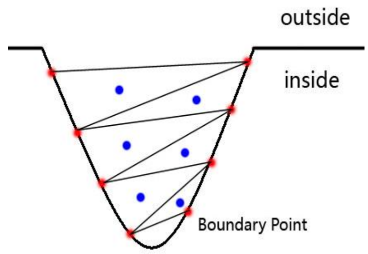



Fig.8. Neighbor-point joining leads to surface topology mistakes of the object which could be solved by remove the tetrahedra whose center points are outside of the object.

### 3.9 Sliver Tetrahedra Elimination

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.

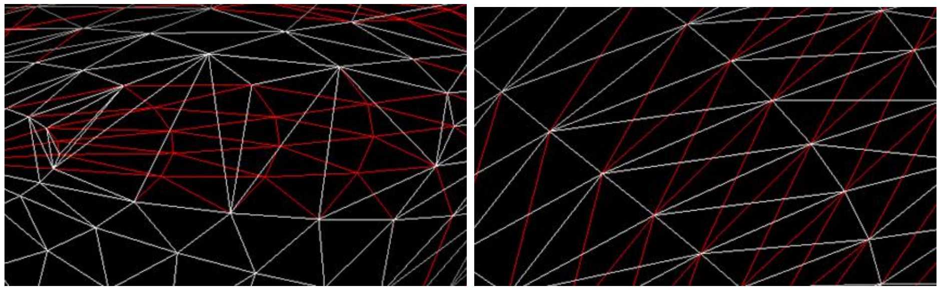
## 4. EXPERIMENTS AND APPLICATIONS

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) [14]. 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.

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 | |

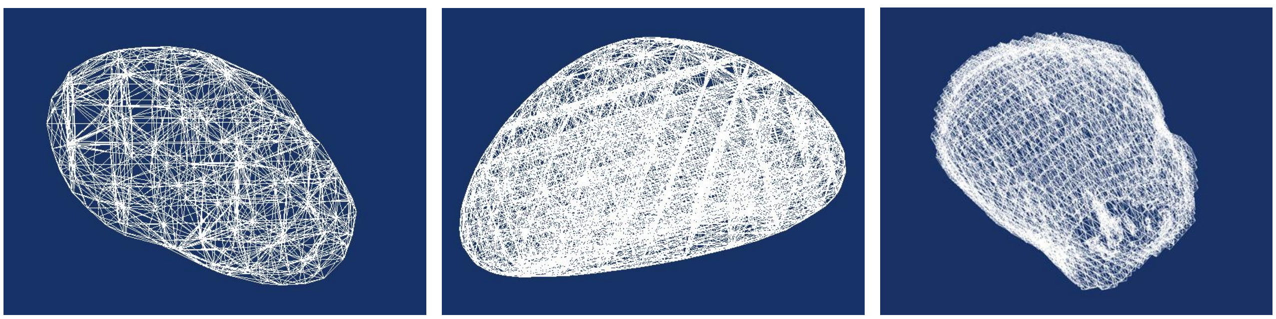
In Fig.9, a comparison of boundary preservation is processed between vtkDelaunay3D and the revised Delaunay algorithm. Fig.10 are the mesh results generated by the revised Delaunay algorithm of the kidney and breast data, and the mesh result generated by Space-Disassembling algorithm of the craniofacial data.



(a) (b)

Fig.9.a The white lines are the boundary triangles generated by vtkDelaunay3D, and the gray lines are the original boundary triangles.

Fig.9.b The white lines are the boundary triangles generated by the revised Delaunay algorithm, and the gray lines are the original boundary triangles.



(a) (b) (c)

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

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

Fig.10.c The mesh result of the craniofacial data by the Space-Disassembling algorithm, the granularity of which is 10.

And some soft-tissue deformation programs are processed based on these tetrahedral meshes. After simulation and analysis, we found good quality tetrahedral mesh is a critical prerequisite of these soft-tissue simulation methods. In FEM, sliver tetrahedra could lead to a program failure easily, because when the force is distributed on vertices of the tetrahedra, sliver tetrahedra could cause stress uneven, which makes the matrix computation more difficult and even program failure. In Mass-spring model, low quality mesh makes the elements iteration very inefficient, and could lead to wrong result of the deformation. If there's a force on the vertex with an obtuse angle of a sliver tetrahedron (Fig.11), the triangle may flip over and never deform back because when it does, the length of the spring can be the same as the original length, which means reaching a new balance state. And obviously, the result is incorrect. In order to eliminate these problems in soft-tissue deformation, we need to use algorithms which generate tetrahedral mesh with good quality. And the revised Delaunay algorithm performs very well on these soft-tissue simulation programs.

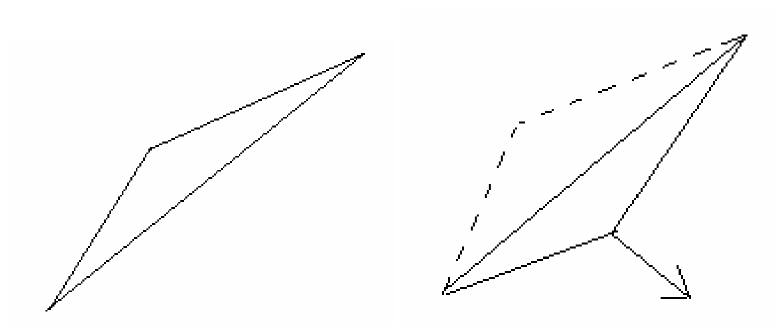


Fig.11 Low quality tetrahedral mesh could lead to wrong result during the simulation.

# 5. CONCLUSION

After analysis and comparison the results of the experiments, one could found out that tetrahedral mesh of soft tissue with good quality is an important pre-requisite of the soft-tissue simulation methods. So a good tetrahedral mesh generation algorithm is very significant in soft-tissue deformation simulation programs like craniofacial plastic surgery and breast reconstruction plastic surgery. Both of the Space-Disassembling and revised Delaunay algorithm work well in these programs. If the organ is not very spatial complicated and boundary preservation is not a main requirement, Space-Disassembling algorithm could be used. If the quality of all tetrahedra should be as good as possible, and the boundary should be preserved, the revised Delaunay algorithm could make a very good balance among these requirements.

# REFERENCES


1. Pengfei Huang et al, "Real-Time Simulation for Global Deformation of Soft Tissue Using Deformable Centerline and Medial Representation", LNCS 4072, Switzerland, July 2006.
2. Nedel, L.P. et al "Real time muscle deformations using mass-spring systems", Computer Graphics International, Jun, 1998.
3. Thomas J. R. Hughes, *The Finite Element Method: linear static and dynamic finite element analysis*, Courier Dover Publications, 2000
4. NIE Ying, "Virtual Human 3-D Medical Image Processing Based on Finite Elements", Beijing Biomedical Engineering, September 2003.
5. Esdras Medeiros, "A Topological Framework for Advancing Front Triangulation", SIBGRAPI'03, 2003.
6. Will Schroeder, *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*, 3rd Edition, Chapter 9
7. Laurent D. COHEN et al, "Deformable Models for 3D Medical Images using Finite Elements and Balloons", IEEE Conference on Computer Vision and Pattern Recognition, June 1992.
8. Nira Dyn et al, "A butterfly subdivision scheme for surface interpolation with tension control", ACM Transactions on Graphics, April 1990.
9. William E. Lorensen et al, "Marching cubes: A high resolution 3D surface construction algorithm", International Conference on Computer Graphics and Interactive Techniques, 1987.
10. Yang Xiaodong et al, "Optimization on the surface mesh of a solid model for FEM analysis", Computer Applications and Software, Aug, 2006.
11. Cavalcanti P R, Mellow U T. “Three-dimensional constrained Delaunay triangulation: A minimalist approach”, Proceedings 7th International Meshing Roundtable, Michigan, 1998.
12. Bonet J, Peraire J. “An alternating digital tree (ADT) algorithm for geometric searching and intersection problem”, Int. J. Number Methods Eng, 1991.
13. 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.
14. Visualization Toolkit 5.0, Kitware, Inc. http://www.vtk.org