Tetrahedral Mesh Generation Based on Contours

Sulei Tian
Northwest A&F University
College of Information Engineering
YangLing, ShannXi, China 712100
Email: suleitian0805@163.com

Zhiyi Zhang*
Northwest A&F University
College of Information Engineering
YangLing, ShannXi, China 712100
Email: liurenzu@sohu.com
Telephone: (+86)29-87092353

Xian Zhang
and Min Chen
Northwest A&F University
College of Information Engineering
YangLing, ShannXi, China 712100
Email: zhangxian-0905@163.com

***Abstract*—We develop a method to construct tetrahedral mesh from contours in this paper. Each tetrahedron is generated under the supervision of the defined quality measure standard .This method reduces possibility of intersection between the segment and the triangle. Intersection test is sufficiently analyzed to guarantee that there is no overlap between tetrahedrons.**

I. INTRODUCTION

The finite element method has been widely used in CAD/CAM/CAE. In finite element analysis, problem domain needs to be partitioned into a lot of small elements for the purpose of computational simulation and analysis. For example, in the field of medicine, biology, geology and so on, planar contours of object can be obtained through computed tomography (CT) or magnetic resonance image (MRI). Then, constructing triangular and tetrahedral meshes from contours could help people visualize the three dimensional object. At last, the geometric or physical properties of object can be analyzed and simulated.

Some research efforts have been made on tetrahedral mesh generation. Delaunay triangulation/tetrahedralization approach and advancing front technique (AFT) approach are widely used for tetrahedral mesh generation. Delaunay approach [1] [2] has the advantage that it has mathematical theory based on empty circle/sphere property as its foundation. Advancing front [3] [4] [5] [6] technique approach is flexible to form high quality tetrahedral mesh, because the point to form tetrahedral mesh can be chosen with quality measure standard.

This paper proposes a new advancing algorithm to construct tetrahedral mesh from adjacent contours. We define a quality measure standard to evaluate the shape of the tetrahedral mesh. Tetrahedral mesh is generated from inside to outside under the supervision of the quality measure standard. The surface is reconstructed automatically under constraint conditions, when this algorithm runs to termination. Intersection test is performed to ensure that all the tetrahedrons are generated without overlap.

II. A QUALITY MEASURE STANDARD FOR TETRAHEDRON

The shape of the tetrahedron is remarkably important in finite element method. In different areas, the requirements for the shape of the tetrahedron are different. A quality measure standard is builded to evaluate the shape of the tetrahedron as following:

If assume geometric center of tetrahedron is defined by equation (1):

$$\mathbf{G} = \frac{\mathbf{P_0} + \mathbf{P_1} + \mathbf{P_2} + \mathbf{P_3}}{4} \tag{1}$$

Here $\mathbf{P_0}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}$ be the vertexes of a tetrahedron, then the quality measure standard can be described as equation (2):

$$L = \frac{|\mathbf{G} - \mathbf{S}|}{R} \tag{2}$$

Here R is the radius of circumsphere of tetrahedron, $\mathbf{S}$ is the center of circumsphere of of tetrahedron. In other words, the metric is the quotient of the distance between geometric center and center of circumsphere and radius of circumsphere. The value of L is in the interval 0 and 1. When $L = 0$, $\mathbf{G}$ and $\mathbf{S}$ coincide, the mesh is a regular tetrahedron, which is the best element in the finite element analysis. The metric value more close to 0 suggests a better of tetrahedron.

III. TETRAHEDRALIZATION

The domain of tetrahedralization is between contours lying on two adjacent slices respectively.

A. Constraint conditions and some definitions

This advancing algorithm has two constraint conditions.

- **Constraint condition 1**: The polygon which represent the contour is a convex polygon.
- **Constraint condition 2**: No more than two adjacent points are collinear in one contour.

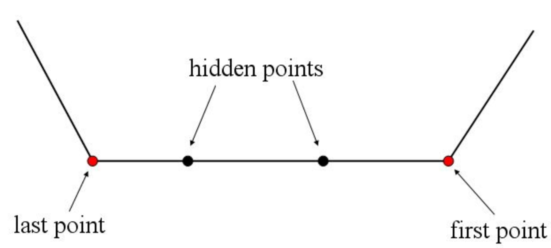

Fig. 1. Constraint condition 2: the first and last points are reserved, hidden points are removed.

Constraint condition 1 and constraint condition 2 guarantee that any three points are not collinear in one contour. Constraint condition 2 is weak. If there are $n(n > 2)$ adjacent

978-1-4244-7873-6/10/$26.00 ©2010 IEEE

points are collinear, the first and last points are reserved only, the other points called hidden points are removed as shown in figure 1. The hidden point are added during post-processing. The advantage of the constraint conditions is illustrated later.

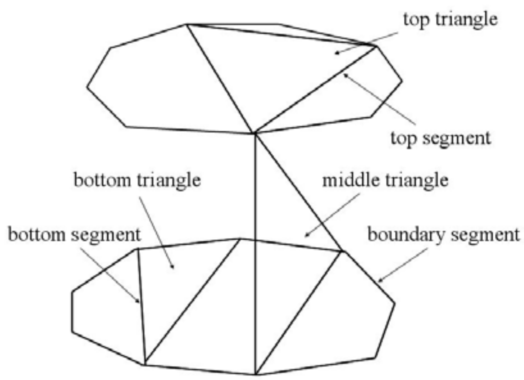

Fig. 2. Bottom, top and middle segment, bottom, top and middle triangle.

Some definitions are presented in order to describe the algorithm conveniently. Figure 2 illustrates this definition 1, 2, 3.

- **Definition 1**: Boundary segment is an edge that is defined by two adjacent points of one contour.
- **Definition 2**: A top segment, or bottom segment is a segment which has two vertexes on the top contour , or bottom contour . A middle segment is a segment which has two vertexes on the top contour and bottom contour respectively.
- **Definition 3**: A top triangle ,or bottom triangle consists three top segments ,or bottom segments .A middle triangle consists two middle segments and one top segment or one bottom segment.
- **Definition 4**: A tetrahedron is called type 0, if it consists three middle triangles and one top triangle or one bottom triangle as shown in figure 3 (a). A tetrahedron is called type 1, if it consists four middle triangles as shown in figure 3 (b).

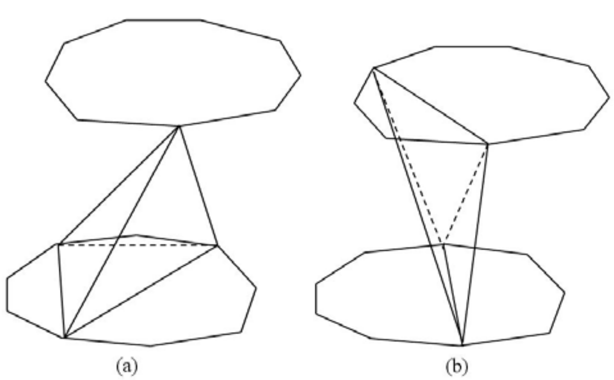

Fig. 3. Two types of tetrahedron: (a) type 0 tetrahedron, (b) type 1 tetrahedron.

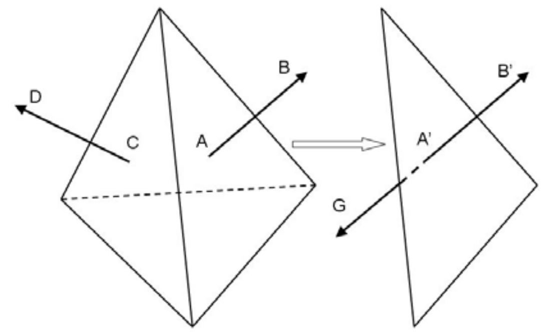

Fig. 4. AB, CD are the normal vector, they point to the outside of the tetrahedron, A'B'points to the outside of the triangle, A'G points to the inside of the triangle

B. Tetrahedralization algorithm

The process of advancing front approach is iterative. The following steps summarize this advancing front approach.

- **Step 1**: Construct the initial triangle. Select the longest boundary segment AB and a point C on the bottom contour to from a triangle ABC by joining point C to segment AB and guarantee the angle ACB maximum [7]. Take the triangle ABC as the initial triangle.
- **Step 2**: Construct the initial tetrahedron and the initial front. Select a point D on the top contour to form a tetrahedron $ABCD$ by joining it to the triangle ABC based on the builded quality measure standard. Calculate the normal vector of the three new triangles whose direction is toward the outside of the tetrahedron as shown in figure 4. The three new generated triangles are defined the initial front.
- **Step 3**: Pick a triangle as the current triangle from the front in a certain sequentially.
- **Step 4**: Select a point on the top contour or the bottom contour based on the defined quality measure standard to form a tetrahedron by joining the point to the current triangle. Then process each new triangle.
- **Step 5**: Update the front.
- **Step 6**: Go to step 3 until the front is empty.

The algorithm is discussed in detail in the rest of this section. The selected point must be on the outside of the current triangle that normal vector points to as shown in figure 4. The current triangle has been a face of a tetrahedron. If the point is not on the outside, the new tetrahedron will overlap with the tetrahedral mesh which consist the current triangle mesh. If there no point on the outside of the current triangle, This triangle is considered as the boundary triangle and remove it from the front. When the algorithm is end, The surface has been constructed from the two adjacent contour automatically. The surface consists all the boundary triangles. Exclude the situation above,the new tetrahedron may overlap with other tetrahedrons that have existed, so overlap test is indispensable. If overlap, abandon this optimal tetrahedron, find the suboptimal tetrahedral mesh, until the new tetrahedral mesh do not overlap with the other tetrahedrons. Except the current triangles, the new tetrahedron has other three new faces. Do the following treatment to each new triangle in the step 4. If the triangle is a top triangle or a bottom triangle, store it directly. One triangle is shared by two tetrahedrons at most. If the triangle is a middle triangle, determine weather it has existed as a face of tetrahedron . Calculate the normal vector of the triangle toward the outside of the new tetrahedron and add it to the front, in the case where this triangle has not existed. Remove this triangular mesh from front, in the case where this triangle has existed.

The front is managed by two operators:

1) Remove the triangle from the front from which the new tetrahedron is generated. Remove the triangle from the front which is coincidence with the new triangle. Also remove the triangle from the front which is a boundary triangle, because the boundary is the surface mesh.
2) Add the new triangle into front, if it has not existed as a face of tetrahedron.

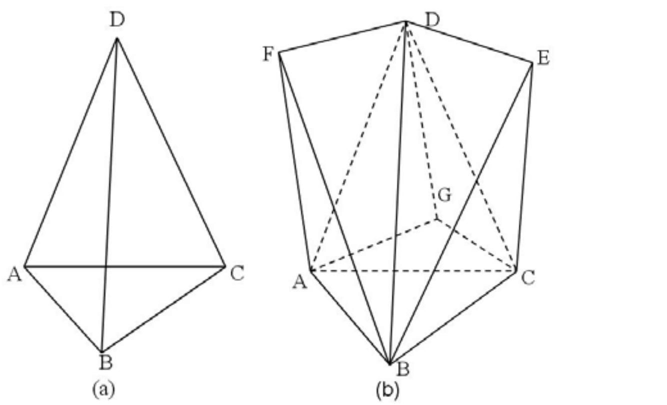

Fig. 5. Tetrahedral mesh generation by each layer from inside to outside: (a) The first layer consists the initial tetrahedron ABCD, the first level front consists triangle ABD, BCD, ACD. (b) The second layer consists tetrahedron ABDF, BCED, ACGD, they are constructed based on the first front triangle ABD, BCD, ACDtriangle ABF, FDB, BDE, BCE, CED, CGD, DAG, ADF form the second level front.

After each iteration, All the front triangles enclose a space together with the top triangles, the bottom triangles and the boundary triangles which have been generated. The enclosed space have been discretize into tetrahedrons. The normal vector of front triangle point to the outside of enclosed space. The new tetrahedrons are created toward outside of enclose space. In order to manage the front uncomplicated, more important to reduce the probability that the new tetrahedron overlap with the existed tetrahedrons, the tetrahedrons are constructed by each layer. Figure 5 shows this idea. The first layer is constructed in the step 2 which just has one tetrahedron. The initial front is considered as the first level front. The $n-th$ layer tetrahedrons is constructed based on the $(n-1)-th$ level front. When the $(n-1)-th$ level front are used up, the $n-th$ layer tetrahedrons are constructed completely. All the front belong to the $n-th$ tetrahedrons form the $n-th$ level front. In the step 3, the sequence to pick a triangle as current triangle from the $(n-1)-th$ level front is counter-clockwise. Succeeding layer can be constructed by this recursively approach. The $q-th$ level front is excepted to have 2.5^q triangles, because the constructed tetrahedral mesh based on the triangular mesh front is either type 0 or type 1, the former is excepted to consist 3 new front triangles, the latter is excepted to consist 2 new front triangles. Because tetrahedrons is constructed by each layer from inside to outside and the current triangle is on the outermost level, many tetrahedrons are on the inside of current current triangle.

This advancing front technique approach is different from the traditional approach. The traditional approach often reconstructs surface first, takes the surface triangles as initial front, constructs the tetrahedron from outside to inside. This advancing front technique approach constructs the initial tetrahedron first, takes the face of tetrahedron as the initial front, constructs the tetrahedron from inside to outside. The surface is reconstructed automatically when the algorithm is end.

C. Intersection test

The new tetrahedron may overlap with the existing tetrahedrons, so overlap test is indispensable after each iteration. Overlap test between tetrahedrons can be transformed into segment-triangle intersection test. Actually ,the process of tetrahedral mesh generation based on advancing front approach accompany extensive intersection test. The intersection test often costs most of time. There are two specific segment -triangle intersection test: 1. The new segment should not intersect with the existing triangle. 2. The existing segment should not intersect with the new triangle. If the vertexes of a existing segment or vertexes of a existing triangle are all on the inside of current triangle, intersection test is unnecessary, because the new segment or new triangle is on the outside of the current triangle.

A method is introduced based on parametric equation which can be used to judge the relationship between line and plane. In three dimensional space, three points that are not collinear determine a plane whose parametric equation is given by:

$$u\mathbf{P_0} + v\mathbf{P_1} + (1-u-v)\mathbf{P_2} \quad (u, v \in R) \tag{3}$$

Here, $\mathbf{P_k} = (x_k, y_k, z_k) \quad k = 0, 1, 2.$

Two distinct points determine a line whose parametric eqution is given by

$$t\mathbf{J_a} + (1-t)\mathbf{J_b} \quad (t \in R) \tag{4}$$

Here, $\mathbf{J_a} = (x_a, y_a, z_a)$, $\mathbf{J_b} = (x_b, y_b, z_b)$.

The point at which the line intersect with the plane is described by setting the line equal to the plane in parametric equation:

$$u\mathbf{P_0} + v\mathbf{P_1} + (1-u-v)\mathbf{P_2} = t\mathbf{J_a} + (1-t)\mathbf{J_b} \tag{5}$$

This can be simplified to

$$(\mathbf{J_b} - \mathbf{J_a})t + (\mathbf{P_0} - \mathbf{P_2})u + (\mathbf{P_1} - \mathbf{P_2})v = \mathbf{J_b} - \mathbf{P_2} \tag{6}$$

Which can be transformed into a system of liner equations using the coordinates of points

$$\begin{cases} (\mathbf{x_b}-\mathbf{x_a})t+(\mathbf{x_0}-\mathbf{x_2})u+(\mathbf{x_1}-\mathbf{x_2})v=\mathbf{x_b}-\mathbf{x_2} \\ (\mathbf{y_b}-\mathbf{y_a})t+(\mathbf{y_0}-\mathbf{y_2})u+(\mathbf{y_1}-\mathbf{y_2})v=\mathbf{y_b}-\mathbf{y_2} \\ (\mathbf{z_b}-\mathbf{z_a})t+(\mathbf{z_0}-\mathbf{z_2})u+(\mathbf{z_1}-\mathbf{z_2})v=\mathbf{z_b}-\mathbf{z_2} \end{cases} \quad (7)$$

The relationship between line and triangle can be judged according to the solution of equations.

If the equations have no solution, then the line parallel to the plane. If the equations have infinite solution, then the line lies on the plane. If the equations have one solution, then the line intersects with the plane at a point.

If the solution satisfies the condition $t \in [0,1]$, then the point of intersection is on the line between $\mathbf{J_a}$ and $\mathbf{J_b}$. Particularly, $t=0$, the point is $\mathbf{J_a}$, $t=1$, the point is $\mathbf{J_b}$.

If the solution satisfies the condition $u,v \in [0,1], u+v \leqslant 1$, then the point of intersection is in the plane inside the triangle spanned by the three points. Particularly, $u=1, v=0$, the point is $\mathbf{P_0}$, $u=0, v=1$, the point is $\mathbf{P_1}$, $u=0, v=0$, the point is $\mathbf{P_2}$.

The test is classified into the following four categories in the view of position of segment and triangle as shown in figure 6. Intersection test is discussed in detail using the method introduced under two constraint conditions as following.

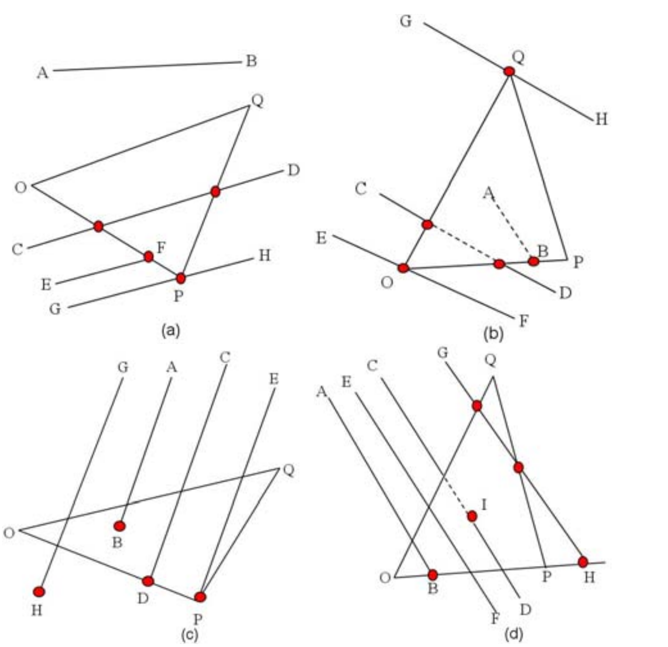

Fig. 6. The relationship between a segment and a triangle: (a) segment and triangle lying on slice, (b) segment lying on slice and middle triangle, (c) middle segment and triangle lying on slice, (d) middle segment and middle triangle.

1. segment and triangle lying on slice: figure 6 (a) illustrates such situation. The top segment is parallel to the bottom triangle, the bottom segment is parallel to the top triangle too. Segment AB is parallel to triangle OPQ. In this case, the equations has no solution. If a segment lies in the plane which is determined by a triangle, the equations have infinite solutions, we can not tell whether the segment intersects triangle, as segment CD intersects with triangle OPQ , while segment MN can't intersect with triangle OPQ. A segment on the slice must be an edge of a middle triangle, so if a segment and a triangle on the same slice intersect , this segment must intersects with a middle triangle, as segment CD must intersect with a middle triangle which consist segment OP.Because no more than two points are collinear in one contour, the case that segment EF intersects with triangle OPQ at point E and segment GH pass the point P can not appear, otherwise point O, F, P and point $G, P, H$are collinear, which conflicts with constraint conditions.

2. segment lying on slice and middle triangle: figure 6 (b) illustrates such situation. Because no more than two points are collinear in one contour,the case that segment AB intersects with triangle OPQ at point B, segment EF pass the point O, and segment GH passes the point Q can not appear, otherwise point O, B, P, point E, O, F, and point G, Q, H are collinear, which conflicts with constraint conditions. when t satisfies the condition $t \in [0,1]$, u, v satisfies one of the following three conditions: 1. $u+v=1, u,v \in (0,1)$, 2. $u=0, v \in (0,1)$, 3. $v=0, u \in (0,1)$,a segment on slice intersects with a middle triangle which consists a segment on the same slice, as segment CD intersects with triangle OPQ at point I.

3. middle segment and triangle lying on slice: figure 6 (c) illustrates such a case. A middle segment and a triangle either intersect at their vertex, or have no intersection, as segment EP intersects with triangle OPQ at vertex P, segment GH and triangle OPQ have no intersection. Because no more than two points are collinear, the case that segment CD intersects with triangle OPQ at point D can not appear, otherwise point O, D, P are collinear. Because a contour is a convex polygon, there is no point in the interior of a triangle on slice, the case that segment AB intersects with triangle OPQ at point B can not appear.

4. middle segment and middle triangle: figure 6 (d) illustrates such a situation. Because no more than two points are collinear, the case that segment AB intersects with triangle OPQ at point B can not be appear, otherwise points O, B, P are collinear, which conflicts with constraint condition. when t satisfies the condition $t \in (0,1)$ and u, v satisfies the condition $u, v \in [0,1]$ a middle segment intersects a middle triangle, as segment CD intersects with triangle OPQ at point I. A middle segment can not be on the plane determined by a middle triangle, as GH is on the plane determined by triangle OPQ, otherwise points O, P, H are collinear. A middle segment may be parallel to a middle triangle, as segment MN is parallel to the triangle OPQ.

From the analysis of these four categories, we conclude that it is unnecessary to do intersection test between segment and triangle on slice, middle segment and triangle on slice under two constraint conditions. The constraint conditions reduces the possibility of intersection.

IV. Post-processing

Tetrahedral mesh have been generated from two adjacent contours under two constraint conditions. It is mentioned that

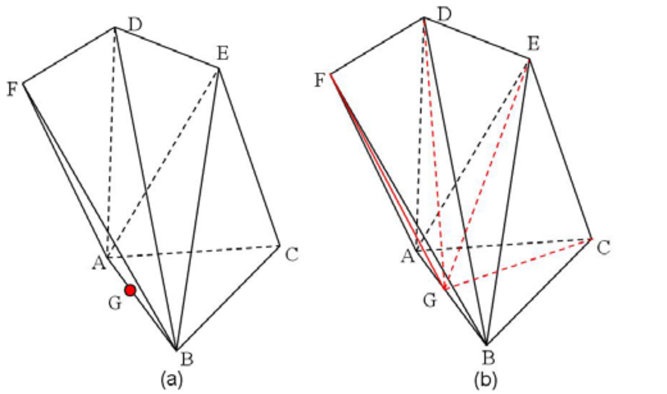

Fig. 7. Add the hidden point: (a) AB is a boundary segment, G is a hidden point, AB is shared by tetrahedron ABCD, ABDE, ABEF which are inadequate tetrahedrons, it is also shared by bottom triangle ABC, middle triangle ABD, ABE, ABF which are inadequate triangles. (b) Each inadequate triangle is split to two new triangles, each inadequate tetrahedron is split to two new tetrahedrons.

constraint conditions 2 is weak. The hidden points which are surly on the boundary segment are added to form new tetrahedrons in this phase. One boundary segment is shared by one type 0 tetrahedrons and several type 1 tetrahedrons called inadequate tetrahedral mesh, so it is shared by a top or bottom triangle and several middle triangles called inadequate triangle as shown in figure 7 (a). Each inadequate triangle is split to two new triangle by joining the hidden point to the vertex of this triangle which is not the vertex of the boundary segment. After split inadequate triangle, in interior of each inadequate tetrahedron, a new triangle is formed whose vertexes are the hidden point and other two vertexes of tetrahedron which are not two vertexes of the boundary segment. This interior triangle split a inadequate tetrahedron into two new tetrahedrons as shown in figure 7 (b). If a middle triangle consisting this boundary segment is not a boundary triangle, it is shared by two tetrahedrons, so after split, each new triangle is shared by two new tetrahedrons. The amount of new tetrahedrons is $\sum_{i=1}^{m} P_i T_i$, Where m is the amount of the boundary segment, pi is the amount of hidden points on the i-th boundary segment, ti is the amount of tetrahedron which consist the i-th boundary segment.

V. Results and Conclusion

Figure 8 shows the results of this algorithm. The results consist the surface meshes and the tetrahedral meshes.

In this paper, we propose a algorithm that can construct tetrahedral mesh from contours. In order to guarantee the quality of tetrahedron, a quality measure standard is builded. After each iteration, the current optimal tetrahedron is generated based on the quality measure standard. Through the sufficient analysis, we get the condition that the segment intersect a triangle in the process of tetrahedral mesh generation. In traditional advancing front approach, intersection test often costs a lot of time, so how to reduce the amount

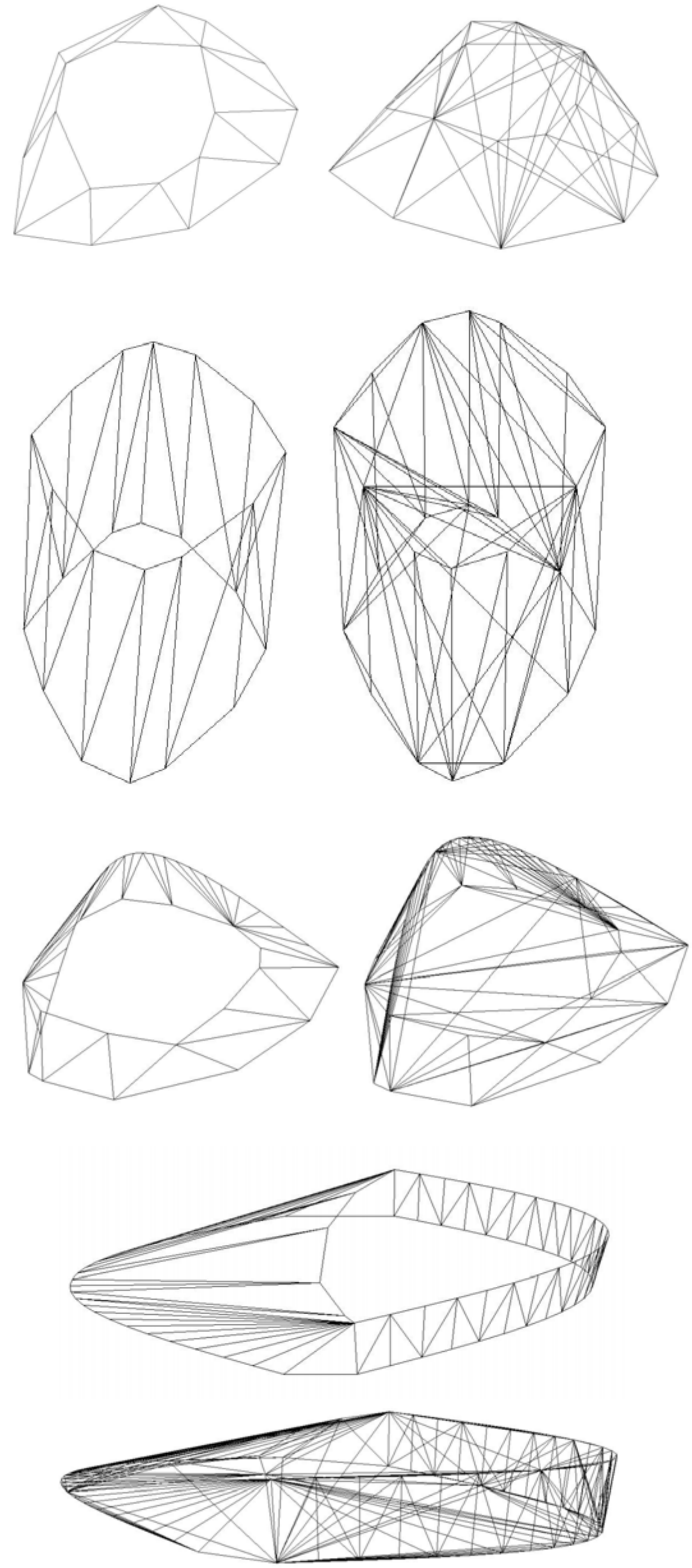

Fig. 8. The result: surface meshes and tetrahedral meshes.

of intersection is important. In our algorithm, the possibility of intersection is decreased, because of constraint conditions and tetrahedral mesh generation by each layer from inside to outside. The surface are reconstructed automatically under constraint condition 1. In the postprocessing, the hidden points are added.

This algorithm can process the situation that there are two contours on top and bottom slice respectively, besides that the contour should be a convex polygon. Many 3D object's shape is complex. The contour of object may be a concave polygon. There may be more than one contours on one slice. In order to process these complex situation, we need develop a algorithm that can decompose a convex polygon into several concave polygons and match these contours on adjacent slice in the future.

ACKNOWLEDGMENT

This work was supported by Northwest A&F University of China No.Z111020721, the Special Fund from the Central Collegiate Basic Scientific Research Bursary No.Z109021004 and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry No.K314020901.

REFERENCES

[1] D.F. Watson, *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*. The Computer Journal, Vol. 24, 1981, pp. 167-172.
[2] T. K. Dey, *Delaunay Mesh Generation of Three Dimensional Domains*. Technical Report OSU-CISRC-9/07-TR64, October, 2007.
[3] Shan Ju, Lin Guan Zhen, Qun Song Chao, *A Reliable and EffectiVe Tetrahedral Meshing Algorithm* . Chinese Journal Of Computers, vol 30, No11 , Nov2007.
[4] George, P.L, *Automatic Mesh Generation: Application to Finite Element Methods*. Wiley, 1991.
[5] Yasushi Ito, Alan M. Shih and Bharat K. Soni, *Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front*. 13th International Meshing Roundtable, Willimasburg, Virginia, USA September 19-22, 2004.
[6] Jin J, Tanner RI, *Generation of unstructured tetrahedral meshes by advancing front technique* . International Journal for Numerical Methods in Engineering, vol. 36, no. 11, 1993, pp. 1805-1823.
[7] Zhiyi Zhang, Kouichi Konno and Yoshimasa Tokuyama, *3D Terrain Reconstruction Based on Contours*. Ninth International Conference on Computer Aided Design and Computer Graphics(CAD/CG2005), pp. 325- 330, 2005.