Research Journal of Applied Sciences, Engineering and Technology 6(10): 1862-1867, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467
© Maxwell Scientific Organization, 2013
Submitted: November 20, 2012 Accepted: January 11, 2013 Published: July 20, 2013

Local Optimum Triangulation for Unorganized Points Cloud

Nan Wang, Qiang Zhang, Donghent Zhou and Xiaopeng Wei
Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Ministry of Education, Dalian, China

Abstract: The triangulation of the unorganized data from the 3D scanner is always an important research subject of the CAD, computer geometry, reverse engineering and other areas. This paper introduces the current mainstream methods of triangulation that based on 3D points cloud data: plane projection method and direct method of division. This paper put forward a local optimization of direct triangulations method.

Keywords: Points cloud, direct triangulations, local optimization

INTRODUCTION

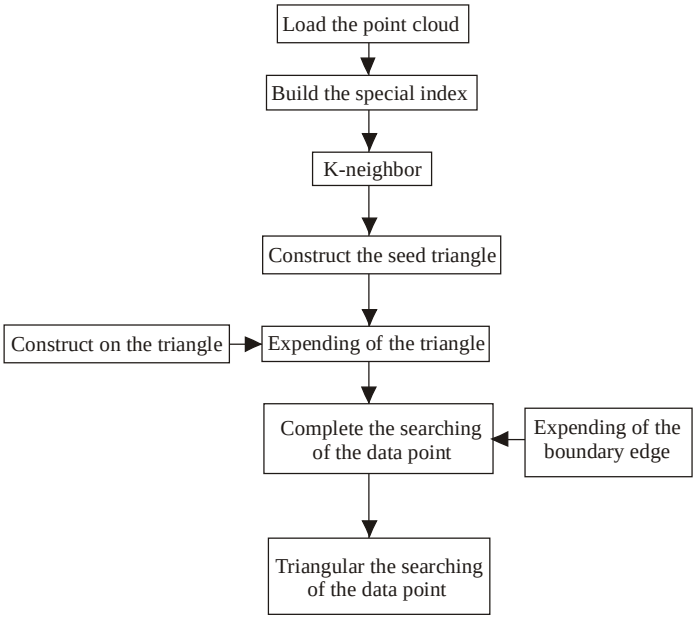
As an important part of the reverse engineering surface reconstruction, the triangulations form the scattered data points in R^3 has received more and more attention with the development of science and technology. Lots of scholars are concentrating on them at home and abroad. According to the difference of the points, the triangulation problem can be divided into 2D triangulation and 3D triangulation (Wei and Su, 2008). The research of the triangulation in plane domain is already quite mature, so we will not go into details in this paper. At present, the 3D triangulation can be divided into two kinds: projection method and the spatial direct triangulation. The projection method is to project the scatted points to the 2D plane and do the triangulations in 2D, then split the triangular mesh after map back to the space. The essence of this method return to the plane domain triangulation problem, and the operation of projected will lead to the distortion of the triangular mesh as well as increase the complexity of the triangulations. For the complex model, this method can't obtain a satisfactory result. Therefore, the research of the spatial direct triangulation is needed.

There have been lots of algorithms about triangulation for points cloud. The more recent exhaustive overviews of 3D triangulation methods are presented.The most common triangulation algorithms can be divided into two main groups: Mesh growing method and Voronoi-based method.

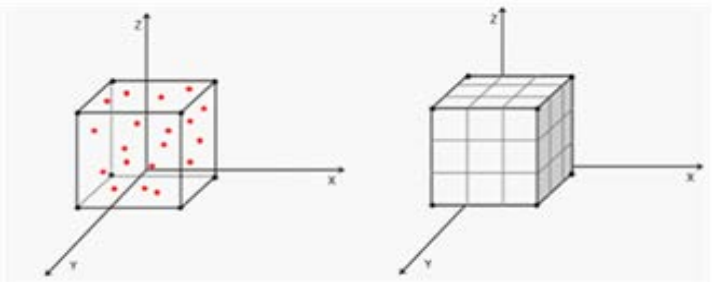
Mesh growing method generate the surface from a seed triangle and grow the meshes by using some criteria. Recent years, a number of algorithms have been proposed. Bernardini *et al*. (1999) put forward the Ball Pivoting algorithm, he grows the front as a ball that around the front edge, the radius is always given by the user, and the triangle is formed when the ball touches another points. This method can afford a correct triangulation for the point cloud as well as the noise data, but it requires normal of the point and is not easy to get the appropriate radius. Huang and Menq (2002) proposed an algorithm which firstly search the K-nearest neighbors of the two points of the front edge, then delete the points which are illegal. He projects the points onto the plane that the triangle belongs to, and the points which generate edges intersect the existing edge will be deleted. In the end, choose the point that is the nearest to the front edge to generate the triangle. Lin *et al*. (2004) found some shortcomings of this method. In Lin *et al*. (2004) introduced the Intrinsic Property Driven (IPD) algorithm, which overcomes the shortcomings by improves the way of looking for the point. In Li *et al*. (2009) improved the Priority Driven by evaluating shape changes from the estimation of the original surface. He proved that the triangulation speed of the method was higher than Ball Pivoting and the Cocone. Di Angelo *et al*. (2011a) put forward a mesh growing algorithm using the Gabriel 2-Simplex criterion. From his experiments result, the quality of the surface is better than Ball Pivoting and the Cocone family in the tessellation of noise points cloud. In order to improve the robust to the area which is not sufficiently sampled, Di Angelo and Giaccari (2011b) improved the algorithm.

The Voronoi-based method is according to the Delaunay tetrahedralization . The common steps are always do the 3D Delaunay triangulation firstly, then remove the tetrahedral and build the triangular mesh. Edelsbrunner and Mucke (1992) proposed the concept of α-shape, its principle is to delete the edge, triangular and tetrahedral which the radius of the circumscribed circle or circumsphere is greater than α and so as to get

Corresponding Author: Qiang Zhang, Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Ministry of Education, Dalian, China, Tel.: +86411-87403733; Fax: +86411-87403733



(a)



(b)

Fig. 1: The process and the Hash tables

the reconstructed surface. Bajaj *et al*. (1995) generated a C^1 continuous surface in his research based on the α-shape. Recent years, the famous method is the Crust which has the theoretical guarantees. The Crust algorithm was proposed by Amenta *et al*. (1998a).

And later Amenta *et al*. (1998b) promoted it to surface reconstruction in 3D space. In Amenta *et al*. (2002) proposed the Cocone to reduce the time and memory consumption of Crust. But the Cocone and Crust have a same shortcoming pointed by Chang *et al*. (2009), that the theoretical guarantee of obtaining a correct reconstruction is only possible as long as the point cloud is well sampled. Dey and Goswami (2003) proposed the Tight Cocone and Dey and Goswami (2006) proposed the Robust Cocone, the first one can provide a watertight closed surface reconstruction and the second can deal with noisy data. The Super Cocone proposed by Dey *et al*. (2001) is good at processing of large scale points cloud. A further evolution of the Crust is the Power Crust proposed by Amenta *et al*. (2001) which can generate a watertighe mesh for any point cloud, but sharp edges or noisy data can result in defectiveness. Ou Yang *et al*. (2011) introduced a method a method of reconstruction of triangular surfaces via clustering of Delaunay spheres.

MATERIALS AND METHODS

The triangle mesh construction method in this paper is based on growth method. The specific process as shown in Fig. 1(a).

In the process of the triangulation, the operation of traverse to the points is needed and it is always a recursive process. But during this process, not every points is required to be traversed for every time, which bringing a challenge to efficiency of the algorithm. In order to improve the efficiency, we should narrow the search scope as much as possible to limit the operation in a local area, and make sure that the optimal point is exist in this local area. The points cloud is lack of the

necessary information of topology and adjacency, we should provide the first level of topological relations by creating k-nearest neighbor (KNN) for every point. We can use the geometric positional relationship of the sample points to create it.

K-nearest neighbor(KNN) (Sarkar and Leong, 2000) can be defined as**:** For a point p which come from the point set P, according to the Euclidean distance of the other points to p, the distance ordered by ascend as$\prod$. We can define the KNN as $N^K{}_P$: $\|P_{\prod(t)} - p\|$ and $\| P_{\prod(t)} - p\| < \| P_{\prod(i+1)}p\|$, i ∈ [1, n - 1] ,

$$N^k{}_p = \{\prod(1), \prod(2), \ldots, \prod(k)\}$$

The K-nearest neighbor search problem is to find the k nearest points, in a point set P containing n points, usually with respect to the Euclidean distance. Guttman (1984) states that it has applications in a wide range of real-world settings, in particular pattern recognition, machine learning, and database querying. In order to speed up the search process as well as the efficiency of the algorithm, we create a spatial index for the points cloud firstly. The spatial index is an auxiliary space data structure which base on the position or shape of the space object. The commonly used spatial index has the following three type: (1) Octree; (2) K-D tree; (3) Hash table. Hash function algorithm divides the space into some little part. By using this we can rapidly get the location of the bounding box and search the k-neighbor. This method is suitable for space object and is simple and efficient. The algorithm in this paper always uses this method to get the K-neighbor of every point. After we get the K-neighbor of every point, the main process of the method is the two steps:

Step 1: Construct the Seed Triangle

A triangle is composed of vertex and edge, so we should construct a vertex firstly, then construct the first edge by this vertex and look for the third vertex in the end. There have a lot of methods about how to construct the seed triangle. In order to improve the efficiency of the algorithm and avoid spending too much time on the judgment, we select the point which the X coordinate value is X_{max} as the first vertex. When we not get only one point, we can select the one which the Y coordinate value is the largest, and so on. The following are the steps to select the seed triangle:

- Select the extreme coordinate value point P1 as the first vertex of the seed triangle
- Searching for the point P2 within P1's K-neighbor which the distance to P1 is the nearest
- Searching for the third point P3 within the K-neighbor of P1 and P2, make sure that P2P3+P1P3 as short as possible. The construction will not stop until the P3 is founded, else go to step (2)

Step 2: Expanding of Boundary Edge

When expanding the boundary edge, we firstly select a boundary edge, then search in the K-neighbor of the vertex to find a new point to construct a new triangle. At the same time put the new triangle to the linked list of the triangle and update the linked list of the boundary edge. In the process of the construction of the new triangle, some condition will be taken into to make the new vertex is the best one and the triangle is locally optimal. How to make the chosen is the best is critical. So, in the growth process, we should pay attention to the following three aspects:

- The mesh won't appear to change drastically
- Avoid the emergence of the narrow triangle
- The new triangle can't overlap the triangle that already have

Constraint 1: In the expanding of the boundary edge, we always get lots of scalable points that are legal but inappropriate. In order to ensure a smooth transition of

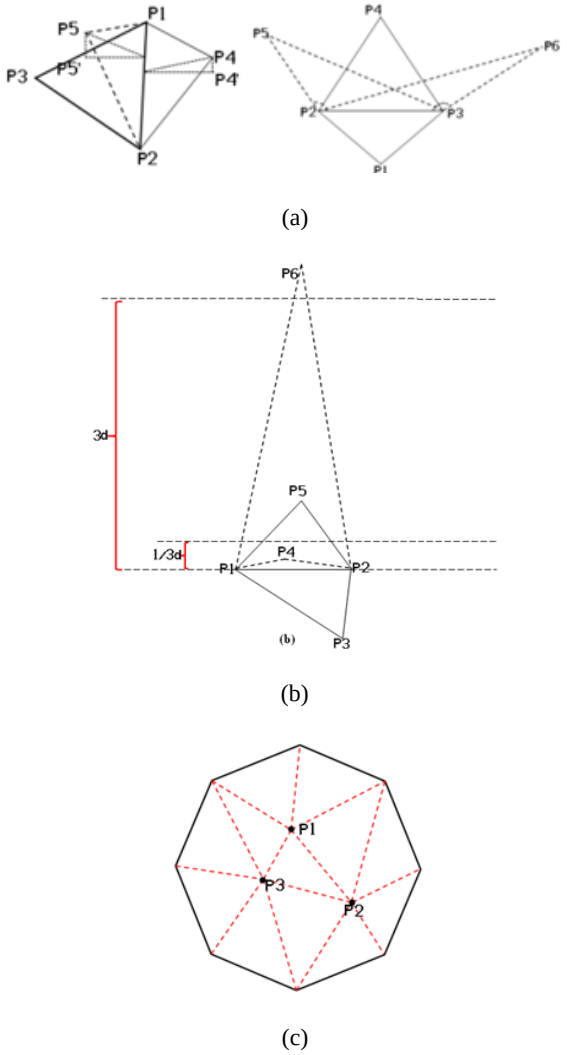


Fig. 2: The constraints needed in the algorithm

the triangle mesh, a constraint on the angle of the two triangles which have the public edge is needed. In general, if the angle of two adjacent grids is greater than 90°, the surface will be smooth. Due to the angle of the grid and the angle of the triangle's normal vector are complementary, this can be transformed into the values of the angle of the normal vector. Here we define a threshold to the angle of the normal vector, the angle between the normal vector of the new triangle and the normal vector of the expanding triangle must greater than 90°.

As shown is Fig. 2(a), expanding edge P1P2. We can find that there are two points P4 and P5 can be chosen, P4' and P5' are the projection of P4 and p5 on the plane that Δ P1P2P3 belongs to. Δ P1P2P4 and Δ P1P2P5 are all good griangles, but we can see that if Δ P1P2P5 is chosen as the new triangle, the transition of the mesh is too drastic and the Δ P1P2P4 will be more smooth. So P4 should be the best choice.

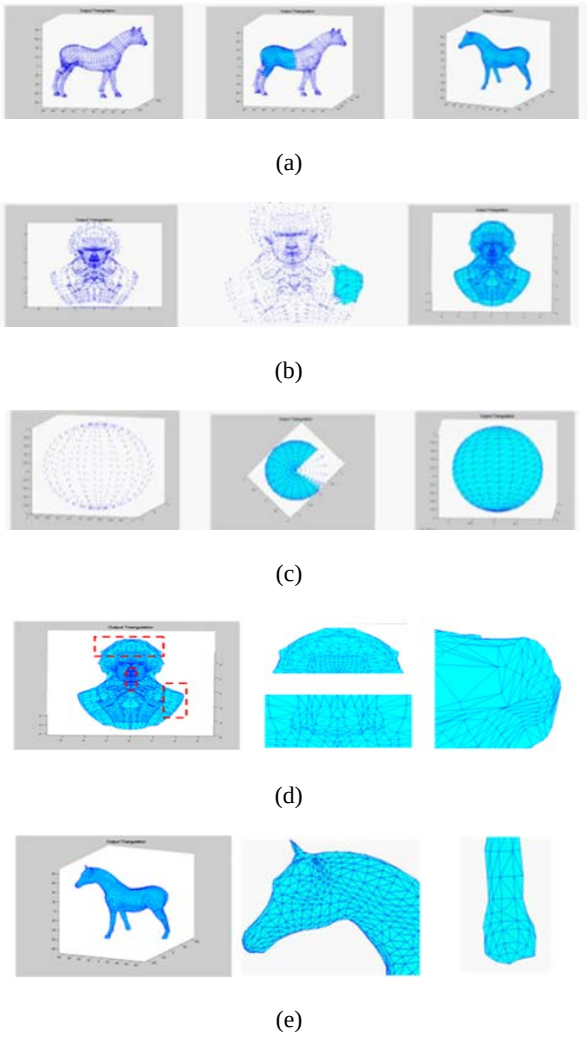
In this paper, we use the vector product to get the normal vector of the triangle. As shown in Fig. 2(a), $\overrightarrow{P1P2} \times \overrightarrow{P2P3}$ is the normal vector of Δ P1P2P3. In the same way, the normal vector of Δ P1P2P4 and Δ P1P2P5 can be calculated. We make the angle in the range of $[0, \pi/3]$.

Constraint 2: We have said front that in the process of triangulation, the emergence of the narrow triangle is forbidden. For the narrow triangle, it has some characterstics: (1) the distance between the expanded point and the edge is closer; (2) compared to an nomal triangle, the minimum interior angle will be smaller. In view of this two characteristics, in this paper we regulate that: the distance between the point and the expanding edge L should be within a certain range, k1. $D < 1 < k2$. d, d is the length of the expanding edge,K1 and K2 are positive number. In this, we define k1 = 1/3, k2 = 3.

In Fig. 2(b), P1P2 is the expanding edge, now has three points P4, P5, P6 can be chosen. But as shown in the figure, P4 is so closed to edge P1P2 and P6 is always too far. Δ P1P2P4 and Δ P1P2P6 are narrow triangles. So we should choose Δ P1P2P5.

Constraint 3: Within the candidate points, the point's projection to the expanding edge may fall on the extension line of the edge. This makes the angle of the two edges too large and the new triangle will be a narrow one. So when we sceening the candidate points, the points with a large angle will be deleted. Here we can make multiplication of the edge vector. According to the sign of the result to determine the relationship of the position.

In this Fig. 2(c), P2P3 is the expanding edge. As is shown, angle P5P2P3 and angle P6P3P2 is too large, Δ P5P2P3 and Δ P6P3P2 are narrow triangle. So P4 is the best one. Make a multiplication of $\overrightarrow{P1P2}$ and $\overrightarrow{P2P5}$:



(a)

(b)

(c)

(d)

(e)

Fig. 3: The experiment results and details

$$\begin{cases} \overrightarrow{P2P3} \cdot \overrightarrow{P2P5} > 0 & \text{is an acute angle} \\ \overrightarrow{P2P3} \cdot \overrightarrow{P2P5} < 0 & \text{is an obtuse angle} \\ \overrightarrow{P2P3} \cdot \overrightarrow{P2P5} = 0 & \text{on the edge or his extension line} \end{cases}$$

Constraint 4: For the structure of the triangle mesh, a vertex can be shared by many triangles but an edge can only be shared by two triangles. This also becomes a constraint that can avoid the overlap of the triangles.

Here we introduce the concept of the interior point. First of all, we define that if one edge is shared by two triangles, this edge is expanded. Now we can define as follows:

Interior Point: if all the edges that treat this point as endpoints have been expanded. We call this point Interior Point. For the point that has been determined as interior point, it cannot be choosen to construct a new triangle. This kind of point can be exculed during each expanding process, which can always improve the efficiency of the search.

As shown below in Fig. 2(d): P1, P2, P3 are all interior point.

In the (e) of Fig. 2, P1 is a interior point. In (f), P1P2 and P2P3 are not expanded, so P1 is not a interior point.

Experimental results: The complexity of the triangulation of this paper is O(KN) because we only search in the K-neighbor each time, K is the number of the points of the K-neighbor, N is the number of the points. In order to prove our algorithm, we realized it in MATLAB. We have used various points cloud. The results are shown in Fig. 3.

DISCUSSION

Figure 3(a), the points cloud of the Horse, has 2302 point and the triangulation result has 4307 triangles; (e) is the details of head and leg of the horse. In these areas, the density of the points cloud is large and the transition is drastic. In our results, we can see that there have no thin and narrow triangles at all, and the points in the ears are triangulated perfectly. In our algorithm, we defined that one edge can only be shared by two triangles, so there are no intersection of triangles in the mesh.

Figure 3(b), the Beethoven points cloud that has 2655 points and the triangulation results, 4971 triangles; (d) shows the shoulder of Beethoven, in this area the angle of the adjacent triangle is larger, from the result, the meshes formed here are good and no holes exists; the head part, the sharp features are retained and have no intersection of triangles; the nose part of the points cloud, the triangulation of this area should be very complex, because the density of the points is large, the transition is always drastic and there always has some sharp features. As it shown, the result is good, no thin triangles, no holes, no single triangle.

Figure 3(c), the points cloud of a ball, has 441 points. The triangulation result has 760 triangles.

CONCLUSION

- In this study, we have presented a new direct triangulation algorithm in 3D, which is computationally and memory efficient. It is always easy to be realized and can obtain a good result.
- From the experiments, there are no illegal triangles in the mesh. In the future work, we will focus on how to remove the noise data during the triangulation process.

ACKNOWLEDGEMENT

This study is supported by the Program for Changjiang Scholars and Innovative Research Team in University(No.IRT1109), the Program for Liaoning Science and Technology Research in University (No.LS2010008), the Program for Liaoning Innovative Research Team in University(No.LT2011018), Natural Science Foundation of Liaoning Province (201102008), the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University and by "Liaoning BaiQianWan Talents Program (2010921010, 2011921009)".

REFERENCES

Amenta, N., M. Bern and D. Eppstein, 1998a. The crust and the beta-skeleton: Combinatorial curve reconstruction. Graph. Mod. Image Proc., 60(2): 125-135.

Amenta, N., M. Bern and M. Kamvysselis, 1998b. A new Voronoi-based surface reconstruction algorithm. Proceeding of the 25th Annual Conference on Computer Graphics and Interactive Techniques, New York, pp: 415-421.

Amenta, N., S. Choi and R. Kolluri, 2001. The power crust. Proceeding of the 6th ACM Symposium on Solid Modeling and Applications, pp: 249-260.

Amenta, N., S. Choi, T.K. Dey and N. Leekha, 2002. A simple algorithm for homeomorphic surface reconstruction. Int. J. Comp. Geom. Appl., 12(1-2): 125-141.

Bajaj, C.L., F. Bernardini and G.L. Xu, 1995. Automatic reconstruction of surfaces and scalar fields from 3D scans. Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp: 109-118.

Bernardini, F., J. Mittleman, H. Rushmeier, C. Silva and G. Taubin, 1999. The ball pivoting algorithm for surface reconstruction. IEEE T. Visual. Comp. Graph., 5(4): 349-359.

Chang, M.C., F.F. Leymarie and B.B. Kimia, 2009. Surface reconstruction from point clouds by transforming the medial scaffold. Comp. Vision Image Underst., 113(11): 1130-1146.

Dey, T.K. and S. Goswami, 2003. Tight cocone: A watertight surface reconstructor. J. Comp. Inform. Sci. Eng., 3(4): 302-307.

Dey, T.K. and S. Goswami, 2006. Provable surface reconstruction from noisy samples. Comp. Geometry, 35(1-2): 124-141.

Dey, T.K., J. Giesen and J. Hudson, 2001. Delaunay based shape reconstruction from large data. Proceeding of the IEEE Symposium on Parallel and Large-data Visualization and Graphics, Ohio State Univ., Columbus, OH, USA, pp: 19-146.

Di Angelo, L. and L. Giaccari, 2011. A fast algorithm for manifold reconstruction of surfaces. Proceeding of the International Conference on Innovative Methods in Product Design, pp: 177-186.

Di Angelo, L., P. Di Stefano and L. Giaccari, 2011. A new mesh-growing algorithm for surface reconstruction. Comp. Aided Design, 43(6): 639-650.

Edelsbrunner, H. and E. Mucke, 1992. Three-dimensional alpha shapes. Proceeding of the Workshop on Volume Visualization, New York, USA, pp: 75-82.

Guttman, A., 1984. R-trees: A dynamic index structure for spatial searching. Proceeding of ACM SIGMOD International Conference on Management of Data, New York, USA, pp: 47-57.

Huang, J. and C.H. Menq, 2002. Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology. Comp. Aided Design, 34(2): 149-165.

Li, X., C.Y. Han and W.G. Wee, 2009. On surface reconstruction: A priority driven approach. Comp-Aided Design, 41(9): 626-640.

Lin, H.W., C.L. Tai and G.J. Wang, 2004. A mesh reconstruction algorithm driven by an intrinsic property of point cloud. Comp-Aid. Design, 36(1): 1-9.

Ou Yang, D. and H.Y. Feng, 2011. Reconstruction of 2D polygonal curves and 3D triangular surfaces via clustering of Delaunay circles/spheres. Comp. Aided Design, 43(8): 839-847.

Sarkar, M. and T. Leong, 2000. Application of k-nearest neighbors algorithm on breast cancer diagnosis problem. Proceeding of the 2000 AMIA Annual Symposium.

Wei, S., 2008. Novel and fast mapping triangulation algorithm for unorganized points cloud. Optic. Eng., 47(11): 117205-117211.