2012 International Workshop on Nonlinear Circuits, Communications and Signal Processing
NCSP'12, Honolulu, Hawaii, USA. March 4-6, 2012

Facet-Growing Approach to Point Cloud Segmentation Using Local Convexity and Octree

Huan Fu[1,2,a] ,Fei Wang[1,b] , Yoshifumi Nishio[3,c] , Yongjian He[1,4,d], Chengwei Hu[5,e]

[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
[2]Institute of Computer Software and Theory, Xi'an Jiaotong University, China
[3]Department of Electrical and Electronic Engineering, Tokushima University, Japan
[4]Xi'an Communication Institute, China
[5]Academy of Space Technology, China
[a]fuhuan.2009@stu.xjtu.edu.cn, [b]wfx@mail.xjtu.edu.cn, [c]nishio@ee.tokushima-u.ac.jp,
[d]heyj@aiar.xjtu.edu.cn, [e]cwh@aiar.xjtu.edu.cn

Abstract An novel method using local convexity and octree is proposed to segment irregular 3D point cloud into surfaces. An modified octree method is used to produce the init facet set which usually contains too many facets and is meaningless, and then the connection of the leaf nodes is found. Finally, the adjacent facets are combined if they satisfied the local convexity. The proposed method is applied to the point cloud of the stone fractured from a tower and the results are discussed.

Keywords Facet-Growing, Point Cloud Segmentation, Local Convexity, Octree

1. Introduction

Point cloud segmentation is an important step of many difficult problems such as object recognition, object registering and so on. In different kinds of applications, we need to handle a large amount of points' data to get different segmentation results. For example, in the application of dealing with laser scanning from planes, we only need to split the roads, buildings from the point cloud. But in another application, we may need to get the tree and its branches from street scene's laser scanning. In our article, we wanted to do more detailed work, which is to partition point cloud of stone with rough appearance into several surfaces.

Traditional point cloud segmentation methods can be divided into three categories: (1) using geometry and topology characteristics for surface segmentation. The common features used in this kind of methods are normal vector, curvature and their derivatives. (2) Get the definition and feature of characteristics in 2D space, extended to 3D space, get its new definition and new feature, and use it to point cloud segmentation. (3) Use the theory in other subject and get similar conclusion in point cloud segmentation. All this methods are well applied in surface segmentation of specific requirements. They platform nearly perfect on relative smooth point cloud, but be terrible on coarse one. Most of the point cloud we get from reality is coarse, so how to handle the coarse point cloud is the main problem we should solve. In this article, we propose a new method to deal with questions of this kind.

In general, the workflow of our approach for surface segmentation from point cloud can be separated in three steps: normal vector estimation, initial segment by octree and facet-growing using local convexity. First, Normal vector estimation is to calculate the normal vector of each point by compute the average of the small triangles the point in. Then, the points are assigned to octree's leaf nodes, and every node contains much information about the points. Finally, we merge two connected nodes by judging whether they satisfied the local convexity requirements.

Our segmentation algorithm is present in Section 2. In Section 3, we display our experiment result and discuss the error and performance of our approach.

2. Our Approach to Surface Segmentation

2.1. Estimation of Normal Vector

Point cloud is composed of a series of scattered points in 3D space, there is no connection between points. So it's impossible to calculate the normal of every point directly. To estimate the normal vectors of the points, the data first need to be triangulated. And the common Delaunay triangulation method is used in our approach. After this step, every point is assigned to at least one triangle, the normal vector of every triangle can be got easily. What we need to do is the compute the normal of point by the triangles include that point.

The normal value of point P is estimated from a group of triangles that using point P as one of their vertexes. Fig.1 shows point P and its triangles. The normal value of point P and normalized pint normal at the same point are calculated by Eqs. (1) and (2), respectively:

$$n_p = \frac{\sum_{i=1}^{m} n_i}{m} \tag{1}$$

$$\hat{n}_p = \frac{n_p}{|n_p|} \tag{2}$$

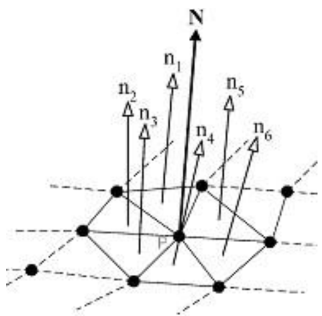


Fig.1 Calculation of the normal value of a point.

As the figure above described, the normal of P is calculated by getting average of 6 triangles.

After this step, the normal of each point is calculated without orientation. And what we should do next is to decide the orientation of the normal. In our approach, the simplest way to solve this problem is to connect the normal to the position of the point in the point cloud. In short, if the point is in upper half of the point cloud's least bounding box, the z normal components must be positive, and be negative in the other case. According to this criterion, we can easily correct the normal vector of each point.

2.2. Initial Segment Using Octree

The scan data after registration keep the normal values for each point. Because the data is coarse, conventional ordering and merging procedure will not play well. In our approach, we apply the octree method to avoid these steps and get the initial segment. This method can be represented as a tree structure, as show in Fig.2. In the octree method, each cubical octant in the pyramidal structure has eight daughters created by halving the parent cell along x, y and z directions based on the criteria defined by the application.

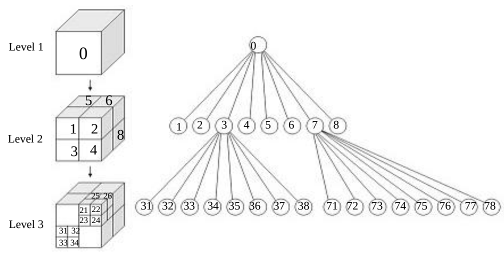


Fig.2 Octree Structure

In this research, the octree method is used for segmenting the point cloud data. The main steps are executed as follow: Firstly, find out the maximum and minimum at each axes, which are marked as $x_{\max}$, $y_{\max}$, $z_{\max}$ and $x_{\min}$, $y_{\min}$, $z_{\min}$, and then, we can get the minimum bounding box of the point cloud. The eight vertexes of the box store in the root node as the frame points, the whole points of the cloud also store in the root as its content point, the average normal is calculated by Eq.(3) as its value.

$$\bar{n} = \sum_{i=0}^{m} n_i / m \qquad (3)$$

Where n_i is the unit normal vector of point i in a node and m is the number of points in the cell.

There are two factors to decide whether a node is continued to be partitioned into eight daughters. One is the standard deviation of point normal vectors, which is calculated by:

$$\sigma = \left| \frac{\sqrt{\sum_{i=1}^{m} (n_i - \bar{n})^2}}{\sqrt{m}} \right| \qquad (4)$$

When the standard deviation is larger than the user-defined tolerance, the node is subdivided into eight. The other one is the number of points in each node. If the number is smaller than the user-defined tolerance, the subdivided procedure stops. The target of the initial segmentation is to get the points into groups with similar property and being connected in geometry. So we set these two parameters: N_{num} and δ_{node} to build the octree.

The ideal result of initial segmentation is that, the standard deviation all leave nodes of the tree is under the user's definition, and the number of points is nearly the same, the tree is a three-level tree. Because of the data, its result is terrible. So we do our best to make the distribution of the point be equal, and platform the next step to optimize the result.

2.3. Facet-Growing Approach

The steps above assign the points into several nodes with a feature value. Then our task is to merge this node with taking advantage of local convexity.

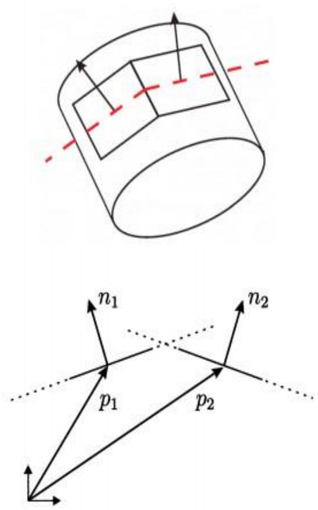


Fig.3 Local convexity holds if the center point of a surface is below the other surface

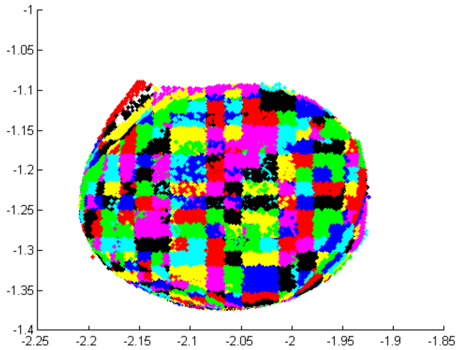
As illustrated in Fig.3, two neighboring surfaces s_i and s_j are termed locally convex to each other, if center point $\vec{p}_i$ lies below the surface s_j. In another word, if their normal vectors approximately have the same direction, two surfaces are considered as locally convex.

The connections between nodes are divided into three classes: facet-connect, line-connect and point-connect, which are judged by the relationship of the minimum bounding boxes of surfaces.

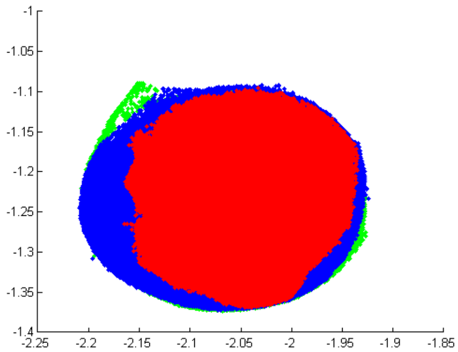
After this merge procedure, the result will be good enough for our application.

3. Experiment and Conclusion

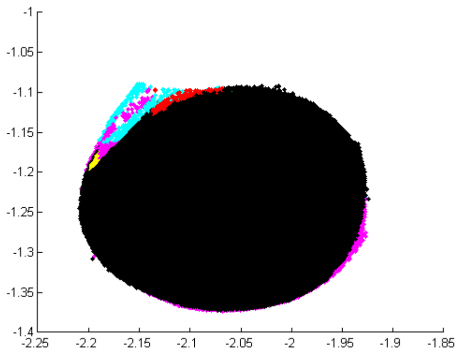
Our approach is used on the point cloud of the stone. And the result is shown as follow. Our method of merging facets can reduce the number of surfaces effectively. And the strategy we used will be different in different application. The Fig.4 shows the result of our experiment. The data we used is a point cloud consisting of more than 100 thousand points.



(a)



(b)



(c)

Fig.4 a) The initial segmentation has 671 facet. b) Get two seeds, and merge the facets into 3 surfaces. c) Combine the facets iteratively and get 10 surfaces.

In this paper, we present a new way to handle irregular point cloud data. But because the data cloud we used is dense in the top surface and sparse at other position. The traditional way to get the normal of the point on the bottom and side faces may has terrible errors, and it's hard to split the side part into surfaces accurately. The next step in our research is to find a new way to estimate the normal of point and find a new way to explain the connection between nodes.

References:

[1] H.Woo,E.Kang,Semyung Wang and Kwan H.Lee. A new segmentation method for point cloud data. International Journal of Machine Tools & Manufacture 42:167-178, 2002.

[2] M.Yang,E. Lee. Segmentation of measured point data using a parametric quadric surface approximation. Computer-Aided Design 31:449-457, 1999.

[3] Frank Moosmann,Oliver Pink and Christoph Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. IEEE Intelligent Vehicle Symposium, 2215-220, 2009.

[4] A.Jagannathan and E.L. Miller. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. IEEE Transaction on Pattern Analysis and Machine Intelligence. 29(12):2195-2204, 2007

[5] D. Steinhauser, O. Ruepp, and D.Burschka. Motion segmentation and scene classification from 3D LIDAR data. Intelligent Vehicles Symposium, 398-403, 2008.

[6] X. Zhu, H. Zhao, Y. Liu, Y. Zhao, and H. Zha. Segmentation and classification of range image from an intelligent vehicle in urban environment. IEEE International Conference on Intelligent Robots and Systems, 1457-1462, 2010.

[7] R. Mario and V. Markus, Point Cloud Segmentation Based on Radial Reflection Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, 5702:955-962, 2009.

[8] R. Rusu, A. Holzbach, N, Blodow and M. Beetz. Fast Geometric Point Labeling using Conditional Random Fields. IEEE International conference on Intelligent Robots and Systems, 7-12, 2009.

[9] C. Tai, M. Huang, The processing of data points basing on design intent in reverse engineering. International Journal of Machine Tools and Manufacture 40:1913-1927, 2000..

[10] C. S. R. Aguiar, S. Druon, and A. Crosnier, 3D datasets segmentation based on local attribute variation IEEE International Conference on Intelligent Robots and Systems, 3205-3210, 2007.