

3D Modeling Algorithm Based on Vehicle-Borne Laser Point Clouds

Xu-Dong Xu^{1,a}, Rui-Jie Wen^{2,b} and Ze Li^{3,c}

^{1,2,3}College of Computer Science, Beijing University of Technology, No.100 Pingleyuan Street, Chaoyang District, Beijing, China

^axuxudong@bjut.edu.cn, ^bmiss_wendy123@163.com, ^c718406390@qq.com

Keywords: Vehicle-Borne Laser Point Clouds; Bounding Box; Octree; Modeling

Abstract. Three-dimensional laser scanning can be used to collect the 3D coordinate information on the object surface to create 3D point clouds, which can reflect the characteristics of the target surface, facilitating the establishment of terrain model. Since point clouds lacunarity is usually caused by occlusion of ground features during the vehicle-borne laser scanning, the existing modeling algorithms are not fully suitable for vehicle-borne laser point clouds. In this paper, a modeling algorithm was designed to be different from the previous ones, in which the model was built by constructing Delaunay triangulation. Instead, a bounding box algorithm was adopted for the modeling of vehicle-borne laser point clouds, which were structured in an octree. This could not only fill the point cloud holes found in modeling, but also improve the storage efficiency of the point clouds.

Introduction

Three-dimensional laser scanning can be used to collect massive 3D point data on the target surface in a continuous, automatic, fast manner, with its unique advantages [2]. However, the load of the vehicle-borne laser point clouds is the vehicle, the actual operations of which will influence the laser point clouds. When a vehicle is moving forward, the sparsity degrees of the resulting point clouds vary at different vehicle speeds. Specifically, the faster the vehicle moves, the sparser the point clouds are, and vice versa. And, in the case of multiple occlusions, point cloud holes can be observed.

In actual production, several point clouds reverse engineering modeling software may be commonly used, such as surfacer, Imageware, Verdict, and Strim. Although all of them can generate desired results of point clouds modeling, these software might not be suitable for point clouds that are uneven and sparse with holes [1]. The other methods of point clouds modeling show various specificities and features in obtaining point clouds with different loads. As the automobile laser scanning has been more and more widely applied, it is undoubtedly of great practical value and theoretical significance to find a modeling method for the vehicle-borne laser point clouds.

In this paper, a bounding box algorithm was designed on the basis of octree for the modeling of the vehicle-borne laser point clouds, modeling the facades of object. The general idea was as follows: According to the obtained vehicle-borne laser point clouds, first, the data scope and partition were determined. Then, by determining the presence of point clouds in the partitioned area, the edge information of the bounding box was taken or kept in the area. The above process would be repeated until the shape of the bounding box was close to the vehicle-borne laser point clouds, and thus the modeling of the point clouds would be completed.

Major steps of modeling: First, the obtained vehicle-borne laser point clouds were pre-treated, to avoid low experimental efficiency due to scattered data and the lack of patterns; next, the vehicle-borne laser point clouds were restructured with the octree and partitioned by using the features of the spatial data; as parts of the obtained point clouds of the object surface might be absent, the lacunarity should be filled, so as to render the modeling more realistic; at last, for the purpose of reducing the resource utilization and thus improving the storage efficiency during the output of the point clouds model, only the edge information of the object surface was stored, so the redundant points on the object surface should be removed.

Preprocessing of Point Clouds

Denoising of point clouds

In obtaining the point clouds, some noises and outliers will be inevitably found in the point clouds, due to the impacts of equipment accuracy, operator experience, and environmental factors. As a crucial step in the processing flow of point clouds, filtering can greatly influence the subsequent processing operations.

The common ordered or partially ordered methods of data denoising include median filter, mean filter, and Gaussian filter algorithms.

For disordered point clouds, there are two methods: (1) Turning the scattered point clouds into a grid model, which is then used for denoising; and (2) directly denoising the disordered, scattered point clouds. Common denoising methods for scattered point clouds include Laplace algorithm, bilateral filtering algorithm, mean curvature flow filter, and mean shift algorithm [4].

Thinning of point clouds

A huge quantity of point clouds can be obtained through vehicle-borne laser scanning, causing unnecessary waste of resources. For this reason, the vehicle-borne laser point clouds should be thinned before modeling. Common thinning methods systematic thinning, thinning based on grid, and thinning based on TIN.

However, both the systematic thinning and that based on grid are random thinning methods that cannot maintain the characteristics of the terrain and buildings, leading to distortions. In the thinning based on TIN, the ground features like the road surface and buildings are fully taken into consideration, and the data points of flat surfaces are thinned, thereby reducing the redundancy of the data and impacts on the data accuracy [3].

Restructuring and Partitioning of Point Clouds

Comparison between k-d tree and octree

Both k-d tree and octree are usually used to process the 3D spatial data. The k-d tree is a data structure of points in the k-dimensional Euclidean space, and it is a binary tree which has all of its nodes in the k-dimension. The k-d tree can partition the data points in the k-dimensional space and is largely used in searching critical data in a multi-dimensional space. The k-d tree structuring the point sets (2,3),(5,4),(9,6),(4,7),(8,1),(7,2) is shown in Fig.1.

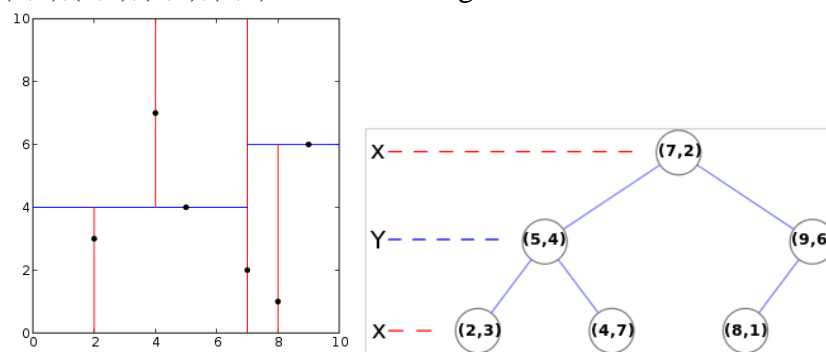


Fig.1 The k-d tree

The octree structure is developed directly from the quadtree. Its principle is to constantly divide a spatial region (a six-sided cube) into eight subregions (smaller cubes) in the same size. The more the number of division, the smaller the subregions. The process continues until there is only one single property in a region [6]. Each node of the octree represents a cube of volume element, and each node has eight children, the summed volume of which is equal to that of the parent node. The octree is shown in Fig.2.

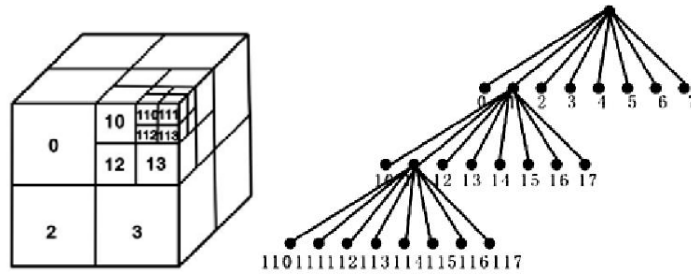


Fig.2 The octree

In general, the k-d tree is more efficient than the octree in data partition. The advantages of k-d tree over the octree is that its data partition can be carried out according to the structural characteristics of the data, particularly effective for irregularly scattered data.

However, for the vehicle-borne laser point clouds, after the preprocessing, the data are so even that the k-d tree's advantage is not obvious. Moreover, in filling the holes, the structure of octree can achieve a very regular partition of all the vehicle-borne laser point clouds, making it more easy to structure the data and to locate the filled coordinates.

Structuring point clouds with octree

In this study, the data were structured and stored in an octree. This approach could not only improve the storage efficiency, but also provide strong support for the subsequent holes-filling and redundancy-removing.

The octree was structured with the vehicle-borne laser point clouds as follows:

- (1) Set the maximum recursion depth;
- (2) Find the maximum size of the scene to establish the minimum enveloping cube;
- (3) Put the vehicle-borne laser point clouds into the current child;
- (4) If the maximum recursion depth is not achieved, partition the leaf node into eight children in the same size, and assign the cube element to the eight children;
- (5) If the elements in the child are the same with the parent node, then stop partitioning this child;
- (6) Repeat (3) until the maximum recursion depth is achieved.

According to the above steps, the preprocessed vehicle-borne laser point clouds could be restructured, and then the point clouds were partitioned.

Given point clouds $P=\{P_i\}(i=1,2,\dots,n)$, $P_i=\{X_i,Y_i,Z_i\}$. First, the entire point cloud was traversed to get the maximum and minimum values on the directions of axis-x, -y, and -z, i.e. X_{min} , X_{max} , Y_{min} , Y_{max} , Z_{min} , Z_{max} . And the cuboid bounding box that initially enveloped the entire point set were structured with these data and used as the root node of the octree; then the bounding box was partitioned respectively in the axis-x, y, and z, that is, $X=X/2$, $Y=Y/2$, $Z=Z/2$; a new bounding box was determined by the newly obtained coordinates. If there were point clouds in this bounding box, then it was kept, and vice versa. The partitioning of the octree was shown in Fig.2.

The termination condition of the partitioning could be controlled by adjusting the size of the minimum enveloping box, which could be determined as follows:

$$L=2*P_i*R/f \tag{1}$$

Where R denotes the distance from the laser scanner to the object, and f denotes the number of the laser pulses emitted during one revolution of the scanner. Given that the laser points fell right between the two scanned boxes, the actual length l of the minimum enveloping box should be greater than the calculated theoretical value L, but smaller than twice of L, that is, $L<l<2L$.

Lacunarity-Filling of Point Clouds

In the vehicle-borne laser scanning, data of the scanned object surface might be lost, mainly

due to the following aspects:

1) The occlusion caused by the object's own spatial topological structure. For instance, the buildings are often occluded by trees. This is the focus of lacunarity-filling.

2) The point cloud holes caused by laser scanner's own defects and errors. For instance, an inadequate scanning distance. In this case the device should be improved [5].

3) The poor reflection property of the scanned object surface. For instance, the specular reflection generated by glass. But it has no effect on modeling and can be excluded.

At present, only the first type of point cloud holes should be filled. Based on the topological relationship of the point clouds, the point clouds could be divided into the following types, according to their recorded number in each bounding box, K :

- 1) $K=1$, the point was on a corner of the object and should not be filled.
- 2) $K=2$, the point was on an edge of the object and should not be filled.
- 3) $K=3$, the point was on a depression of the object and should be filled.
- 4) $K=4$, the point was on a surface of the object and should not be filled.

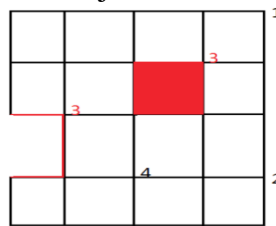


Fig.3 Areas to be filled on the object surface

Therefore, only the third type of points should be filled. By traversing the entire point cloud, the point with $K=3$ was found and filled. In this process, if a new point with $K=3$ was generated, the filling should be repeated. The hole filling was shown as follows:

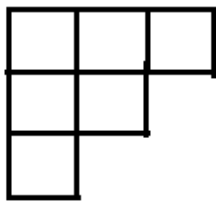


Fig.4 Before filling lacunarity

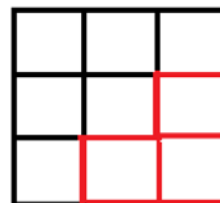


Fig.5 After filling lacunarity

Redundancy-Removing of Point Clouds

We only had to obtain the corner and edge information of the buildings, and the remaining ones were redundant, leading to unnecessary resource utilization and lowering the storage efficiency. To fix this, the following solution was taken: since one point might be recorded in different bounding boxes for several times, the redundancy could be removed by eliminating the points that had been repeatedly recorded.

According to the recorded number of one point in different bounding boxes, M , the redundant points could be divided into the following types: $M=4$, the point was on the surface of the building; $M=6$, the point was on the depression of the building; $M=8$, the point was inside the building; and these redundant points should be eliminated. The removing of redundant points was shown as follows:

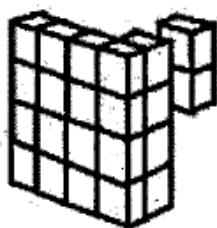


Fig.6 Before removing redundant points

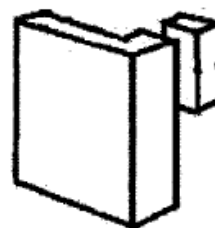


Fig.7 After removing redundant points

The unnecessary topological relationship between the children and their parent nodes in the octree were terminated, in order to reduce the size of data stored and improve the storage efficiency of the model. Each child was determined to be a redundant point or not, according to its recorded number of one point in different bounding boxes, M . If it was not a redundant point, it should be kept; if it was, the newly generated nodes and topological relationship should be recorded to the original parent nodes, and the children of this parent node should be simultaneously deleted.

Conclusion

Based on the above theoretical research, by using the development tool Visual Studio and the graphics development package OpenGL, the modeling of the point clouds was achieved by the vehicle-borne laser. In particular, Fig.8 shows the original data, while Fig.9 shows the result of processing with the above algorithm.

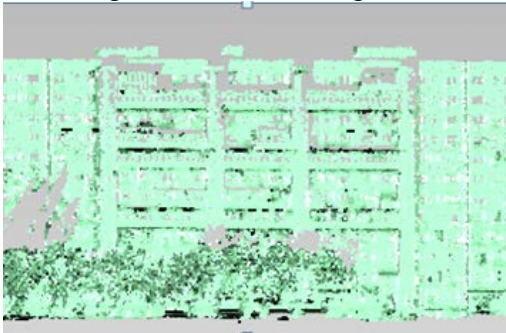


Fig.8 Original point clouds

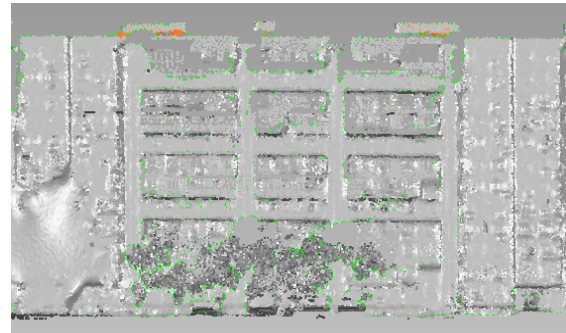


Fig.9 Experimental result

The proposed modeling algorithm of the bounding box based on octree was feasible. The algorithm could not only fill the point cloud holes during the modeling, but also improve the storage efficiency of the point clouds.

However, although the above algorithm is suitable for the building facades in regular shapes, its modeling result may deviate from the actual shape for irregular discrete point clouds. The solution to this problem is to first categorize the vehicle-borne laser point clouds before the modeling steps with the above algorithm.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under grant No.41371434.

References

- [1] H.X.Zhang, Y.J.Chen, and G.B.Liu. Campus building modeling based on 3D laser scanning[J]. *Engineering of Surveying and Mapping*, 2010, 19(1): 32-34.
- [2] J.Wu, F.X.Jin, and J.Wang. 3D modeling of buildings based on 3D laser scanner data[J]. *Engineering of Surveying and Mapping*, 2007,16(5): 57-60.
- [3] A.M.Deng. The method of vehicle-borne laser point cloud data streaming thinning[D]. Southwest Jiaotong University,2011.
- [4] J.L.Dai. Preprocessing algorithm of massive point cloud[D]. Zhejiang University, 2006.
- [5] B.Dai. Research on 3D reconstruction of data based on vehicle-borne laser scanning[D]. Capital Normal University, 2011.
- [6] K.Yang, Z.Y.Zhang, and Y.Dong. Surface model reconstruction based on point cloud subdivision with adaptive octree[J]. *Computer Applications and Software*, Vol.30, No.6, 2013.