Research on Surface Reconstruction from 3D Scattered Points

Fu Yan[1], Zhai Jin Lei[2]
Xi'an University of Science and Technology
Xi'an, Shaanxi 710054, China
Zhai Jin Lei
zhaileicool@163.com

ABSTRACT: *The Power Crust algorithm is widely used in the 3D reconstruction. But when we directly use it in the process of reconstruction, there are some problems, such as the large calculating quantity and poor denoising ability, etc. This paper puts forward a new method to solve the problems of the Power Crust algorithm. In stage I, we use the KD tree algorithm to get every point's neighborhood information within the points cloud. In stage II, we use the Laplace method to denoise the points sets. In stage III, we use the neighborhood average method to simplify the points sets when the points are too many. At last, we use the Power Crust algorithm to accomplish the reconstruction. The result is a 3D model constructed from the simplifying points sets. In addition, compared with the previous method, this paper's method systematically solves the denoising problem and simplifies the points cloud. It has a fast reconstruction speed and a good reconstruction model.*

Keywords: The Power Crust algorithm; KD tree algorithm; The Laplace method; The neighborhood average method

Received: 15 August 2016, Revised 26 September 2016, Accepted 30 September 2016

1. Introduction

3D modeling [1] plays an essential role in developing various applications including urban planning, cartography, environmental monitoring, medical imaging and computer gaming. The challenge of developing fast and automated 3D model reconstructions has attracted many researchers due to its vast potential and significance in many real world applications. The data acquisition for 3D models, however, is complicated and time consuming. Existing large scale systems can take several months to create 3D models and require a lot of manual intervention. In addition, this process is computationally expensive and not suitable for applications like 3D rendering, so it is necessary to study the technology of reconstruction from the 3d scattered points.

In the field of surface reconstruction of 3d scattered points, based on the Delaunay triangulation and Voronoi diagram algorithms are few theories that have a prove. There are three main kinds of this kind of algorithm: Crust algorithm, Power Crust algorithm and Cocone algorithm. The advantage of the Crust algorithm [2] is that the output of the discrete surface in the detail area has dense points and in the absence of characteristic area only has sparse points. Its weakness is large amount of calculation. These years, some researchers propose some new methods about 3d reconstruction like Amenta's Power Crust[3] algorithm. The Power Crust algorithm uses the power cell instead of tetrahedron. This paper mainly solve the problems when use the Power Crust algorithm that large computation, low reconstruction speed and rough reconstruction result. This paper gets the neighbors of any point within the points sets through constructing the KD tree of points sets, then uses the Laplace method to denoise and the neighborhood average method to simplify the points sets to improve the reconstruction speed. At last, we use the Power Crust algorithm to reconstruct the points cloud. The experiment result shows that the method in this paper can deal with the large amount of points. In addition, the speed and reconstruct effect is better than directly using the Power Crust algorithm.

2. The KD Tree Algorithm

The KD [4] tree is a well-known space-partitioning data structure for organizing points in k-dimensional space. As an acceleration structure, it has been used in a variety of graphics applications, including triangle culling for ray-triangle intersection tests in ray tracing, nearest photon queries in photon mapping, and nearest neighbor search in point cloud modeling and particle-based fluid simulation.

In order to deal with the points cloud before reconstruction, we should get the neighborhood relationship information of the points sets. There are mainly two ways to do it, either KD tree algorithm or octree algorithm. Because the KD tree is easy to construct and has high search speed. We choose the KD tree method in this paper to get the neighbors of any point in the points sets.

The process of construction the KD tree:

1. According to the *x* value, we sort the points sets and get the middle point p_0 , then insert it into the root of the KD tree.

2. Respectively sort the left and right side of p_0 according to the *y* value and get the middle point p_1 and p_2, then insert them to the left and right child of p_0.

3. Respectively sort the left and right side of p_1 and p_2, according to the *z* value and get middle point p_3, p_4, p_5, p_6, then insert them to the left and right child of p_1 and p_2.

4. In a similar way,we continuously partition the points sets based on the *x*, *y* and *z* value until every child space only has one point, then we will have constructed the whole KD tree.

3. The Laplace Method

These years, domestic and foreign scholars put forward a variety of smoothing denoising algorithms of points cloud model. Most of these algorithms [5] are from denoising algorithm in image processing and mesh smoothing algorithms. From the algorithm complexity analysis they can be divided into the method based on optimization method, simple non_iterative method and the method based on Laplace. Because the Laplacian operator is easy to implement and can achieve the purpose of smoothing denoising. So this paper uses Laplace algorithm to deal with the noise in point cloud model.

Laplace algorithm description:

1. According to section 2, we build the KD tree and get any point's neighbors.

2. According to the type (1), we deal with each point in the points cloud until all points have been processed.

$$L(p_i) = p_i + \lambda\left(\frac{\sum_{j=1}^{k}\omega_j q_j}{\sum_{j=1}^{k}\omega_j} - p_i\right) \tag{1}$$

The p_i is the point that is to be processed. q_j is any point in the neighbors of p_i. ω_j is the weight value of q_j. λ is a small number. Laplacian filtering moves the current point to the geometric center of the neighborhood through many iterations to get the purpose of denoising.

4. Simplify the points cloud

Scattered points cloud to streamline is the key technology of post-processing like data management, points cloud visualization, etc. To simplify the points cloud, it can improve the calculation efficiency and reduce the storage space. This article adopts the method of proposed in literature [6] to streamline the points cloud. This method is similar to the principle of image median filtering, its implementation is simple and can meet the requirements of general point cloud data sets. Firstly it requires the user to define a sampling cube of length d, as well as percentage ϕ. Set point P is any point within the point set. The sampling cube Q has a points set: $Q = \{Q_i(x_i, y_i, z_i), i = 1, 2, ..., n\}$. Suppose that the coordinates of point P is P_x, P_y, P_z then any point located at the cube satisfies these conditions:

$$P_x - \frac{d}{2} \le Q_{ix} \le P_x + \frac{d}{2} \tag{2}$$

$$P_y - \frac{d}{2} \le Q_{iy} \le P_y + \frac{d}{2} \tag{3}$$

$$P_z - \frac{d}{2} \le Q_{iz} \le P_z + \frac{d}{2} \tag{4}$$

Calculate the distance between the point P and any point Q_i with the points set.

$$\left|PQ\right| = \sqrt{(P_x - Q_{ix})^2 + (P_y - Q_{iy})^2 + (P_z - Q_{iz})^2} \tag{5}$$

Add the distance value:

$$D = \left|PQ_1\right| + \left|PQ_2\right| + ... + \left|PQ_n\right| \tag{6}$$

Get the average value:

$$\overline{D} = \frac{D}{n} \tag{7}$$

Implement the above calculation for all points. At last, according to the user-defined data lean percentage ϕ, we will delete the percentage ϕ points that its average distance is smaller than others. So we can simplify the point cloud.

5. Power Crust

The Power Crust algorithm generates the surface meshes and approximate axis of the object from the points sets. This algorithm has a strong theoretical support. It can get the object surface from any points sets. At the same time it can also fill the hole in the cloud. It has a quite strong robustness. This algorithm has a good treatment effect for the points sets with uneven density and high noise. The computation can be broken down into several steps, as shown in figure 1.

Power Crust Algorithm steps:

1. Do the Delaunay triangle subdivision of the points sets and compute the Voronoi graph.

2. Compute the pole point of each sample point.

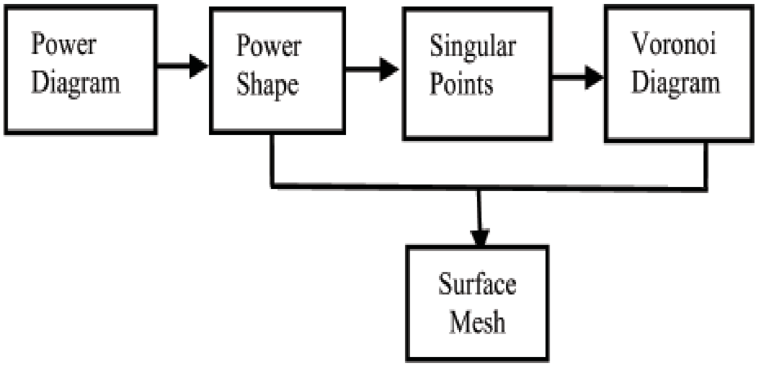

Figure 1. Amenta and Kolluri’s algorithm

3. Calculate the pole’s Power graph.

4. Mark the pole point belonging to the internal or external.

5. Separate the plane of Power graph of internal and external pole point and generate the approximate surface.

More details on the algorithms by Amenta et al., including the theoretical derivation, sampling assumptions, and convergence guarantees can be found in[3].

6. Experiment and Analysis

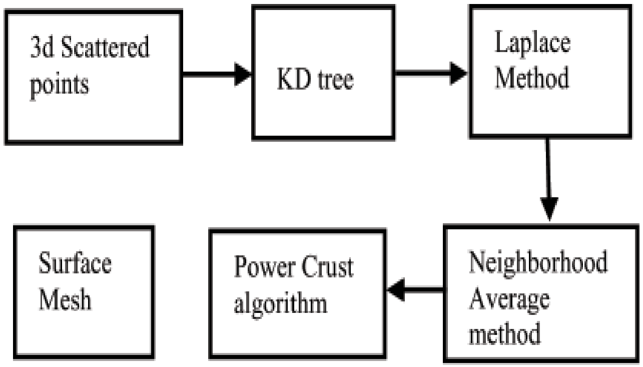

Figure 2. The whole process of algorithm in this paper

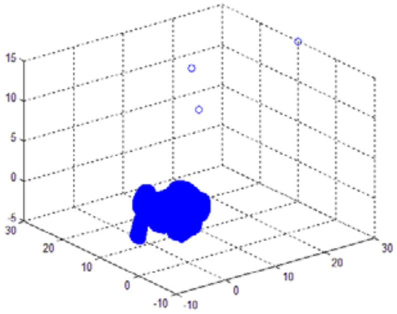

Figure 3.1 Bunny points cloud

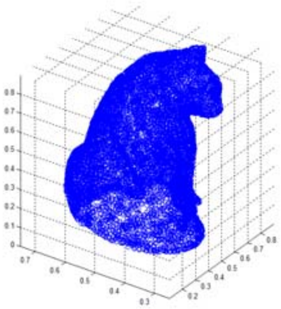

Figure 3.2 Cat points cloud

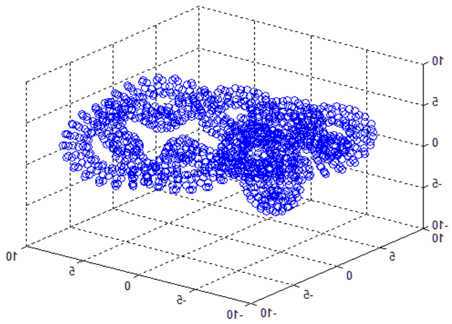

Figure 3.3 Knots points cloud

Figure 3. The different points cloud

Figure 4.1 Bunny model

Figure 4.2 Cat model

Figure 4.3 Knots model

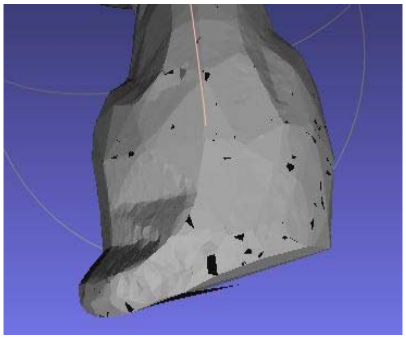

Figure 4.4 Local graph with noise

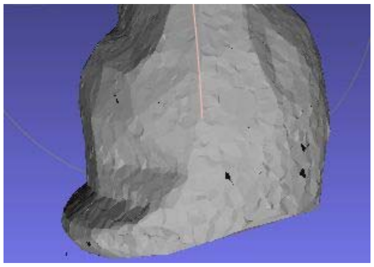

Figure 4.5 Local graph after denoising
Figure 4. The different 3D models

Model	Bunny	Cat	Knot
Points	34838	10000	1218
Simplification	22000	8000	800
Power Crust(sec)	40.23	20.62	1.22
This Paper(sec)	20.58	9.86	0.68

Table 1. The result of the experiment

6. Conclusion

We get the neighborhood information of every point through building the KD tree of the scattered points sets, then use the neighborhood average method to eliminate the noise points of the cloud. We use the neighborhood average method to simplify the points cloud. At last, we adopt the Power Crust algorithm to reconstruct the surface meshes. The algorithm in this paper can deal with some complex shape structure and high noise points sets. It has a high robustness and it can truly reconstruct the appearance of original object.

References

[1] Duraisamy, Prakash. (2012). 3D Reconstruction using LIDAR and visual images. Universoty of NORTH TEXAS.

[2] Song Da Hu, Li Zhong Ke, Sun Yu Chun. (2013). Fast surface reconstruction algorithm for 3D scattered points. Application research of Computers.

[3] Amenta, Choi., Kolluri, (2001). The Power Crust, 6th ACM Symposium on Solid Modeling.

[4] Kun Zhou, Qiming Hou, Rui Wang. (2008). Real-Time KD-Tree Construction on Graphics Hardware.

[5] Lin, Sun Zheng. (2011. Research on Filtering Method of Tree-Dimensional Laser Scanning Point Cloud Data. China Univ of Central South University.

[6] Jun, Wan., Yue, Ju Lu. (2004). Research on streamline method of Point cloud data in reverse engineering. *Journal of Shanghai University.*