# Simplification of 3D point cloud data based on ray theory

**Changsu Liao[1], Xiaojing Niu[1], Meili Wang[1*], Dongjian He[2]**

[1]*College of Information Engineering, Northwest A & F University, Yangling, Shaanxi 712100, China*

[2]*Mechanical and Electronic Engineering, Northwest A & F University, Yangling, Shaanxi 712100, China*

*Received 1 July 2014, www.cmnt.lv*

**Abstract**

To effectively reduce the amount of 3D point cloud data, whose shape is symmetrical or spherical, this paper proposes an efficient simplification algorithm based on ray theory. Meanwhile, a boundary retention method based on the distribution uniformity of neighbouring data points is used to keep the model complete. Avoiding time-consuming recursion and curvature estimation, the proposed method is much efficient and achieves good simplification results.

*Keywords:* 3D Point Cloud, Data Simplification, Ray Generation, Boundary Retention

## 1 Introduction

In recent years, with the decrease in cost and the increased precision of 3D scanners, 3D point clouds have become an important data representation form in graphics, reverse engineering, and industrial fields [1]. However, the huge of original 3D point cloud data causes great difficulties in 3D reconstruction. As a result, denoising and simplification of the 3D point cloud are important pre-processing steps [2]. Algorithms for simplifying 3D point cloud data include simplification based on clustering [3-5] and simplification based on curvature [6-8], amongst others. Although these methods can reduce the size of the point cloud, retain characteristics of the point cloud model, and increase the reconstruction efficiency of the model to some extent, most of them require recursive calculations and curvature estimation. Thus, these algorithms are quite time-consuming, especially for models with enormous point cloud data.

This paper aims at reducing 3D point cloud data whose shape is symmetrical or spherical, such as head, fruit, mechanical devices et al. In this paper, in order to simplify 3D point cloud data, we assume, based on the ray principle, that the centre of a 3D point cloud model generates several rays, and that data locates at a certain distance from the rays will be reduced. Meanwhile, a boundary retention method based on distribution uniformity of neighbouring data points is used to retain the boundaries of the model [9, 10], thereby guaranteeing the completeness of the data after simplification.

To conduct a rapid local search of data points, a kd-tree is constructed to establish the topological relations among the point cloud data. The organized point data structure of the kd-tree in a k-dimensional Euclidean space is a special binary tree. There are many studies on search of k nearest neighbours in a kd-tree, please refer to [11].

## 2 Foundation of the proposed algorithm

### 2.1 RAY PRINCIPLE

The ray principle is simple and easy to understand. First, suppose that the central point of the point cloud data generates rays evenly in all directions, as shown in Figure 1(a), with the rays filling the entire 3D space. Regarding the point cloud model (as shown in Figure 1(b, c)) in this space, if the distance from a certain point in the model to the nearest ray is smaller than a given distance, this point will be simplified. The denser the rays are, the greater the distance is, and the more data points in the 3D point cloud model can be reduced. Therefore, different degrees of simplification results can be achieved by controlling the number of rays and other parameters.

a. Ray in the space　b. The cat point cloud model　c. Schematic diagram of simplification method

FIGURE 1 Generation principle of rays

### 2.2 RAY GENERATION

Since the ray principle is the foundation of this paper, the simplification result of the point cloud model is directly related to the way that the rays are generated. Basically, two points make up one line, the centre of the 3D point cloud model is set as one point for all rays. Thus, the

*__Corresponding author__ e-mail *meili_w@nwsuaf.edu.cn*

coordinates of the other points must be generated according to a specific rule, to form all rays. For convenience of generating coordinate points, a minimum cube $V_m$ that encircles the point cloud model is first generated. Suppose that the length of this cube is $L$, and the maximum and minimum values of the scattered point cloud in the directions of the X-, Y- and Z-axes are $X_{\max}$, $Y_{\max}$, $Z_{\max}$, $X_{\min}$, $Y_{\min}$, and $Z_{\min}$, respectively. Then, the length of the cube is calculated as

$$L = Max((X_{\max} - X_{\min}), (Y_{\max} - Y_{\min}), (Z_{\max} - Z_{\min})). \quad (1)$$

A cube that entirely encircles the point cloud model is created by setting $L$ as the length and $(X_{\min}, Y_{\min}, Z_{\min})$ as the vertex. Then, the required points are generated on the six sides of the cube. By taking a plane parallel with the *XOY* plane as an example as shown in Figure 2, suppose that the two sides of this plane parallel with the X- and Y-axes are $L1$ and $L2$, respectively, and the end vertexes of the two sides are $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_2$ and $P_3(x_3, y_3, z_3)$, respectively. The coordinates of the generated point $P_i(x_i, y_i, z_i)$ are calculated as

$$x_i = (i/n) \times (x_1 - x_2) + x_2 \quad (i = 0,1,2...,n-1,n) \quad (2)$$

$$y_i = (i/n) \times (y_3 - y_2) + y_2 \quad (i = 0,1,2...,n-1,n) \quad (3)$$

$$z_i = z_1 = z_2 = z_3 \quad (i = 0,1,2...,n-1,n) \quad (4)$$

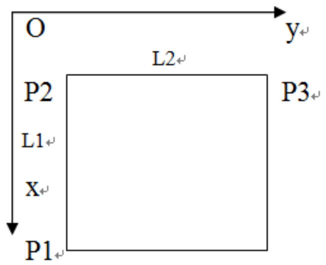



FIGURE 2 Schematic diagram of x-y plane of cube $V_m$

The number of points generated on each plane is $(n-1)^2$, and $(n+1)^2 + 4$ rays can be formed. In this way, the number of rays can be controlled by adjusting parameter $n$.

## 2.3 DETERMININATION OF THE DATA POINTS FOR SIMPLIFICATION

To increase efficiency, the equation of the line on which the ray is located is not calculated, instead, the distance from the point to the line is calculated using the relation between the vectors. Suppose that the direction vector of line $L$ is $s = (n, m, q)$, $M$ is a point on the line and $p$ is a point beyond the line. Thus, the distance from $p$ to $L$ is $d = \frac{|\overrightarrow{MP} \times s|}{s}$. If $d$ is less than the given distance $dis$, then $p$ should be reduced; otherwise $p$ will be retained. $dis$ is used as the judgment distance to control the degree of simplification. In this paper, locations far away from the central point, lines will be sparse, and thus the simplification degree will be low. Therefore, $dis$ should vary with a change in the distance from the data point to the central point. The greater the distance, the greater the value of $dis$ should be. Suppose that the central point is $Mid$ and the data point is $p_i$; $dis_i$ is used as the judgment distance when carrying out the simplification operation for each data point, $dis_i$ is computed by Equation (5) , where *x* is an input, used to control the value of $dis_i$ and the point cloud quantity is $Max$. Let

$$dis_i = x \times Distance(Mid, p_i)^2, \quad (i = 0,1,2...Max), \quad (5)$$

where $Distance(Mid, p_i)$ denotes the distance from $Mid$ to $p_i$. The simplification degree can be controlled by $x$.

## 3 Boundary retention

It is important to retain the boundary of the model during the data simplification procedure in that the boundary information is vital to keep the final reconstructed model complete. Thus, keeping the boundary points is also discussed in this paper.

FIGURE 3 Result without boundary retaining treatment

FIGURE 4 Result with boundary retaining treatment

Figure 3 and Figure 4 illustrate the data simplification results with and without applying the boundary retention method. It can be seen that using the boundary retention method, the boundary is complete, thereby improving the effects of data simplification effectively. There are proposed several methods for retaining the boundary points of a model. For example, when determining whether a point is a boundary point [12], several neighbouring points must be searched firstly using the topological relations among the point cloud, and then a least squares plane is constructed in the space based on the neighbouring points. Whether a point is a boundary point is determined by the distribution uniformity of the neighbouring point projection on the least squares plane. Meanwhile, the boundary retention degree can be controlled through a proportionality coefficient $e$. In this method, when calculating the distribution uniformity of the neighbouring point projection, either the standard deviation of the included angle must be calculated or the coordinate values must be compared. Besides, a boundary point judgment must be made for each data point.

We proposed a method to improve the method for determining boundary points by a way of comparing point cloud coordinate values. In this method, boundary point determination is not carried out for every data point, if a certain point is a non-boundary point, then several neighbouring points of this point are probably not boundary points either. We define that $Num$ denotes the number of point clouds in which neighbouring points are treated as non-boundary points without any judgment, if a certain point is detected as a non-boundary point. This value increases with an increase in the data size of the 3D point cloud model. Through combined control of the proportionality coefficient $e$ and $Num$, this method has little influence on the boundary retention effect. Figure 5 and Figure 6 show the results of making a boundary judgment, respectively, for each point or only for certain points using the proposed method. The value of the proportionality coefficient $e$ in Figure 5 and Figure 6 is 0.9 and 0.85, respectively, while the threshold value $Num$ is set to 1. It can be seen that there are no obvious differences in the results. However, execution of the former method takes 3.75 s, while the latter only requires 2.70 s. Therefore, the proposed method effectively improves the performance.

FIGURE 5 Result of making a boundary judgment for each point

FIGURE 6 Result of making a boundary judgment for some points

## 4 Simplification results and analysis

The experiments were conducted on Intel CORE i5-3210M, 2.50 Hz processor and 4.00 GB memory. Table 1 gives the test data for simplification under different parameters, while Figure 7 shows the corresponding simplification results, The proportionality coefficient $e$ is only used to control the boundary retention degree of the point cloud model, and therefore its value remains unchanged in general situations, $e$ is set to be 0.85.

TABLE 1 Test results under different parameters

| Test number | Model | Input points | $d$ | $x$ | Output points | Execution time (s) |
|---|---|---|---|---|---|---|
| test1 | Test model | 4000 | 10 | 0.11 | 2592 | 0.96 |
| test2 | | | 15 | | 2036 | 1.72 |
| test3 | | | 20 | | 1514 | 2.25 |
| test4 | | | 10 | 0.23 | 1729 | 1.19 |
| test5 | | | | 0.35 | 1125 | 1.21 |

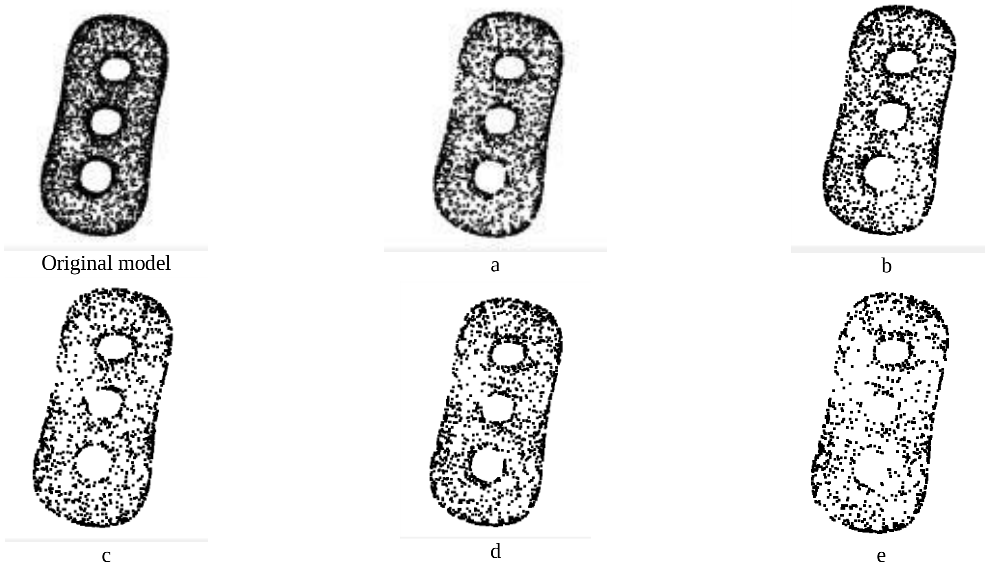



FIGURE 7 Simplification results

According to Table 1 and Figure 7(a, b and c), since the simplification degree is controlled by the number of generated rays, a more number of rays requires a longer execution time. According to Table 1 and Figure 7(a, d and e), the simplification degree is controlled by the value of $x$, a larger $x$ varies the simplification degree, the execution time has not change much. However, a larger $x$ will lead to holes in the model. Therefore, under the premise that the simplification effect is not affected, threshold $x$ can be increased as much as possible to improve data simplification efficiency.

Table 2 presents the test results for a model using the proposed algorithm and the traditional data simplification algorithm, which involves curvature estimation and recursion. By adjusting the relevant parameters, the number of output points is approximately achieved. Figure 8 shows the simplification effects of Model1 of Table 2, respectively.

TABLE 2 Comparison of the proposed algorithm and traditional curvature estimation algorithm

| Model | Input points | Output points | | Execution time(s) | |
|---|---|---|---|---|---|
| | | The traditional algorithm | The proposed algorithm | The traditional algorithm | The proposed algorithm |
| Model1 | 4102 | 1162 | 1196 | 2.02 | 0.94 |

a. Original model

b. Effect of simplification involving curvature estimation and recursion

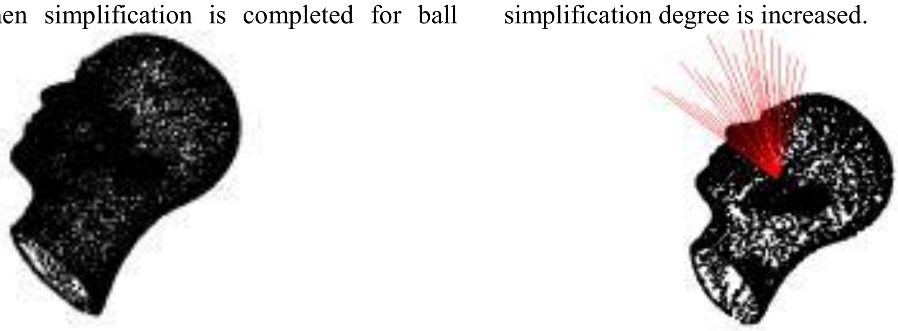
c. Simplification effect using proposed algorithm

FIGURE 8 Comparison of simplification effects for Model1

It can be seen from Table 2 and Figure 8 that the proposed algorithm is more efficient than the traditional curvature estimation algorithm, while good simplification effects are guaranteed.

The proposed algorithm is based on the ray principle, and thus is more suitable for simplification under a ball model or point cloud model with symmetrical structure. Figure 9 and Figure 10 depict a schematic diagram of the simplification effects as well as some of the rays when applying the proposed algorithm to a ball model and a symmetrical model.

As shown in Figure 9 and Figure 10, according to the definition of the central point in the ray generation method used in this paper, all central points used to generate rays are approximately located in the central position of the entire model when data simplification is carried out for Model 1 of Figure 9 or Model 2 of Figure 10. In this situation, results of data simplification are more even. Regarding approximate ball models, the distance from the central point to each surface of the model is almost the same, and thus every part of the model can be reduced by a similar degree. Besides, the central point will not be too close to a certain surface of the model when simplification is completed for ball models. According to the ray principle, rays are dense at places near the central point. If the central point is too close to a certain surface of the model, this part can be reduced more easily than other parts when the simplification degree is increased.



a. Original model b. Schematic diagram of simplification effects and some rays

FIGURE 9 Model 1(Ball model)

a. Original model b. Schematic diagram of simplification effects and some rays

FIGURE 10 Model 2(Symmetrical model)

## 5 Conclusion

In this paper, we proposed a novel and efficient simplification algorithm for 3D point cloud data, whose shape is symmetrical. The experimental results have shown that within a certain degree of simplification, this algorithm achieves good simplification effects and high efficiency. The proposed algorithm is especially suited to ball and symmetrical models. To increase the adaptability to solve the problem whereby an unreasonable central point of the model negatively affects the simplification effect, the central point of the model will be set as a flexible point in the future research. In this way, the position of the central point can be adjusted according to the actual simplification model, thereby improving the simplification results and the efficiency.

## References

[1] Wang Y, Li H, Ning X, Shi Z 2011 A new interpolation method in mesh reconstruction from 3D point cloud *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry* 235-42

[2] Nguyen H, Kim J, Lee Y, Ahmed N, Lee S 2013 Accurate and fast extraction of planar surface patches from 3D point cloud *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication* **84**

[3] Shi B, Liang J, Liu Q 2011 Adaptive simplification of point cloud using k-means clustering *Computer-Aided Design* **43**(11) 910-22

[4] Song H, Feng H 2008 A global clustering approach to point cloud simplification with a specified date reduction ratio *Computer-Aided Design* **40**(3) 281-92

[5] Yu Z, Wong H, Hong P, Ma Q 2010 An adaptive simplification method for 3d point-based models *Computer-Aided Design* **42**(7) 598-612

[6] Miao Y, Pajarola R, Feng J 2009 Curvature-aware adaptive re-sampling for point-sampled geometry *Computer-Aided Design* **41**(6) 395-403

[7] Wang Y, Feng H, Delorme F, Engin S 2013 An adaptive normal estimation method for scanned point clouds with sharp features *Computer-Aided Design* **45**(11) 1333-48

[8] Zhu Y, Kang B, Li H, Shi F 2012 Improved algorithm for point cloud data simplification *Journal of Computer Applications* **32**(2) 521-3

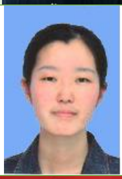
[9] Song H, Feng H 2009 A progressive point cloud simplification algorithm with preserved sharp edge data *The International Journal of Advanced Manufacturing Technology* **45**(5-6) 583-92

[10] Su J, Srivastava A, Huffer F W 2013 Detection, classification and estimation of individual shapes in 2D and 3D point clouds *Computational Statistics and Data Analysis* **58** 227-41

[11] Li H, Zhang X, Jaeger M, Constant T 2010 Segmentation of forest terrain laser scan data *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry* 47-54

[12] Chen Feizhou, Chen Zhiyang, Ding Zhan, et al. 2006 Filling Holes in Point Cloud with Radial Basis Function *Journal of Computer-Aided Design & Computer Graphics* **18**(9) 1414-9

## Authors


| | |
|---|---|
|  | **Changsu Liao**<br><br>**University studies:** undergraduate student at College of Information Engineering, Northwest A&F University.<br>**Scientific interests:** computer animation, computer graphics and geometric modelling. |
|  | **Xiaojing Niu**<br><br>**University studies:** graduate student at College of Information Engineering, Northwest A&F University.<br>**Scientific interests:** image processing, computer graphics and geometric modelling. |
|  | **Meili Wang**<br><br>**Current position, grades:** lecturer at College of Information Engineering, Northwest A&F University.<br>**University studies:** Ph.D. degree at the National Centre for Computer Animation, Bournemouth University.<br>**Scientific interests:** computer graphics, geometric modelling, image processing, visualization and virtual reality. |
|  | **Dongjian He**<br><br>**Current position, grades:** Professor at Mechanical and Electronic Engineering, Northwest A&F University.<br>**Scientific interests:** intelligent detecting and control, image analysis and recognition, agricultural information technology, digital plant and Virtual technology. |