Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5713137, 13 pages
https://doi.org/10.1155/2020/5713137

Research Article

A Point Cloud Simplification Method Based on Modified Fuzzy C-Means Clustering Algorithm with Feature Information Reserved

Yang Yang,[1] **Ming Li**,[1] **and Xie Ma**[1,2]

[1]*Key Laboratory of Intelligent Manufacturing and Robotics,*
School of Mechatronic Engineering and Automation Shanghai University, Shanghai 200072, China
[2]*School of Mechanical and Electrical Engineering College, Ningbo University of Finance and Economics, Ningbo 315175, China*

Correspondence should be addressed to Yang Yang; mryyfi@i.shu.edu.cn

Received 17 April 2020; Accepted 24 August 2020; Published 21 October 2020

Guest Editor: Ching-Hsin Wang

To further improve the performance of the point cloud simplification algorithm and reserve the feature information of parts point cloud, a new method based on modified fuzzy *c*-means (MFCM) clustering algorithm with feature information reserved is proposed. Firstly, the normal vector, angle entropy, curvature, and density information of point cloud are calculated by combining principal component analysis (PCA) and *k*-nearest neighbors (*k*-NN) algorithm, respectively; Secondly, gravitational search algorithm (GSA) is introduced to optimize the initial cluster center of fuzzy *c*-means (FCM) clustering algorithm. Thirdly, the point cloud data combined coordinates with its feature information are divided by the MFCM algorithm. Finally, the point cloud is simplified according to point cloud feature information and simplified parameters. The point cloud test data are simplified using the new algorithm and traditional algorithms; then, the results are compared and discussed. The results show that the new proposed algorithm can not only effectively improve the precision of point cloud simplification but also reserve the accuracy of part features.

1. Introduction

The development of digital measurement technology provides an efficient and accurate method for the scanning process of mechanical parts. Laser scanning technology has enriched the inspection methods, which can quickly and accurately show the shape information of the parts through the point cloud data obtained by the equipment [1]. According to the data form of the point cloud, it mainly includes line scan, triangular grid, uniform grid, and scattered point cloud data [2]. Among them, scattered point cloud is a kind of widely used data form. Although there are different point cloud data types, the essence of point cloud data is to represent geometric information of parts by obtaining discrete coordinates on parts' surfaces. On the basis of coordinate information, relevant algorithms of point cloud data were adopted for postprocessing analysis, so as to analyze and evaluate the actual state of parts quickly, accurately, and completely.

As mentioned in the above research background, point cloud is one of the basic measurement data sources at this stage and has been widely used in various industries such as 3D-LiDAR [3], additive manufacturing [4], and unmanned aerial vehicle [5]. However, the point cloud obtained by laser scanner has a lot of redundant data, which affect the point cloud computing analysis process. The large amount of point data will lead to lower work efficiency and heavy workload. Therefore, initial point cloud data need to be simplified during the preprocess of point cloud. For the simplified process, the traditional irregular point cloud simplification algorithm will affect the reconstruction of features to some extent. Therefore, in order to improve the simplification performance and the accuracy of the reconstruction model, it is necessary to simplify the point cloud data on the basis of ensuring the geometric characteristics of parts. Moreover, according to the distribution of point clouds, the point cloud simplification algorithm is also different. At present, the classic point cloud simplification methods include curvature sampling method [6], point spacing algorithm [7], and bounding box algorithm [8] according to the form of point cloud. Generally, a good point cloud simplification method can use the least number of points to reflect the characteristics of the measured object. In the field of point cloud simplification algorithms, different types of point cloud data have different simplified algorithms, compared to other more regular point cloud data, the simplification algorithms that suitable for scattered point cloud data still need to be studied in depth. In this paper, the simplification algorithm for scattered point cloud is studied and designed. The main contributions of this paper are as follows:

(1) The initial clustering center of FCM is optimized by GSA to improve the clustering accuracy and avoid algorithm falling into local optimization. In addition, related algorithms are used to obtain the normal vector, angle entropy, curvature, and density information of point cloud, to provide geometric feature information data for point cloud simplification.

(2) On the basis of the above data calculated in point (1), the strong feature information of point cloud is first reserved, and then the multidimensional point cloud data is divided through the MFCM, so as to simplify the point cloud based on point cloud feature information and simplified parameters. The data volume of point cloud is reduced while the point cloud features are reserved.

(3) We carried out simulation experiments and compared the results with the related algorithm. The results showed that the algorithm is effective to some extent.

The rest structure of this article is as follows. In Section 2, the related work of point cloud simplification algorithms are introduced. In Section 3, the experimental data of point cloud simplification are described. In Section 4, the geometric information of point cloud data are calculated in detail in the part. In Section 5, the simplification method based on MFCM algorithm is presented. The simulation experiments of the proposed algorithms are carried out in Section 6. In Section 7, the conclusion of this work is summarized and the future work is prospected.

2. Related Work

For point cloud simplification algorithms, Lee et al. implemented the point cloud simplification process based on geometric information and proved the advantages of the algorithm through experimental data [9]. Dyn et al. simplified the point cloud by using an adaptive refinement strategy, and experiments proved the effectiveness of the algorithm [10]. Shi et al. designed an adaptive point cloud simplification method based on k-mean clustering algorithm [11]. On the basis of data accuracy, Wang et al. applied Akima spline interpolation algorithm to the line point cloud simplification method, which can ensure the accuracy while reducing the amount of point cloud data [12]. Based on the fuzzy entropy iterator, Sun and Da designed a point cloud simplification algorithm, which was claimed to have the advantage of preserving the details of the parts and ensuring the computational speed of data simplification [13]. Han et al. simplified point cloud data by constructing the topology of the points and adopted different sampling strategies based on point location [14]. Chen et al. proposed a point cloud simplification algorithm based on the normal vector included angle local entropy model, and the experimental results showed that the algorithm can achieve the optimal computational accuracy and efficiency [15]. Wang et al. built a vehicle size measurement platform based on improved point cloud simplification algorithm. This platform designed three point cloud preprocessing data algorithms, which were said to be able to simplify point cloud data in a large scale and maintain part features and model quality [16]. Zang et al. proposed a point cloud simplification algorithm based on multilevel adaptation, which can be evaluated through the constructed point cloud grid, and the effectiveness of the algorithm is proved by experiment [17]. By designing a new sampling rule, Zhang simplified the point cloud data and ensured the sampling quality of the point cloud [18]. Xuan et al. first obtained the angle between the normal vectors of each point, then introduced the information entropy to determine the importance of each point, and finally achieved the simplification process of point cloud by reserving the important points with certain simplified rules [19]. Chen et al. designed a point cloud simplification algorithm based on dynamic k-nearest neighbors (k-NN) search to improve the simplification accuracy of point clouds [20]. Sayed et al. designed a point cloud simplification algorithm based on weighted graph, which simplified the point cloud by preserving feature points [21]. Markovic et al. adopted support vector machine to simplify scanning point cloud data in different regions [22]. Wang et al. designed a point cloud simplification algorithm based on feature perception, which can reduce the simplification error while ensuring the original geometric accuracy of parts [23]. Chang et al. explained that k-means clustering was applied to point cloud simplification process and modified the simplification process by combining the boundary extraction algorithm [24]. Ji et al. designed a point cloud simplification algorithm based on k-NN search with multifeature measurement, which can guarantee the simplification accuracy and improve the simplification efficiency [25]. Mahdaoui et al. combined k-NN and clustering algorithm to simplify the point cloud and achieved good results [26]. Li et al. designed a point cloud simplification algorithm based on k-means and Hausdorff distance [27].

Based on the above literatures, many scholars have conducted in-depth studies on point cloud simplification; among them, the clustering algorithm which is represented by k-means has been widely applied to simplify the point cloud [11, 26]. At present, FCM has better clustering results as a soft clustering algorithm compared to other clustering algorithm. As to the point cloud simplification process, it can be seen that, although there are a lot of research results on point cloud simplification algorithm; it is necessary to consider not only the removal of redundant information but also it need to reserve the feature information of the point cloud, so further studies are needed. Therefore, based on the FCM clustering algorithm, this paper firstly optimized the initial clustering center through GSA, then combined the point cloud coordinates with feature information to divide the point cloud, finally reserved strong feature information, and simplified the point cloud data in different regions. Four groups of commonly used point cloud data are applied to the final simulation experiment process.

3. Materials

For the point cloud simplification process, different simplification algorithms will have different simplification results. However, for the comparison and verification process of the algorithm, the same point cloud data should be used as standard dataset, which can objectively test the simplification performance of the proposed algorithm. Therefore, in order to better verify the effectiveness of the simplified algorithm designed in this paper, the four groups of classical point cloud data are adopted for experimental verification in this paper, as shown in Figure 1.

Figure 1(a) is the Bunny data and the number of points is 35947. Figure 1(b) is the Chair data and the number of points is 49960. Figure 1(c) is the Dino data and the number of points is 23982. Figure 1(d) is the Gargo50 K data and the number of points is 25038. The above models are classic point cloud test data, with the rich characteristic properties; this paper will use the above model and combined them with the relevant simplified algorithms for comparison and verification analysis.

4. Point Cloud Geometric Information Calculation

In order to reserve point cloud data feature information, it not only needs the spatial coordinate value of point cloud but also the geometric information of point cloud. The geometric information is mainly divided into the normal vector and curvature of points; in addition, angle entropy and point cloud density are added in our method. Scattered point cloud obtained by optical measuring equipment usually has no topological relations between points. Therefore, the normal vector, curvature, angle entropy, and density of point clouds need to be estimated by related algorithms.

4.1. K-d Tree and k-NN. From the existing research results, the point cloud data can greatly improve its computational analysis efficiency through the data structure to establish a relationship [14, 27], and *k-d* tree [28] is one of the most commonly used at this stage, where k represents the dimensions of data. For three-dimensional point cloud data, the *k-d* tree construction process is as follows: calculating the mean value of x-directional coordinate for the three-dimensional dataset $P_i(x_i, y_i, z_i)$, then finding the point data which is closest to the mean value, and using a plane to divide the space into two parts; then, each part of y is split by the same principle as shown in x-directional coordinate, z-direction is the same method, then x direction is split, and so on. All points in the space are divided into the subspaces which they belong to.

When point cloud is the research object, it is necessary to obtain the neighborhood information of points, which is the basis for analyzing point cloud data. The most commonly used method is to calculate the distance d between one point cloud and other point cloud data and then select the nearest k points as the neighborhood of the selected point, which is also known as the k-nearest neighbor algorithm [28], as shown in point q in Figure 2. Generally, Euclidean space is selected for distance operation:where (x_a, y_a, z_a) and (x_b, y_b, z_c) are the three-dimensional Euclidean space coordinates of point a and point b, respectively. d is the Euclidean distance between the two points. The normal vector and curvature information of each point can be calculated and obtained through its k-NN data structure.

$$d = \sqrt{\left(x_a - x_b\right)^2 + \left(y_a - y_b\right)^2 + \left(z_a - z_b\right)^2}, \quad (1)$$

4.2. Point Cloud Normal Vector and Curvature Calculation. Principal component analysis (PCA) is a classical point cloud normal vector calculation method which is proposed by Hoppe [29]. The method constructs the covariance matrix through the neighborhood information of the point, so as to obtain the geometric information of the point. At present, the k-NN data structure of the point can be used to obtain its normal vector. For point cloud set $P = (p_1, p_2, \ldots p_k)^T$, a certain point p_i constructs its k-nearest neighbor through *k-d* tree, to build the least square fitting plane, which can obtain the normal vector information of the point cloud, as shown in formula (2). The eigenvectors corresponding to its minimum eigenvalue are the normal vectors of the point. At the same time, due to the randomness of the normal vector direction of the point cloud, it is necessary to adjust the normal consistency through formula (3):

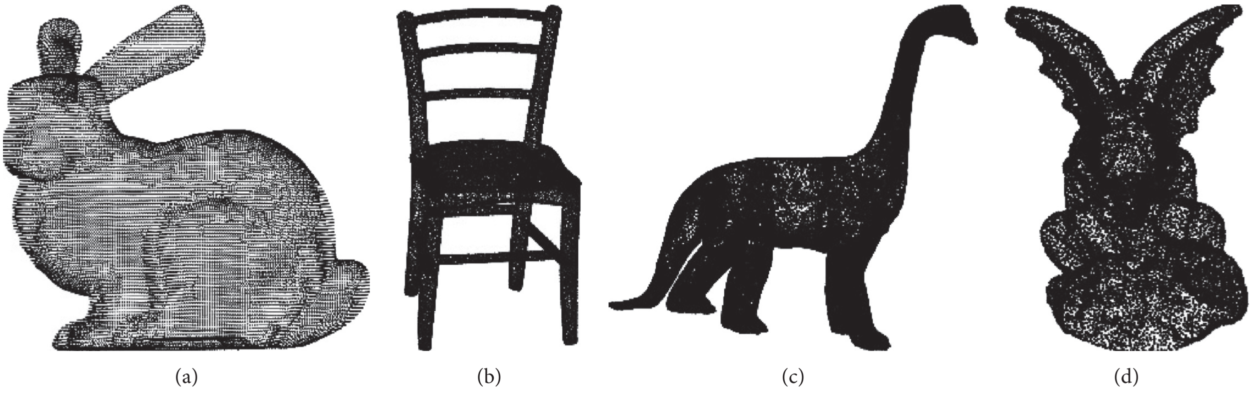

Figure 1: Experimental data in this paper. (a) Bunny. (b) Chair. (c) Dino. (d) Gargo50 K.

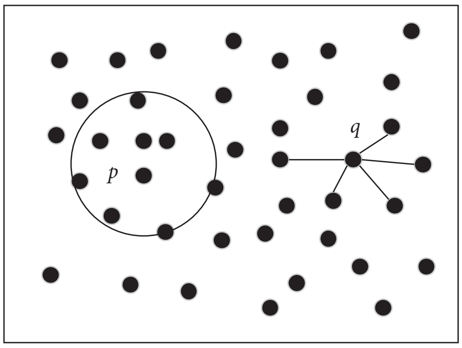

Figure 2: Diagram of k-NN.

$$C^{3\times 3}=\sum_{i=1}^{k}(p_i-\overline{p})(p_i-\overline{p})^T=\begin{bmatrix}\sum_{i=1}^{k}(x_i-\overline{x})(x_i-\overline{x}) & \sum_{i=1}^{k}(x_i-\overline{x})(y_i-\overline{y}) & \sum_{j=1}^{k}(x_i-\overline{x})(z_i-\overline{z})\\ \sum_{j=1}^{k}(x_i-\overline{x})(y_i-\overline{y}) & \sum_{j=1}^{k}(y_i-\overline{y})(y_i-\overline{y}) & \sum_{j=1}^{k}(y_i-\overline{y})(z_i-\overline{z})\\ \sum_{j=1}^{k}(x_i-\overline{x})(z_i-\overline{z}) & \sum_{j=1}^{k}(y_i-\overline{y})(z_i-\overline{z}) & \sum_{j=1}^{k}(z_i-\overline{z})(z_i-\overline{z})\end{bmatrix}, \tag{2}$$

$$\vec{\gamma}'=\begin{cases}\vec{\gamma}\,\vec{\gamma}(l-p_i)\geq 0,\\ -\vec{\gamma}\,\vec{\gamma}(l-p_i)<0,\end{cases} \tag{3}$$

where k is the number of the nearest point; x_i, y_i, and z_i are the coordinates values of point p_i; $\overline{p}$ is the mean value of the $P=(p_1,p_2,\ldots,p_k)^T$; $\vec{\gamma}$ is the normal vector before the adjustment; $\vec{\gamma}'\ (h,m,n)$ is the adjusted normal vector; and l is the location of the view.

For point cloud curvature, it can be obtained by combining the eigenvalues of formula (2) on the basis of PCA which is used to solve the point cloud normal vector [30]. The curvature of point p_i is shown as

$$\sigma_i=\frac{\tau_0}{\tau_0+\tau_1+\tau_2}, \tag{4}$$

where σ_i is the point cloud curvature and τ_0, τ_1, and τ_2 are the eigenvalues of formula (2) which satisfy $\tau_0\leq\tau_1\leq\tau_2$.

4.3. Point Cloud Angle Entropy. In addition, angle entropy is a new geometric parameter proposed by Chen et al [15]. By building the k-NN of the point p_i and combining the normal vector of each internal k point, the mean value of the normal vector deviation between p_i and the points in its k-neighborhood is calculated [31], as shown in formula (5). Then, calculate the angle entropy of the point according to formula (6):

$$\overline{\theta}_i = \frac{1}{k}\sum_{j=1}^{k} \arccos\left(\frac{\vec{\gamma}_i \cdot \vec{\gamma}_j}{\left|\vec{\gamma}_i\right|\left|\vec{\gamma}_j\right|}\right), \tag{5}$$

$$\begin{aligned} En_i = & -\frac{\overline{\theta}_i}{\overline{\theta}_i + \sum_{j=1}^{k}\overline{\theta}_j}\log_2\frac{\overline{\theta}_i}{\overline{\theta}_i + \sum_{j=1}^{k}\overline{\theta}_j} \\ & - \sum_{j=1}^{k}\frac{\overline{\theta}_j}{\overline{\theta}_i + \sum_{j=1}^{k}\overline{\theta}_j}\log_2\frac{\overline{\theta}_j}{\overline{\theta}_i + \sum_{j=1}^{k}\overline{\theta}_j}, \end{aligned} \tag{6}$$

where $\vec{\gamma}_i$ is normal vector of p_i, $\overline{\theta}_i$ is the angle of point p_i, $\overline{\theta}_j$ is the angle of point in k-NN of p_i, and En_i is the angle entropy of point p_i.

4.4. Point Cloud Density Calculation. Point cloud density also can reflect its location to a certain extent. Generally speaking, the number of points in the feature area is large, so is the density of point cloud. On the contrary, the number and density of noncharacteristic regions are smaller. At the same time, for point cloud data, it is a three-dimensional spatial data. Therefore, the spatial density can better reflect its density information. This paper introduced the point cloud density calculation formula proposed by Yang et al. [32] for

$$\rho_i = \frac{4k}{3\pi d_{\max}^3}, \tag{7}$$

where the ρ_i is the density parameter and $d_{\max}$ is the farthest distance in the k-NN.

5. The Proposed Point Cloud Simplification Algorithm

5.1. Reservation of Point Cloud Strong Features. For each set of measurement point cloud data, strong features can reflect its unique nature of the point cloud. These points not only can reserve the overall characteristics of the point cloud but also affect the quality of point cloud reconstruction; in order to retain these points, this paper uses the point cloud curvature σ_i as the evaluation parameter and adopts the statistical principle in the way of reservation.

The curvature σ_i of each point p_i is obtained from the above calculation, for the point set $P(p_1, p_2, \ldots, p_n)$, the average curvature σ_i of all point cloud, and the standard deviation σ_{std}, as shown in the formulas,

$$\overline{\sigma} = \frac{1}{n}\sum_{i=1}^{n}\sigma_i, \tag{8}$$

$$\sigma_{\text{std}} = \sqrt{\frac{\sum_{i=1}^{n}\left(\overline{\sigma} - \sigma_i\right)^2}{n}}, \tag{9}$$

determines whether the curvature σ_i of each current p_i is greater than the set threshold σ_0. When $\sigma_i > \sigma_0$, the retention point p_i as a strong feature point, as shown in the formula,

$$\sigma_0 = \overline{\sigma} + \beta \times \sigma_{\text{std}}, \tag{10}$$

where β is a constant from 1 to 5.

5.2. Basic Gravitational Search Algorithm. The GSA is a new heuristic optimization algorithm proposed in 2009 [33]. The idea of GSA is derived from the law of universal gravitation and Newton's second law in physics. In GSA, every particle not only has a certain mass but also can move without resistance under the gravitational force between particles in the solution space. For the whole population, the particle will move to the one with a larger mass, until it reaches the position of the maximum mass, which is the optimal location of the problem. The flow of basic GSA is shown below:

Step 1: define algorithm parameters and initialize population. The particles in the solution space can be described as

$$x_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,d}\right), \quad i = 1, 2, \ldots, n, \tag{11}$$

where i is the sequence number of particles in the population; n is size of the population; and d is the dimension of the problem.

Step 2: calculate the fitness function value of each particle. The fitness value is generated by the target function of the problem. Update universal gravitational constant, and find the worst and the best particle in the population. For the minimization problem, the calculation of relevant parameters is shown in formulas (12)–(14), respectively:

$$G(t) = G_0 \times e^{(-\alpha(t/T))}, \tag{12}$$

$$\text{best}_t(x) = \min f_t(x_i), \quad i = 1, 2, \ldots, n, \tag{13}$$

$$\text{worst}_t(x) = \max f_t(x_i), \quad i = 1, 2, \ldots, n, \tag{14}$$

where $G(t)$ is the universal gravitational constant; $G_0 = 100$; T is the maximum number of iterations; t is the current number of iterations; and $\alpha = 20$.

Step 3: calculate the mass and acceleration of each particle. The mass of each particle is calculated by the worst and best particles, as shown in formulas (15) and (16). The acceleration of each particle is calculated according to formulas:

$$m_t(x_i) = \frac{f_t(x_i) - \text{worst}_t(x)}{\text{best}_t(x) - \text{worst}_t(x)}, \tag{15}$$

$$M_t(x_i) = \frac{m_t(x_i)}{\sum_{j=1}^{N} m_t\left(x_j\right)}, \tag{16}$$

$$F_{ij}^d(x) = G(t) \times \frac{M_t\left(x_j\right)M_t(x_i)}{R_t(i, j) + \varepsilon}\left(x_t^d(j) - x_t^d(i)\right), \tag{17}$$

$$F_i^d(x) = \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{rand}_j F_{ij}^d(x), \tag{18}$$

$$a_i^d = \frac{F_i^d(x)}{M_t(x_i)}, \tag{19}$$

where ε is a constant value, a_i^d is the acceleration of particles in dimension d, and $R_t(i, j)$ is the Euclidean distance between i and j.

Step 4: update the velocity and position of the particle under the action of the acceleration as

$$v'_{t,d}(x_i) = \text{rand}_i \times v_t(x_i) + a_t(x_i), \tag{20}$$

$$x'_i = x_i + v'_{t,d}(x_i), \tag{21}$$

where rand_i takes a value of 0 to 1.

Step 5 (judgment of stop criteria): if the algorithm reaches the maximum iterative times, the algorithm terminates and outputs the optimal solutions; if not, return to Step 2. GSA algorithm flow chart is shown in Figure 3.

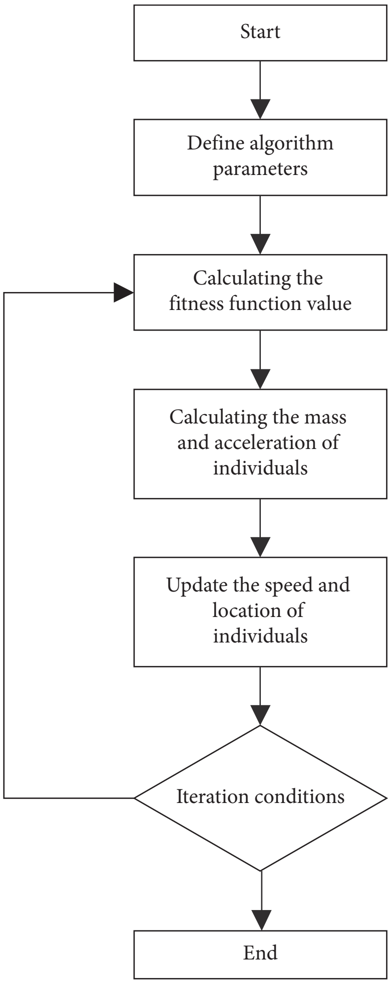
Start
Define algorithm parameters
Calculating the fitness function value
Calculating the mass and acceleration of individuals
Update the speed and location of individuals
Iteration conditions
End

Figure 3: GSA algorithm flow chart.

5.3. Data Clustering Based on MFCM. FCM is a soft computing clustering algorithm which is widely used in many fields [34–36]. Suppose point set $p = \{p_1, p_2, \ldots, p_n\}$, its subsample data is $p_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,j}\}$; when the value of membership function $F(p_i)$ is close to 1, it means the $p_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,j}\}$ belong to the same set. Otherwise, the data need to be regrouped until the threshold requirement of clustering is met. However, FCM algorithm also has problems such as unstable initial clustering center, easy to fall into local optimal, resulting in poor clustering results [37]. Therefore, GSA algorithm is introduced into FCM in this paper to optimize the initial clustering center of FCM, which has strong global search capability, thus further improving the clustering accuracy of FCM, and the flow of the MFCM algorithm is as follows.

Step 1 (parameter definition): input dataset $p = \{p_1, p_2, \ldots, p_n\}$. Define population number N, dimension of the problem d, $G_0 = 100$, $\alpha = 20$, number of cluster c, fuzzy index m, iteration number t and T, random initialization of the cluster center $V = \{v_1, v_2, \ldots, v_c\}$, which is $c \times d$ vectors, and membership matrix $F = [f_{ki}]_{c\times n}$, which need to satisfy the formula

$$f_{ki} \in [0, 1], \sum_{k=1}^{c} f_{ki} = 1, \quad 0 < \sum_{i=1}^{n} f_{ki} < n, \tag{22}$$
$$\forall i = 1, 2, \ldots, n, \forall k = 1, 2, \ldots, c.$$

Step 2: calculate the initial cluster center by GSA as shown in Section 5.2, where the objective function is shown in formula

$$Y = \frac{1}{J(F, V)} = \frac{1}{\sum_{k=1}^{c} \sum_{i=1}^{n} f_{ki}^{m} p_i - v_k{}^2}. \tag{23}$$

Step 3: input the initial clustering center obtained by Step 2, and update the membership matrix according to formula

$$f_{ij} = \frac{1}{\sum_{k=1}^{c} \left(d_{ij}/d_{kj}\right)^{2/m-1}}. \tag{24}$$

Step 4: update the new cluster center $V' = \{v'_1, v'_2, \ldots, v'_c\}$:

$$v_i = \frac{\sum_{j=1}^{n} f_{ij}^{m} p_i}{\sum_{j=1}^{n} f_{ij}^{m}}. \tag{25}$$

Step 5: judge whether the calculation process meets the threshold requirements. When the number of iterations reaches the set requirements, the calculation process is completed and the clustering result is output. Otherwise, return Step 3 until the termination condition is met.

The MFCM calculation process is shown in Figure 4.

5.4. The Proposed Point Cloud Simplification Algorithm Flow. From what has been discussed above, the process of the new proposed point cloud simplification algorithm is shown below.

Step 1: import the point cloud dataset $P = (p_1, p_2, \ldots p_k)^T$; then, go to step 2.

Step 2: build the k-d tree and k-NN data structures of point cloud data as described in Section 4.1; then, go to step 3.

Step 3: obtain the normal vector by PCA and k-NN, according to formulas (2) and (3). Then, the curvature information and angle entropy of point cloud are calculated by combining the normal vector information of point cloud as described in Sections 4.2 and 4.3, respectively. In addition, point cloud density ρ_i is obtained by formula (7), which calculation process is described in Section 4.4; then, go to step 4.

Step 4: reserve the strong feature points of the point cloud as described in Section 5.1; then, go to step 5.

Step 5: firstly, coordinate information and geometric feature information obtained in step 3 of point cloud are built into a nine-dimensional data structure of point cloud $p' (x, y, z, h, m, n, \sigma, En_i, \rho_i)$. Secondly, FCM is optimized by GSA to obtain the optimal initial clustering center as described in Sections 5.3. Then, the constructed multidimensional data point cloud is segmented by MFCM; then, go to step 6.

Step 6: according to the point cloud simplification parameters and percentages in different characteristic region, the point cloud data in different regions are simplified to complete the simplification process of point cloud based on the angle entropy in each region; then, go to step 7.

Step 7: complete the simplification process and export the simplified point cloud.

The simplification algorithm flowchart is shown in Figure 5.

6. Results and Discussion

To validate the performance of the point cloud simplification algorithm proposed, this section provided comparative experiments and analysis through four groups of point cloud data and related point cloud simplification algorithms. Experimental data are shown in Section 3. The platform for the experiment is Windows 10 system and Intel (R) Core (TM) i5-7300 processor. The point cloud simplification algorithms included grid sampling [38], Poisson-disk sampling [39], and algorithm in literature [11]. The experiment had a 91% simplification rate, which represents the only 9% of the original point cloud data is reserved. Figures 6–9 show the simplification results.

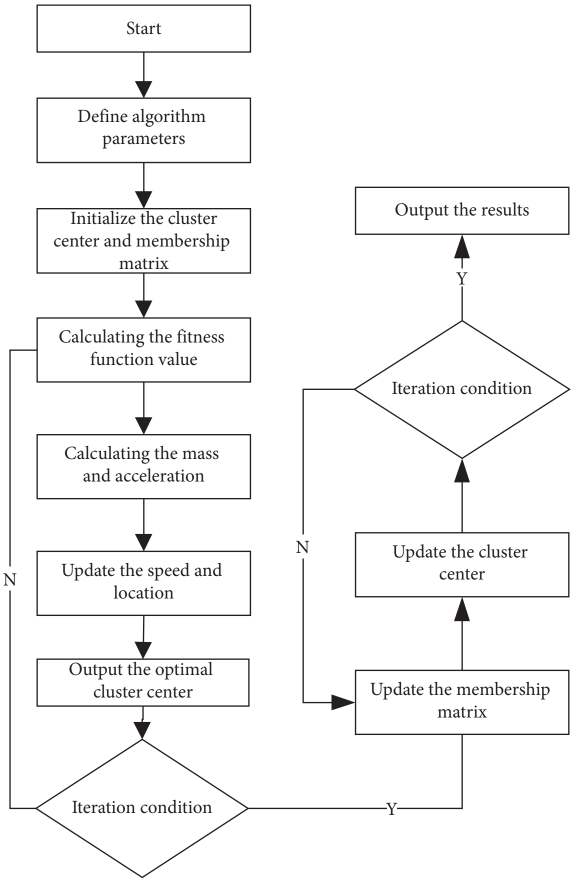
Start
Define algorithm parameters
Initialize the cluster center and membership matrix
Calculating the fitness function value
Calculating the mass and acceleration
Update the speed and location
Output the optimal cluster center
Iteration condition
N
Y
Update the membership matrix
Update the cluster center
Iteration condition
N
Y
Output the results

Figure 4: MFCM algorithm flow chart.

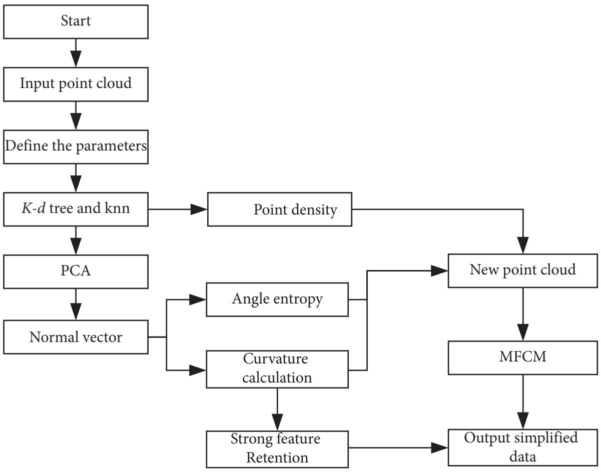

Figure 5: Flowchart of simplification algorithm.

Figures 6–9 show the simplification results of point cloud in Section 3 using different simplification algorithms on the vision. In Figures 6–9, (a) is the original data, (b) is the calculation result by grid sampling algorithm, (c) is the calculation result of Poisson-disk sampling, and (d) is the

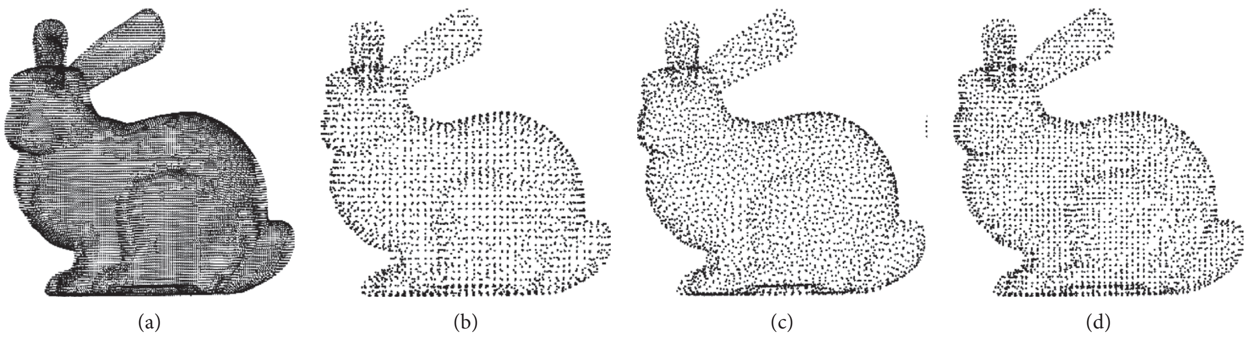

Figure 6: Bunny simplification results. (a) Bunny. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

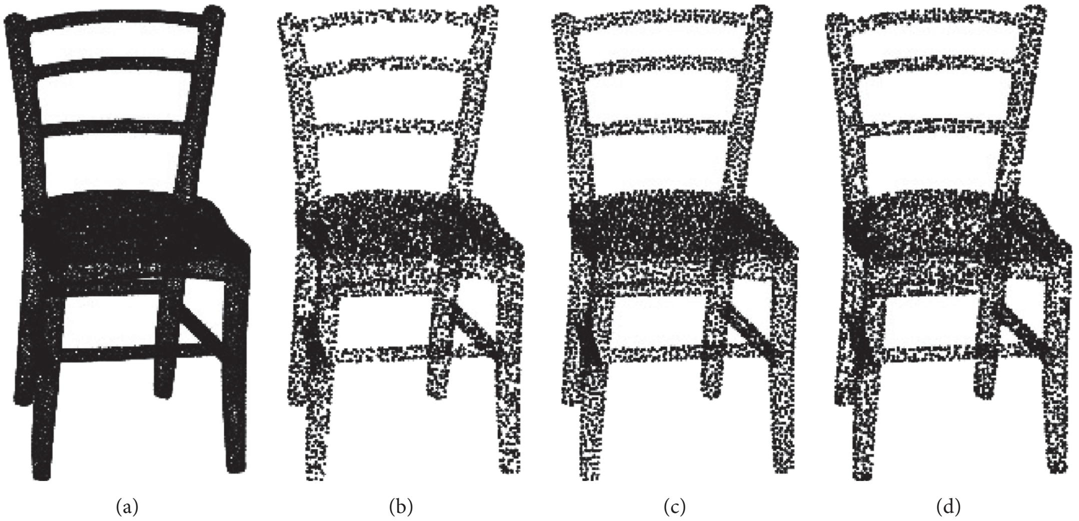

Figure 7: Chair simplification results. (a) Chair. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

calculation result of MFCM algorithm. The simplification result by grid sampling distributed evenly cannot reserve the feature information accurately as shown from (b) in Figures 6–9. Poisson-disk sampling has a better simplification effect than grid sampling and can reflect the feature information of point cloud in a certain degree. Hence, our algorithm can reserve the feature information, especially strong feature information.

However, these experimental results cannot reflect the effect of the simplification postprocessing of point cloud. Therefore, inspired by the literature [25], Geomagic Studio 12 is used to reconstruct the point cloud to observe the reconstruction results. The reconstructed models are shown in Figures 10–13.

Figures 10–13 show the simplified point cloud reconstruction diagram by different algorithms. As shown in Figures 10–13, at a reduction ratio of 91%, the traditional point cloud simplification results have a large area of holes in reconstruction model, while our algorithm also has holes in Figures 11–13, but the area is relative smaller than other algorithms. In order to evaluate the effect of point cloud simplification algorithm more accurately, we analyze the 3D deviation of the reconstructed model. The results are shown in Figures 14–17 and Table 1.

Figures 14–17 show the 3D deviation cloud diagram of the original point cloud and the reconstructed surface. For the 3D deviation result of point cloud, we choose the average distance and standard deviation as the error evaluation basis shown in Table 1. It can be seen that MFCM has the smallest 3D deviation for the simplified bunny and Dino, which means it has the highest simplification accuracy on these models. However, for the chair and gargo50 k, Poisson sampling and grid sampling perform better.

In order to further compare the effectiveness of the algorithm, this paper compares the simplified results with literature [20]. The data for comparison are bunny and chair. The error calculation method is shown in literature [11], which is a classical point cloud error calculation method. The calculation results are shown in Table 2.

As can be seen in Table 2, for the simplified point cloud data, the MFCM algorithm performs better in bunny by comparison with the error calculation; the maximum error (0.042 mm) and average error (0.015 mm) are smaller than the algorithm in the literature [20]. However, for the chair model, the error of the algorithm in this paper is worse than the calculation results in the literature [20]. Therefore, the effect of the algorithm still needs to be further improved. In combination with the reconstructed

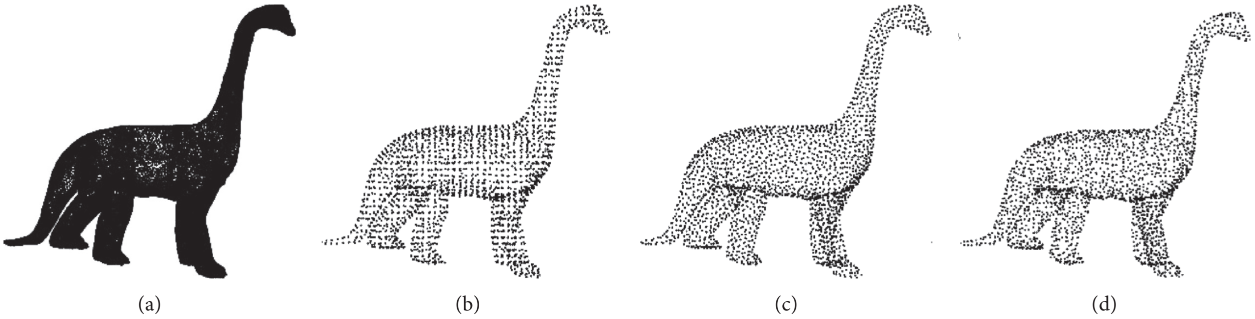

Figure 8: Dino simplification Results. (a) Dino. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

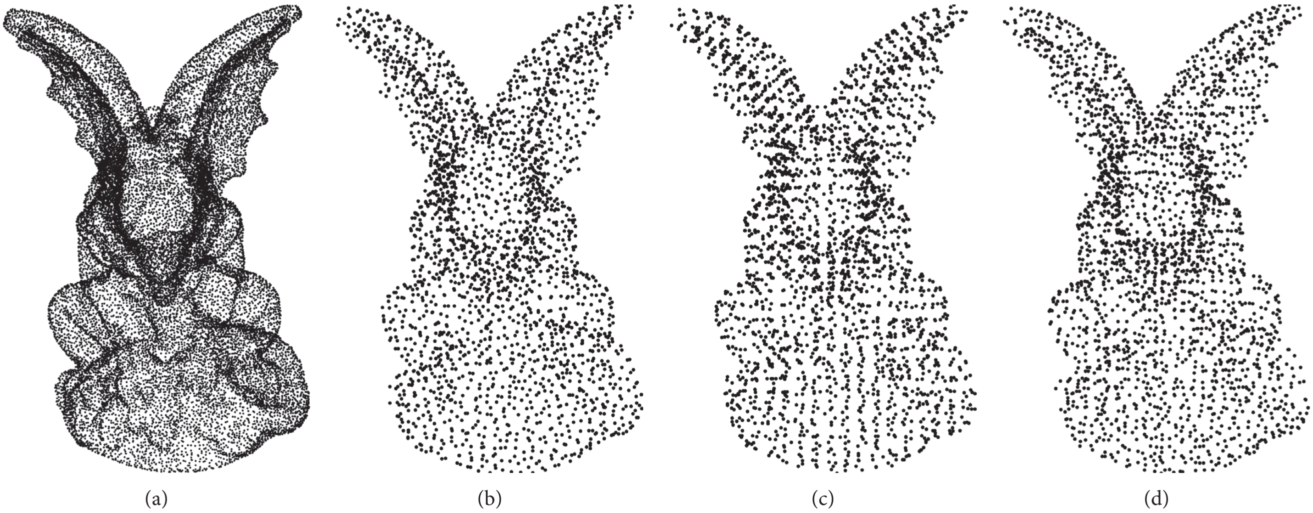

Figure 9: Gargo50 K simplification results. (a) Gargo50 K. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

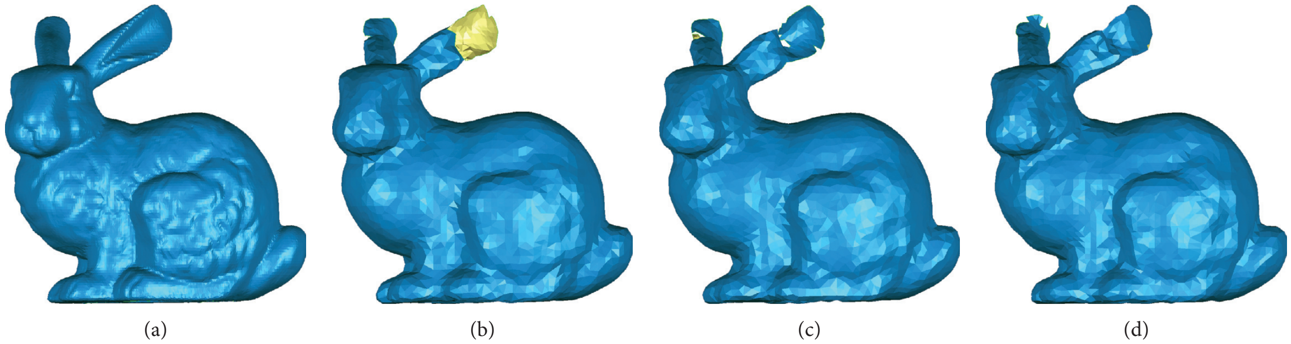

Figure 10: Point cloud reconstruction for Bunny. (a) Bunny. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

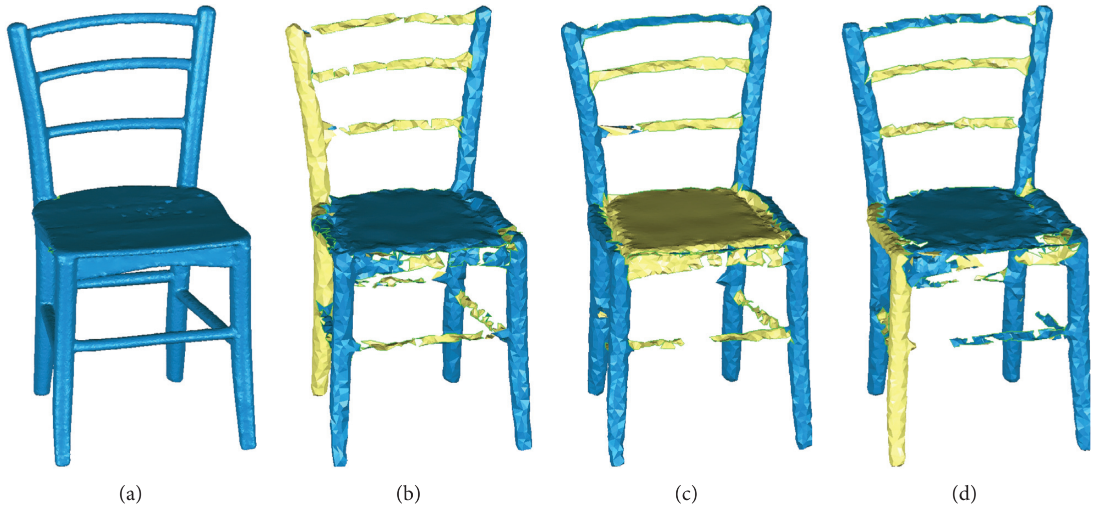

Figure 11: Point cloud reconstruction for Chair. (a) Chair. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

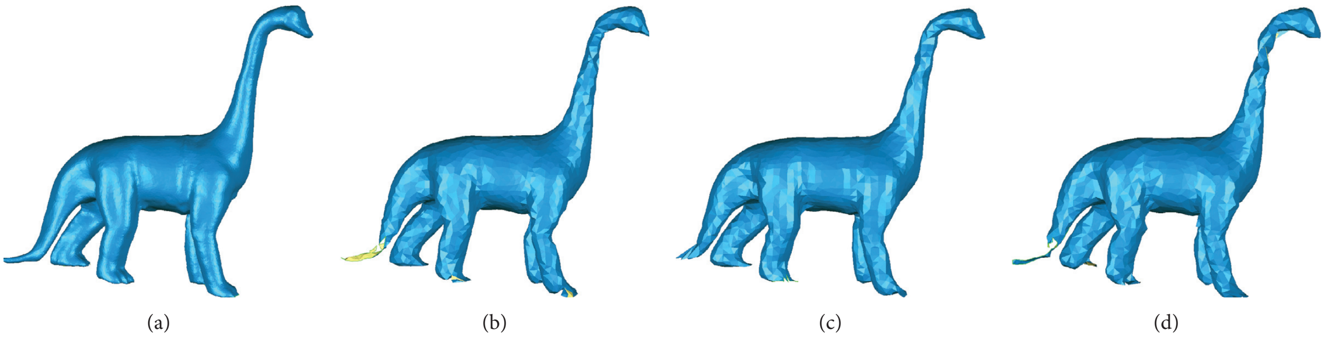

Figure 12: Point cloud reconstruction for Dino. (a) Dino. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

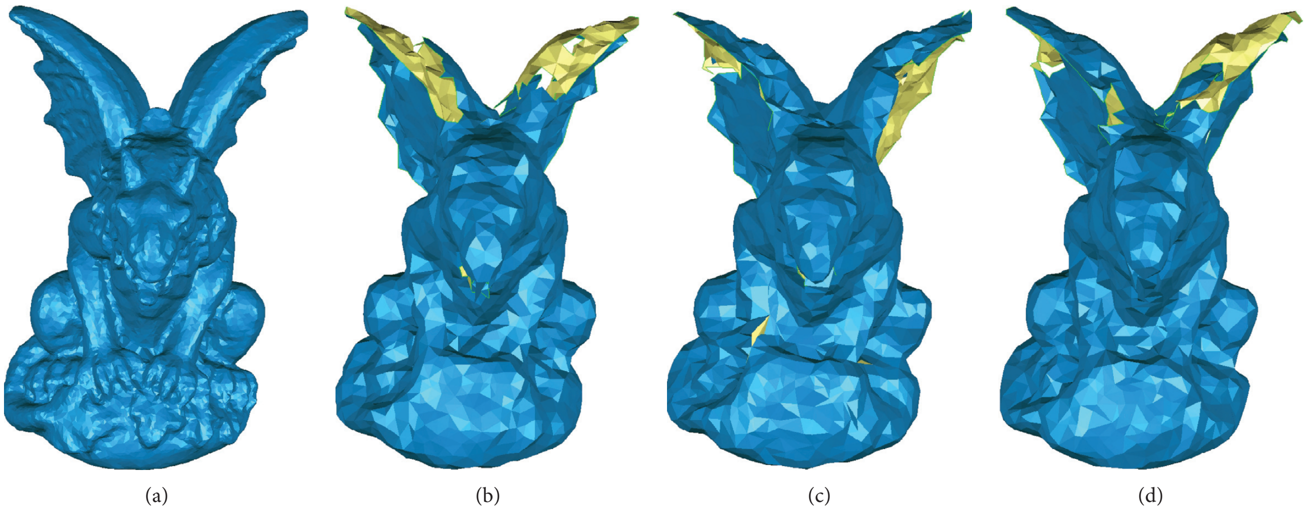

Figure 13: Point cloud reconstruction for Gargo50 K. (a) Gargo50 K. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

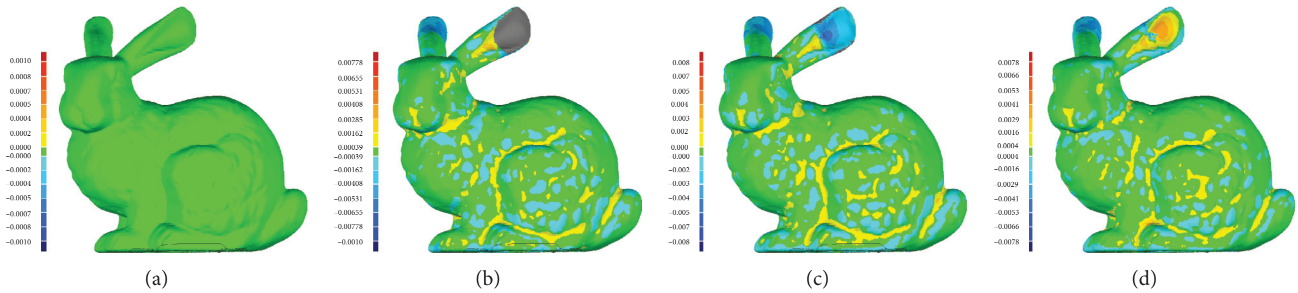

Figure 14: Point cloud 3D deviation for Bunny. (a) Bunny. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

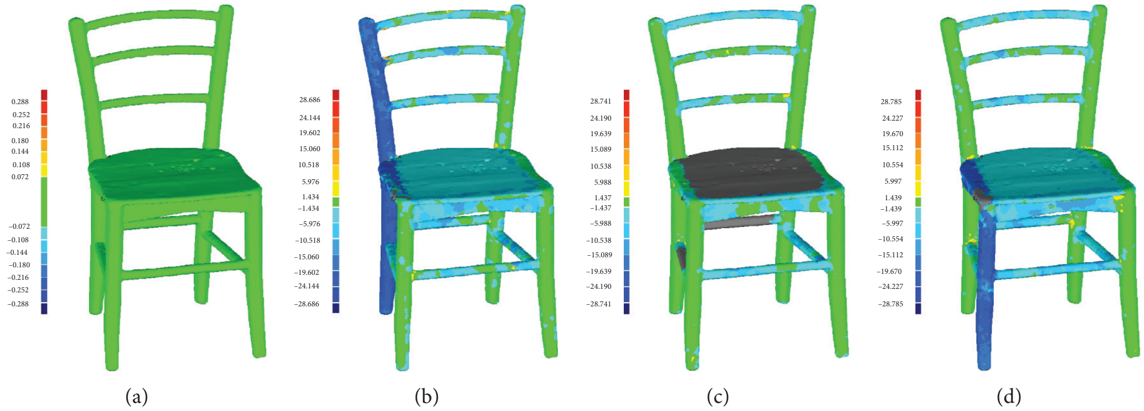

Figure 15: Point cloud 3D deviation for Chair. (a) Chair. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

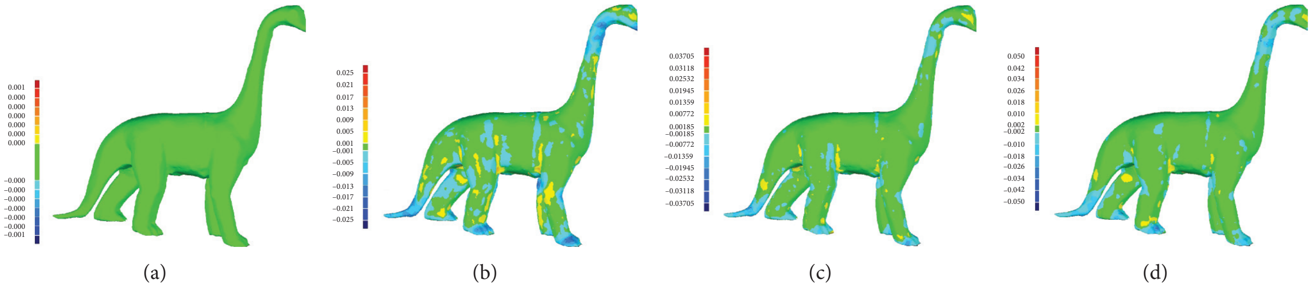

Figure 16: Point cloud 3D deviation for Dino. (a) Dino. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

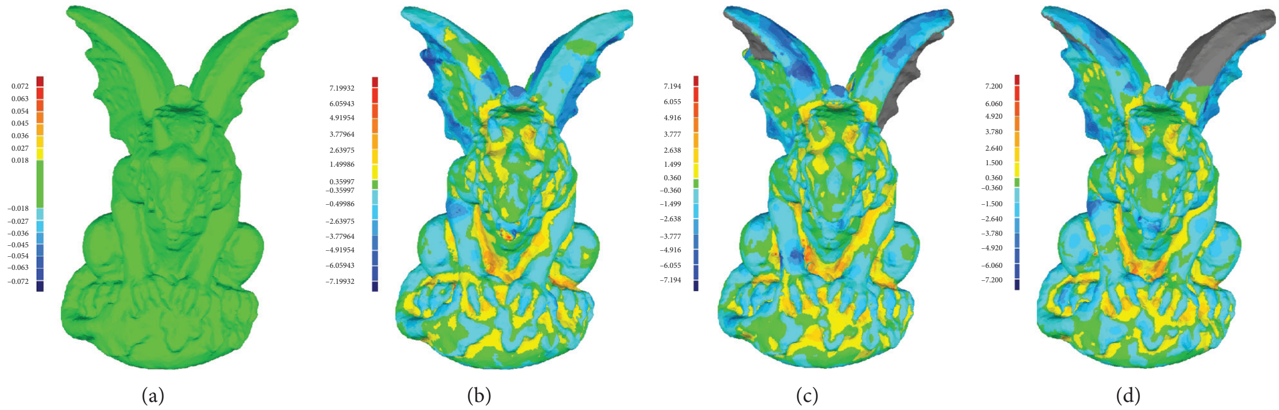

Figure 17: Point cloud 3D deviation for Gargo50 K. (a) Gargo50 K. (b) Grid sampling. (c) Poisson-disk sampling. (d) MFCM.

Table 1: Comparison of results (/mm).

Data	Parameters	Grid sampling	Poisson-disk sampling	MFCM
Bunny	Mean value	−0.00012	−0.00012	−0.00010
	Standard deviation	0.00066	0.00065	0.00064
Chair	Mean value	−5.246	−1.459	−3.893
	Standard deviation	7.604	2.161	6.400
Dino	Mean value	−0.00086	−0.00079	−0.00026
	Standard deviation	0.00271	0.00266	0.00211
Gargo50 K	Mean value	−0.351	−0.379	−0.388
	Standard deviation	1.329	1.252	1.311

Table 2: Comparison of results (/mm).

Data	Simplification rate (%)	Bunny	Chair
Literature [20]	91	Max error: 0.044	Max error: 6.2186
		Mean error:0.017	Mean error:3.2151
MFCM		Max error: 0.042	Max error: 6.4039
		Mean error:0.015	Mean error:3.4442

model, it is necessary to further retain the feature information from the model.

7. Conclusion and Future Work

To further improve the performance of the point cloud simplification algorithm and ensure the accuracy of the features, a new algorithm based on modified FCM with feature information reserved algorithm is proposed. By constructing the data structure of the point cloud, the geometric feature information of the point cloud were calculated, and then the point cloud was divided by the MFCM algorithm to provide a simplified data basis for the point cloud. Finally, the point cloud was simplified according to the simplification rules. The point cloud test data were simulated by using the simplified algorithm and the results were compared, and the results showed that our proposed algorithm can effectively improve the precision of point cloud simplification results and ensure the accuracy of part features.

Simplifying point cloud data is usually the first step of point cloud preprocessing. When this process is completed, further operations are required, such as registration process. Therefore, how to accurately move point cloud data into the actual working coordinate system is the next topic.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the Natural Science Foundation of Ningbo, China (Grant no. 2016A610039).

References

[1] Y. L. Tang, "A Typical Classification Method Based on Vehicle Laser Point Cloud," pp. 5-6, Beijing University of Technology, Beijing, China, 2015, Master's thesis.

[2] J. M. Dong, "Research on Simplification of Point Cloud with Preserved Features," vol. 6, Taiyuan University of Technology, Taiyuan, China, 2019, Master's thesis.

[3] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods," *Autonomous Robots*, vol. 44, no. 2, pp. 147–164, 2020.

[4] J. Xu, W. Hou, Y. Sun, and Y.-S. Lee, "PLSP based layered contour generation from point cloud for additive manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 1–12, 2018.

[5] J. Y. Huang, Y. F. Qu, and J. X. Jiang, "Method for identifying the landing area of unmanned aerial vehicle," *Chinese Journal of Liquid Crystals & Displays*, vol. 33, pp. 238–244, 2018.

[6] C. C. Chu and A. J. Aggarwal, "The integration of region and edge-based segmentation," in *Proceedings of the Third International Conference on Computer Vison*, pp. 117–120, Washington DC, USA, December 1990.

[7] R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 58–68, 1984.

[8] W. Sun, C. Bradley, Y. F. Zhang, and H. T. Loh, "Cloud data modelling employing a unified, non-redundant triangular mesh," *Computer-Aided Design*, vol. 33, no. 2, pp. 183–193, 2001.

[9] K. H. Lee, H. Woo, and T. Suk, "Point data reduction using 3D grids," *The International Journal of Advanced Manufacturing Technology*, vol. 18, no. 3, pp. 201–210, 2001.

[10] N. Dyn, M. S. Floater, and A. Iske, "Adaptive thinning for bivariate scattered data," *Journal of Computational and Applied Mathematics*, vol. 145, no. 2, pp. 505–517, 2002.

[11] B.-Q. Shi, J. Liang, and Q. Liu, "Adaptive simplification of point cloud using -means clustering," *Computer-Aided Design*, vol. 43, no. 8, pp. 910–922, 2011.

[12] Y.-Q. Wang, Y. Tao, H.-J. Zhang, and S.-S. Sun, "simple point cloud data reduction method based on Akima spline interpolation for digital copying manufacture," *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 9–12, pp. 2149–2159, 2013.

[13] Z. W. Sun and F. P. Da, "3D point cloud simplification algorithm based on fuzzy entropy iteration," *Acta Optica Sinica*, vol. 33, pp. 161–167, 2013.

[14] H. Han, X. Han, F. Sun, and C. Huang, "Point cloud simplification with preserved edge based on normal vector," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 19, pp. 2157–2162, 2015.

[15] X. J. Chen, G. Zhang, and X. H. Hua, "Point cloud simplification based on the information entropy of normal vector angle," *Chinese Journal of Lasers*, vol. 42, pp. 336–344, 2015.

[16] Q. Wang, H. Y. Wang, B. Q. Sun, G. Xia, and C. Xu, "Research and improvement of point cloud simplification algorithm in vehicle surface reconstruction," *Journal of Electronic Measurement and Instrumentation*, vol. 31, pp. 1693–1701, 2017.

[17] Y. Zang, B. Yang, F. Liang, and X. Xiao, "Novel adaptive laser scanning method for point clouds of free-form objects," *Sensors*, vol. 18, no. 7, pp. 2239–2267, 2018.

[18] Q.-N. Xiao, Z.-C. Huang, Z. Xu, and H.-B. Shang, "Study on sampling rule and simplification of LiDAR point cloud based on terrain complexity," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 11, pp. 1773–1784, 2018.

[19] W. Xuan, X. Hua, X. Chen, J. Zou, and X. He, "A new progressive simplification method for point cloud using local entropy of normal angle," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 4, pp. 581–589, 2018.

[20] H. Chen, X. M. Huang, and W. Q. Liu, "Study of laser point cloud simplification algorithm based on dynamic grid K-nearest neighbors searching," *Control and Decision*, vol. 34, pp. 1–6, 2019.

[21] A. R. E. Sayed, A. E. Chakik, H. Alabboud, and A. Yassine, "An efficient simplification method for point cloud based on salient regions detection," *RAIRO-Operations Research*, vol. 53, pp. 487–504, 2019.

[22] V. Markovic, J. Zivana, and Z. Miljkovic, "Feature sensitive three-dimensional point cloud simplification using support vector regression," *Tehnički vjesnik*, vol. 26, no. 4, pp. 985–994, 2019.

[23] C. F. Wang, G. H. Geng, J. B. Hu, and Y. J. Zhang, "Feature-aware three-dimensional point cloud simplification algorithm," *Laser & Optoelectronics Progress*, vol. 56, pp. 138–145, 2019.

[24] J. F. Chang, J. M. Zhao, and H. B. Wang, "Research on point cloud reduction algorithm based on boundary reservation," *Geomatics & Spatial Information Technology*, vol. 42, pp. 136–140, 2019.

[25] C. Ji, Y. Li, J. Fan, and S. Lan, "A novel simplification method for 3D geometric point cloud based on the importance of point," *IEEE Access*, vol. 7, pp. 129029–129042, 2019.

[26] A. Mahdaoui and E. H. Shai, "3D point cloud simplification based on k-nearest neighbor and clustering," *Advances in Multimedia*, vol. 2020, pp. 1–10, Article ID 8825205, 2020.

[27] J. Li, Y. Cao, Z. M. Wang, and G. Y. Wang, "Scattered point cloud simplification algorithm integrating k-means clustering and hausdorff distance," *Geomatics and Information Science of Wuhan University*, vol. 45, pp. 250–257, 2014.

[28] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[29] H. Hoppe, T. Duchamp, J. McDonald, and W. Stuetzle, "reconstruction from unorganized points," *ACM Siggraph Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.

[30] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the IEEE Visualization VIS 2002*, pp. 163–170, Boston, MA, USA, November 2002.

[31] T. F. DeRose, "Research on Complex Free-form Surface Parts Quality Inspection with 3D Registration Method," pp. 21-22, Huazhong University of Science and Technology, Wuhan, China, 2017, Doctor's thesis.

[32] Y. Yang, K. Zhang, G. Huang, and P. Wu, "Outliers detection method based on dynamic standard deviation threshold using neighborhood density constraints for three dimensional point cloud," *Journal of Computer-Aided Design & Computer Graphics*, vol. 30, no. 6, pp. 1034–1045, 2018.

[33] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[34] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy *c*-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[35] W. Zhang, J. Hu, G. Zhou, and M. He, "Detection of apple defects based on the FCM-NPGA and a multivariate image analysis," *IEEE Access*, vol. 8, pp. 38833–38845, 2020.

[36] H. Wang and W. Fei, "Failure recognition method of rolling bearings based on the characteristic parameters of compressed data and FCM clustering," *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 1–9, 2019.

[37] X. H. Wang, L. S. Wu, H. W. Chen, and G. L. Shi, "Region segmentation of point cloud data based on improved particle swarm optimization fuzzy clustering," *Optics and Precision Engineering*, vol. 25, pp. 1095–1105, 2020.

[38] P. Gurram, S. Hu, and A. Chan, "Uniform grid upsampling of 3D lidar point cloud data," *Three-Dimensional Image Processing (3DIP) and Applications 2013*, vol. 8650, 2013.

[39] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 6, pp. 914–924, 2012.