

Article

An Advanced Vehicle Body Part Inspection Scheme Based on Scattered Point Cloud Data

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; robotlib@shu.edu.cn (M.L.); maxie@nbufe.edu.cn (X.M.)

² School of Mechanical and Electrical Engineering College, Ningbo University of Finance and Economics, Ningbo 315175, China

* Correspondence: mryyfi@i.shu.edu.cn

Received: 28 May 2020; Accepted: 29 July 2020; Published: 4 August 2020



Abstract: To further improve the efficiency and accuracy of the vehicle part inspection process, this paper designs an accurate and efficient vehicle body part inspection framework based on scattered point cloud data (PCD). Firstly, a hybrid filtering algorithm for point cloud denoising is designed to solve the problem of multiple noise points in the original point cloud measurement data. Secondly, a point cloud simplification algorithm based on Fuzzy C-Means (FCM) is designed to solve the problems of a large amount of data and many redundant points in the PCD. Thirdly, a point cloud fine registration algorithm based on the Teaching-Learning-based Optimization (TLBO) algorithm is designed to solve the problem where the initial point cloud measurement data cannot be located properly. Finally, the deviation distance between the PCD and Computer-Aided-Design (CAD) model is calculated by the K-Nearest Neighbor (KNN) algorithm to inspect and analyze the point cloud after preprocessing. On the basis of the design algorithm, four groups that contain measurement data for eight vehicle body parts are analyzed and the results prove the effectiveness of the algorithm, which is very suitable for the inspection process of vehicle body parts.

Keywords: scattered point cloud data; vehicle body part; point cloud denoising; point cloud simplification; point cloud registration; vehicle parts inspection

1. Introduction

In the field of vehicle engineering, the design, manufacturing, inspection and evaluation of body parts constitute the whole life cycle of vehicle part production under the background of “Industry 4.0” [1]. Each of the above links will have a great impact on the final matching process of the vehicle body. In the production of the parts, the inspection and evaluation processes of the parts, in later stages of production, is the key to ensuring the accuracy of the parts and the quality of late matching. According to the actual engineering experience, rework or repair caused by the incorrect matching of parts accounts for a relatively high proportion [2]. On the other hand, given the continuous growth in the market demand for automobile quality, the matching quality requirements of the parts that have the greatest influence on body shape are correspondingly increasing. Generally speaking, the inspection process of the parts at this stage mainly includes two aspects—one is the manufacturing status of parts themselves, such as the evaluation of dimensional tolerance, form and position error; the other is the matching state of parts with other parts—that is, gap and interference evaluations between the body parts. However, no matter the necessary inspection and evaluation processes, the technical parameters of the parts themselves should be satisfied first.

Above all, as a key link of each product's lifecycle, the quality inspection of mechanical parts has been given increasing attention. In the process of vehicle assembly, during the actual late matching process, some parts of the body will be deformed during stamping, welding and painting prior to final assembly, which will directly affect the assembly accuracy of the whole vehicle [3]. Therefore, to ensure the quality of the late assembly of the parts, engineers and technicians estimate the matching state of the parts by means of measurements and later data processes to predict and adjust the parts in the later stage. However, traditional manual inspection techniques have low matching accuracy, a tedious matching process and a large workload. To further improve the matching precision of parts and guarantee the quality of the body, a new technical scheme based on scattered point cloud data (PCD) has been increasingly used in the assembly process [3,4]. This new inspection process relies on high data quality. Poor data quality which contain many invalid and redundant points can lead to many problems, such as high computational demand of point cloud processing, low work efficiency and large errors in the matching results. Therefore, the inspection process of vehicle body parts has been a research subject with practical engineering value.

The development of computer technology and three-dimensional (3D) point cloud measurement technology has provided a reliable technical tool for vehicle parts inspection. As part of an interdisciplinary field, the basic principle of this technology is to obtain PCD coordinate information for manufacturing molded parts by scanning them on their surface and to preprocessing them through denoising, simplification, registration and so forth, to provide reliable PCD data for subsequent quality inspection analyses. However, for the pre-processing step of PCD, selecting or designing effective algorithms can improve the inspection accuracy of vehicle body parts.

Therefore, further design and research of related algorithms are needed. On the basis of the above factors, this paper aims to study a set of computing processes to improve the inspection efficiency and accuracy of the vehicle body parts based on scattered PCD. We used the actual project data to analyze the location relationship between the scattered PCD to provide important data support and technical guarantee for the inspection of vehicle body quality. The main contributions of this paper are as follows:

- (1) For the inspection process of vehicle body parts, an analytical workflow based on point cloud data is established, which is an effective method for the inspection of vehicle body parts.
- (2) For the denoising process of the PCD of vehicle body parts, a hybrid denoising algorithm based on straight-through filtering and statistical filtering is proposed, which can extract the target point cloud data accurately.
- (3) For the simplification process of PCD, a point cloud subsampling method based on the Fuzzy C-Means (FCM) algorithm is designed, which can keep the vehicle part features while simplifying the point cloud.
- (4) For the fine registration process of PCD, the Teaching-Learning-based Optimization (TLBO) algorithm is applied to solve the mathematical model based on the Iterative Closest Point (ICP) algorithm, which can further improve the precision of the fine registration.

The structure of this paper is as follows. In Section 1, the research background is introduced. Section 2 describes and summarizes the research on parts assembly. In Section 3, the materials and methods of this paper are described in detail. In Section 4, the algorithm is tested and discussed. The final section summarizes the paper.

2. Related Work

2.1. Preprocessing of Point Clouds

For point cloud denoising, Fleishman et al. applied a bilateral filtering algorithm based on image denoising to the PCD filtering process [5]. In 2005, Fleishman et al. used the moving least squares (MLS) method to construct a point cloud neighborhood and then designed a forward search paradigm

for PCD smoothing, which achieved good results [6]. Gu et al. first removed abnormal points through statistical filtering and then filtered and smoothed the point cloud using an improved tri-lateral filtering algorithm [7]. Hermosilla et al. used unsupervised learning to denoise large-scale point clouds and achieved good results [8]. Liu et al. used the double-tensor voting algorithm to detect the features of PCD and combined Ransac and the multi-normal strategy to carry out point relocation to complete the point cloud denoising [9]. Zhou et al. designed a non-iterative double-threshold de-noising process based on the Canny detection algorithm for image processing [10]. This algorithm performed phased de-noising of the point cloud through the threshold and improved the denoising efficiency.

For the point cloud simplification process, Chen et al. introduced a triangular grid into the point cloud processing process and then simplified the grid using a vector weighting algorithm to complete the point cloud simplification process [11]. Lee et al. simplified the point cloud data through the geometric information of parts and validated the algorithm by using two sets of part PCD [12]. Dyn et al. designed an adaptive point cloud simplification strategy and the experiment proved the accuracy and efficiency of the algorithm [13]. Han et al. used the octree to construct the topological relationship between points and designed different sampling strategies for edge points and non-edge points, respectively, to simplify the point cloud data [14]. The test set was analyzed and verified. Shao et al. introduced the improved particle swarm optimization algorithm to the average distance method in the simplified point cloud algorithm and applied the algorithm to the online measurement of the 3D scanning of the granary [15]. The experiment shows the effectiveness of the algorithm.

For the point cloud registration process, Senin et al. provided a point cloud registration method based on different levels for a measurement scheme of multiple sensors. This method registers the data according to the accuracy requirements of the point cloud and improves the registration accuracy and efficiency of the point cloud [16]. Huang et al. designed a point cloud registration algorithm based on spherical feature constraints for curved point cloud data. By constructing virtual overlapping areas, the point cloud movement parameters were calculated and the effectiveness of the algorithm was verified by experiments [17]. Du used deep learning to register three-dimensional images with noise and the effectiveness of the algorithm was proven through experiments [18]. Li et al. first carried out a rough registration of point clouds using a weighted l_q -norm and then completed the process of fine registration by using topology diagram theory. This algorithm is considered to have good stability [19]. Based on the registration algorithm of the fast point feature histogram, Wu optimized the neighborhood density of points to complete the coarse registration process and ultimately carried out fine registration with ICP to improve the registration accuracy of the point cloud [20].

It can be seen from the relevant literature that there are various pre-processing methods for point clouds. For point cloud denoising algorithm, although bilateral filtering has a great advantage in feature retention, it needs to calculate the feature information of point cloud. When the noise points are relatively dense, the error cannot be controlled. However, clustering and other algorithms cannot effectively identify target point cloud and outlier groups.

For point cloud simplification algorithm, it not only need to consider the number of point cloud in the downsampling process but also retain the point cloud features. In addition, in the field of vehicle body part inspection, the number of point cloud data is large, it is necessary to compress the point cloud, which can improve the efficiency of analysis and reduce workload in further exploration and research.

For point cloud registration, it is still mainly based on ICP algorithm, although various improved method has improved the efficiency of the point cloud registration process. However, there are still the problems such as a lower calculation precision. Therefore, fine registration process is important to improve its computation accuracy.

Based on above analysis, it is important to select an appropriate algorithm for the point cloud data of vehicle body parts.

2.2. Parts Inspection

In the field of parts inspection, Ramaswamy et al. designed a robotic virtual disassembling system based on Agent technology in 1999 [21]. Tching et al. developed a virtual assembly system based on virtual reality (VR) technology and computer aided design (CAD). This system first used computer graphics to position the parts and then applied the kinematic principle to assemble the parts [22]. Liang et al. built a set of virtual assembly modeling systems for a satellite. This technique improved the assembly efficiency of the satellite parts through a tree structure [23]. Jiang et al. combined Unity 3D and Computer Aided Three-dimensional Interactive Application (CATIA) to build a virtual assembly training platform based on the Kinect V2 technology. The platform was tested and good experimental results were obtained [24]. For the above techniques, parts inspection technology is a more design-oriented process and based more strongly on an ideal model than measured data; thus, it cannot simulate a real state. Thus, the actual state of the parts cannot be evaluated. At present, the development of non-contact laser scanning technology has helped improve inspection objectivity and simulate a real matched state based on PCD. This research is developing rapidly and has made significant advancements in many areas, there are many scholars and engineers apply it to all walks of life [25,26], Li provided new technology for a complex curved surface quality inspection method based on PCD [27]. An experimental analysis of complex parts proved the efficiency of the algorithm. Bao et al. proposed a quasi-physical assembly method and successfully applied it to the aerospace cabin inspection process [28]. Zhu et al. used PCD inspection technology, such as the trial assembly process, for mining machinery equipment, thereby providing a new method for the inspection process of mining machinery and equipment [29]. Yu studied composite part inspection based on PCD [30]. Zhao applied PCD inspection technology to the automobile body inspection field by using a non-contact measuring device to obtain the PCD [31].

The essence of the part inspection process is the processing and analysis of PCD. The PCD obtained by non-contact measurements can reflect the actual state of the parts more truly and can help analyze the actual matching position of the parts more accurately. For the initial PCD, the problems of noise data, large data volumes and inaccurate spatial positions greatly affect the quality of PCD.

In the above studies, most PCDs were handled by manual operation or commercial software. Although these methods function, they are usually inefficient and require a large workload. Moreover, at present, most of the research in this field is limited to one of the modules such as denoising or registration and there are few studies on the systematic process of body parts inspection. Therefore, based on the relevant research, a new vehicle body part inspection framework is established in this paper. Firstly, a hybrid filtering denoising algorithm is designed for point cloud noise. Secondly, a point cloud simplification algorithm based on FCM is designed to solve the lightweight point cloud problem. This method simplifies the point cloud by obtaining the normal vector, curvature and point density of the PCD based on the 3D coordinates of the point cloud and constructs eight-dimensional PCD sets using the FCM clustering algorithm to classify the characteristics area and non-characteristics area of the PCD. The curvature method and bounding box method are used to simplify the different areas. Finally, a point cloud fine registration algorithm based on the TLBO algorithm is designed to solve the point cloud positioning problem. This method constructs a mathematical model of point cloud registration based on the Iterative Closest Point (ICP) principle and then uses the TLBO algorithm for further calculations to obtain the best registration position. In addition, four groups of point cloud measurement data were verified and analyzed to assess the proposed method, including the processes of denoising, simplification and registration. Moreover, *K*-Nearest Neighbor (KNN) is introduced into the inspection of the vehicle body parts.

3. Materials and Methods

3.1. Materials

In this study, the data of vehicle body parts were mainly obtained via optical non-contact measurement equipment. The data obtained by this method are provided in a 3D format using spatial coordinates (x , y , z) and the principle of this method is to obtain the spatial coordinate information of the measured part's surface by laser scanning. In general, according to the distribution of the PCD, the PCD can be divided into regular PCD and irregular PCD which is also called scattered PCD shown in Figure 1 [32], Where Figure 1a shows the regular PCD, Figure 1b shows the irregular PCD. Regular PCD is generally acquired by a single scan. However, in order to obtain more sufficient PCD, it is usually necessary to carry out random multiple scans on parts, so the PCD become messy and irregular. Scattered PCD is the most common data distribution form.

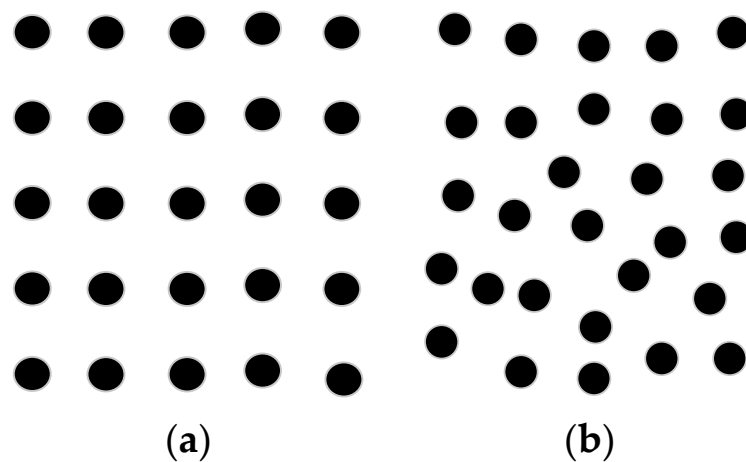



Figure 1. The point cloud data (PCD) classification, (a) is regular PCD, (b) is irregular PCD.

In this paper, the measurement equipment used to obtain the scattered PCD and its parameters is shown in Table 1. In the point cloud collection process, Man, Machine, Material, Method and Environments are required to follow the equipment parameters and relevant international standards [33]. The CAD and scattered PCD of the data studied in this paper (based on the project data source and application analyses) are shown in Figure 2 to Figure 3 and need to be further processed.

Table 1. Measurement parameters.

	Measurement Parameters	Value
	Fineness of Scanning	7640 (point/line)
	Scan Rate	458,400 (point/s)
	Measurement Accuracy	0.0240 (mm)
	Frequency	60 (Hz)
	Resolution Ratio	0.01370 (mm)
	Temperature	10–40 (°C)

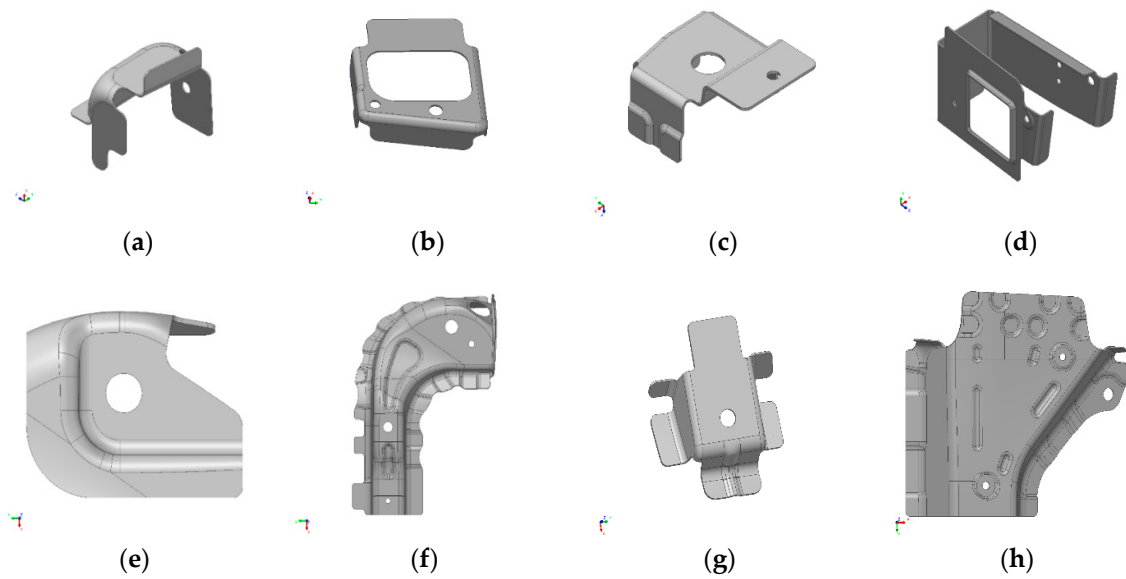


Figure 2. The computer aided design (CAD) model. (a)–(h) are a series of CAD model of vehicle body parts, respectively.

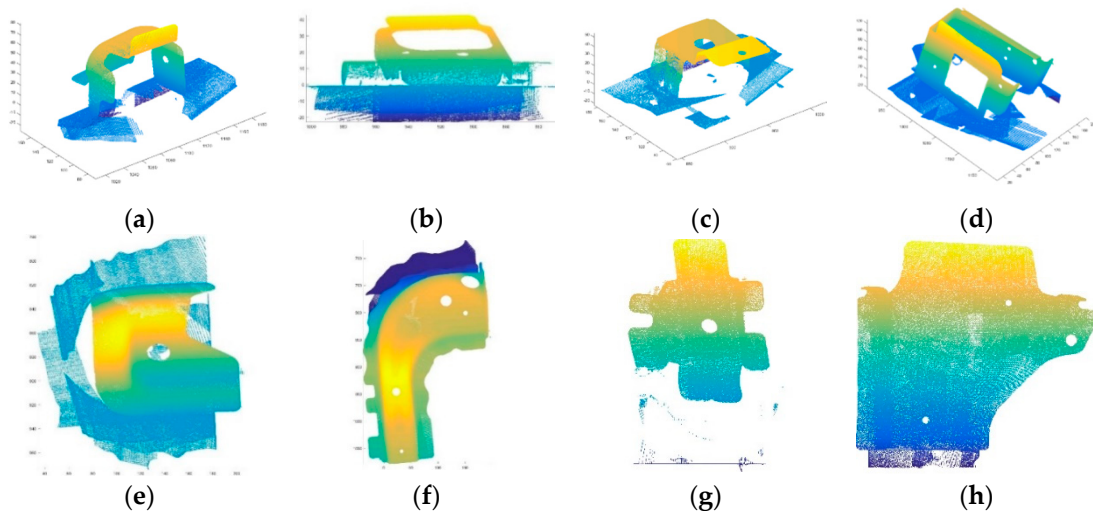


Figure 3. The experimental data. (a)–(h) are a series of measurement data of vehicle body parts, respectively.

3.2. Methods

3.2.1. The Scheme Proposed

For the inspection process of vehicle body parts, this paper preprocesses the initial PCD to obtain and analyze the data in high quality. Based on the PCD data, this paper proposes the following framework.

- (1) A new hybrid filtering algorithm is proposed for point cloud denoising.
- (2) A new point cloud simplification algorithm is proposed to reduce the amount of point cloud data.
- (3) A new point cloud registration strategy is used to the measurement point cloud in the vehicle coordinate system.
- (4) The *K*-Nearest Neighbor (*KNN*) algorithm performs point cloud error inspection, which is also applied to the above three computing processes.

The scheme flow proposed in this paper for the inspection process of vehicle body parts is shown in Figure 4. This paper processes the initial point cloud data acquired by non-contact measurement equipment through pre-processing steps such as denoising, simplification and registration and ultimately assesses the real quality of the parts. The specific algorithm will be discussed in detail in the following section.

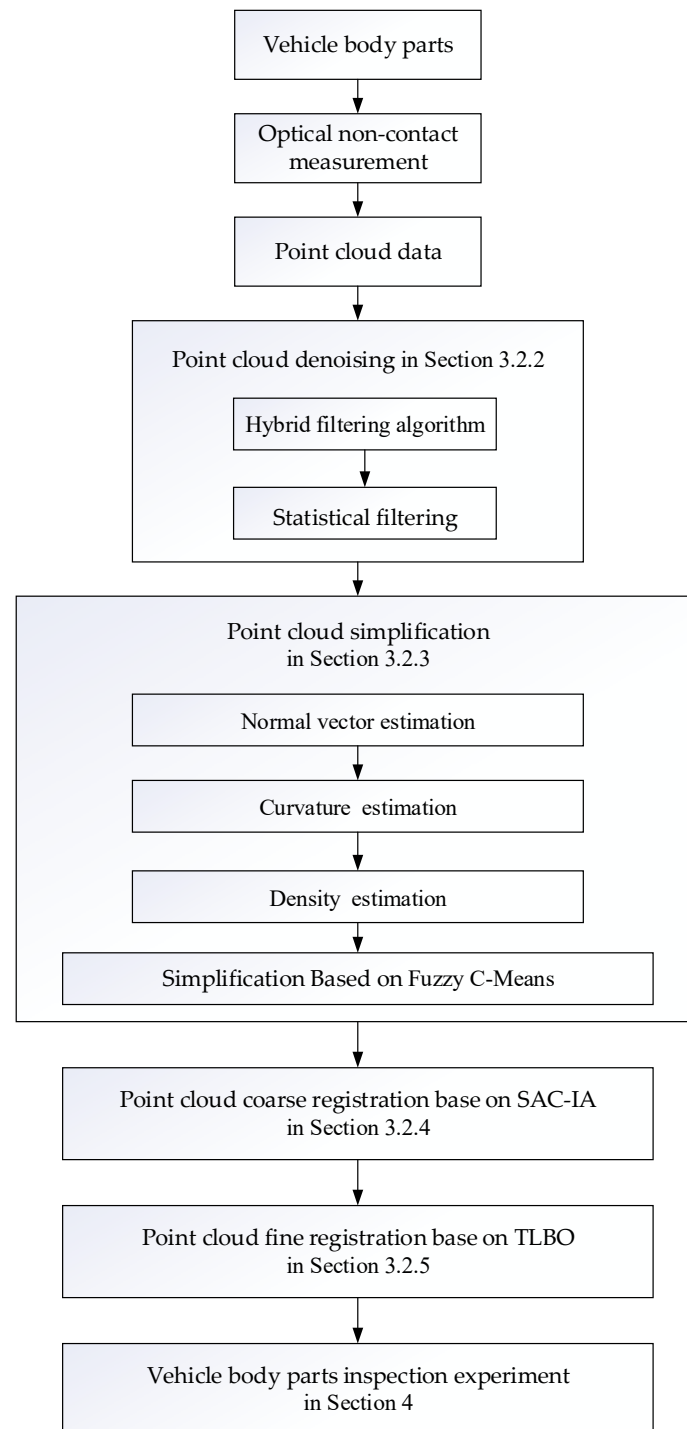


Figure 4. The scheme flow of inspection process.

3.2.2. Denoising of PCD Based on a Hybrid Filtering Algorithm

The initial PCD generally features a great deal of noise and many outliers, as shown in Figures 2 and 3. These factors will affect subsequent analyses of the point cloud. Therefore, to avoid the impact of noise points on the simplified registration process, the PCD first needs to be filtered. The noise-denoising process is performed using the following steps.

Step 1. Invalid point removal based on straight-through filtering [34]; the calculation process is shown in Equation (1).

$$\begin{cases} x_{min} \leq x_i \leq x_{max} \\ y_{min} \leq y_i \leq y_{max} \\ z_{min} \leq z_i \leq z_{max} \end{cases}, \quad (1)$$

where x_i , y_i and z_i are the coordinate values of point p_i ; x_{min} , y_{min} and z_{min} are the minimum coordinate values in the x , y and z directions; and x_{max} , y_{max} and z_{max} are the maximum coordinate values in the x , y and z directions. Thus, the invalid point cloud is removed by setting the part range of the PCD.

However, the pass-through filtering algorithm needs to get the specific size of the parts according to the drawing information firstly and set the parameters manually. This method is subjective to some extent.

Step 2. Outlier removal based on statistical filtering. Because of the large amount of the PCD, the k -dimensional (k -d) tree [35] and KNN [36] are introduced into the traditional statistical filtering algorithm to further improve the efficiency of the algorithm [37]. The algorithm flow is as follows.

- (1). Read into the PCD set $P_i(p_1, p_2, \dots, p_n)$ and build the k -d tree;
- (2). For each point p_i , calculate the average distance between point p_i and its k neighborhood point. d_{ij} is the distance between point p_i and p_j , and \bar{d}_i is the mean distance of the k neighborhood point in p_i . This process is shown in Equations (2) and (3).

$$d_{ij} = \sqrt{(p_{i,x} - p_{j,x})^2 + (p_{i,y} - p_{j,y})^2 + (p_{i,z} - p_{j,z})^2} \quad (2)$$

$$\bar{d}_i = \frac{1}{k} \sum_{j=1}^{i=n} \sum_{j=1}^{j=k} d_{ij}. \quad (3)$$

- (3). Calculate the mean value \bar{d} and the standard deviation d_{std} of all \bar{d}_i . This is shown in Equations (4) and (5).

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{i=n} \bar{d}_i \quad (4)$$

$$d_{std} = \sqrt{\frac{\sum_{i=1}^n (\bar{d}_i - \bar{d})^2}{n-1}}. \quad (5)$$

- (4). For each selected point p_i , if its \bar{d}_i is greater than the set threshold L , remove it from $P_i(p_1, p_2, \dots, p_n)$. If not, keep point p_i . Threshold L is shown in Equation (6).

$$L = \bar{d} + \sigma \times d_{std}, \quad (6)$$

where σ a coefficient.

- (5). After calculations are complete, output the PCD after noise removal.

Algorithm 1 provides pseudo-code of the hybrid denoising algorithm.

Algorithm 1 Pseudocode of hybrid denoising algorithm**Begin**

1. **Input** $P_i(p_1, p_2, \dots, p_n)$;
2. **Initialization** Parameter $k, \sigma, x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}$;
3. **Denoise** point data according to Equation (1);
Output $P_i(p_1, p_2, \dots, p_{n'})$;
4. **Create** k - d tree
For $i = 1$ to n'
Build all point k - d tree;
End;
5. **Crear** KNN
For $i = 1$ to n'
Build all point kd -tree;
End;
6. **Statistical filter**
For $i = 1$ to n'
Calculate point KNN mean distance \bar{d}_i according to Equations (2) and (3);
Calculate mean value \bar{d} and standard deviation d_{std} of all \bar{d}_i according to Equations (4) and (5);
Determine whether point p_i is a noise point according to the Equation (6);
End;
Output $P_i(p_1, p_2, \dots, p_{n''})$;

End

Based on the above discussion, this paper uses the measured part data and related denoising algorithm to verify the hybrid denoising method. The experimental results are shown in Figure 5.

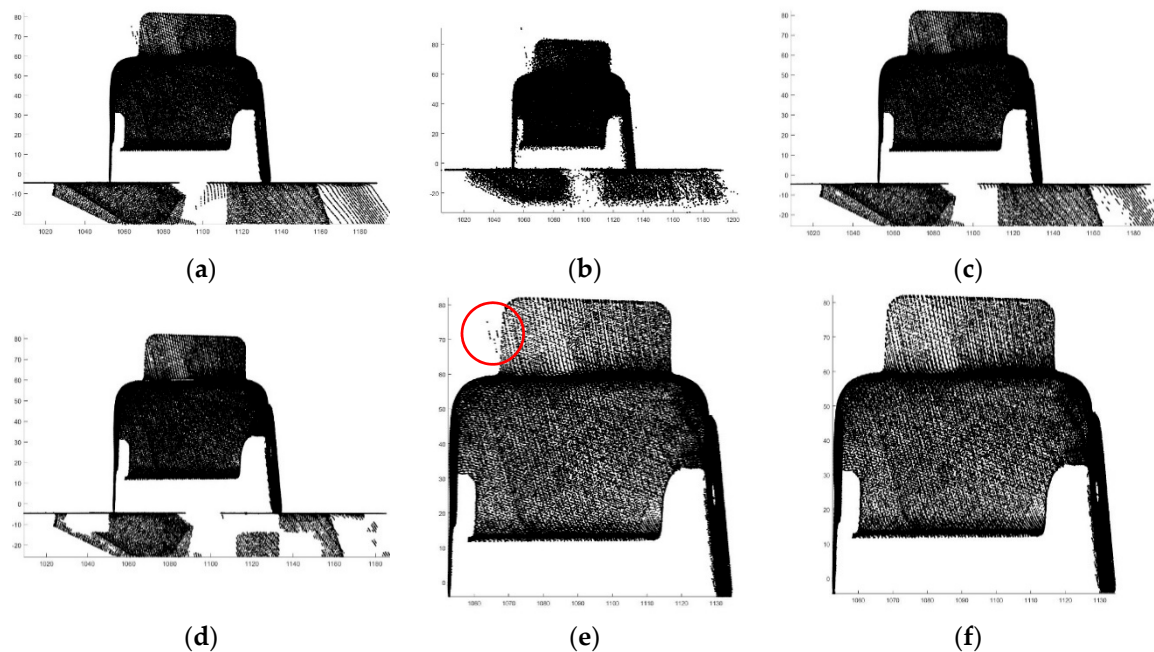


Figure 5. Point cloud denoising experiment. (a) is the initial point cloud data, (b) is the calculation result of Laplace filtering, (c) is the calculation result of neighborhood filtering, (d) is the calculation result of statistical filtering, (e) is the calculation result of straight-through filtering, and (f) is the calculation result of hybrid filtering in this paper.

From the experimental results, in the results of Figure 5b, the Laplace denoising effect is poor. Figure 5c present the neighborhood filter and Figure 5d shows the statistical filtering denoising results.

Here, the external data noise density distribution characteristics are similar to those of the target point cloud and the area is larger. The two algorithms with good performance in the random denoising process cannot effectively remove the noise and invalid points outside the target point cloud. Therefore, it is necessary to adopt different strategies to remove different types of noise data from the point cloud. It can be seen from the experimental results in Figure 5e that the straight-through filtering algorithm can effectively filter invalid point cloud data. As shown in Figure 5f, the hybrid algorithm can remove large-scale noise points and outliers from the initial point cloud. The new algorithm can ensure the precision and accuracy of denoising for the vehicle point cloud data measured in this paper.

3.2.3. Simplification of PCD Based on FCM

Initial point cloud data usually feature a large amount of data, which will greatly affect the efficiency of subsequent point cloud analyses. To simplify the point cloud data, this paper designs a point cloud simplification algorithm based on feature retention. The specific calculation process is as follows.

Normal Vector Estimation of PCD

Principal component analysis (PCA) [38] is a dimensionality reduction algorithm for data analysis without parameter limitations. The normal vector of a point is estimated by fitting the neighborhood of that point into a plane. For a point cloud $P = (p_1, p_2, \dots, p_k)^T$ with a certain point p_i first build a k -d tree and KNN; then, constructs the local plane E by using a fitting process for these points, as shown in Equations (A1), (A2), (7) and (8).

$$C_0 = \frac{C}{n} = \frac{1}{n} \sum_{i=1}^n (p_i - \bar{p})^T (p_i - \bar{p}) \quad (7)$$

$$C_0 \cdot e_m = \gamma_m \cdot e_m, \quad (8)$$

where C_0 is a symmetric semidefinite matrix; eigenvalues $\gamma_0, \gamma_1, \gamma_2$ which satisfy that $\gamma_0 \leq \gamma_1 \leq \gamma_2$ and the three eigenvalues are non-negative real numbers. The eigenvector corresponding to the minimum eigenvalue is a normal vector $l(i, j, k)$ of the least-squares fitting plane required.

Although the direction estimated by the point cloud has a certain randomness, the surface fitting result will divide the normal vector into two directions inside and outside, meaning that the same surface normal cannot be unified and affecting subsequent analyses. However, it can be corrected with an algorithm [38], as shown in Equation (9).

$$l' = \begin{cases} ll(v-p) \geq 0 \\ -ll(v-p) < 0 \end{cases} \quad (9)$$

where, l is the normal vector and v is the viewpoint direction.

Curvature Estimation of the PCD

The curvature information of the point also reflects the degree of bending deformation in the point position. In general, the characteristic value can be obtained by the covariance matrix of the local point cloud by PCA. Then, the characteristic value can be estimated by the eigenvalues. Because the normal vector was adjusted by PCA in Equation (9). The eigenvalues change accordingly. On the other hand, the accurate point cloud normal vector provides parameter information for the local surfaces in Equations (7), (8) and (9). Based on the above analysis and discussion, this paper uses the move least square method (MLSM) [39] to calculate the point cloud curvature. The calculation process is as follows.

Step 1. Input the point cloud coordinate $p_i(x_i, y_i, z_i)$ and the normal vector $l(i, j, k)$;

Step 2. Establish a surface implicit function that enables the local energy function $E(x)$ to be minimal in the n -direction vector field as shown in Equations (A3) and (A4);

Step 3. Further calculations can obtain Gaussian curvature, as shown in Equation (10):

$$GC = - \frac{\det \begin{bmatrix} \nabla(\nabla f(x)) & \nabla^T f(x) \\ \nabla f(x) & 0 \end{bmatrix}}{\|\nabla f(x)\|^4} \quad (10)$$

The mean curvature is shown in Equation (11):

$$MC = \frac{\|\nabla f(x)\|^2 \text{tr} \nabla - \nabla f(x) \nabla(\nabla f(x)) \nabla^T f(x)}{\|\nabla f(x)\|^3}, \quad (11)$$

where ∇ is the gradient operator and $\nabla(f(x))$ is the black plug matrix. In this paper, the average curvature is taken as the calculation parameter of the curvature of the point cloud.

Point Cloud Density Calculation

The density of the point cloud is also an important parameter to measure the location of the point cloud. There are many ways to estimate the density of the point cloud. In this paper, the reciprocal of the average distance between the point and the neighborhood point k is taken as the density parameter ρ_i of point p_i , as shown in Equation (12):

$$\rho_i = \frac{1}{\frac{1}{k} \sum_{i=1}^n \sum_{j=1}^{j=k} d_{ij}}, \quad (12)$$

where ρ_i is the density parameter of point p_i .

Feature Retention Point Cloud Simplification Algorithm Based on Fuzzy C-Means (FCM)

At present, the clustering algorithm is one of the important methods for data classification. In this paper, the point cloud is segmented by clustering and then different regions are simplified to achieve point cloud sampling.

Unlike a traditional hard clustering algorithm which is represented by K -means clustering [40], FCM [41] adopts a soft partition method. Similarly, unlike traditional deterministic sets, the data ownership of fuzzy sets is uncertain and the degree of data ownership is measured by the degree of data membership. For point set $p = \{p_1, p_2, \dots, p_n\}$, combining the point cloud coordinate data and point cloud geometric information calculated, point cloud data set is expanded from three-dimensional coordinate data $p_i(x, y, z)$ to eight-dimensional data $p_i(x, y, z, i, j, k, MC_i, \rho_i)$. Then, data set p is divided into c classes by clustering segmentation. The subsample data is $p_c = \{p_{c,1}, p_{c,2}, \dots, p_{c,i}\}$, where the membership function of the fuzzy set is $F(p_c)$. The closer $F(p_c)$ is to 1, the greater the degree that it belongs to the same set and vice versa. Fuzzy set is closer to a real description of the objective world. The basic calculation process of the point cloud is as follows.

Step 1. Input the point data set $p = \{p_1, p_2, \dots, p_n\}$, where p_i is $(x, y, z, i, j, k, MC_i, \rho_i)$. Then go to **Step 2**;

Step 2. Parameter initialization. These parameters mainly include the cluster number C , fuzzy index m , iteration termination threshold ε and maximum iteration number T . The random initialization membership matrix $U = [u_{ki}]_{c \times n}$, which satisfies Equation (13). Then, go to **Step 3**;

$$\begin{cases} u_{ki} \in [0, 1], \forall i = 1, 2, \dots, n, \forall k = 1, 2, \dots, c \\ \sum_{k=1}^c u_{ki} = 1, \forall i = 1, 2, \dots, n \\ 0 < \sum_{i=1}^n u_{ki} < n, \forall k = 1, 2, \dots, c \end{cases}, \quad (13)$$

where $U = [u_{ki}]_{c \times n}$ is the membership matrix.

Step 3. Calculate the clustering center c , as shown in Equation (14).

$$c_i = \frac{\sum_{j=1}^n u_{ij}^m x_i}{\sum_{j=1}^n u_{ij}^m}. \quad (14)$$

Step 4. The value of the objective function is calculated according to Equation (15).

$$J(U, V) = \sum_{k=1}^c \sum_{i=1}^n u_{ki}^m \|x_i - v_k\|^2. \quad (15)$$

$c \in [2, n]$ is the clustering number, $\|x_i - v_k\|$ is the Euclidean distance between sample i to clustering center k and $m \in [1, +\infty)$ is the fuzzy index used to control data partitioning. The value of this parameter is generally set as 2. and $V = \{v_1, v_2, \dots, v_c\}$ is the clustering center.

Step 5. Calculate the new membership matrix J , as shown in Equation (16).

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}}. \quad (16)$$

Step 6. The termination judgment is calculated. When the maximum iteration number meets the requirements or reaches the threshold, the algorithm is terminated and the result is output. If not, return to **Step 2.** until the iteration number meets the requirements.

In this article, PCD is clustered by FCM and the feature area and non-feature area of the PCD are divided by the mean curvature of each clustered area. The average curvature \overline{MC}_i of each area is obtained from the above calculation and the overall average curvature of the \overline{MC} and the standard deviation MC_{std} can be calculated, as shown in Equations (17) and (18):

$$\overline{MC} = \frac{1}{n} \sum_{i=1}^c \overline{MC}_i \quad (17)$$

$$MC_{std} = \sqrt{\frac{\sum_{i=1}^c (\overline{MC} - MC_i)^2}{n-1}}. \quad (18)$$

This determines whether the area is a feature part. When $\overline{MC}_i > L$, this area is a feature area and the other area is a non-featured area, as shown in Equation (19):

$$L = \overline{MC} + \sigma \times MC_{std}, \quad (19)$$

where the value of σ is 1 to 5.

Considering the above discussion, the calculation process of the new point cloud algorithm based on FCM is shown below.

Step 1. Input point cloud data $P = (p_1, p_2, \dots, p_k)^T$. Initialize the algorithm parameters and then go to **Step 2.**

Step 2. Calculate the normal vector $l(i, j, k)$ of the PCD using the PCA algorithm and correct the direction of the normal vector, then go to **Step 3.**

Step 3. Based on the results in **Step 2** calculate the mean curvature and point density of the point cloud data by MSLM; then, go to **Step 4.**

Step 4. Divide the point cloud by the coordinates of the point cloud combined with the vector, the mean curvature and density, then go to **Step 5.**

Step 5. The non-feature area simplification of the PCD. The non-feature area point clouds are relatively flat and evenly distributed. It is useful to simplify the distribution of non-featured areas by

using the bounding box algorithm [42], which can provide a reliable way to simplify non-feature areas by setting the required simplification parameters. The algorithm flow is as follows:

Step (1). Input the non-feature area point cloud and the minimum and maximum values for x , y and z in each direction.

Step (2). For the data in **Step (1)**, obtain the bounding box edge length and split the bounding box at a simplified scale.

Step (3). Reserved the center point of each bounding box and output it.

Step 6. Point cloud feature area simplification. For feature areas, the curvature sampling principle is used to calculate the average curvature MC_i of the points. as shown in the Equation (20).

$$|MC_i - \overline{MC}| > \varepsilon, \quad (20)$$

where a set of point cloud data is retained by value ε .

Step 7. Calculating complete. Output simplified point cloud.

The Algorithm 2 is pseudocode of the proposed simplification algorithm.

Algorithm 2 Pseudocode of Simplification Algorithm

Begin

Input $P_i(p_1, p_2, \dots, p_n)$;

Define parameter k, C, m, ε, T ;

For $i = 1$ to n

 Calculated the normal vector l_i by PCA as Equations (7) and (8);

 Correction point cloud normal vector as Equation (9);

 Build the new point cloud $p_i(x_i, y_i, z_i, i, j, k)$;

For $i = 1$ to n

 Calculated the Mean curvature MC_i and point density ρ_i as Equations (10)–(12), respectively;

 Build the new point cloud $p_i(x_i, y_i, z_i, i, j, k, MC_i, \rho_i)$;

End

 Segment the point cloud data $p_i(x_i, y_i, z_i, i, j, k, MC_i, \rho_i)$ by FCM as Equations (13) to (19);

 Simplify point cloud data as bounding box algorithm and Equation (20);

 Output the simplified point cloud data.

End

End

This set of point cloud experimental data is derived from the actual measurements of vehicle body parts. The total number of vehicle body parts contained in this file is 227,970 in txt format. The experiment provided a 90% point cloud simplification rate, which means that only 10% of the original point cloud data were retained. The simplified results are shown in Figure 6.

Figure 6 illustrates the calculation results of the vehicle body part point cloud under different simplified algorithms. In Figure 6, from (b) to (f), the number of remaining data after the point cloud was simplified were 22797. In Figure 6, a is the original point cloud data for the vehicle part, b is the simplified result of random sampling [43], c is the simplified result of curvature sampling [44], d is the simplified result of grid sampling [45], e is the simplified result of octree-based sampling [46] and f is the simplified result of our sampling algorithm.

As can be seen from Figure 6, the points clouds simplified by random sampling (Figure 6b) and the curvature method (Figure 6c) are not evenly distributed, while the grid sampling (Figure 6d) and octree-based sampling (Figure 6e) results are too evenly distributed and cannot accurately reflect the feature area. Compared to the above algorithm, our algorithm (Figure 6f) can fully reflect the feature areas and non-feature regions.

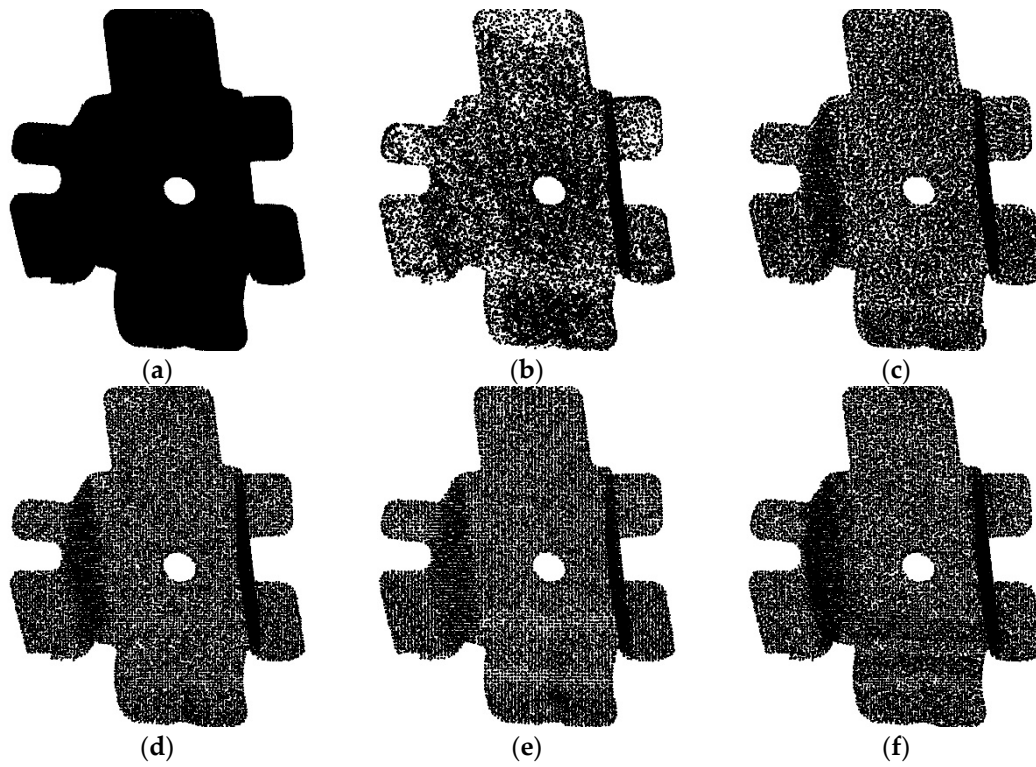


Figure 6. Point cloud simplification experiment. (a) is the initial point cloud data, (b) is the calculation result of random sampling, (c) is the calculation result of curvature sampling, (d) is the calculation result of grid sampling, (e) is the calculation result of octree-based sampling, and (f) is the calculation result of our sampling algorithm in this paper.

To further improve the effectiveness of the FCM algorithm in simplifying point clouds, we again adopt the application of curvature information and curvature information entropy commonly used by other scholars [46] to study the computational effects of different simplified algorithms. Curvature information entropy is used to quantitatively express the features of point clouds and its entropy function En_i is calculated by Formulas (21) and (22) [47].

$$En_i = -\frac{MC_i}{MC_i + \sum_{j=1}^k MC_j} \log_2 \frac{MC_i}{MC_i + \sum_{j=1}^k MC_j} - \sum_{j=1}^k \frac{MC_j}{MC_i + \sum_{j=1}^k MC_j} \log_2 \frac{MC_j}{MC_i + \sum_{j=1}^k MC_j} \quad (21)$$

$$En = \sum_{i=1}^n En_i. \quad (22)$$

Figure 7 shows the information entropy of PCD under different simplified algorithms, as can be seen from the Figure 7, according to the calculation results of entropy under the condition of the same reduction rate, the FCM simplified point cloud entropy value is the largest and the retained point cloud feature information is the most. Therefore, compared with the other four algorithms, the FCM point cloud algorithm proposed in this paper has better point cloud simplification effect.

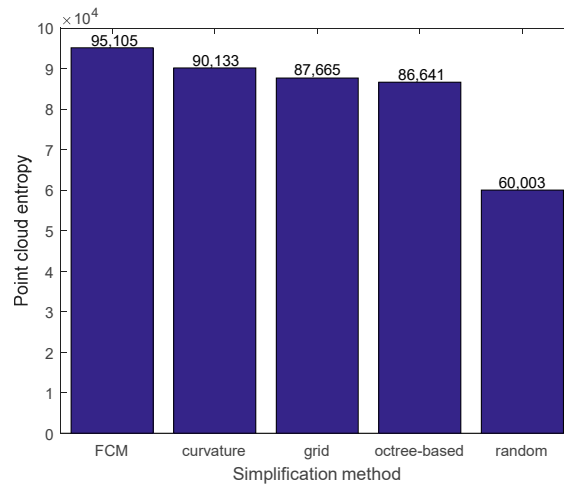


Figure 7. Entropy under different simplified algorithms.

3.2.4. Registration of PCD

The ICP (Iterative Closest Point) algorithm is a classical point cloud registration algorithm proposed by Besl [48], which established the foundations of this field. The basic principle of this algorithm is to find the closest corresponding location between two sets of point clouds. The ICP algorithm does this by calculating the mean square of the distances between the target point cloud and the measurement point cloud in Euclidean Space. Specifically, through matrix transformation, this method transfer the moving point cloud to the target point cloud's location and completes the point cloud registration via a number of iterations of point cloud movements. To move a spatial point, the point's initial coordinate is defined as $p = (x, y, z)$, while the coordinate after moving is defined as $p' = (x', y', z')$, When p is shifted in space, its motion is as follows:

Based on the ICP calculation principle, its general error function [48], F , can be described as:

$$F(\alpha, \beta, \gamma, d_x, d_y, d_z) = \text{Min}(\frac{1}{N} \sum_{i=1}^N \|V_i - (Rp_i + T)\|^2), \quad (23)$$

where V_i is the target point cloud. When F is the minimum value, the final position of the point cloud registration and the spatial motion parameters $(\alpha, \beta, \gamma, d_x, d_y, d_z)$ of the measurement point cloud are calculated. In this way, point cloud registration is transformed into a minimization problem. The remaining parameters are shown in the Appendix A (A5)–(A9).

However, the point cloud registration process is a complex nonlinear problem, which generally needs to be solved in two stages.

Coarse Registration of PCD

The coarse registration process of the point cloud is an important step to complete the spatial alignment. The coarse registration process can provide a good initial position for fine registration and avoid the process of fine registration falling into the local optimum too early. There are many kinds of coarse registration algorithms. We chose the sample consensus initial alignment (SAC-IA) [49] because of its good coarse registration performance. In this paper, the SAC-IA algorithm is used to complete the coarse registration process of the point cloud. The calculation process is as follows:

Step 1. For the measurement point cloud $P(p_1, p_2, \dots, p_n)$, first, the normal vector information of the point cloud is obtained from the Equations (7)–(9), and m sampling points are selected from P to construct their fast point feature histograms (FPFH). To avoid similarities among the selected feature

points, the distance threshold value d_ε is set to distinguish them. $F(p_{i,k})$ is the FPFH feature of point p_i , as determined in Equation (24):

$$F(p_{i,k}) = \text{SPFH}(p_{i,k}) + \frac{\sum_{j=1}^k \omega_j \cdot B(p_{i,k})}{k}, \quad (24)$$

where, $B(p_{i,k})$ is the relative relationship between point p_i and its KNN; ω_j is the characteristic weighted value of SPFH of point j .

Step 2. Similarly, for the reference point cloud $P'(p'_1, p'_2, \dots, p'_n)$, the FPFH is calculated via the normal calculations and a group of points with similar features in the measurement point cloud $P(p_1, p_2, \dots, p_n)$ is searched through the FPFH corresponding to the sampling points from the measurement points.

Step 3. This step calculates the transformation matrix between the corresponding points and then judges the performance of the current registration transformation by solving the distance error function after the corresponding point transformation. The distance error function here is mostly represented by the Huber penalty function, denoted as:

$$H(\varepsilon_i) = \begin{cases} 0.5 \times \varepsilon_i^2 & \|\varepsilon_i\| < d_\varepsilon \\ 0.5 \times d_\varepsilon (2 \|\varepsilon_i\| - d_\varepsilon) & \|\varepsilon_i\| \geq d_\varepsilon \end{cases}, \quad (25)$$

where, ε_i is a preset value and d_ε is the distance difference after transformation of the point corresponding to the i th group.

Step 4. Returns **Step 1~Step 3** until the maximum number of iterations is reached and outputs the transformation matrix and the spatial coordinates after coarse registration.

Fine Registration of the PCD

Although many algorithms have been applied in the field of point cloud registration, they all show a strong initial value dependency and easily fall into local optimization [48]. In addition, their convergence speed is slow, among other issues. In this paper, an intelligent optimization algorithm that provides a new technical tool for this kind of nonlinear optimization problem is adopted to solve the issue of point cloud registration problem.

The Teaching-learning-based optimization (TLBO) algorithm is a classic intelligent optimization algorithm proposed by Rao in 2012 [50]. The TLBO algorithm has few parameters. Its basic principle is to simulate the communication process of teachers and students in the class. The calculation process of fine registration using TLBO is as follows:

Step 1. Input the PCD $P(p_1, p_2, \dots, p_n)$ after coarse registration and CAD model PCD $P'(p'_1, p'_2, \dots, p'_n)$. Then go to **Step 2**.

Step 2. Algorithm initialization. Initialize the class of students $X = (x_1, x_2, \dots, x_n)$, where n is the number of students. Student x_i in the class is defined as $x_i = (x_{i1}, x_{i2}, \dots, x_{ij})$, $j = 1, 2, \dots, D$, $i = 1, 2, \dots$ where x_{ij} represents the subject j learned by student i , D is the total subjects learned (6 in this paper), the upper and lower limits of each subject are U and L , the total number of iterations is T and x_{best} is the optimal solution. The objective function is shown in Equation (23). The students are randomly generated as shown in Equation (26). Then go to **Step 3**.

$$x_{ij} = L + rand \times (U - L), \quad (26)$$

where, $rand$ is the random number generation function.

Step 3. Teaching process. The fitness value is calculated for the students in the class using Equation (23). This process primarily simulates the teaching process of the class from the teacher to the

students. The teacher transmits knowledge to the students through teaching to continuously improve the students' performance as follows:

$$x'_{ij} = x_{ij} + rand \times (x_{best,j} - round(1 + rand)M_j), \quad (27)$$

$$M_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \quad (28)$$

where x'_{ij} is the updated score; x_{ij} is the current score; $rand$ is a random number between 0 and 1; $x_{best,j}$ is the result of the teacher's subject j ; M_j is the average score of subject j ; and $round$ is the round function. Then go to **Step 4**.

Step 4. Learning process. The learning process is a process in which students communicate with each other and update their scores. Based on the teacher's teaching, students learn from their classmates. The updated formula for the minimization problem after comparing the fitness values is shown in Equation (29).

$$x'_a = x_b + rand \times |x_a - x_b|, \quad (29)$$

where: x'_a is the updated academic performance; x_a is the current academic performance; x_b is student b 's current academic performance. Then go to **Step 5**.

Step 5. Terminate the conditional judgment. After each iteration, judge whether the total number of iterations T is reached. When the number of iterations reaches, the calculation is completed and the global optimal value and global optimal solution $(\alpha, \beta, \gamma, d_x, d_y, d_z)$ are output. When the number of iterations is less than T , return to **Step 3** and continue the iterations.

Algorithm 3 is the pseudocode of fine registration.

Algorithm 3 Pseudocode of fine registration

Begin

Input data $P(p_1, p_2, \dots, p_n)$ and $p' = (x', y', z')$

Initialize U, L, D, T, N

Generate initial solutions randomly

While $t < T$ **do**

For $i = 1$ to N

$x'_i = x_i + rand \times (x_{best} - round(1 + rand)M)$

If $f(x'_i) < f(x_i)$

$x_i = x'_i$

End if

$x_{a,b} = \text{random select}(\text{solutions})$

If $f(x_b) < f(x_a)$

$x'_a = x_b + rand \times |x_a - x_b|$

End if

End for

End while

Output $(\alpha, \beta, \gamma, d_x, d_y, d_z)$

For the fine registration process of the point cloud, this paper uses the measured point cloud data as a case for simulation experiments.

Figure 8 illustrates the calculation results of the vehicle body part point cloud under different registration algorithms. In Figure 8, a is the original point cloud data position of the vehicle part, b is the coarse registration result of FPFH, c is the fine registration result of ICP, d is the fine registration result of Harmony search (HS) [51], e is the fine registration result of Dragonfly algorithm (DA) [52] and f is the fine registration result of TLBO algorithm. Table 2 shows the parameters and error of the fine registration calculation results under different algorithms. The TLBO algorithm has the smallest

error which the value is 0.0132, so the registration effect is the best and the accuracy is the highest compared with the other three algorithms.

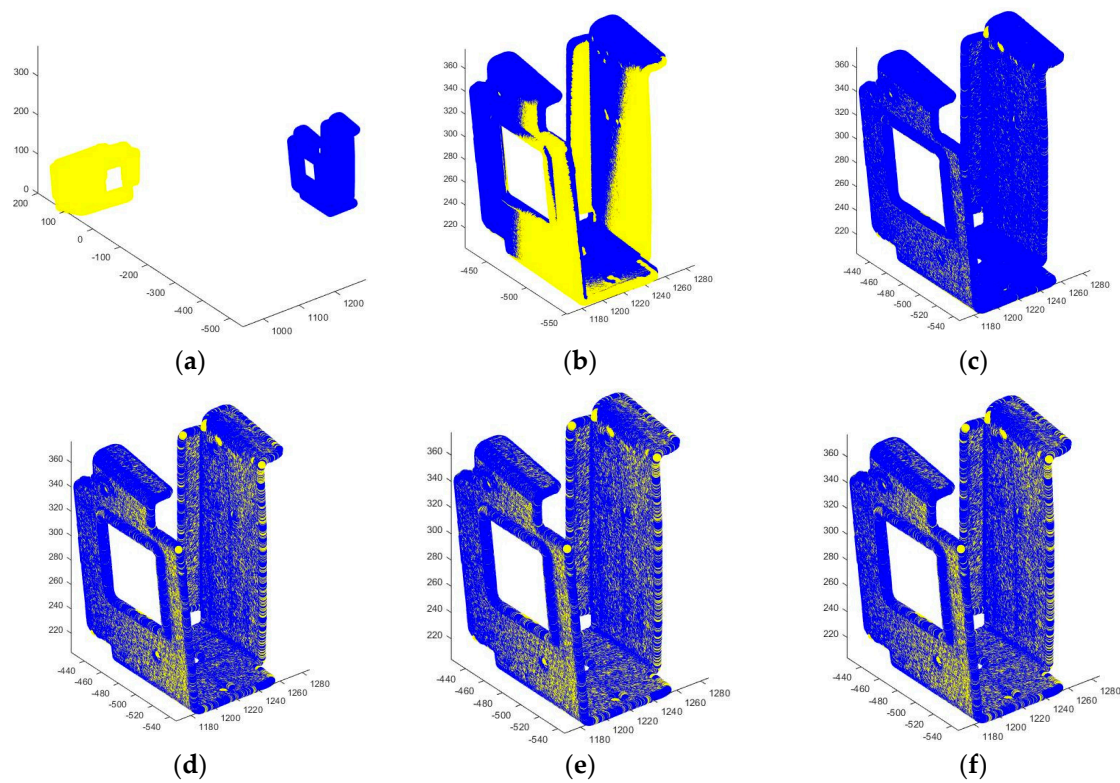


Figure 8. Point cloud fine registration experiment. (a) is the initial point cloud data position of the vehicle part, (b) is the coarse registration result of FPFH, (c) is the fine registration result of ICP, (d) is the fine registration result of Harmony search (HS), (e) is the fine registration result of Dragonfly algorithm (DA) and (f) is the fine registration result of TLBO algorithm.

Table 2. PCD after registration in Figure 3a.

	Part	α	β	γ	d_x	d_y	d_z	F
Fine registration	ICP	0.135	−1.393	6.823	−4.142	−1.486	−2.198	0.331
	HS	0.133	−1.341	6.790	−4.139	−1.450	−2.137	0.1132
	DA	0.132	−1.347	6.797	−4.140	−1.433	−2.139	0.0740
	TLBO	0.131	−1.332	6.792	−4.148	−1.455	−2.139	0.0132

4. Results and Discussion

To verify the method of the vehicle body part inspection scheme, four group experimental parts as shown in Section 3.1, were used to analyze the algorithms. Among them, the point cloud measurement data in Section 3.1 included 134,724 points (Figure 3a), 169,424 points (Figure 3b), 186,483 (Figure 3c), 326,535 (Figure 3d), 214,834 (Figure 3e), 437,752 (Figure 3f), 239,626 (Figure 3g) and 319,955 (Figure 3h). In addition to the PCD of the measured vehicle parts, this paper also includes CAD model data of the vehicle parts which are input in the form of point clouds. The reference point cloud data is obtained by converting CAD file format into STereoLithography (STL) model and then sampling from STL model by Meshlab. In this experiment, the computer's operating system was Windows 7 64 bit and the processor was an i5-4200U CPU @ 1.60GHz.

4.1. Denoising Experiment

To reduce the influence of noise points on the subsequent point cloud processing process, the point cloud denoising algorithm described in Section 3.2.2 is first used to denoise the initial point cloud data. The denoising results of the PCD are shown in Figure 9.

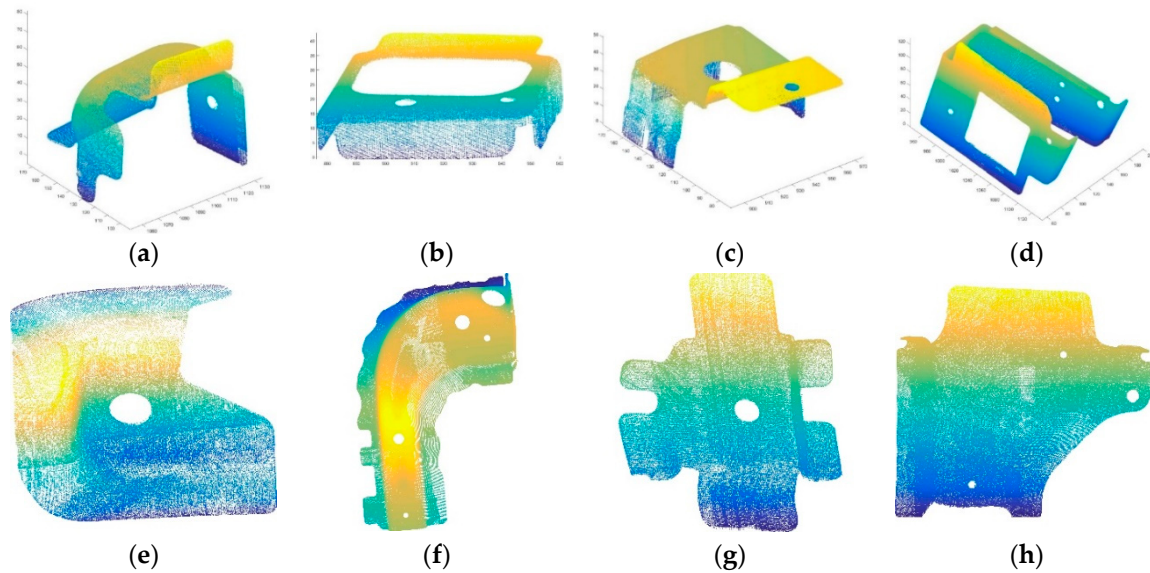


Figure 9. PCD after denoising. (a)–(h) are a series of denoising results of vehicle body parts PCD in Figure 3, respectively.

In Figure 9, the amount of PCD after denoising was 97,868 (Figure 9a), 75,456 (Figure 9b), 143,516 (Figure 9c), 267,308 (Figure 9d), 164,473 (Figure 9e), 410,265 (Figure 9f), 228,189 (Figure 9g) and 289,115 (Figure 9h). Therefore, the number of data removed by the point cloud denoising algorithm was 36,856 (Figure 9a), 25,908 (Figure 9b), 42,967 (Figure 9c), 59,227 (Figure 9d), 50,361 (Figure 9e), 27,487 (Figure 9f), 11,437 (Figure 9g), 30,840 (Figure 9h). The pass-through filtering parameters are defined according to different parts, while the statistical filtering parameters are $k = 10$, $\sigma = 1$. As can be seen in Figure 9, the point cloud noise data were largely removed through the denoising algorithm designed in this paper and the target PCD was thus accurately extracted, providing an accurate data foundation for the subsequent point cloud processing.

4.2. Simplification Experiment

The target PCD extracted after denoising retained a large amount of data and redundant point clouds, resulting in a heavy workload for subsequent analysis and low analytical efficiency. Therefore, the point cloud simplification algorithm designed in Section 3.2.3 was used to simplify the PCD. The simplified point cloud model is shown in Figure 10. For the point cloud simplification algorithm, the relevant calculation parameters are $k = 15$, $m = 2$, $C = 20$, $\sigma = 2$, $\varepsilon = 0.02$ and $T = 50$.

To reduce the influence of redundant points on the subsequent analysis of PCD, a feature preserving retention FCM algorithm was designed to simplify the PCD. Figure 10 illustrates the simplified point cloud model. The number of point clouds after simplification was 13,911 (Figure 10a), 9,794 (Figure 10b), 13,576 (Figure 10c), 30,502 (Figure 10d), 15,150 (Figure 10e), 41,367 (Figure 10f), 22,019 (Figure 10g), 28,528 (Figure 10h). The number of redundant points removed was 83,957, 65,662, 129,940, 236,806, 149,323, 368,898, 206,170 and 260,587. The simplified rate equation is as:

$$rate = \frac{N_{before} - N_{after}}{N_{before}} \times 100\%, \quad (30)$$

where, N_{before} is the number of point clouds before simplification and N_{after} is the number of point clouds after simplification.

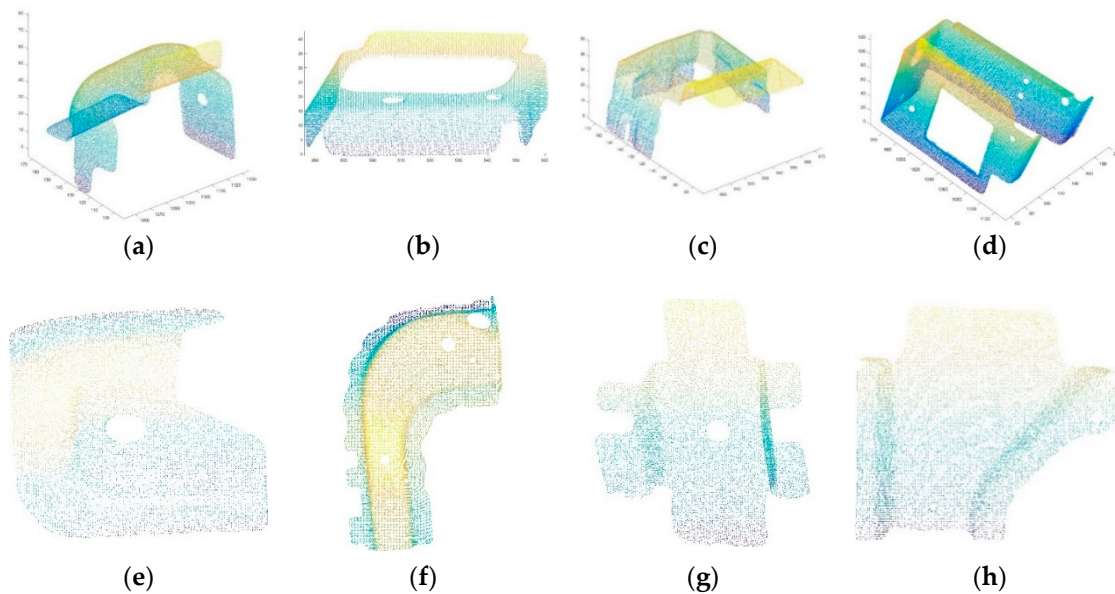


Figure 10. PCD after simplification in Figure 9. (a)–(h) are a series of simplification results of vehicle body parts PCD in Figure 9, respectively.

According to Equation (30), the point cloud simplification rate was 85% (Figure 10a), 87% (Figure 10b), 90% (Figure 10c), 88% (Figure 10d), 90% (Figure 10e), 90% (Figure 10f), 90% (Figure 10g) and 90% (Figure 10h). These simplification rates can maximize the removal of redundant data points while retaining the features of the point cloud model, providing a good data foundation for subsequent point cloud processing to reduce the workload and improve the computing efficiency.

4.3. Registration Experiment

After denoising and simplifying of the PCD, the PCD was registered with its CAD model. The registration process was first performed through coarse registration based on SAC-IA and placed in a good initial position. Then the fine registration process based on TLBO was used to complete the whole point cloud registration process. The parameters of the registration process were set as follows: $T = 500$, $U = 100$, $L = 100$, $N = 30$, $D = 6$. The registration process is shown in Figures 11–17 where the data (yellow) is the PCD of CAD model and the blue data comprise the simplified measurement PCD.

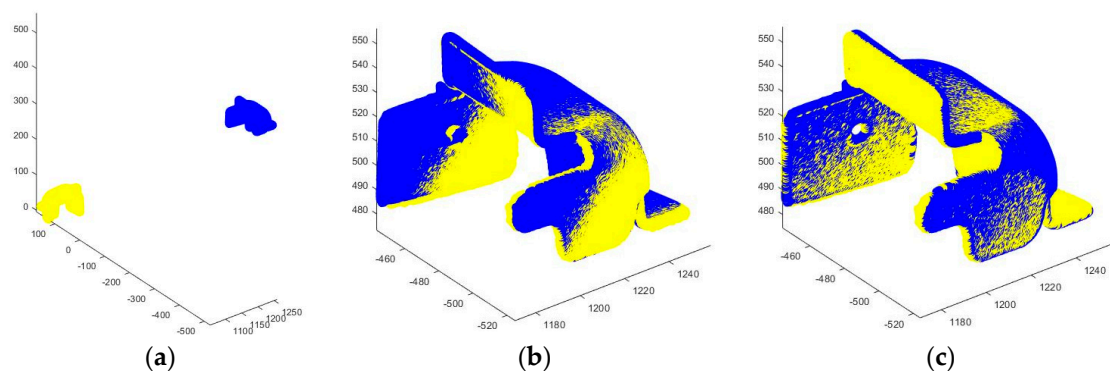


Figure 11. PCD after registration in Figure 10a. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

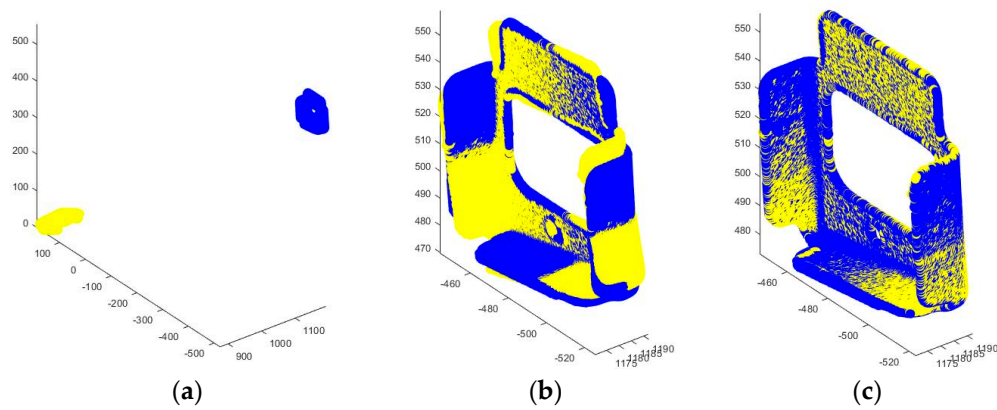


Figure 12. PCD after registration in Figure 10b. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

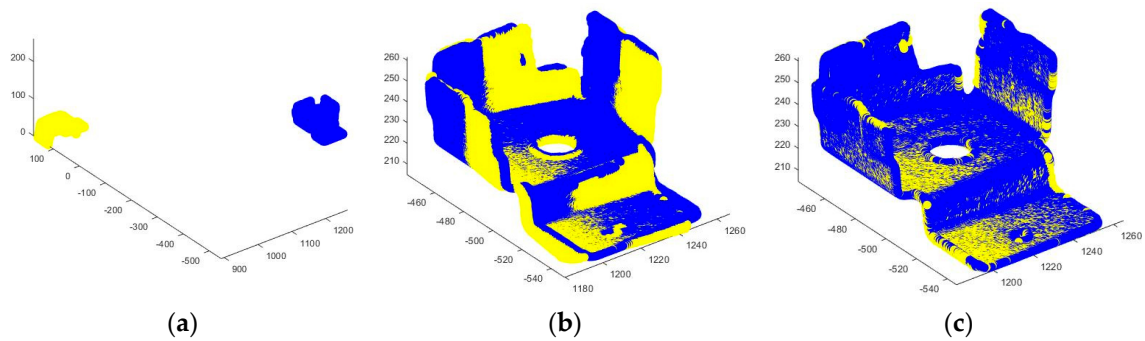


Figure 13. PCD after registration in Figure 10c. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

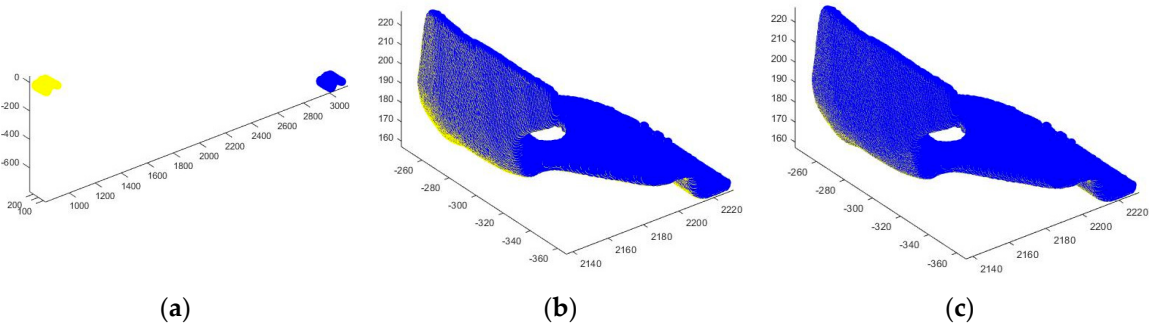


Figure 14. PCD after registration in Figure 10d. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

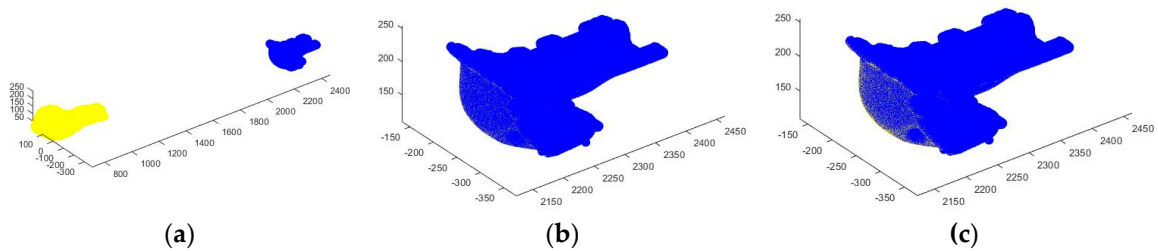


Figure 15. PCD after registration in Figure 10f. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

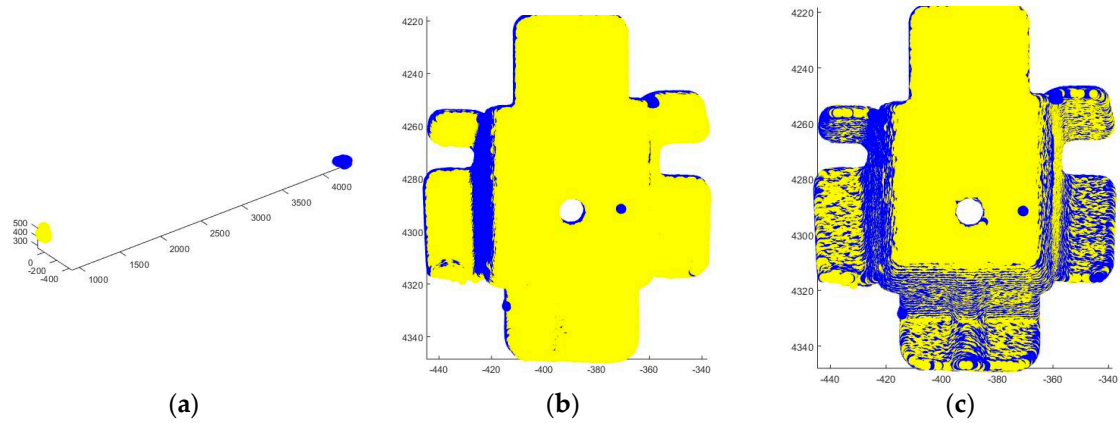


Figure 16. PCD after registration in Figure 10g. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

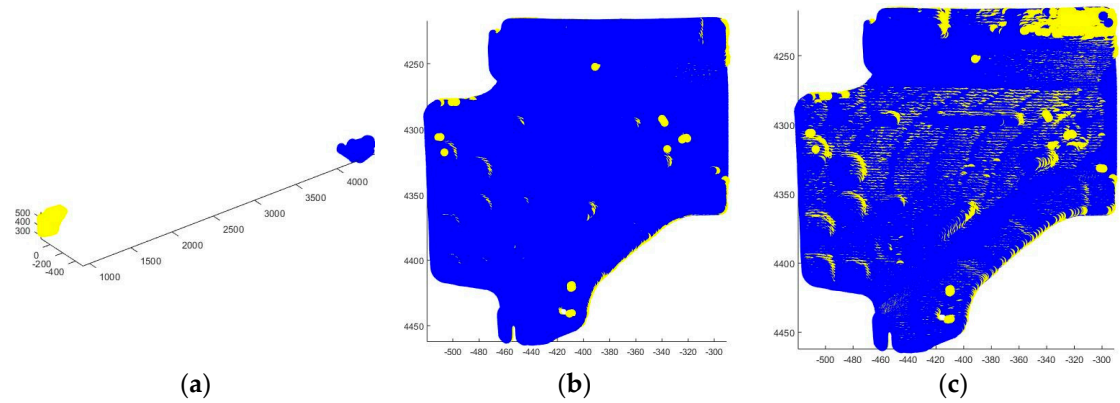


Figure 17. PCD after registration in Figure 10h. (a) is the initial position of the point cloud, (b) is the result of coarse registration, and (c) is the result of fine registration.

From Figures 11–17, (a) is the initial PCD location, (b) is the PCD location after coarse registration and (c) is the PCD location after fine registration. From the data fusion state of PCD, we can see that the fusion result of point cloud and CAD model is more uniform after fine registration, while the coarse registration result is more biased but it provides a good initial position for the calculation process of fine registration and avoids the process of fine registration from falling into the local optimal prematurely. Take the parts in Figure 11a as an example, the spatial transformation matrix from the coarse to fine registration process is shown as Equation (31). Table 3 is the registration parameters of PCD in this paper.

$$\begin{bmatrix} -0.014 & 0.006 & 1.000 & 1196.752 \\ 0.998 & 0.066 & 0.014 & -1590.586 \\ -0.066 & 0.998 & -0.007 & 460.298 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \rightarrow \begin{bmatrix} 1.000 & -0.011 & 0.001 & -6.414 \\ 0.011 & 1.000 & -0.009 & -9.384 \\ -0.001 & 0.009 & 1.000 & 6.170 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \quad (31)$$

Table 3. Registration parameters.

	Part	α	β	γ	d_x	d_y	d_z
Coarse registration	Figure 11	9.0387	3.793	90.823	1196.752	−1590.586	460.298
	Figure 12	−65.745	−20.340	−94.093	1166.474	426.493	314.908
	Figure 13	177.283	2.508	173.269	2132.986	−728.748	291.238
	Figure 14	138.172	−77.432	143.629	1299.793	−697.308	−732.713
	Figure 15	146.283	5.299	100.222	2432.986	−511.439	543.314
	Figure 16	155.333	−54.520	166.877	1115.545	−555.358	−753.234
	Figure 17	−88.344	−9.453	−13.342	2563.111	−657.546	300.455
Fine registration	Figure 11	0.500	0.079	0.685	−6.414	−9.381	6.173
	Figure 12	8.161	0.056	−0.165	1.157	7.854	7.963
	Figure 13	0.132	−1.342	6.782	−4.144	−1.451	−2.137
	Figure 14	1.397	−0.537	2.823	−2.060	−5.086	1.711
	Figure 15	3.445	0.043	−0.133	0.160	3.444	2.323
	Figure 16	0.111	−0.554	2.820	−1.155	−0.775	−1.444
	Figure 17	1.337	−0.666	2.823	−1.041	−2.168	0.911

4.4. Inspection Results

According to the above analysis process of PCD, the initial PCD were denoised, simplified and registered, respectively. The PCD after the pre-processing is further analyzed, mainly including the dimensional deviation of the part itself and the matching state analysis with other parts by KNN. The inspection results are shown in the Figures 18 and 19. Figure 18 shows the color deviation diagram of the inspection result. Figure 19 shows the histogram of the inspection results.

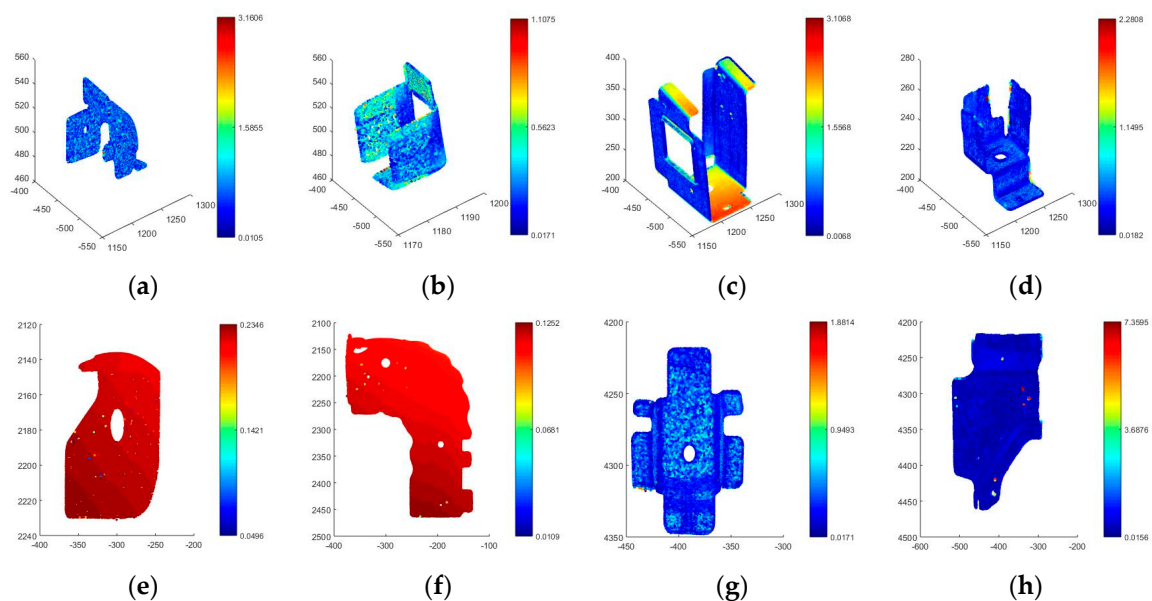


Figure 18. The color deviation diagram of inspection results in Figure 3. (a–h) are a series of inspection results color deviation diagram of vehicle body parts in Figure 3, respectively.

Figures 18 and 19 show the inspection results of four groups of vehicle body parts based on the denoised PCD and the spatial matrix registered, where Figure 18 is the dimensional deviation of the single part and Figure 19 the deviation histogram of vehicle parts. Take Figure 18a as an example, the minimum, maximum, mean and standard deviation of part deviation are 0.0105 mm, 3.1606 mm, 0.6576 mm and 0.3838 mm in Table 4, respectively. The inspection results of other vehicle parts are also shown in Figures 18 and 19 and Table 4.

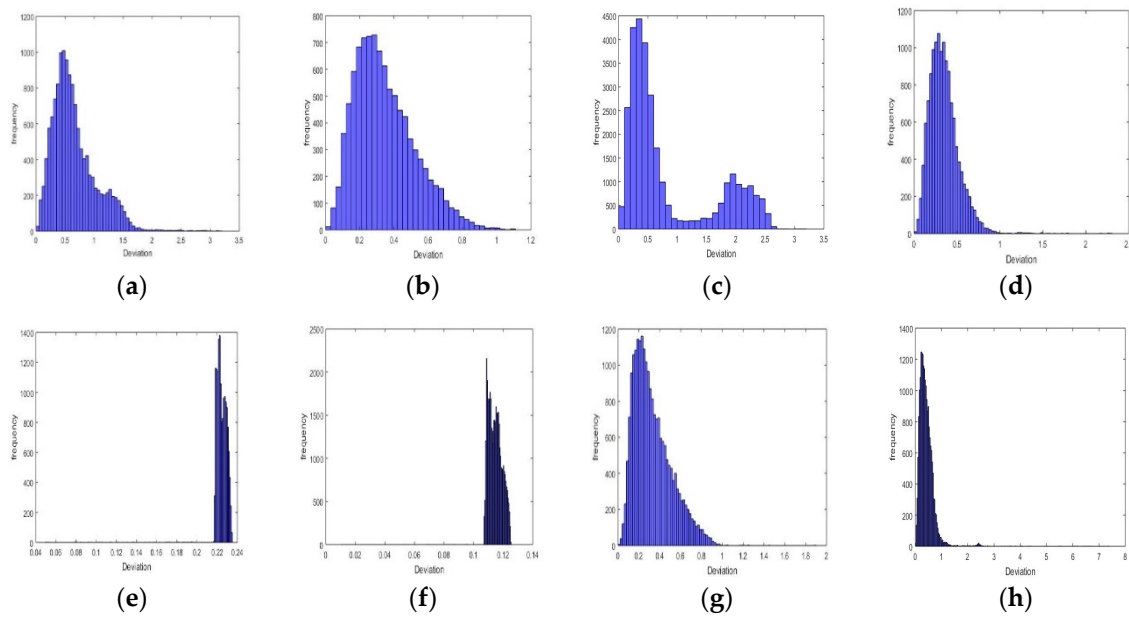


Figure 19. The histogram of inspection results in Figure 3. (a)–(h) are a series of inspection results histogram of vehicle body parts in Figure 3, respectively.

Table 4. Inspection results (mm).

Part	Min	Max	Mean	Std
Figure 18a	0.0105	3.1606	0.6576	0.3838
Figure 18b	0.0171	1.1075	0.3469	0.1761
Figure 18c	0.0068	3.1068	0.8512	0.7559
Figure 18d	0.0182	2.2808	0.3487	0.1715
Figure 18e	0.0496	0.2346	0.2243	0.0070
Figure 18f	0.0109	0.0125	0.1146	0.0046
Figure 18g	0.0171	1.8814	0.3311	0.1806
Figure 18h	0.0156	7.3595	0.4190	0.3115

From the point cloud deviation of the above parts shown in Figure 18a–h, although the maximum and minimum values are relatively large, their mean values and standard deviations are within the standard value range. This is due to the large amount of point cloud data, the processing process is very complex, there may be some individual noise points, affecting the results of the maximum and minimum values. Therefore, the mean value and standard deviation of statistical data have more practical guiding significance.

Therefore, in order to more objectively reflect the actual state of the measured data, the deviation histogram is shown in Figure 19a–h. It can be seen from the figure that, except for Figure 19c, the deviation distribution of the other point cloud is relatively uniform, indicating that the algorithm designed has a good computing effect and can accurately complete the point cloud pre-processing process and carry out the actual state analysis of the point cloud. As for the deviation histogram of vehicle body part shown in Figure 18c, some points have large deviation (1.5 mm~2.5 mm), so it is necessary to further inspect and analyze whether the dimension of parts meets the design requirements.

Finally, we compare this algorithm framework with other algorithms. Considering that any link in the inspection process will have an impact on the calculation results, in this test, TLBO algorithm in the process of fine registration is replaced by the HS algorithm, while the rest of the calculation process remains unchanged. The vehicle part in Figure 2h is taken as the research object. The deviation histogram of the two algorithm framework is shown in Figure 20.

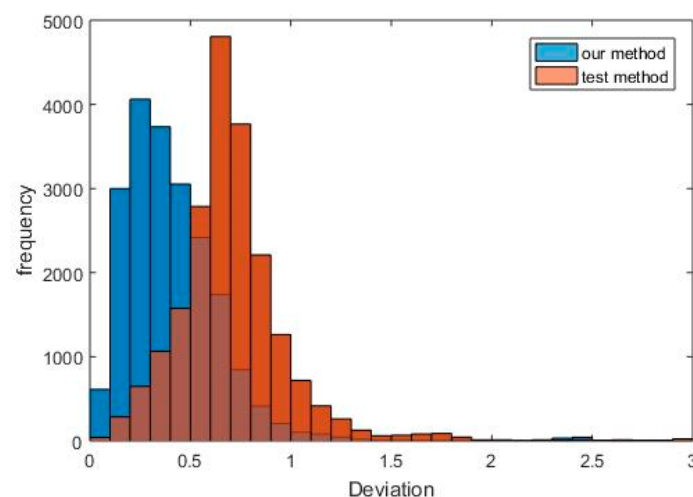


Figure 20. The inspection results in Figure 2h.

The histogram of point cloud deviation can be seen in Figure 20. The deviation distribution of point cloud tends to be larger under the HS algorithm. This reflects that, although the deviation of parts has a manufacturing error, the algorithm selected also greatly affects the evaluation results of parts in the inspection process. Therefore, the vehicle part inspection workflow in this paper provides a reliable method for vehicle body quality inspection.

5. Conclusions and Future Work

Under the background of “Industry 4.0,” it was found that the whole vehicle factory interfered in the measurement and matching of vehicle parts, resulting in the reworking of parts and assembly failure, which greatly increased the workload of engineers and technicians and reduced production efficiency. Therefore, using optical non-contact measurements and computer science technology, combined with an analysis of the research status of virtual matching technology at home and abroad, we proposed an inspection technology for vehicle body parts based on point cloud data. Optical measurement equipment is used as a tool based on the measured point cloud data of the acquired parts. Firstly, the point cloud noise data were removed by the hybrid filtering algorithm and the target point cloud data were extracted. Secondly, to reduce the subsequent calculation workload and improve the efficiency of the analysis, a fuzzy clustering simplification algorithm based on feature retention of the PCD was designed. Thirdly, in combination with the CAD model data of the parts, the SAC-IA and TLBO algorithms for point cloud data were used from the coarse registration to the registration process to complete the complete data alignment operation. In the final analysis after pretreating the actual state PCD for the parts, this paper offers an effective technical proposal for an examination process of automobile body parts.

In future work, we will continue to carry out quantitatively research the matching of vehicle parts, especially when the interference is found in the parts, the interference value is obtained in advance through the related algorithm design to pre-adjust the parts and reduce the assembly error rate. In addition, the influence of point cloud noise should be further reduced during the point cloud pretreatment process, so future work is also needed in this area.

Author Contributions: Algorithm design and Writing—original draft, Y.Y.; Writing—review and editing, M.L.; Funding acquisition—X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Natural Science Foundation of Ningbo, China (Grant No. 2016A610039) and Major Research Plan of Ningbo, China (Grant No. 201601ZDA01097).

Acknowledgments: The author thanks the editors and anonymous reviewers for their constructive comments on the manuscript and for the basic research of relevant scholars in the references.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

$$ax + by + cz = \quad (A1)$$

$$C = \begin{bmatrix} \sum_{i=1}^n (x_i - \bar{x})^2 & \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^n (x_i - \bar{x})(z_i - \bar{z}) \\ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^n (y_i - \bar{y})^2 & \sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z}) \\ \sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z}) & \sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z}) & \sum_{i=1}^n (z_i - \bar{z})^2 \end{bmatrix}, \quad (A2)$$

where (a, b, c) is the normal vector of selected point.

$$E(x) = \sum_{i=1}^n ((x - \varepsilon_i)^T n(x))^2 G(x, \varepsilon_i) \quad (A3)$$

$$n(x) = \frac{\sum_{i=1}^n n(\varepsilon_i) G(x, \varepsilon_i)}{\| \sum_{i=1}^n n(\varepsilon_i) G(x, \varepsilon_i) \|}, \quad (A4)$$

where G is Gaussian function; x is projection point; ε_i is the neighborhood of x .

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A5)$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (A6)$$

When p is rotated in space, its motion is as follows:

$$R = \begin{bmatrix} \cos \beta \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\cos \beta \sin \gamma + \sin \alpha \sin \beta \sin \gamma & \sin \beta \cos \alpha & 0 \\ \cos \alpha \sin \gamma & \cos \alpha \cos \gamma & -\sin \alpha & 0 \\ -\sin \beta \cos \gamma + \sin \alpha \sin \beta \sin \gamma & \sin \beta \sin \gamma + \sin \alpha \cos \beta \cos \gamma & \cos \alpha \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A7)$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = R \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (A8)$$

Therefore, when p moves arbitrarily in space, its motion process can be written as:

$$p' = R \cdot p + T, \quad (A9)$$

where T is the translation matrix in space; R is the rotation matrix; d_x , d_y and d_z are the translation distances in the directions of X , Y and Z , respectively; and α , β and γ are the rotation angles on the X , Y and Z axes, respectively.

References

1. Shu, Z. The Industry 4.0 and Intelligent Manufacturing. *Mach. Des. Manuf. Eng.* **2013**, *43*, 1–5.
2. Guo, J. Study on Production Process Quality Control and Evaluation in Vehicle Manufacturing Enterprise. Ph.D. Thesis, Wuhan University of Technology, Wuhan, China, 2012.
3. Liu, T. Research on Key Technologies for Automated Measurement of Automotive Complex Parts. Ph.D. Thesis, Tianjin University, Tianjin, China, 2018.
4. Tran, T.T.; Ha, C.K. Non-contact Gap and Flush Measurement Using Monocular Structured Multi-line Light Vision for Vehicle Assembly. *Int. J. Control Autom. Syst.* **2018**, *16*, 2432–2445. [[CrossRef](#)]
5. Fleishman, S.; Drori, I.; Cohen-Or, D. Bilateral mesh denoising. *ACM Trans. Graph.* **2003**, *22*, 950–953. [[CrossRef](#)]
6. Fleishman, S.; Cohen-Or, D.; Silva, C. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.* **2005**, *24*, 544–552. [[CrossRef](#)]
7. Gu, X.Y.; Liu, Y.S.; Wu, Q. Research on a Denoising Smoothing Algorithm for 3d Scattered Point Cloud. *ICIC Express Lett.* **2014**, *8*, 2403–2409.
8. Hermosilla, P.; Ritschel, T.; Ropinski, T. Total Denoising: Unsupervised Learning of 3D Point Cloud Cleaning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October 2019; pp. 52–60.
9. Liu, Z.; Xiao, X.W.; Zhong, S.S.; Wang, W.N.; Li, Y.L.; Zhang, L.; Xie, Z. A feature-preserving framework for point cloud denoising. *Comput. Aided Des.* **2020**, *127*, 102857. [[CrossRef](#)]
10. Zhou, S.T.; Liu, X.L.; Wang, C.Y.; Yang, B. Non-iterative denoising algorithm based on a dual threshold for a 3D point cloud. *Opt. Laser. Eng.* **2020**, *126*, 105921. [[CrossRef](#)]
11. Chen, Y.; Ng, C.; Wang, Y. Data reduction in integrated reverse engineering and rapid prototyping. *Int. J. Comput. Integr. Manuf.* **1999**, *12*, 97–103. [[CrossRef](#)]
12. Lee, K.H.; Woo, H.; Suk, T. Point Data Reduction Using 3D Grids. *Int. J. Adv. Manuf. Technol.* **2001**, *18*, 201–210. [[CrossRef](#)]
13. Dyn, N.; Floater, M.S.; Iske, A. Adaptive thinning for bivariate scattered data. *J. Comput. Appl. Math.* **2002**, *145*, 505–517. [[CrossRef](#)]
14. Han, H.; Han, X.; Sun, F.; Huang, C. Point cloud simplification with preserved edge based on normal vector. *Opt. Int. J. Light Electron Opt.* **2015**, *126*, 2157–2162. [[CrossRef](#)]
15. Qing, S.; Tao, X.; Tatsuo, Y.; Yujie, Z.; Wenting, Y.; Hang, Z. Point cloud simplification algorithm based on particle swarm optimization for online measurement of stored bulk grain. *Int. J. Agric. Biol. Eng.* **2016**, *9*, 71–78.
16. Senin, N.; Colosimo, B.M.; Pacella, M. Point set augmentation through fitting for enhanced ICP registration of point clouds in multisensor coordinate metrology. *Robot. Comput. Integr. Manuf.* **2013**, *29*, 39–52. [[CrossRef](#)]
17. Huang, J.; Wang, Z.; Gao, J.; Huang, Y.; Towers, D.P. High-Precision Registration of Point Clouds Based on Sphere Feature Constraints. *Sensors* **2016**, *17*, 72. [[CrossRef](#)] [[PubMed](#)]
18. Du, Q. 3D point cloud registration denoising method for human motion image using deep learning algorithm. *Multimed. Syst.* **2019**, *26*, 75–82. [[CrossRef](#)]
19. Li, J.; Zhao, P.; Hu, Q.; Ai, M. Robust point cloud registration based on topological graph and Cauchy weighted l_q -norm. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 244–259. [[CrossRef](#)]
20. Wu, L.S.; Wang, G.L.; Hu, Y. Iterative closest point registration for fast point feature histogram features of a volume density optimization algorithm. *Meas. Control* **2020**, *53*, 29–39. [[CrossRef](#)]
21. Ramaswamy, S.; Yan, Y. Interactive modeling and simulation of virtual manufacturing assemblies: An agent-based approach. *J. Intell. Manuf.* **1999**, *10*, 503–518. [[CrossRef](#)]
22. Tching, L.; Dumont, G.; Perret, J. Interactive simulation of CAD models assemblies using virtual constraint guidance. *Int. J. Interact. Des. Manuf.* **2010**, *4*, 95–102. [[CrossRef](#)]
23. Liang, M.Y.; Song, Y.; Wang, Y.; Peng, X.D.; Nie, D.H. Assembly modeling technology for satellite virtual assembly. In Proceedings of the 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD), Wellington, New Zealand, 26 April 2017; pp. 562–566.

24. Jiang, S.Q.; Liu, P.; Gao, D.W.; Xu, Y.; Meng, X.; Liu, Z.Y.; Huang, Z.; Xu, R.L. Research on low cost virtual assembly training platform based on somatosensory technology. In Proceedings of the IEEE International Conference on Industrial Engineering & Engineering Management (IEEM), Singapore, 10 December 2017; pp. 250–254.
25. Chen, Z.; Li, L. A new pose estimation method for non-cooperative spacecraft based on point cloud. *Int. J. Intell. Comput. Cybern.* **2019**, *12*.
26. Wen, L.; He, L.; Gao, Z. Research on 3D Point Cloud De-distortion Algorithm and Its Application on Euclidean Clustering. *IEEE Access* **2019**, *7*, 86041–86053. [\[CrossRef\]](#)
27. Li, T.F. Research on Complex Free-form Surface Parts Quality Inspection with 3D Registration Method. Ph.D. Thesis, Huazhong University of Science and Technology, Wuhan, China, 2017.
28. Bao, J.S.; Li, Z.Q.; Xiang, Q.; Wu, D.L. The Modeling, Evolutionary and Application of Quasi-physical virtual assembly. *J. Mech. Eng.* **2018**, *54*, 61–69. [\[CrossRef\]](#)
29. Zhu, X.H.; Liu, Y.M.; Liu, J.L. Application of reverse engineering to assembly and correction based on non-contact measurement. *Min. Process. Eq.* **2019**, *46*, 46–49.
30. Yu, A.X. Object Scanning Mode Based Composite Product Repair Scheme Design. Master's Thesis, Zhejiang University, Hangzhou, China, 2019.
31. Zhao, Y.; Zhang, L.; Xue, Q.; Zhang, Y.T. Application of virtual technology in vehicle dimension matching. *Auto Eng.* **2015**, 48–51. [\[CrossRef\]](#)
32. Sun, Z.L. Research on Filtering Method of Three-Dimensional Laser Scanning Point Cloud Data. Master's Thesis, Central South University, Changsha, China, 2011.
33. Ehrlich, C. Terminological aspects of the Guide to the Expression of Uncertainty in Measurement (GUM). *Metrologia* **2014**, *51*, S145. [\[CrossRef\]](#)
34. He, D.J.; Shao, X.N.; Wang, D.; Hu, S.J. Denoising Method of 3-D Point Cloud Data of Plants Obtained by Kinect. *Trans. Chin. Soc. Agr. Machi.* **2016**, *47*, 331–336. [\[CrossRef\]](#)
35. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517. [\[CrossRef\]](#)
36. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.
37. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Dolha, M.; Beetz, M. Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.* **2008**, *56*, 927–941. [\[CrossRef\]](#)
38. Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. In Proceedings of the 19th Annual Conference and Exhibition on Computer Graphics and Interactive Techniques (CGIT), New York, NY, USA, 27–31 July 1992; pp. 71–78.
39. Lancaster, P.; Salkauskas, K. Surfaces generated by moving least squares methods. *Math. Comput.* **1981**, *37*, 141–158. [\[CrossRef\]](#)
40. Hartigan, J.A.; Wong, M.A. A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C* **1979**, *28*, 100–108.
41. Bezdek, J.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [\[CrossRef\]](#)
42. Gill, B.; Sariel, H. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms* **2001**, *38*, 91–109.
43. Pauly, M.; Gross, M.; Kobbelt, L.P. Efficient simplification of point-sampled surface. In Proceedings of the IEEE Visualization (VIS 2002), Boston, MA, USA, 27 October–1 November 2002.
44. Du, X.; Zhuo, Y. A Point Cloud Data Reduction Method Based on Curvature. In Proceedings of the IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design, Wenzhou, China, 26–29 November 2009.
45. Gurram, P.; Hu, S.; Chan, A. Uniform grid upsampling of 3D lidar point cloud data. In Proceedings of the Three-Dimensional Image Processing (3DIP) and Applications 2013, Burlingame, CA, USA, 12 March 2013.
46. Schnabel, R.; Klein, R. Octree-based Point-Cloud Compression. *SPBG* **2006**, *6*, 111–120.
47. Wu, J.J. Research of Point-based Techniques on Unorganized Point Cloud. Ph.D. Thesis, Huazhong University of Science and Technology, Wuhan, China, 2004.

48. Besl, P.J.; McKay, H.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [[CrossRef](#)]
49. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. 2009 IEEE International Conference on Robotics and Automation. In Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Kobe, Japan, 17 June 2009; pp. 3212–3217.
50. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided. Des.* **2011**, *43*, 303–315. [[CrossRef](#)]
51. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony Search. *Simulation* **2001**, *2*, 60–68. [[CrossRef](#)]
52. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective discrete, and multi-objective problems. *Neural. Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).