




## Article

# A Hybrid Improved SAC-IA with a KD-ICP Algorithm for Local Point Cloud Alignment Optimization

Yinbao Cheng <sup>1</sup>, Haiman Chu <sup>1</sup>, Yaru Li <sup>1,\*</sup>, Yingqi Tang <sup>2</sup>, Zai Luo <sup>1</sup> and Shaohui Li <sup>3,\*</sup>

<sup>1</sup> College of Metrology Measurement and Instrument, China Jiliang University, Hangzhou 310018, China; cyb@cjlu.edu.cn (Y.C.); s23020804013@cjlu.edu.cn (H.C.); luozai@cjlu.edu.cn (Z.L.)

<sup>2</sup> School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China; 7520230008@bit.edu.cn

<sup>3</sup> National Industrial Measurement and Testing Centre for Water Transport Monitoring Equipment, Tianjin Research Institute for Water Transport Engineering, Tianjin 300400, China

\* Correspondence: liyaru@cjlu.edu.cn (Y.L.); lishh@tiwte.ac.cn (S.L.)

**Abstract:** To overcome incomplete point cloud data obtained from laser scanners scanning complex surfaces, multi-viewpoint cloud data needs to be aligned for use. A hybrid improved SAC-IA with a KD-ICP algorithm is proposed for local point cloud alignment optimization. The scanned point cloud data is preprocessed with statistical filtering, as well as uniform down-sampling. The sampling consistency initial alignment (SAC-IA) algorithm is improved by introducing a dissimilarity vector for point cloud initial alignment. In addition, the iterative closest point (ICP) algorithm is improved by incorporating bidirectional KD-tree to form the KD-ICP algorithm for fine point cloud alignment. Finally, the algorithms are compared in terms of runtime and alignment accuracy. The implementation of the algorithms is based on the Visual Studio 2013 software configuring point cloud library environment for testing experiments and practical experiments. The overall alignment method can be 40%–50% faster in terms of running speed. The improved SAC-IA algorithm provides better transformed poses, combined with the KD-ICP algorithm to select the corresponding nearest neighbor pairs, which improves the accuracy, as well as the applicability of the alignment.

**Keywords:** point cloud alignment; SAC-IA; KD-ICP; laser scanner



**Citation:** Cheng, Y.; Chu, H.; Li, Y.; Tang, Y.; Luo, Z.; Li, S. A Hybrid Improved SAC-IA with a KD-ICP Algorithm for Local Point Cloud Alignment Optimization. *Photonics* **2024**, *11*, 635. <https://doi.org/10.3390/photonics11070635>

Received: 27 May 2024

Revised: 26 June 2024

Accepted: 1 July 2024

Published: 2 July 2024



**Copyright:** © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The 3D laser scanner uses a non-contact measurement method to quickly obtain dimensional information of the measured workpiece and store it as a 3D spatial point cloud model. The laser scanner is capable of reproducing one-to-one 3D data of the measured workpiece and construct 3D visualization models. For measured workpieces with complex surfaces and large dimensions, the laser scanner is only able to obtain localized information. It is necessary to scan from multiple viewpoints to obtain a complete 3D data model of the measured workpiece for quality inspection in industrial production processes or for robot attitude estimation. The point cloud data obtained by scanning from different viewpoints cannot be used directly because they have different coordinate systems. So, the coordinate conversion is realized by aligning the local data obtained from any viewpoint scanning with the ideal CAD model. Therefore, the alignment of the local point cloud data from the laser scanner to the ideal CAD model is an integral part.

Scholars at home and abroad have proposed various methods for point cloud local feature extraction or the improvement of point cloud alignment algorithm. Wei S et al. [1] improved the random sample consistency algorithm for attitude estimation of a target ship to extract the plate side planes of a moving ship. Mao Y et al. [2] extracted step feature lines of open pit mining clouds through tracking step feature control nodes using seed growth tracking algorithm. Wang X et al. [3] dealt with scattered point clouds based on a clustering algorithm and generated feature curves through calculating local feature weights based on the mean curvature of the point cloud. This method can effectively

extract edge and sharpness features for different density point clouds. Zhao T et al. [4] performed fragment matching for low overlap point clouds and fused fragments and points at the feature and result layers. Shi J et al. [5] coalesced global features to iteratively interact with local geometric features of the point cloud in order to obtain matching features with high discriminative ability. Li P et al. [6] addressed the problem of spatial attitude detection of pipeline welding through aligning the measured point cloud with the CAD model with the help of a point cloud alignment method to guide pipeline adjustment. Scholars have different ways of extracting local features for different experimental subjects, and the alignment methods are more complex and time-consuming. In the case where alignment accuracy requirements are consistent, industrial production is more concerned with alignment speed. For example, when point cloud alignment is used for industrial production robot gripping and localization, the shorter the time spent on alignment, the faster the robot arm responds and the more efficient the gripping is. Therefore, the main improvement is proposed in the speed of operation of the algorithm, using more accessible local point clouds as the object of study for experiments to obtain quality-assured and more efficient point cloud alignment methods.

Considering the diversity of sizes and shapes of actual production workpieces, the coordinates of the point cloud data output by the laser scanner are based on its own coordinate system. Therefore, it is necessary to use a scanner for multi-view scanning and alignment. Aiming at the above problems, an improved method is proposed for local feature point cloud alignment. Point cloud alignment is divided into two main steps, initial and fine alignment. Currently, the commonly used initial alignment algorithms are SAC-IA and random sample consensus initial alignment (RANSAC). Where the RANSAC algorithm is highly randomized, or even worse, when the number of iterations is limited, it may not obtain the optimal solution. In contrast, the SAC-IA algorithm is more pervasive, but it is slower to compute fast point feature histograms (FPFH) features when the number of points is high, making it inefficient. Thus, the SAC-IA algorithm was chosen for the initial alignment of the point cloud and the introduction of dissimilarity vectors improves the efficiency. The improved SAC-IA algorithm, with the principle of geometric consistency between the scanned and the target point cloud, is able to exclude the hypotheses with large differences in advance, thus speeding up the initial alignment of the local point cloud. The iterative closest point (ICP) is the most classical algorithm for fine point cloud alignment. However, the algorithm is quite dependent on the initial position of the point cloud and a one-to-many situation occurs when finding neighboring pairs of points, which causes the algorithm to run into local optimum results. As a result, fusing a bidirectional KD-tree not only accelerates the search and identification of nearest points, but also identifies one-to-one pairs of nearest points to be used for iteration, which improves the algorithm alignment speed and accuracy.

This paper is based on the point cloud library 1.8.0 environment of Visual Studio 2013 software for algorithm testing. The classical sample and the laser scanner scanned workpiece model were used as experimental objects for point cloud alignment respectively. Finally, the root mean square error (RMSE) is used as the evaluation index of point cloud alignment accuracy, comparing the runtime and alignment accuracy to analyze the reliability and feasibility of the alignment methodology.

## 2. Point Cloud Alignment

### 2.1. Mathematical Model

The coordinate information of the point cloud data output by the laser scanner is based on its own scanning coordinate system [7,8]. Therefore, the point cloud data obtained from scanning the same workpiece from different viewpoints cannot be used directly. It is necessary to convert the results from different viewpoints to the same coordinate system based on the feature points. In the production of workpiece parts, there must be a standard CAD model. Using the ideal model of the workpiece enables faster conversion of point cloud coordinates for different viewpoints (i.e., local point cloud alignment).

The essence of point cloud alignment is a coordinate system transformation. In the 3D point cloud dataset, take any point in the set and write it as  $P_i(x, y, z)$ , and denote the coordinate transformed points as  $P'_i(x', y', z')$ . Noting the rotation matrix  $R$ , translation matrix  $T$ , and transformation matrix  $H$ , where  $R$  is a  $3 \times 3$  matrix,  $T$  is a  $3 \times 1$  matrix, and  $H$  is a  $4 \times 4$  matrix. The system of equations for 3D spatial coordinate transformation is as follows [9]:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R & 0 \\ V & U \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} = H \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, H = \begin{bmatrix} R & T \\ V & U \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where  $t_x$  denotes the offset of point  $P$  on the X-axis ( $t_y, t_z$  in the same vein);  $V$  denotes the proportion of its projection transformation in each coordinate direction;  $U$  refers to the scale factor of the overall point cloud. The point cloud data generated by laser scanning does not produce singular deformation, so  $V$  is a zero vector,  $U = 1$ ;  $\alpha$  denotes the angle of rotation of the point clockwise around the X-axis ( $\beta, \theta$  in the same vein).

There are two main steps in the realization of local point cloud alignment. On the one hand, the local point cloud features are identified to find the corresponding features in the ideal model for initial alignment. On the other hand, searching for the neighboring point pairs of the ideal and scanned point clouds continuously based on the preliminary transformed positional state. Then, computing transformation matrices for coordinate approximation to improve point cloud alignment accuracy. For the implementation algorithms of point cloud alignment, what needs to be concerned is its alignment accuracy and speed.

### 2.2. Improved SAC-IA Point Cloud Initial Alignment Algorithm

The SAC-IA [10,11] algorithm is able to perform the initial alignment of the point cloud by identifying the point cloud features, which provides a good initial value of coordinate transformation for fine alignment. However, when facing point cloud data with a large data volume and a large search radius [12], the running time of the SAC-IA algorithm grows. It becomes more difficult to reach the optimal transformation within the maximum number of iterations. Therefore, it is worthwhile to continuously optimize the speed of the initial alignment. Meanwhile, it is important to ensure that the algorithm is able to achieve the required alignment accuracy provided by the fine alignment initial position.

Based on the SAC-IA algorithm, the introduction of dissimilarity vectors  $\delta$  excludes the hypothesis of large differences in advance. The scanned point cloud is geometrically consistent with the target point cloud. In other words, the Euclidean distance between points of the overlapping parts is constant in their respective spaces. By using this principle, the corresponding point pairs of the selected features are denoted as  $p_i$  and  $q_i$ . They form virtual polygons with neighboring points in their respective spaces. There are  $n$  sides to this polygon, calculate the length of the  $j$ -th side of each of the two polygons:  $d_{p,j} = \|p_{i+j} - p_i\|$ ,  $d_{q,j} = \|q_{i+j} - q_i\|$ ,  $j = 1, 2, \dots, n$ . After that, the two polygon side lengths of the pairs of points are sorted in descending order and the corresponding difference is obtained. Then, we compare with the maximum difference to obtain the relative dissimilarity vectors  $\delta$ . The equation is as follows:

$$\delta = \frac{|d_{p,1} - d_{q,1}| \cdot |d_{p,2} - d_{q,2}| \cdots |d_{p,n} - d_{q,n}|}{\max(|d_{p,j} - d_{q,j}|)} \quad (2)$$

When the two polygons of the feature-corresponding point pairs can match exactly, the relative dissimilarity vector  $\delta = 0$ . Meanwhile, the SAC-IA algorithm we use is based on fast point feature histograms (FPFH) [13–16]. It describes the geometric attributes of points in the form of multidimensional histograms with the help of spatial differences between

points in the point cloud and their neighboring points, thus reflecting the point cloud data characteristics. The specific steps of the improved SAC-IA algorithm are as follows:

**Step 1:** Sample  $x$  data sample points randomly in the scanned point cloud  $Q$  and ensure that the minimum distance between two by two of the sample points is greater than the set threshold distance  $\omega_d$ .

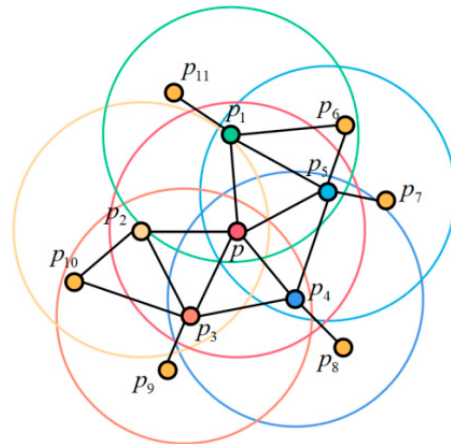
**Step 2:** Compute the feature descriptor FPFH for the scanned point cloud  $Q$  and the ideal point cloud  $S$ , respectively. First, for any point  $p$  in the point cloud, set the normal vector estimation radius  $d_f$ , with a local neighborhood  $\{p_1, p_2, \dots, p_k\}$ . The eigenvector corresponding to the smallest eigenvalue of the covariance matrix is taken as the normal vector of the point  $p$ .

**Step 3:** With point  $p$  as the center of the circle, construct a spherical neighborhood of point  $p$  through setting the neighborhood radius  $d_l$ . Construct a local coordinate system with the normal vectors of two points in the neighborhood. Then, calculate the horizontal angle, the pitch angle, and the angle of the connecting line of the two points as the point features (PFH).

**Step 4:** In order to reduce the computational complexity, calculating the PFH value of a point  $p$  with the points in its neighborhood is called SPFH. Then, the SPFH values of the neighboring points of point  $p$  are computed separately for weighting to obtain the final FPFH value. The sphere of influence is shown in Figure 1. The weighting equation is as follows:

$$FPFH(p) = SPFH(p) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_i} SPFH(p_i) \tag{3}$$

where  $\omega_i$  is the Euclidean distance from point  $p$  to a neighboring point  $p_i$ .



**Figure 1.** Scope of influence.

**Step 5:** FPFH visualization. Preliminary observations of the characterizers of a scanned point cloud  $Q$  and an ideal point cloud  $S$ . If the feature extraction is poor, you can choose to return to **Step 2** to change  $d_f$  or to **Step 3** to change  $d_l$  and search again. However, it should be noted that the search radius should satisfy  $d_l > d_f$ .

**Step 6:** Finding point cloud data with FPFH features similar to the scanned point cloud by KD-tree nearest neighbor search in an ideal point cloud  $S$ . And randomly select a point from the similar points as a candidate counterpart for this feature.

**Step 7:** Calculate the relative dissimilarity vectors  $\delta$  of feature counterparts with respect to polygon edge lengths. Set the deviation threshold  $t = 0.25$ , i.e., allow 25% edge variation. If  $\|\delta\| \geq t$ , then return to **Step 6** for resampling.

**Step 8:** Calculate the transformation parameters between pairs of points. For each pair of points in **Step 7** that passes the dissimilar vectors test, a transformation relation is determined, yielding a translation matrix as well as a rotation matrix. Moreover, use

the Huber equation (see Equation (4)) to determine whether the transformation matrix is optimal or not.

$$H(a) = \begin{cases} \frac{1}{2}a^2, & \|a\| \leq \omega_d \\ \omega_d(\|a\| - \frac{1}{2}\omega_d), & \|a\| > \omega_d \end{cases} \quad (4)$$

where  $a$  denotes the difference between the distances of the sample data point pairs after performing the transformation;  $\omega_d$  indicates the set threshold distance in **Step 1**.

**Step 9:** Return to **Step 1** for iteration, set the maximum number of iterations, and output the transformation matrix with the smallest error function as the optimal transformation matrix.

The improved SAC-IA algorithm mainly adds the computation and comparison of dissimilarity vectors. The time cost of this step will be much smaller than the time cost of calculating the error function for each complete output and determining whether to exclude it. The efficiency and results of the specific improvements will be illustrated by experimental demonstrations in the subsequent sections of this paper.

### 2.3. ICP Algorithm of Point Clouds Incorporating Bidirectional KD-Tree Optimization for Fine Alignment

The primary requirement for fine point cloud alignment is alignment accuracy, with the ultimate goal of matching the local point cloud exactly to the ideal point cloud. ICP algorithm is a classical algorithm for fine alignment of point clouds [17–20]. The algorithm is essentially an optimal matching algorithm based on least squares. It continuously searches for the corresponding points between the two point clouds, as well as the optimal coordinate transformation, to minimize the distance between the point sets of the corresponding points until a certain convergence condition is met and the iteration terminates. However, the scanned point cloud is close to the position of the ideal point cloud after the initial alignment transformation. Therefore, there will be a situation that the nearest point pair exists many-to-one during the calculation process, which leads to the inefficiency of the alignment of the ICP algorithm. So, we incorporate the bidirectional KD-tree to optimize the ICP algorithm [21].

KD-tree, abbreviated as K-dimension, is commonly used for range or nearest-neighbor searches in high-dimensional spaces. Using it for 3D point cloud data, the dimension of the KD-tree is 3. The construction of KD-tree is to order the disordered point cloud according to the process of determining the segmentation axes and nodes. Meanwhile, we refer to the idea of cross-validation to construct the KD-tree of ideal and scanned point clouds respectively, and save the nearest point pairs with one-to-one correspondence, i.e., bidirectional search. The search time for a bidirectional KD-tree is twice as long as that of a KD-tree, but determining one-to-one pairs of nearest points allows the algorithm’s iteration time to be drastically reduced. Therefore, the KD-ICP algorithm incorporating bidirectional KD-tree is particularly suitable for aligning data-heavy point cloud models.

Denote the data points in the ideal point cloud  $S$  as  $S_i$  ( $i = 1, 2, \dots, n$ ), the data points in the scanned point cloud  $Q$  as  $Q_i$  ( $i = 1, 2, \dots, n$ ). The algorithm finds the transformation matrix of the scanned point cloud to minimize the objective function  $F(\mathbf{R}, \mathbf{T})$ . The mathematical model is represented as follows:

$$\min F(\mathbf{R}, \mathbf{T}) = \sum_{i=1}^m \|S_i - (\mathbf{R}Q_i + \mathbf{T})\|^2 \quad (5)$$

The specific steps of the KD-ICP algorithm for fusing bidirectional KD-tree optimization are as follows:

**Step 1:** Construct the KD-tree of the ideal point cloud  $S$  and the scanned point cloud  $Q$ , respectively. Search for the nearest point  $S_i$  of  $Q_i$  within the ideal point cloud  $S$ , while searching for the nearest point  $Q_j$  of  $S_i$  within the scanned point cloud  $Q$ .

**Step 2:** If  $Q_j = Q_i$  and both are unique, then  $Q_i$  and  $S_i$  are identified as a pair of nearest point pairs with one-to-one correspondence and retained, and vice versa, discarded.

Then, return to **Step 1** to continue the bidirectional search, traversing the points within the scanned point cloud  $Q$  and terminating.

**Step 3:** The points of the scanned point cloud  $Q$  in the nearest point pairs with one-to-one correspondence constitute the set of nearest neighbor points  $Q_i^k$  ( $k$  is the number of iterations) makes  $\|S_i^k - Q_i^k\|$  minimum.

**Step 4:** According to Equation (5), the rotation matrix  $R$  and translation matrix  $T$  are computed by the method of singular value decomposition. The rigid body transformation of  $Q_i^k$  yields  $Q_i^{k'} = R^k Q_i^k + T^k$ .

**Step 5:** Calculate the average distance  $d^k = \frac{1}{m} \|Q_i^{k'} - S_i^k\|$  between the transformed point cloud set  $Q_i^{k'}$  and the theoretical point cloud set  $S_i^k$ . If  $d^k$  is greater than the given iteration termination error threshold  $\tau$  then return to **Step 1** and continue iterating until  $d^k$  is less than the given threshold or the number of iterations has been greater than the preset number of terminations.

The accuracy of point cloud alignment needs to be measured by an evaluation index. Root mean square error (RMSE) can measure the deviation between the predicted value and the real value, which is used for evaluating the accuracy of point cloud registration. The equation is written as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (D_i - \hat{D}_i)^2}{n}} \tag{6}$$

where  $n$  denotes the number of corresponding point pairs;  $D_i$  denotes the Euclidean distance between the  $i$ -th counterpart to the counterpart after alignment; and  $\hat{D}_i$  denotes the truth value of the Euclidean distance between corresponding points. However, the use of RMSE for point cloud alignment accuracy evaluation is also essentially an algorithm to find nearest neighbor pairs. The distance between the two is then calculated, so there is no theoretically absolute true value for the distance between pairs of points. To facilitate the computational processing, we take the distance between the corresponding points in the absolute ideal state, after the point cloud is fully aligned, as an estimate of the true value, i.e.,  $\hat{D}_i = 0$ . However, it is undeniable that the RMSE is closely related to the alignment results, and the higher the alignment accuracy, the smaller the value of the RMSE must be. Therefore, we are able to use the RMSE as an evaluation index to evaluate the alignment results, which is also a more common evaluation index in the field of point cloud alignment at present.

#### 2.4. The Overall Flow of the Improved Alignment Method

In the measurement process of laser scanners, the direct scanned data may be noisy or redundant due to lighting factors or viewing angle obstructions. Therefore, it is necessary to preprocess the point cloud data. Common point cloud preprocessing methods include point cloud noise reduction, point cloud streamlining, and point cloud stitching [22,23].

Point cloud noise reduction is the process of removing outliers and noisy points from point cloud data. Common denoising methods include voxel filtering, statistical filtering, bilateral filtering, conditional filtering, Gaussian filtering, etc. [24,25]. At the same time, too large and redundant point cloud data will affect the accuracy and speed of data processing. It is necessary to streamline point cloud data. Currently, mature point cloud streamlining algorithms include the enveloping box (circle, sphere, body) method, uniform mesh method, angle method, triangular mesh method, etc. [26–28].

After the preprocessing step is completed, the alignment of the local feature point cloud with the ideal CAD model is performed. Dissimilarity vectors are introduced to remove feature correspondences with large differences in advance to improve the SAC-IA algorithm as the initial alignment algorithm for point clouds. Then, the ICP algorithm is

optimized using the bidirectional KD-tree to accelerate the nearest-neighbor search with precise correspondence as a point cloud fine alignment algorithm. The specific algorithm flow is shown in Figure 2.

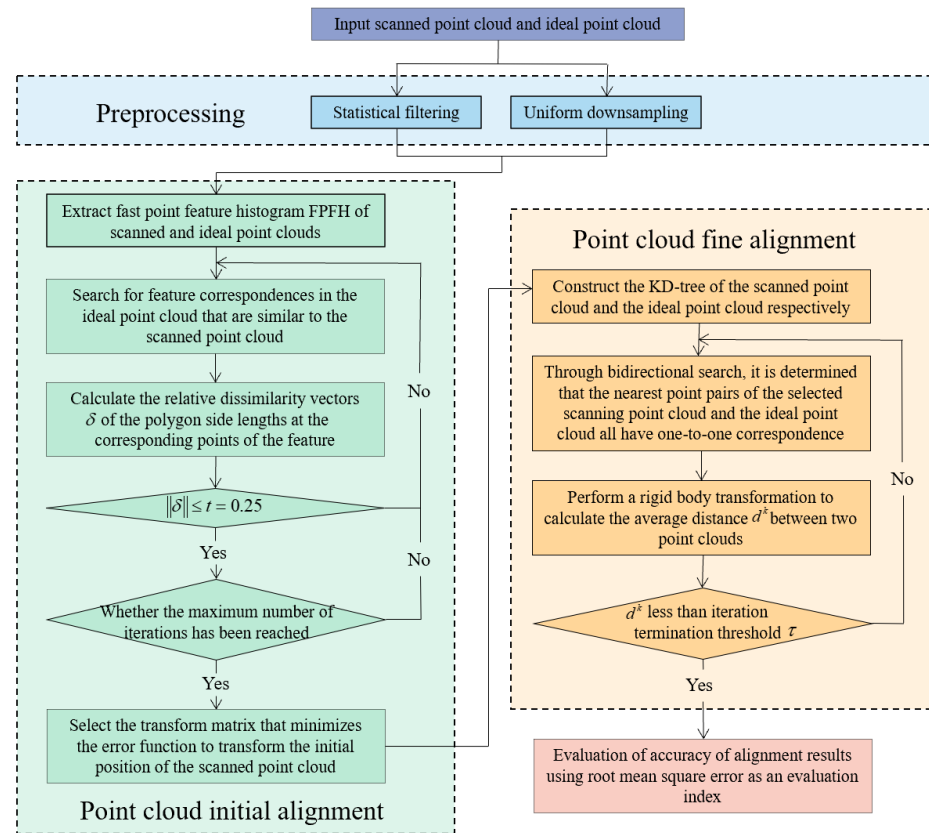


Figure 2. Flowchart of improved alignment algorithm.

### 3. Test Experiment

#### 3.1. Classical Sample Data

The reliability and feasibility of the alignment algorithm needs to be proved experimentally using point cloud data. The proposed alignment algorithm was tested for reliability using data from the Stanford bunny and the dragon 3D model, which is highly recognized and used in academia. The specific point cloud data visualization shown in Figure 3a is a partial point cloud with localized features to be aligned for the bunny sample, as well as the dragon sample; Figure 3b is the corresponding complete target point cloud.

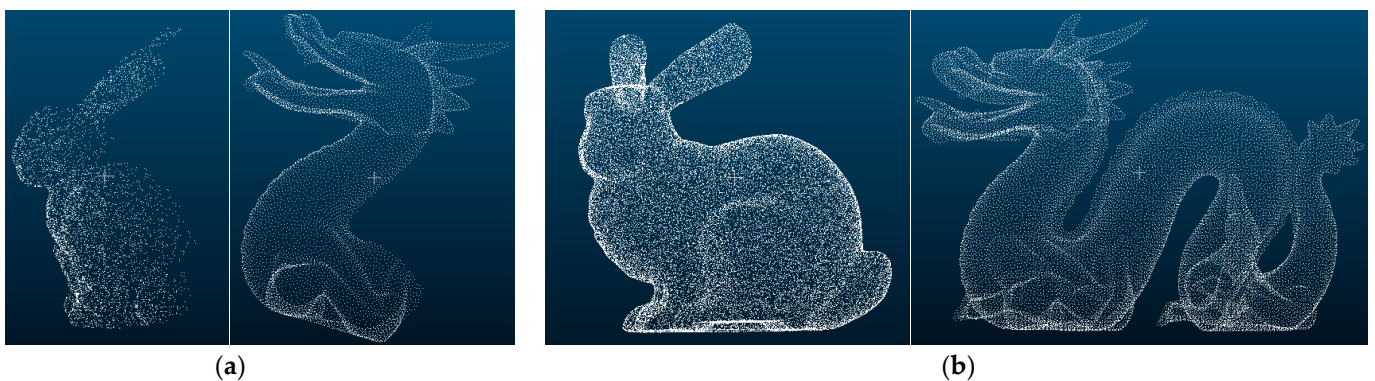


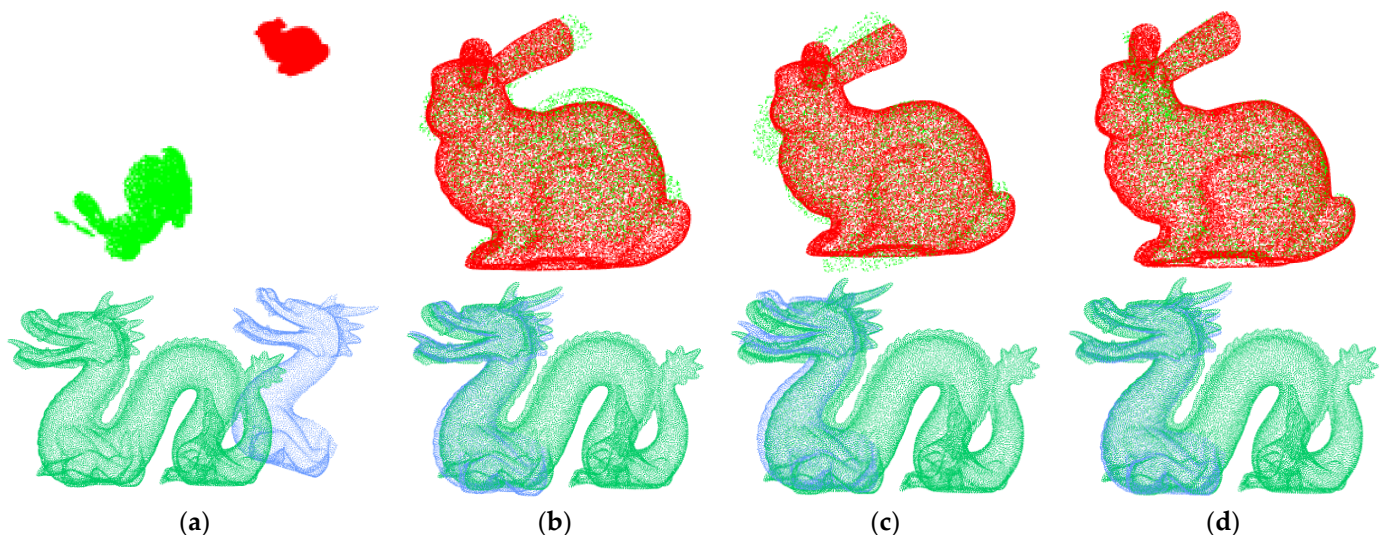
Figure 3. Classical sample. (a) Partial point cloud; (b) Complete point cloud.

### 3.2. Initial Alignment of Classical Sample

The algorithms were all implemented on a computer with a processor of AMD Ryzen5 3550H with Radeon Vega Mobile Gfx 2.10 GHz processor and 16.0 GB of RAM (14.9 GB available). The algorithm software is based on Visual Studio 2013 assembling the Point Cloud Library 1.8.0 (PCL) Environment. Following the basic process of the alignment method, the bunny and dragon point cloud data were first preprocessed. Then, the initial alignment of the point cloud is performed based on the improved SAC-IA algorithm. In order to test the specific enhancement of the improved SAC-IA algorithm, both the SAC-IA algorithm as well as the commonly used RANSAC algorithm are used for the initial alignment of the point cloud of the classical sample.

The calculation of normal vector and FPFH in the algorithm is based on PCL, so the threshold of the initial point sampling does not need to be set separately in the algorithm implementation, and PCL will be implemented automatically. We set the radius of the normal vector estimate to 1/10 of the total size of the origin cloud. In general, the neighborhood search radius for performing FPFH calculations is larger than the normal vector estimation radius, which we set to be 1/8 of the total size of the origin cloud. Then, make small adjustments to the setpoints based on the results. The iteration termination threshold of the initial alignment algorithm is set to 1, i.e., the iteration stops when the minimum distance between the selected samples reaches 1 mm. Of course, this setting does not guarantee that all points are optimally aligned, but only iterates until the alignment of some of the selected points is complete, so fine alignment is still needed afterwards.

Figure 4 is a visualization of the experimental results of the initial alignment algorithm. Where, (a) shows the initial position of the two point clouds, and it can be found that there is no corresponding overlap between them. (b–d) are the positional figures of the converted point cloud and the target point cloud after the different algorithms are aligned, and the good or bad alignment can be judged most intuitively by observing the amount of overlap between the two point clouds.

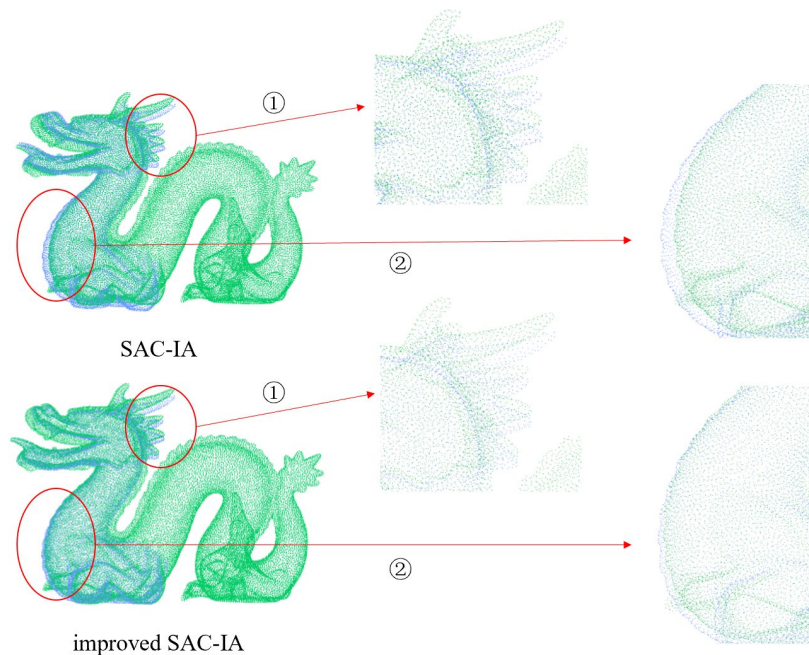


**Figure 4.** Initial alignment of classical sample. (a) Initial position; (b) SAC-IA; (c) RANSAC; (d) Improved SAC-IA.

It can be found that our improved SAC-IA algorithm has the most overlapping portion of point cloud results compared to SAC-IA algorithm and RANSAC algorithm for both rabbit and dragon samples. For example, the difference between the SAC-IA algorithm and the modified SAC-IA algorithm dragon point cloud alignment result visualization local zoom is shown in Figure 5. Among them, at ① and ②, the non-overlapping part of the alignment results is more obvious. Through local zoom, it can be more intuitively seen that

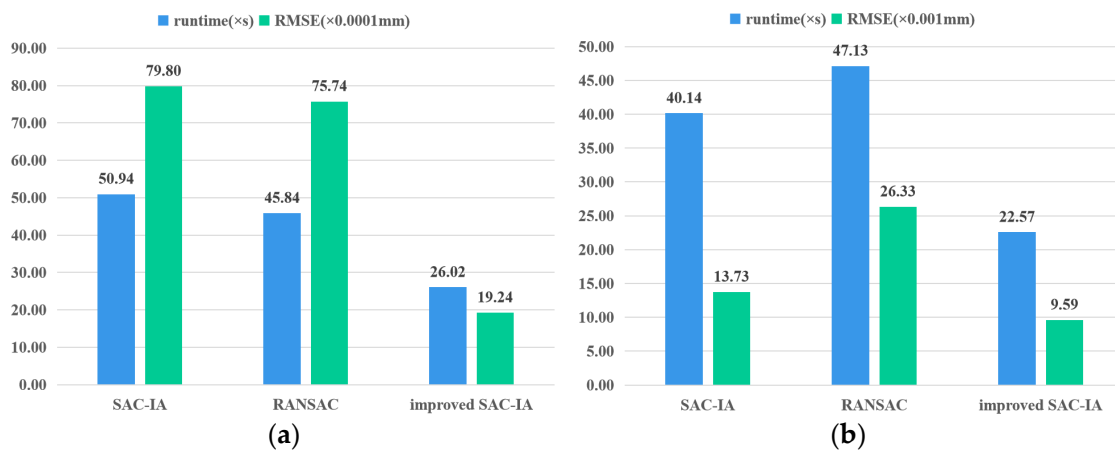


the improved SAC-IA algorithm alignment results in more overlapping parts of the point cloud and better alignment.



**Figure 5.** Local differences in the initial alignment of the dragon point cloud.

The visualization results can intuitively see that the improved SAC-IA algorithm has the best initial alignment results for both bunny and dragon samples. And the SAC-IA algorithm has poor alignment results with the RANSAC algorithm. Of course, a more reliable way to compare results is the runtime of the algorithm and the accuracy of the alignment. The running time of each algorithm under the same computer configuration and the alignment accuracy using the RMSE as an evaluation metric are listed. Comparison of algorithm performance with reference to the above data. The data are shown in Figure 6.



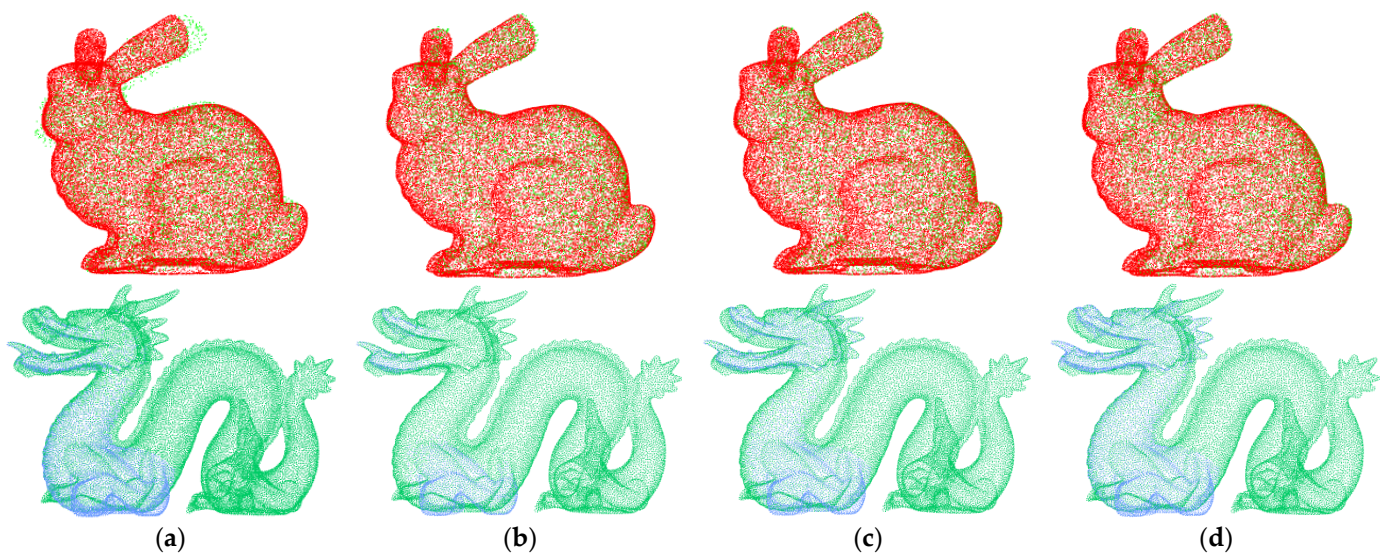
**Figure 6.** Comparison of initial alignment algorithms for classical sample. (a) Bunny sample; (b) Dragon sample.

The initial alignment speed of the improved SAC-IA algorithm is 48.9% higher than the SAC-IA algorithm and 43.3% higher than the RANSAC algorithm in the bunny sample, and 43.7% and 52.1% higher in the dragon sample, respectively. It has been experimentally proven that the improved SAC-IA algorithm has a more significant improvement in alignment speed and also has a greater improvement in alignment accuracy. The results show

that the algorithm removes the error sampling of the SAC-IA algorithm in advance using the dissimilarity vector, allowing the algorithm to find more possible counterparts again with the same number of iterations as the termination constraints. Improvements make the algorithm faster and better at determining the optimal transformation matrix.

### 3.3. Fine Alignment of Classical Sample

Following the alignment process, point cloud fine alignment is performed after the initial point cloud alignment is completed. Both the SAC-IA algorithm alignment results and the improved SAC-IA algorithm alignment results are finely aligned using the ICP algorithm, as well as the KD-ICP algorithm. The visualization of the alignment results is shown in Figure 7.



**Figure 7.** Fine alignment of classical sample. (a) SAC-IA+ICP; (b) SAC-IA+KD-ICP; (c) Improved SAC-IA+ICP; (d) The proposed algorithm.

The alignment results of the ICP algorithm are very dependent on the initial position of the point cloud. For the same original point cloud of bunny sample data, Figure 7a,b are compared with the same number of iterations and other parameters set by the ICP, as well as KD-ICP algorithms. In (a), it is possible to visualize the less-than-perfect positional situation obtained from the initial alignment based on the SAC-IA algorithm, where the ICP algorithm is not able to align successfully. However, (b) shows that the KD-ICP algorithm incorporating the bidirectional KD-tree improvement can be realized. (d) is a graph of the results obtained according to the flow of the alignment method proposed in this paper, in which the bunny sample results can be directly compared to observe its excellent alignment. Yet, the alignment of the dragon sample data resulted in all four sets of experiments performing well enough that no difference could be directly observed. When the visual result figure does not intuitively show the difference in the alignment of different algorithms, we still analyze it by more reliable data results. Therefore, we compare and analyze the running time with the alignment accuracy in conjunction for the specific algorithms. The data are shown in Table 1, where the proposed method has the highest alignment accuracy, i.e., the smallest RMSE, and the shortest algorithm runtime.

**Table 1.** Comparison of classical sample alignment algorithms.

Clarification	Algorithm	Bunny Sample		Dragon Sample		
		Runtime (s)	RMSE (mm)	Runtime (s)	RMSE (mm)	
Initial alignment	SAC-IA	50.939	$7.980 \times 10^{-3}$	40.138	$1.373 \times 10^{-2}$	
	RANSAC	45.844	$7.574 \times 10^{-3}$	47.134	$2.633 \times 10^{-2}$	
	Improved SAC-IA	26.016	$1.924 \times 10^{-3}$	22.573	$9.594 \times 10^{-3}$	
Fine alignment	SAC-IA result	ICP	$7.471 \times 10^{-3}$	3.485	$7.609 \times 10^{-7}$	
		KD-ICP	4.404	$9.573 \times 10^{-4}$	2.031	$5.395 \times 10^{-7}$
	Improved SAC-IA result	ICP	6.000	$7.261 \times 10^{-4}$	3.193	$7.492 \times 10^{-7}$
		KD-ICP	2.960	$7.245 \times 10^{-4}$	1.422	$2.878 \times 10^{-7}$

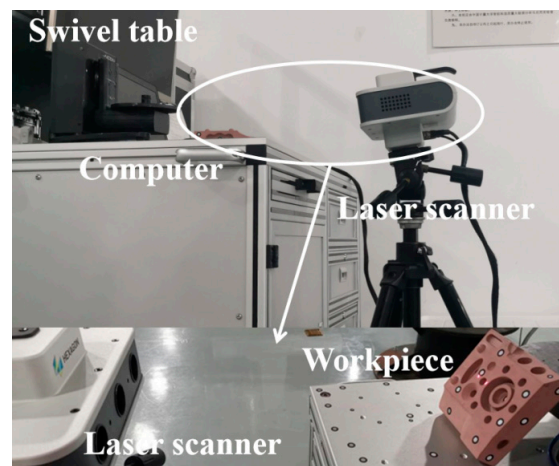
There is independence in the evaluation metrics of point cloud alignment in terms of run time as well as RMSE, because these metrics are affected by factors such as the size of different point cloud objects, the size of the data volume, and so on. So, they are both only suitable for longitudinal evaluation of the results of different algorithmic processing of the same point cloud object. As can be seen in Table 1, according to the flow of the alignment method proposed in this paper, the accuracy of the final alignment results of the initial alignment with the improved SAC-IA algorithm is higher than that using the SAC-IA algorithm. This shows that the improved SAC-IA algorithm can indeed provide a better transformed bit position for fine alignment, which in turn improves the final alignment accuracy. The KD-ICP algorithm optimized with bidirectional KD-tree for fine alignment shows significant improvement in runtime as well as alignment accuracy for both bunny and dragon samples. The proposed alignment method reduces the RMSE of the bunny sample from  $7.860 \times 10^{-3}$  to  $7.245 \times 10^{-4}$  compared to the base algorithm; the total running time of the algorithm was reduced from 58.410 s to 28.976 s, i.e., the speed of alignment was increased by 50.392%. The improvement in dragon sample alignment results is also intuitively obvious. Its RMSE decreased from  $7.609 \times 10^{-7}$  to  $2.878 \times 10^{-7}$ ; the total running time of the algorithm was reduced from 43.623 s to 23.995 s, i.e., the speed of the alignment was improved by 44.995%.

The results show that the proposed alignment method improves both the initial alignment and the fine alignment algorithm to optimize the final alignment results from two aspects. First, the initial alignment is improved by the introduction of a dissimilarity vector, so that the initial alignment provides a better transformed bit position, which in turn affects the alignment results of the fine alignment. Then, the nearest point pairs with one-to-one correspondence are identified by a bidirectional KD-tree for subsequent iterations of the algorithm. It makes the KD-ICP algorithm not only accelerate the search of nearest neighbor pairs to improve the alignment speed, but also make the selection of nearest point pairs more accurate to improve the alignment accuracy. The experimental results on classical samples prove the feasibility of the alignment method proposed in this paper, but practical experiments are needed to align the scanned point cloud data to prove the reliability of the method proposed in this paper.

## 4. Practical Experiment

### 4.1. Scanned Sample Data Acquisition

PrimeScan, a portable scanning system from Hexagon AICON Measurement Systems (see Figure 8) was used to scan and obtain 3D data information about the workpiece. Its resolution is 5 and 8 Mpx. The 3D scanner consists of two high-resolution industrial CCD cameras distributed on the left and right sides, as well as a central raster projection unit, which uses structured light measurement.

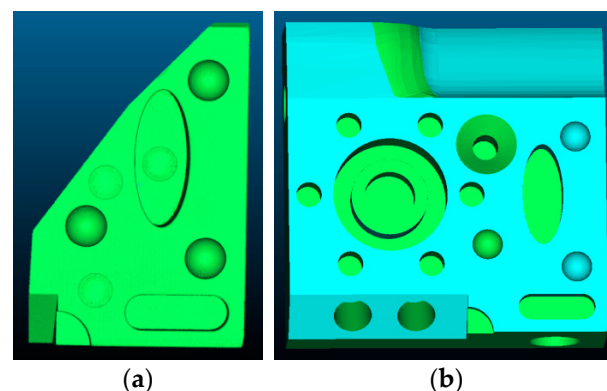


**Figure 8.** Laser scanner experimental setup.

Structured light is laser light emitted from a laser and converged into a narrow band after passing through a cylindrical lens. By means of a grating projection unit, a set of grating stripes with phase information is projected onto the surface of the measuring workpiece and synchronized by two industrial cameras, left and right. The 3D laser scanner is calibrated and ready for actual part measurement. In order to perform the measurement, the sensor needs to be placed in front of the measuring object. The ideal operating distance depends on the selected measuring area of the sensor. However, in reality the optimal working distance is where the two laser red dots coincide, which needs to be adjusted to the center of the object to be measured.

The point cloud data generated by the laser scanner [29–31] does not produce singular deformations. It can be used to analyze the actual condition of the surface of the workpiece by reproducing the 3D data of the measured workpiece one-by-one, and it can also be used directly for the alignment study with the ideal CAD model.

However, the scanned result of the laser scanner is affected by the scanned angle and the shape of the measured workpiece. It is impossible to generate the complete surface information of the measured workpiece from only one viewpoint for a workpiece with complex curved surfaces. Therefore, partial measurements are performed to obtain localized point cloud data of the measured workpiece and simply fill the holes created by the marked points. The specific scanned results are shown in Figure 9. Figure 9a shows the point cloud of local features to be aligned; Figure 9b is the ideal target point cloud.

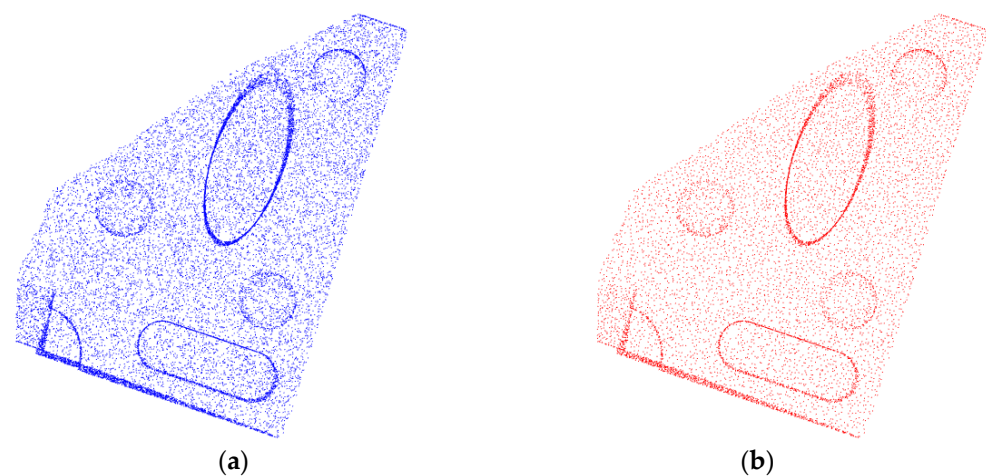


**Figure 9.** Scanned workpiece model. (a) Scanned data; (b) Ideal CAD data.

#### 4.2. Scanned Sample Data Preprocessing

The data obtained by laser scanner measurement are affected by the measurement environment and other factors during the measurement process. The directly obtained

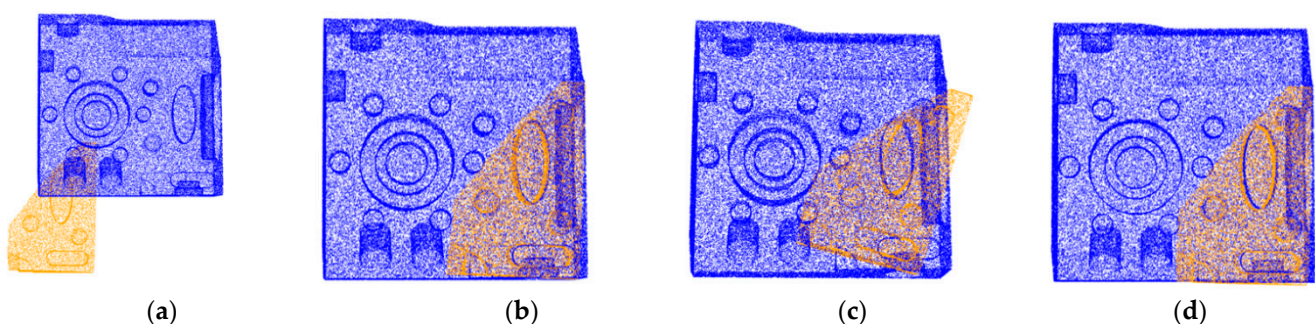
data may have outliers or data redundancy, etc., which requires point cloud preprocessing. To begin with, the scanned point cloud is filtered to remove noise points and outliers. Statistical filtering is capable of processing point clouds of different shapes with a wide range of applicability. Moreover, the statistical filtering is based on the distance distribution characteristics from the point to all neighboring points, and each point field in the point cloud data is statistically analyzed to filter out some outliers that do not meet the requirements. At the same time, the uniform grid method is used to streamline the point cloud data, which is a better way to reduce data redundancy, while preserving the point cloud features. A specific before and after comparison of the scanned point cloud data preprocessing is shown in Figure 10. The original point cloud has 20,001 data points, which is streamlined to 9418 data points after preprocessing. The point cloud is arranged more regularly, removing redundancy, while better preserving the data characteristics of the original point cloud.



**Figure 10.** Comparison of scanned workpiece before and after preprocessing. (a) Original scanned point cloud; (b) Preprocessed point cloud.

#### 4.3. Initial Alignment of Scanned Sample

According to the local feature point cloud alignment process, the initial alignment of the scanned workpiece point cloud data is performed. The SAC-IA algorithm, the RANSAC algorithm, and the modified SAC-IA algorithm were also used for the initial alignment of the point cloud. The visualization of the alignment results is shown in Figure 11.



**Figure 11.** Scanned workpiece model initial alignment. (a) Initial position; (b) SAC-IA; (c) RANSAC; (d) Improved SAC-IA.

From Figure 11, it can be seen that both the SAC-IA algorithm and the improved SAC-IA algorithm can be used for the alignment of point cloud data with a large data volume and a large search radius. The algorithms are also better aligned with the measured workpiece. The difference between the specific SAC-IA as well as the improved SAC-IA algorithms

in terms of runtime and alignment results is shown in Figure 12. The improved SAC-IA algorithm aligns 37.8% faster than the SAC-IA algorithm and has a significant improvement in alignment accuracy, while the RANSAC algorithm has poorer alignment results.

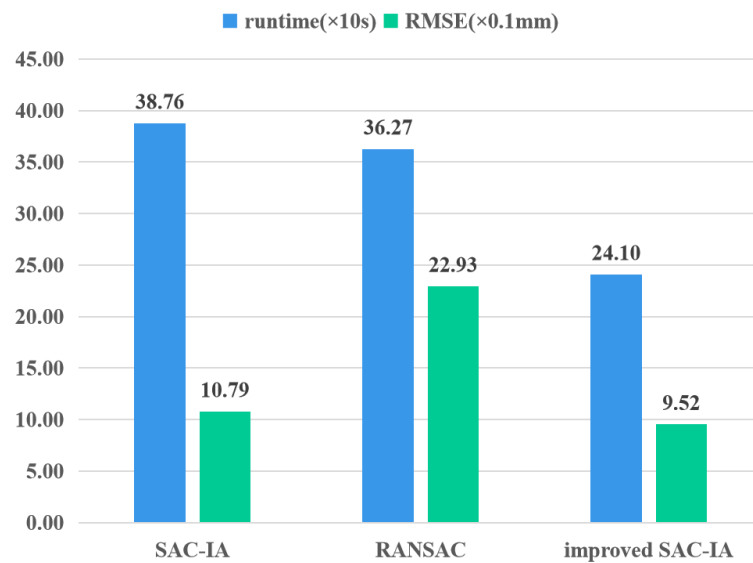


Figure 12. Comparison of initial alignment algorithms for scanned workpiece model.

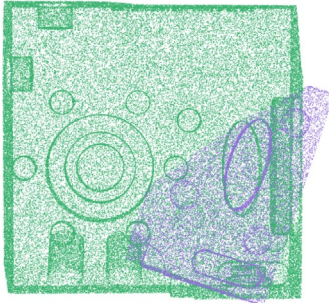
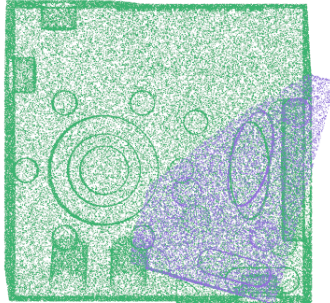
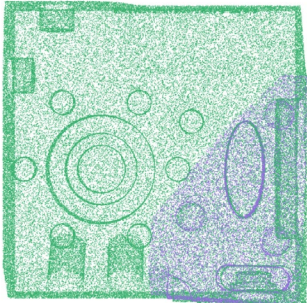
The transformation matrix for the initial alignment of the workpiece model is as follows:

$$H = \begin{bmatrix} 0.998 & 0.011 & -0.066 & 82.365 \\ -0.008 & 0.999 & 0.052 & 52.218 \\ 0.066 & -0.051 & 0.996 & 39.678 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4.4. Fine Alignment of Scanned Sample

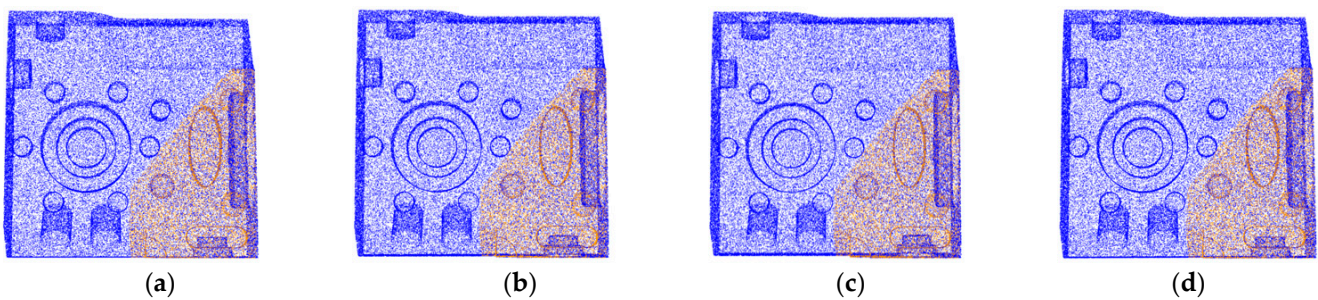
When the classical sample data alignment is performed in Section 3.3, it has been shown that the KD-ICP algorithm, which incorporates bi-directional KD-tree improvements, also has good alignment results against poor initial positional attitude. Also, to increase persuasiveness, we use the GICP alignment algorithm based on principal component analysis proposed in the literature [32] as a comparative. Its essence is also the optimization of the most classical ICP algorithm. We borrowed the initial alignment results obtained by the RANSAC algorithm with insufficient overlap (see Figure 11c) for the test. For the scanned point cloud, all three fine alignment algorithms set the maximum distance between corresponding pairs of points to 10, which means that the algorithms can find corresponding pairs of points with a maximum distance of 10 mm. We set the maximum number of iterations to 50 and the convergence threshold to 1, i.e., we stop iterating when the number of convergences is reached, or the mean square error sum is less than the threshold value of 1 mm. Under such conditions, the ICP algorithm, the GICP algorithm, and the KD-ICP algorithm were used for the initial alignment results of the RANSAC algorithm (Figure 11c data) to perform fine alignment. The visualization of the alignment results and the accuracy of the alignment are shown in Table 2.

**Table 2.** Comparison of ICP, GICP, and KD-ICP algorithms for scanned workpiece.

	ICP	GICP	KD-ICP
Visualization of alignment results			
RMSE (mm)	2.2538	2.1763	0.7480

This can be illustrated by the visualization graph of the alignment results shown in Table 2, as well as the corresponding results of the alignment accuracy data. We can see that the alignment results of the GICP algorithm are better than the ICP algorithm, indicating that the improvement of the literature [32] is effective, but the improvement is not significant. It may be due to the fact that the improvement principle of the GICP algorithm is based on principal component analysis, which considers the whole point cloud data as a whole. Therefore, it is less applicable for overall and local point cloud data alignment. In contrast, the proposed improved KD-ICP algorithm based on bidirectional KD-tree is more applicable and has the best alignment effect among the three algorithm comparisons. This is because our bidirectional KD-tree is optimized for the selection of the nearest point pairs in the most basic and important part of the algorithm, which makes the selection of point pairs more accurate and thus improves the alignment accuracy and applicability. In the scanning point cloud alignment, the improved KD-ICP algorithm incorporating bidirectional KD-tree can optimize the selection of the nearest point pairs and make the point pair selection more accurate to improve the alignment accuracy, which is more applicable.

Similar to the classical sample alignment, the SAC-IA algorithm alignment results for the scanned workpiece and the modified SAC-IA algorithm alignment results were aligned using the ICP algorithm as well as the KD-ICP algorithm, and the alignment results are visualized in Figure 13. Since the SAC-IA algorithm and the improved SAC-IA algorithm have good initial alignment results and provide a good change of position, both the ICP algorithm and the KD-ICP algorithm are better at achieving fine alignment of scanned parts. From Figure 13, it appears that the differences are not significant, so the data were used for further analysis, as shown in Table 3.



**Figure 13.** Scanned workpiece model fine alignment. (a) SAC-IA+ICP; (b) SAC-IA+KD-ICP; (c) Improved SAC-IA+ICP; (d) The proposed algorithm.

**Table 3.** Comparison of scanned workpiece model alignment algorithms.

Clarification		Algorithm	Runtime (s)	RMSE (mm)
Initial alignment		SAC-IA	387.591	1.079
		RANSAC	362.716	2.293
		Improved SAC-IA	240.964	0.953
Fine alignment	SAC-IA result	ICP	27.775	0.605
		KD-ICP	14.276	0.598
	Improved	ICP	17.499	0.601
	SAC-IA result	KD-ICP	9.822	0.597

From Table 3, it can be seen that according to the proposed alignment method, the RMSE of the scanned point cloud is reduced from 0.605 to 0.597 compared to the base algorithm; the total running time of the algorithm was reduced from 415.366 s to 250.786 s, i.e., the speed of alignment was improved by 39.623%. As can be seen from the results of the previous test experiments, the overall algorithmic alignment speed of the rabbit point cloud data is improved by 50.392%; the dragon point cloud data is improved by 44.995%. By combining all of the experimental results, it is roughly possible to assess that the algorithm proposed in this paper operates with a speedup of 40% to 50% compared to the base algorithm. Since SAC-IA and the improved SAC-IA algorithm give good initial transformed poses, which affect the fine alignment, the alignment accuracies of the four sets of experiments are in the upper and lower ranges of 0.6, and the KD-ICP algorithm improves, but not by much. The reason for this is that the scanned workpiece itself has surface roughness and is labeled with marking points, while the target model is an ideal CAD model. It is impossible for the two to achieve absolute coincidence with the RMSE of zero. In terms of operating speed, the large size of the scanned part corresponds to a large search radius of the algorithm. The proposed alignment method reduces the time required for alignment well and can be applied to large size point cloud alignment.

The transformation matrix for the fine alignment of the workpiece model is as follows:

$$H = \begin{bmatrix} 0.999 & -0.021 & -0.025 & 3.524 \\ 0.021 & 1.000 & 0.004 & -1.503 \\ 0.024 & -0.005 & 1.000 & -3.038 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

### 5. Conclusions

This paper proposes an improved point cloud alignment method for the local feature point cloud alignment problem. First, the improved SAC-IA algorithm is used for the initial alignment of the point cloud, and the introduction of dissimilarity vectors can remove erroneous feature point pair selection in advance. The bidirectional KD-tree can accelerate the nearest-neighbor pair search and optimize the selection of nearest-neighbor pairs, which improves the ICP algorithm’s heavy dependence on the initial positional attitude and makes it more applicable. Moreover, the well-recognized classical sample data as well as the actual workpiece model scanned by the laser scanner were selected for the experiment. Experimental analysis shows that the improved SAC-IA is able to increase the operation speed by 30% to 50% compared to the SAC-IA algorithm; KD-ICP is able to increase the speed of the ICP algorithm by 30~50%; and the overall running speed of the algorithm can be increased by 40% to 50%. In terms of alignment accuracy, for all the experimental subjects, the alignment results obtained by the proposed alignment method in this paper have the smallest RMSE and the best alignment results. Through test experiments and practical experiments, the feasibility and reliability of the proposed improved point cloud alignment method applied to local point cloud data alignment are demonstrated. The alignment efficiency of this method is also significantly improved compared to conventional methods.



**Author Contributions:** Conceptualization, Y.C., H.C. and Y.L.; methodology, Y.C., H.C. and Y.L.; software, Y.C., Y.L. and Y.T.; validation, Z.L. and S.L.; formal analysis, H.C.; investigation, H.C.; resources, Y.T. and S.L.; data curation, Y.T. and S.L.; writing—original draft preparation, H.C. and Y.L.; writing—review and editing, Y.C. and Z.L.; visualization, Y.T.; supervision, Y.C. and Z.L.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Science Foundation of Zhejiang Province of China under grant No. LY22E050005, the Science and Technology Plan of State Administration for Market Regulation of China under grant No. 2021MK188, and the Zhejiang Province Industry University Cooperation Collaborative Education Project.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Wei, S.; Xiao, Y.; Yang, X.; Wang, H. Attitude Estimation Method for Target Ships Based on LiDAR Point Clouds via An Improved RANSAC. *J. Mar. Sci. Eng.* **2023**, *11*, 1755. [[CrossRef](#)]
- Mao, Y.; Wang, H.; Cao, W.; Fu, Y.; He, L.; Bao, N. Extraction of Step-Feature Lines in Open-Pit Mines Based on UAV Point-Cloud Data. *Sensors* **2022**, *22*, 5706. [[CrossRef](#)] [[PubMed](#)]
- Wang, X.; Chen, H.; Wu, L. Feature extraction of point clouds based on region clustering segmentation. *Multimed. Tools Appl.* **2020**, *79*, 11861–11889. [[CrossRef](#)]
- Zhao, T.; Li, L.; Tian, T.; Ma, J.; Tian, J. Patch-guided point matching for point cloud registration with low overlap. *Pattern Recognit.* **2023**, *144*, 109876. [[CrossRef](#)]
- Shi, J.; Ye, H.; Yang, B.; Cao, F. An iteration-based interactive attention network for 3D point cloud registration. *Neurocomputing* **2023**, *560*, 126822. [[CrossRef](#)]
- Li, P.; Zhang, Q.; Wang, X.; Zhao, T.; Wang, Y. Method for detecting pipeline spatial attitude using point cloud alignment. *Measurement* **2023**, *219*, 113279. [[CrossRef](#)]
- Cheng, L.; Chen, S.; Liu, X.; Xu, H.; Wu, Y.; Li, M.; Chen, Y. Registration of Laser Scanning Point Clouds: A Review. *Sensors* **2018**, *18*, 1641. [[CrossRef](#)] [[PubMed](#)]
- Marshall, G.F.; Stutz, G.E. (Eds.) *Handbook of Optical and Laser Scanning*, 2nd ed.; CRC Press: London, UK, 2011.
- Bedkowski, J. Benchmark of multi-view Terrestrial Laser Scanning Point Cloud data registration algorithms. *Measurement* **2023**, *219*, 113199. [[CrossRef](#)]
- Sahillioğlu, Y.; Kavan, L. Scale-Adaptive ICP. *Graph Models* **2021**, *116*, 101113. [[CrossRef](#)]
- Yao, Z.; Zhao, Q.; Li, X.; Bi, Q. Point cloud registration algorithm based on curvature feature similarity. *Measurement* **2021**, *177*, 109274. [[CrossRef](#)]
- He, Y.; Liang, B.; Yang, J.; Li, S.; He, J. An Iterative Closest Points Algorithm for Registration of 3D Laser Scanner Point Clouds with Geometric Features. *Sensors* **2017**, *17*, 1862. [[CrossRef](#)] [[PubMed](#)]
- Hu, C.; Ru, Y.; Fang, S.; Zhou, H.; Xue, J.; Zhang, Y.; Li, J.; Xu, G.; Fan, G. A Tree Point Cloud Simplification Method Based on FPFH Information Entropy. *Forests* **2023**, *14*, 1507. [[CrossRef](#)]
- Singh, D.-P.; Yadav, M. 3D-MFDNN: Three-dimensional multi-feature descriptors combined deep neural network for vegetation segmentation from airborne laser scanning data. *Measurement* **2023**, *221*, 113465. [[CrossRef](#)]
- Ge, P.; Wang, Y.; Zhou, J.; Wang, B. Point cloud optimization of multi-view images in digital image correlation system. *Opt. Lasers Eng.* **2024**, *173*, 107931. [[CrossRef](#)]
- Su, Z.; Hu, C.; Hao, J.; Ge, P.; Han, B. Target Detection in Single-Photon Lidar Using CNN Based on Point Cloud Method. *Photonics* **2023**, *11*, 43. [[CrossRef](#)]
- Mao, Z.; Zhang, C.; Guo, B.; Xu, Y.; Kong, C.; Zhu, Y.; Xu, Z.; Jin, J. The Flatness Error Evaluation of Metal Workpieces Based on Line Laser Scanning Digital Imaging Technology. *Photonics* **2023**, *10*, 1333. [[CrossRef](#)]
- Wang, W.; Zhao, C.; Zhang, H. PR-Alignment: Multidimensional Adaptive Registration Algorithm Based on Practical Application Scenarios. *Machines* **2023**, *11*, 254. [[CrossRef](#)]
- Zhao, Y.; Cheng, Y.; Xu, Q.; Luo, Z.; Wang, X.; Li, H. Uncertainty modeling and evaluation of profile measurement by structured light scanner. *Meas. Sci. Technol.* **2022**, *33*, 5018. [[CrossRef](#)]
- Li, X.; Chen, X.; Li, W.; Yin, X.; Yuan, X.; Chen, H.; Zhou, J.; Ma, X. High-accuracy calibration method for an underwater one-mirror galvanometric laser scanner. *Opt. Express* **2023**, *31*, 5973–5989. [[CrossRef](#)]
- He, Y.; Yang, J.; Hou, X.; Pang, S.; Chen, J. ICP registration with DCA descriptor for 3D point clouds. *Opt. Express* **2021**, *29*, 20423–20439. [[CrossRef](#)]

22. Yu, X.; Rao, Y.; Wang, Z.; Lu, J.; Zhou, J. AdaPoinTr: Diverse Point Cloud Completion with Adaptive Geometry-Aware Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14114–14130. [[CrossRef](#)]
23. Wang, N.; Ma, D.; Du, X.; Li, B.; Di, D.; Pang, G.; Duan, Y. An automatic defect classification and segmentation method on three-dimensional point clouds for sewer pipes. *Tunn. Undergr. Space Technol.* **2024**, *143*, 105480. [[CrossRef](#)]
24. Gou, F.; Li, Y.; Hou, C.; Wang, H. 3D point cloud global denoising algorithm based on different noise characteristics. *Meas. Sci. Technol.* **2024**, *35*, 5032. [[CrossRef](#)]
25. Zhou, S.; Liu, X.; Wang, C.; Yang, B. Non-iterative denoising algorithm based on a dual threshold for a 3D point cloud. *Opt. Lasers Eng.* **2020**, *126*, 105921. [[CrossRef](#)]
26. Baiocchi, A.; Giagu, S.; Napoli, C.; Serra, M.; Nardelli, P.; Valleriani, M. Artificial neural networks exploiting point cloud data for fragmented solid objects classification. *Mach. Learn. Sci. Technol.* **2023**, *4*, 5025. [[CrossRef](#)]
27. Li, M.; Nan, L. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 135–150. [[CrossRef](#)]
28. Wang, S.; Hu, Q.; Xiao, D.; He, L.; Liu, R.; Xiang, B. A new point cloud simplification method with feature and integrity preservation by partition strategy. *Measurement* **2022**, *197*, 111173. [[CrossRef](#)]
29. Cheng, X.; Li, Z.; Zhong, K.; Shi, Y. An automatic and robust point cloud registration framework based on view-invariant local feature descriptors and transformation consistency verification. *Opt. Lasers Eng.* **2017**, *98*, 37–45. [[CrossRef](#)]
30. Si, H.; Qiu, J.; Li, Y. A Review of Point Cloud Registration Algorithms for Laser Scanners: Applications in Large-Scale Aircraft Measurement. *Appl. Sci.* **2022**, *12*, 10247. [[CrossRef](#)]
31. Fang, L.; You, Z.; Shen, G.; Chen, Y.; Li, J. A joint deep learning network of point clouds and multiple views for roadside object classification from lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2022**, *193*, 115–136. [[CrossRef](#)]
32. Zhao, W.; Zhang, D.; Li, D.; Zhang, Y.; Ling, Q. Optimized GICP registration algorithm based on principal component analysis for point cloud edge extraction. *Meas. Control* **2024**, *57*, 77–89. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.