



## Article

# Super Edge 4-Points Congruent Sets-Based Point Cloud Global Registration

Shikun Li <sup>1</sup> , Ruodan Lu <sup>2</sup>, Jianya Liu <sup>3,\*</sup> and Liang Guo <sup>1</sup>

<sup>1</sup> School of Mathematics & Statistics, Shandong University, Weihai 264209, China; shikun.li@mail.sdu.edu.cn (S.L.); Liangguo@sdu.edu.cn (L.G.)

<sup>2</sup> Darwin College, University of Cambridge, Cambridge CB3 9EU, UK; rl508@cam.ac.uk

<sup>3</sup> Data Science Institute, Shandong University, Jinan 250100, China

\* Correspondence: jyliu@sdu.edu.cn; Tel.: +86-631568-8927

**Abstract:** With the acceleration in three-dimensional (3D) high-frame-rate sensing technologies, dense point clouds collected from multiple standpoints pose a great challenge for the accuracy and efficiency of registration. The combination of coarse registration and fine registration has been extensively promoted. Unlike the requirement of small movements between scan pairs in fine registration, coarse registration can match scans with arbitrary initial poses. The state-of-the-art coarse methods, Super 4-Points Congruent Sets algorithm based on the 4-Points Congruent Sets, improves the speed of registration to a linear order via smart indexing. However, the lack of reduction in the scale of original point clouds limits the application. Besides, the coplanarity of registration bases prevents further reduction of search space. This paper proposes a novel registration method called the Super Edge 4-Points Congruent Sets to address the above problems. The proposed algorithm follows a three-step procedure, including boundary segmentation, overlapping regions extraction, and bases selection. Firstly, an improved method based on vector angle is used to segment the original point clouds aiming to thin out the scale of the initial point clouds. Furthermore, overlapping regions extraction is executed to find out the overlapping regions on the contour. Finally, the proposed method selects registration bases conforming to the distance constraints from the candidate set without consideration about coplanarity. Experiments on various datasets with different characteristics have demonstrated that the average time complexity of the proposed algorithm is improved by 89.76%, and the accuracy is improved by 5 mm on average than the Super 4-Points Congruent Sets algorithm. More encouragingly, the experimental results show that the proposed algorithm can be applied to various restrictive cases, such as few overlapping regions and massive noise. Therefore, the algorithm proposed in this paper is a faster and more robust method than Super 4-Points Congruent Sets under the guarantee of the promised quality.

**Keywords:** global registration; boundary segmentation; overlapping regions; volumetric information



**Citation:** Li, S.; Lu, R.; Liu, J.; Guo, L. Super Edge 4-Points Congruent Sets-Based Point Cloud Global Registration. *Remote Sens.* **2021**, *13*, 3210. <https://doi.org/10.3390/rs13163210>

Academic Editor: Joaquín Martínez-Sánchez

Received: 3 July 2021

Accepted: 10 August 2021

Published: 13 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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

Point cloud registration is a fundamental task that aims to calculate the optimal full scene coverage between the multiple scans from limited viewpoints by estimating the transformation parameters, that is, rotation matrix and translation vector [1]. Researchers developed the combination of fine alignment as typified by Iterative Closest Point (ICP) [1] and coarse alignment represented by the 4-Points Congruent Sets (4PCS) algorithm [2]. Some tools such as Vercator Cloud and CloudCompare also follow the same coarse-to-fine registration strategy. However, limited to the initial approaching pose conditions of the ICP algorithm, it is essential to afford accurate initial values provided by the coarse alignment, which can accelerate the convergence of the fine alignment to a global minimum.

Therefore, coarse registration is valuable in many scenarios [3]. Specifically, the coarse registration intends to estimate an optimal rigid transformation matrix to align the source point cloud  $P$  to target point cloud  $Q$ . The state-of-the-art methods of class 4PCS, including

Super 4PCS [3], V4PCS [4] adopt a completely different strategy from RANSAC [5] for point cloud registration. These algorithms employ the four-point registration bases combined with the invariants in the rigid transformation to perform registration. However, this kind of methods are still time-consuming due to the lack of reduction in data scale and the existence of many false corresponding pairs. Furthermore, the methods of class 4PCS do not reveal good adaptability to scenes with restrictive conditions like few overlapping regions.

To tackle the challenges, this paper proposes a more efficient global registration algorithm called Super Edge 4PCS. Benefiting from the dramatic reduction of the scale of point clouds by boundary segmentation and overlapping regions extraction, the proposed algorithm is more efficient than the popular Super 4PCS algorithm. Besides, the noise elimination and the robustness to the overlap rates also ensure the broader application. Furthermore, the novel acquisition of corresponding bases does not only use affine invariants to screen the registration bases like the Super 4PCS. Instead, point cloud volumetric information can ensure to eliminate many unsuitable candidate registration bases. As a result, our proposed algorithm possesses the merit of more strong robustness and adaptability than the state-of-the-art Super 4PCS algorithm, as long as the points in the point cloud have the same distribution. The contributions of this paper are summarized as follows:

(1) The boundary information of point clouds is taken into account in the separated overlapping regions extraction, which supports the proposed method free from unnecessary internal points, and thus reduces the computational cost to an amazing 7.1% of the original compared with the state-of-the-art Super 4PCS.

(2) We propose a method of cross-selection of registration bases in overlapping regions. This method only uses the distance information of the point pairs to filter the registration base points. Therefore, a small number of accurate candidate registration base pairs can be quickly generated. Therefore, compared with the Super 4PCS, there is an 89.8% speedup overall without an increase in the registration error.

The rest of this paper is organized as follows: Section 2 reviews previous literature and background. Section 3 describes the principle of the Super Edge 4PCS algorithm. Both synthetic artifacts and large-scale building scans were tested in Section 4 to demonstrate the advantages of our framework. Finally, Section 5 makes a complete summary of the work. We exhibit the structure of our paper and the experiment part in Figures A1 and A2 in Appendix A.

## 2. Related Work

Point cloud registration is divided into pairwise registration and multi-cloud registration [6]. The latter is based on pairwise registration and assisted by global optimization methods such as graph optimization. Therefore, this paper focuses on pairwise matching, which primarily impacts the effectiveness of the multi-cloud registration [6]. Pairwise registration includes fine registration and coarse registration. The result of coarse registration is taken as the initial value of fine registration to make a more accurate estimation of transformation parameters [7–9]. The following texts provide the review of fine registration and coarse registration, respectively.

### 2.1. Fine Registration

The most representative method in fine algorithms is the Iterative Closest Points (ICP) algorithm [7,10] which mainly includes two steps: searching corresponding points in two point clouds and calculating the optimal transformation parameters iteratively [6,11]. ICP algorithm does not need any prior knowledge about the point cloud [12] but concentrates on seeking the point-to-point correspondences [13]. However, it also proposes the strict request to the point cloud data. The convergence accuracy of the ICP algorithm mainly depends on the ratio of overlapping regions [14,15]. Previous works stated that corresponding points were difficult to be extracted correctly if the overlapping ratio was under 50% [16,17]. Therefore, the ICP algorithm requires data to complete rough alignment in advance to avoid acquiring a less qualified initial value from the local optimization process. Therefore,

the ICP algorithm takes an excellent initial value as input [7,14,16]. Numerous researchers improved the ICP algorithm from different aspects [18–21] after Besl and McKay proposed the original ICP algorithm in 1992 [1]. For example, some methods change the objective function, and thus simplify the computation [11]. Chen et al. [22] used the distance from the point to plane instead of point to point in the original ICP algorithm. Although the generalization ability of the algorithm is improved, a lot of noise in the environment will affect the extraction of normal vectors, resulting in unsatisfactory results [23,24]. Segal [25] proposed a probabilistic version of the ICP algorithm called generalized ICP. This method draws the definition of a surface-to-surface distance. However, the application ability in outdoor scenes is also greatly affected by noise and outliers due to the employment of normal vectors [26,27]. Yang et al. [28] proposed the GO-ICP algorithm, which combined the ICP algorithm with the branch and bound method. This method performs well on small samples, but the computational complexity becomes large once the overlap rate is less than 70%. [29]. Among these improved objective functions, correspondences from points to planes hold the highest convergence accuracy [30,31]. Despite the improved performance, the convergence guarantee of the above methods can only take effect after receiving a good initial value from coarse registration.

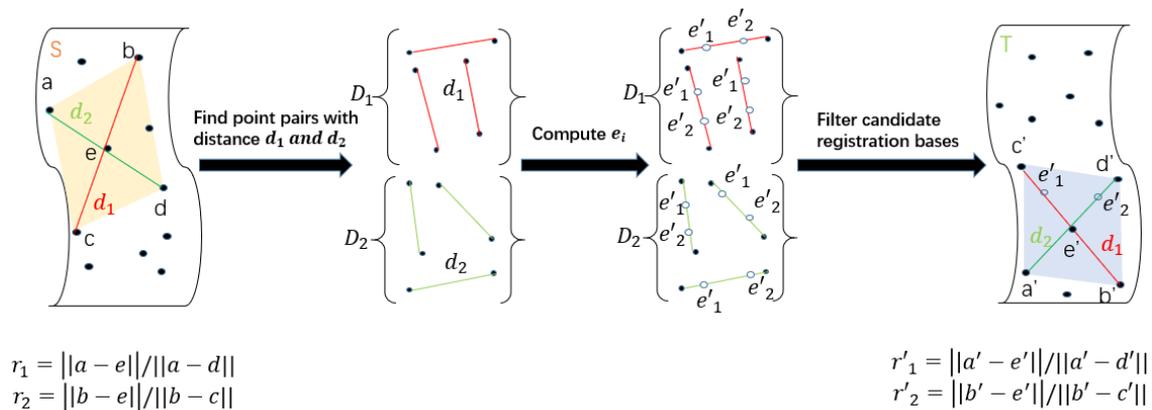
## 2.2. Coarse Registration

Coarse registration starts with arbitrary initial poses and is divided into feature-based methods and non-feature-based methods [32]. The feature-based methods are mainly divided into two steps: features extraction and calculation of transformation parameters [33,34]. Features used in the coarse registration include but not limited to point features, line features, surface block features, specific models, structural features, and higher-level shape descriptors [18]. For the point features part, the most famous one is the 3D SIFT keypoints extended from 2D SIFT [35]. The authors considered the points curvature rather than the pixels size. Other 3D keypoints include FAST (Features From Accelerated Segment Test) [36], SURF (SpeededUp Robust Features) [37], ORB (oriented fast and rotated brain) [38]. Without exception, these point features are sensitive to the noise and occlusion of the scene. Instead of point features, Yang and Zhang (2014) [39] proposed to use spatial curves as registration primitives. They extracted crest lines by clustering visual salient points selected according to geometric curvature. However, it is challenging to extract crest lines in the nature scene with less obvious curvature change. Dold and Brenner (2004) pointed out that the transformation parameters between two point clouds can be calculated according to three corresponding plane pairs [40]. Therefore, they then proposed to extract the largest plane block from the two sets of scans respectively and constructed the registration triplet to calculate the rotation and translation parameters [41]. The above feature-based methods put forward strict requirements for the environment. Especially for line features and surface features, these methods often require the environment to provide rich line targets and smooth planes to provide enough candidates for registration [7,42]. Although some methods based on intensity value relax the requirements of the environment [43], the intensity information also increases the amount of data to be processed. Moreover, the environment cannot contain too much noise interfering with features extraction [42,44].

RANSAC and 4PCS are considered as the representations of non-feature-based algorithms. The RANSAC framework [5] forms a registration base-pair, including three points randomly selected from the source and the target point cloud. The 4PCS algorithm improves the RANSAC framework from two aspects. The 4PCS algorithm is shown in Figure 1. Firstly, it uses two four-point coplanar bases to construct congruent candidate sets instead of two sets of triplets in RANSAC [45]. Secondly, the invariant of intersection ratio in affine transformation is utilized to filter congruent candidate sets. We can summarize the registration process as the following five steps:

- (1) Select the coplanarity registration base  $b_1$  in the source point cloud  $S$ . Calculate the distance among the point pairs,  $d_1, d_2$ , and the affine invariant,  $r_1, r_2$ .

- (2) Obtain two sets of point pairs whose lengths are  $d_1$  and  $d_2$  in the target point cloud  $T$  according to the distance between the point pairs in  $b_1$ .
- (3) Calculate the intersection points  $e'_i$  of point pairs based on affine invariants  $r_1$  and  $r_2$ .
- (4) If the intersection points of two point pairs are coincident, then these two point pairs are selected as the registration base  $b_2$ .
- (5) Use  $b_1$  and  $b_2$  to calculate the rigid transformation parameters  $T$  and return it.



**Figure 1.** The flowchart of 4PCS. Given the registration base composed of four coplanar points  $a, b, c,$  and  $d$  in source point cloud  $S$ , calculate the intersection point  $e$ , affine invariants  $r_1, r_2$  and the distance  $d_1$  and  $d_2$  between diagonal points. Then point pairs extracting in target point cloud  $T$  and the intersection points  $e'_i$  calculating according to  $r_i$  are performed in turn where  $D_1$  and  $D_2$  are point pair sets with distance  $d_1$  and  $d_2$  among point pairs, respectively. Finally, construct the target registration base pair according to the intersection point  $e'_i$ .

The above adjustments reduce the time complexity of the algorithm from  $O(n^3)$  of RANSAC to  $O(n^2 + k)$ , where  $n$  is the number of points in the point cloud and  $k$  is the number of candidate registration bases. The theory of rigid transformation used in the 4PCS method includes distances  $d_i$  among diagonal points  $a, d$  and  $b, c$ , and affine invariants  $r_i$  which is shown in Equation (1) where  $a, b, c,$  and  $d$  are the four points of the registration base,  $e$  is the intersection point of line segments  $ad$  and  $bc$ . The affine invariant  $r_i$  defined in 4PCS remains preserved under any affine transformation, thus it is afforded to filter the ambiguities and unnecessary registration bases.

$$d_1 = \|a - d\| \quad d_2 = \|b - c\| \quad r_1 = \frac{\|a - e\|}{\|a - d\|} \quad r_2 = \frac{\|b - e\|}{\|b - c\|} \quad (1)$$

However, the 4PCS algorithm has three drawbacks. Firstly, the rigid transformation invariants often require finding out all the point pairs satisfying the distance constraint effectively. Secondly, a large number of candidate registration bases thus are generated since only one affine invariant is used as the filter condition. Therefore, it is necessary to find out other conditions. Thirdly, the 4PCS algorithm cannot be widely used in scenes with a vast number of points because a large number of points call for efficient filter techniques. To deal with the first two challenges, Nicolas and Miley [3] proposed the Super 4PCS algorithm in 2014. They employed a rasterization approach to extract point pairs with a fixed distance. The rasterization method simply rasterizes a sphere of radius  $r$  centered at the interest point and counts the points located at common regions between cells and sphere. Besides, the Super 4PCS algorithm takes the angle between point pairs as the filter condition, further reducing the search space. The time complexity of this method is reduced to  $O(n + k + c)$ , where  $n$  is the number of points,  $k$  is the number of reported pairs, and  $c$  is the number of extracted candidate bases. However, the Super 4PCS algorithm can not show great performance when dealing with the registration between partially overlapped point clouds [46]. Besides, noise and overlapping rates have an undeniable influence on the registration results of the Super 4PCS algorithm, which is confirmed in the later

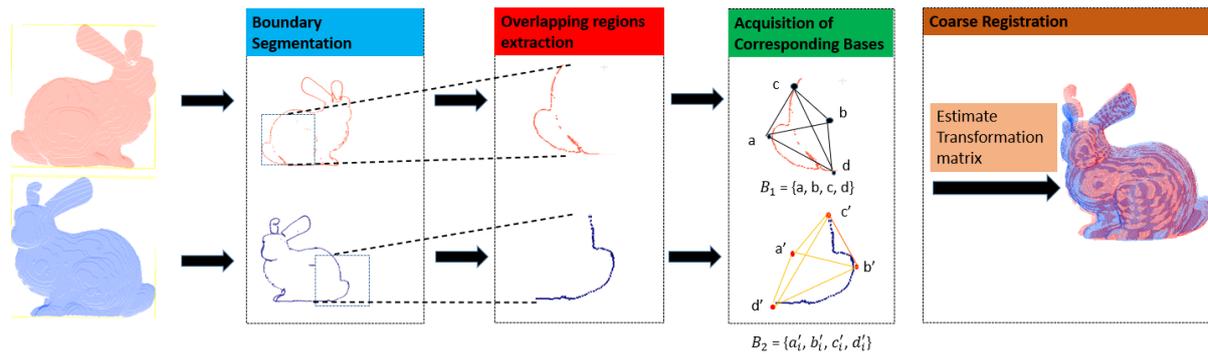
experiments part. Therefore, some developments that are proposed in recent years [5,45,46]. Generalized 4PCS [47] and Super Generalized 4PCS [48] relaxed the coplanar constraint and considered more abundant geometric information, such as the intersection distance of line segments in different planes. Huang et al. [4] and Sun et al. [46] considered integrating volume information into the Super 4PCS algorithm to speed up the acquisition of congruent sets. They used the distance between selected points to filter point pairs. These improved strategies screen registration bases furthermore, but the input clouds are still not be thinned out, thus limits the application in large-scale scenes. To solve the third drawback, Theiler (2014) et al. [49] preprocessed the original data by extracting the keypoints, 3D SIFT and the 3D difference of Gaussian, thus significantly reducing the size of the point cloud in large-scale environments. Ge [50] used the extracted semantic feature points to form two groups of registration bases, which further reduced the scale of the point clouds. Xu et al. [32] investigated the embedment of sparse multiscale features to optimize the construction of registration bases obtained by 4PCS. As we reviewed in feature-based methods, the extraction of semantic feature points and other keypoints are always sensitive to the noise of environment despite cutting the cardinality of point clouds down.

This paper aims to design a new method, leading to fast acquisition of candidate registration bases set after effective overlapping regions extraction. The shape distribution and volume information of the point clouds are embedded in the framework, which help to shrink the search space. Experiments based on various datasets illustrate that this algorithm received satisfactory performance in different scenes, especially for the scene with strict constraints like few overlapping regions.

### 3. Super Edge 4-Points Congruent Sets

#### 3.1. Overview

This section mainly introduces three major steps of the proposed algorithm: Boundary Segmentation(BS), Overlapping Regions Extraction(ORE), and the Acquisition of Corresponding Bases(ACB). We observed that accurate registration can still be finished even if points within the surface were excluded and kept only boundary points. In other words, we can only rely on the boundary points to complete the registration. Figure 2 depicts the framework of the proposed algorithm whose pseudocode is given in Algorithm 1. It can be seen from Figure 2 that the Super Edge 4PCS algorithm takes the source point cloud and the target point cloud as inputs, and finally outputs the estimated rigid transformation parameters. We firstly perform the boundary segmentation on original point clouds by calculating the angle between the projection vectors. Similar to the space division in the rasterization method, the overlapping regions  $S''$  and  $T''$  are obtained by continuously cutting the boundary point clouds. After extracting the overlapping regions, we can heuristically obtain the corresponding registration base set  $S_{base}$  with the consideration of volumetric information. For each pair of registration bases in  $S_{base}$ , we need to calculate the rigid transformation matrix  $T_i$  and the corresponding  $RMSE_i$ . If the  $RMSE_i$  decreases, update  $T$  to the current  $T_i$ , and  $M$  to the current  $RMSE_i$ . Finally,  $T$  and  $M$  are the rigid transformation matrix and RMSE for point cloud registration. Super Edge 4PCS algorithm uses an identity matrix as the initial value of transformation matrix. The values and meanings of the parameters involved in the following steps will be described in Section 3.3.2 especially. More details of each step are provided in the following sections.



**Figure 2.** The flowchart of the Super Edge 4PCS. The original point clouds are taken as the inputs. After boundary segmentation, overlapping regions extraction, corresponding bases acquisition, we can obtain registration base pair set  $S_{base}$  including base  $B_1$  and  $B_2$ . The transformation matrix is estimated from  $S_{base}$ , then the final transformed point cloud is outputed.

---

### Algorithm 1 the Super Edge 4PCS.

---

**Input:** Source and target point sets  $S$  and  $T$ ;

**Output:** The optimal transformation matrix and the minimum RMSE;

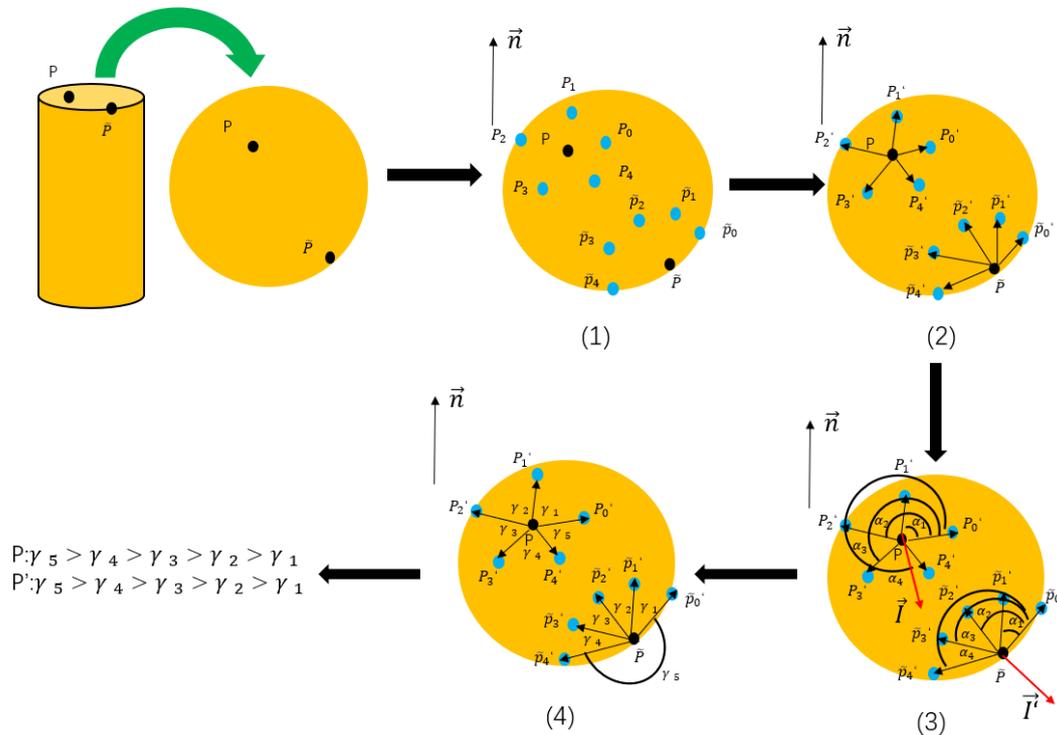
- 1: //Boundary Segmentation(BS);
  - 2: Estimate the normal of the surface;
  - 3: Calculate the angle between mapping vectors;
  - 4: Extract boundary points  $S'$  and  $T'$ ;
  - 5: //Overlapping Regions Extraction(ORE);
  - 6: Perform local extraction on two boundary point clouds  $S'$  and  $T'$
  - 7: Perform global extraction on  $S'$  and  $T'$
  - 8: Get overlapping region  $S''$  and  $T''$
  - 9: //Acquisition of corresponding base(ACB);
  - 10: obtain base set  $S_{base} = \{(b_1, b'_1), (b_2, b'_2), (b_3, b'_3), \dots\}$
  - 11:  $T$ : the best rigid transform whose initial value is identity matrix  $I_T$ .
  - 12:  $M$ : the minimum of all RMSE whose initial value is 10,000
  - 13: **for all** base pair  $(b_i, b'_i) \in S_{base}$  **do**
  - 14:     The transformation parameters and RMSE between two registration bases are calculated.
  - 15:      $T_i \leftarrow \text{rigid transform}$
  - 16:      $M_i \leftarrow \text{RMSE}_i$
  - 17:     **if**  $(M_i \leq M)$  **then**
  - 18:          $T \leftarrow T_i$
  - 19:          $M \leftarrow M_i$
  - 20:     **end if**
  - 21: **end for**
  - 22: **return**  $T$  and  $M$ ;
- 

### 3.2. Algorithm Steps

#### 3.2.1. Boundary Segmentation (BS)

The first step is to perform boundary segmentation(BS), which aims to remove unnecessary internal points of original two point clouds and only retain the boundary points. The input of this step is original point clouds, and the output is boundary points set. In this paper, K-nearest neighbor search and R-nearest neighbor search are set to determine the search neighborhood. Specifically, the K search mode searches k nearest neighbor points of interest points  $P$  to obtain adjacent points set  $p_k$ . By contrast, the R search mode explores the adjacent points of  $P$  within the radius  $r$ . The core idea of boundary segmentation is to take the angle between vectors as the evaluation metric. If the reference point  $P$  is a boundary point, the maximum angle between projection vectors must be larger than the threshold  $h$ , which is an angle threshold used to measure the angle between the mapping vectors. The flowchart of point cloud boundary segmentation is shown in Figure 3, whose

framework is shown in Algorithm 2. The specific process of detecting boundary points is shown below:



**Figure 3.** The flowchart of boundary segmentation.  $P$  and  $\tilde{P}$  are interest points,  $P_0, \dots, P_4, \tilde{P}_0, \dots, \tilde{P}_4$  are neighbor points of interest points  $P$  and  $\tilde{P}$ , respectively.  $P'_0, \dots, P'_4$  and  $\tilde{P}'_0, \dots, \tilde{P}'_4$  are the mapping points of  $P_0, \dots, P_4, \tilde{P}_0, \dots, \tilde{P}_4$  on the tangent plane  $\pi$  of point  $P$ .  $\vec{n}$  is the normal vector of the tangent plane  $\pi$ .  $\alpha_1, \dots, \alpha_4$  are the angles between the vectors  $PP_i$  and  $PP'_0$ , and  $\gamma_1, \dots, \gamma_5$  are the angles between two adjacent mapping vectors.

- (1) Set the parameters of BS. First, select one search mode. Here we take K-nearest neighbor search as an example. Next, set the neighborhood range of normal vector  $N_V$  and neighborhood range of reference point  $N_B$ . Here  $N_V$  and  $N_B$  are set to  $k$ . Then the adjacent points around reference point  $p$  can be obtained as  $p_k = \{p_0, p_1, \dots, p_{k-1}\}$  within the neighborhood  $N_B$  of point  $P$ . The boundary segmentation algorithm uses the tangent plane of point  $P$  as the fitting plane  $\pi$ . The corresponding normal vector  $\vec{n}$  can also be obtained within the neighborhood range of  $N_V$ . Set the angle threshold  $h$  as  $\frac{\pi}{2}$ . If the angles between vectors are greater than  $h$ , the reference point corresponding to the angle is regarded as the boundary point. Last, set boundary points set as an empty set  $B$ .
- (2) The mapping points of adjacent points set  $p'_k = \{p'_0, p'_1, \dots, p'_{k-1}\}$  on the plane  $\pi$  can be obtained by calculating the intersection points between the plane and the straight line started with  $p_k$  whose direction is parallel to the normal vector  $\vec{n}$ . After obtaining the mapping points  $p'_k$ , mapping vectors set  $\vec{pp}'_i = \{pp'_0, pp'_1, \dots, pp'_{k-1}\}$  can be gained.
- (3) Solve the angles between starting vector and the rest mapping vectors:
  - (3.1) Take  $p$  as the starting point and all the points in  $p'_k$  set as the ending point to create  $K$  vectors. Randomly select one of the vectors as the starting vector, we select  $\vec{pp}'_0$  as the starting vector here. Calculate the cross product of the vector  $\vec{pp}'_0$  and the plane normal vector as vector  $\vec{I}$ .
  - (3.2) Calculate the angles  $\alpha_i$  between mapping vector  $\vec{pp}'_i (i = 1, 2, 3, \dots, k - 1)$  and vector  $\vec{pp}'_0$ . The angles  $\beta_i$  between  $\vec{pp}'_i$  and vector  $\vec{I}$  are also calculated aiming to ensure  $\alpha_i < \alpha_{i+1}$ , that is, If  $\beta_i > 90^\circ$ ,  $\alpha_i = 360 - \alpha_i$ .

- (4) Solve the angles between  $\vec{pp}'_i$  and  $\vec{pp}'_{i+1}$ :
- (4.1) Arrange  $\alpha_i$  calculated in (3.2) in ascending order to get  $S' = (\alpha'_1, \alpha'_2, \dots, \alpha'_{k-1})$ . Calculate the angles  $\gamma_i$  between  $\vec{pp}'_i$  and  $\vec{pp}'_{i+1}$ .
- $$\gamma_i = \begin{cases} \alpha'_i, & i = 1 \\ \alpha'_i - \alpha'_{i-1}, & i = 2, 3, \dots, k-2, k-1 \\ 360 - \alpha'_{k-1}, & i = k \end{cases} \quad (2)$$
- (4.2) The maximum value of  $\gamma_i$  is compared with the threshold value  $h$ . If it is greater than  $h$ , point  $P$  is added to the boundary points set  $B$ .
- (5) return boundary points set  $B$ .

---

**Algorithm 2** Boundary Segmentation.
 

---

**Input:** Point cloud  $S$ ;

**Output:** The boundary points set  $B$  of point cloud  $S$ ;

- 1: Set the boundary point set  $B$  to be empty
- 2: **for all** point  $P \in S$  **do**
- 3:   Set the search mode  $M$ , here is  $k$  nearest neighbor search.
- 4:   Set the neighborhood range of normal vector  $N_V$  and neighborhood range of reference point  $N_B$ . Here  $N_V$  and  $N_B$  are set to  $k$
- 5:   Set the neighborhood  $p_k = \{p_0, p_1, \dots, p_{k-1}\}$  of the reference point  $P$  to obtain the neighborhood points set
- 6:   Set the angle threshold  $h$  of  $P$ ;
- 7:   Get the tangent plane  $\pi$  of  $P$ ;
- 8:   Calculate the normal vector  $\vec{n}$  of  $\pi$  based on  $N_V$ ;
- 9:   Projected the neighborhood points set  $p_k = \{p_0, p_1, \dots, p_{k-1}\}$  to the plane  $\pi$ ;
- 10:   Select the starting vector  $PP'_0$  and cross-multiply the plane normal vector  $\vec{n}$  to obtain the vector  $\vec{I}$ ;
- 11:   Calculate the angle  $\alpha_i$  between  $\vec{pp}'_i$  and  $\vec{pp}'_0$  and the angle  $\beta_i$  between  $\vec{pp}'_i$  and  $\vec{I}$ , ( $i = 1, 2, 3, \dots, k-1$ );
- 12:   **if**  $\beta_i \leq 90^\circ$  **then**
- 13:      $\alpha_i = 360^\circ - \alpha_i$
- 14:   **end if**
- 15:   Sort  $\alpha_i$  in ascending order recorded as  $\alpha'_i$ ;
- 16:   Calculate the angle between adjacent vectors  $\gamma_i$ ;

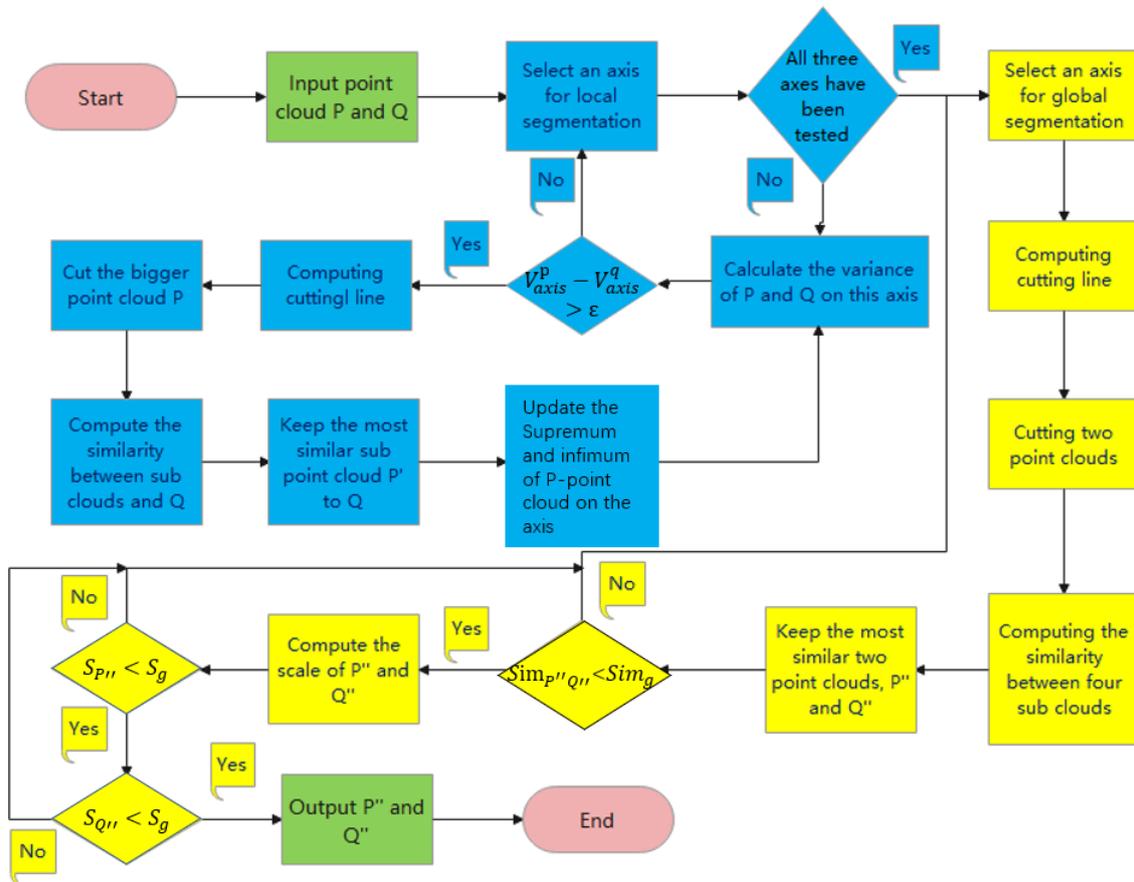
$$\gamma_i = \begin{cases} \alpha'_i, & i = 1 \\ \alpha'_i - \alpha'_{i-1}, & i = 2, 3, \dots, k-2, k-1 \\ 360 - \alpha'_{k-1}, & i = k \end{cases} \quad (3)$$

- 17:   Get the biggest  $\gamma_{max}$ ;
  - 18:   **if**  $\gamma_{max} > h$  **then**
  - 19:     append point  $P$  to  $B$
  - 20:   **end if**
  - 21: **end for**
  - 22: **return**  $B$ ;
- 

### 3.2.2. Overlapping Regions Extraction(ORE)

The registration process can be streamlined if the search process of the corresponding registration bases can be constrained to the overlapping regions. ORE aims to get the overlapping regions, thus further reduce the scale of boundary points. ORE takes the boundary points  $S', T'$  obtained in BS as the input and outputs the overlapping regions  $S'', T''$  of two boundary clouds. Figure 4 shows the complete flowchart of the ORE method, whose pseudocode is shown in Algorithm 3. We can summarize the entire extraction process into two main steps: local extraction and global extraction. In the local extraction, the cutting object is the point cloud with a larger size. Here is  $S'$ . We perform a continuous

division along the X-Y-Z axis, aiming to make the size of the larger point cloud close to the smaller one, thus obtain the divided point cloud  $S'$ . After local extraction, we cut the point clouds  $S'$  and  $T'$  simultaneously, which is global extraction. A circular cutting is executed until the similarity between the sub point clouds is less than the threshold  $Sim_g$ . Finally, we get the overlapping regions by reserving similar sub point clouds. Compared with the original point clouds, the scale of point cloud after BS has significantly been reduced. Coupled with the exponential velocity of ORE, the cardinality of the point cloud is further reduced.



**Figure 4.** Overlapping regions extraction. Blue part represents local segmentation, yellow part represents global segmentation and green part represents the input and output of the method. It finally outputs two divided sub point clouds as the overlapping regions.  $V_{axis}^p$  and  $V_{axis}^q$  represent the variance of  $P$  point cloud and  $Q$  point cloud on three coordinate axis;  $Sim_{P''Q''}$  represents the similarity of the overlapping regions  $P''$  and  $Q''$ ;  $Sim_g$  represents the similarity threshold for stopping segmentation;  $S_{P''}$  and  $S_{Q''}$  represent the scale of overlapping regions  $P''$  and  $Q''$ ;  $S_g$  represents the scale threshold to stop segmentation.

Next, we focus on the local extraction and global extraction in Figure 4. The 2D demonstration of local extraction is visually shown in Figure 5. We describe the process of local extraction as follows, which corresponding to the local overlapping regions extraction part of Algorithm 3:

(1) This step starts by calculating the variance of the two point clouds  $V_x^s, V_y^s, V_z^s, V_x^t, V_y^t, V_z^t$  on the x-axis, y-axis, and z-axis, respectively. Suppose each point cloud has  $n$  points,  $(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)$ . So  $V_x^s = ((x_1 - \bar{x}) + (x_2 - \bar{x}) + \dots + (x_n - \bar{x}))/n$ . The calculation method of  $V_y^s$  and  $V_z^s$  is the same as  $V_x^s$ , except that the calculation objects are replaced with y-axis and z-axis. Then, the differences between the corresponding variance are calculated as  $D_x, D_y, D_z$ , where  $D_x = V_x^s - V_x^t, D_y = V_y^s - V_y^t, D_z = V_z^s - V_z^t$ . Select the coordinate axis where the largest one of  $D_x, D_y, D_z$  is located as the cutting direction. Take the x-axis as an example here. The Supremum and inimum of the coordinates of the point

cloud in the x-axis are denoted as  $x_{sup}$  and  $x_{inf}$ , respectively. Then the cutting line  $x_{cutting}$  is calculated as  $x_{cutting} = 1/2(x_{sup} + x_{inf})$ .

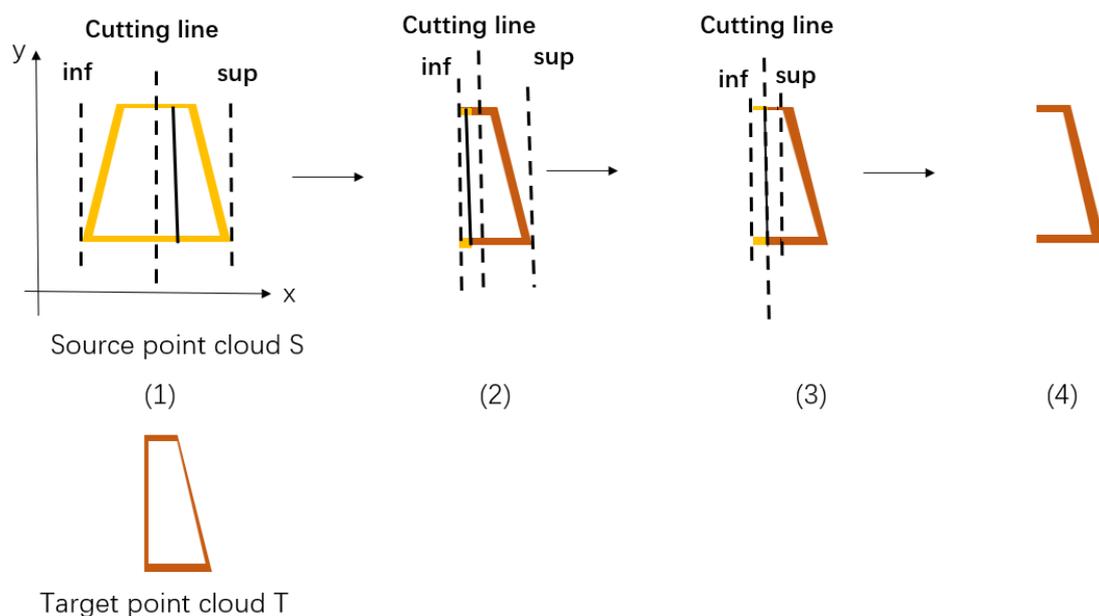
(2) We split the original cloud in two subparts along the cutting line. Then we keep the point cloud which is more similar to the target point cloud  $T$ . The method of comparing similarity between point clouds will be described in the end of Section 3.2.2. Next, we need to update the  $x_{sup}$  and  $x_{inf}$ . If the variance of divided point clouds in x-axis direction is greater than that of target point cloud  $T$ , we increase infimum  $x_{inf}$  to  $x_{cutting}$ . Otherwise, the supremum  $x_{sup}$  is decreased to  $x_{cutting}$ . Then we recalculate the cutting line  $x_{cutting}$  according to the updated  $x_{sup}$  and  $x_{inf}$ .

(3) Repeat steps (1) and (2) until the variances in all directions are equal to the target point cloud  $T'$ . This step outputs target point cloud  $T'$  and divided source point cloud  $S''$ .

After local extraction, we need to perform the global extraction of Algorithm 3. Figure 6 describes the process of global extraction.  $ESF_i$  in Figure 6 represents the global features of the point cloud. The part colored in red is the sub-point cloud that is reserved for further extraction. The specific extraction process is shown below.

- (1) Select one axis to divide two point clouds.
- (2) Split the point cloud  $S''$  and  $T'$  along one selected axis of  $X, Y, Z$  according to the cutting line whose calculation method is the same as local extraction.
- (3) Calculate the ESF of four divided point clouds as  $ESF_i$
- (4) Preserve the most similar two point clouds according to the distance between  $ESF_i$
- (5) Judge whether the scale of selected divided point clouds arrives at the indivisible level  $S_g$ . If true, skip to step (7).
- (6) if false, return to step (2).
- (7) Judge whether the similarity between two selected point clouds meets the constraint  $Sim_g$ . If true, skip to step (9).
- (8) If false, return to step (1).
- (9) Judge whether we have selected all three axes. If true, skip to step (11).
- (10) If false, return to step (1)
- (11) Output the final selected sub-point cloud  $S''$  and  $T''$ .

Where  $S_g$  is calculated as the ratio of the number of points in overlapping regions to original point clouds. Finally, ORE takes the final sub point clouds  $S''$  and  $T''$  as the overlapping regions.



**Figure 5.** Local overlapping regions extraction of the larger point cloud. This figure employs two-dimensional object segmentation as an example to show the process of local overlapping regions extraction. The three-dimensional situation can be derived accordingly. The cutting line approach the ground-truth segmentation axis by updating the supremum and infimum of cutting direction where  $inf$  and  $sup$  represent the upper and lower bounds of the point cloud S on the x-axis.

The calculation of similarity between two point clouds is based on the global features extracted by the Ensemble of shape functions(ESF). As a global feature describing the shape of a three-dimensional(3D) model of an object, ESF was first proposed by Osada et al. [51]. The ESF is a 640-dimensional vector, which includes three aspects: distance, angle, and area [52]. Its basic principle is converting the problem of shape matching to comparing probability distributions and using three indicators to summarize the object's geometric information. The ESF feature maintains a strong invariance to any rotation and translation of the point cloud. Therefore, the similarity of the two point clouds is obtained by calculating the L2 distance between two ESF features.

---

**Algorithm 3** Overlapping Regions Extraction(ORE)
 

---

**Input:** Two boundary point clouds  $S'$  and  $T'$ ;

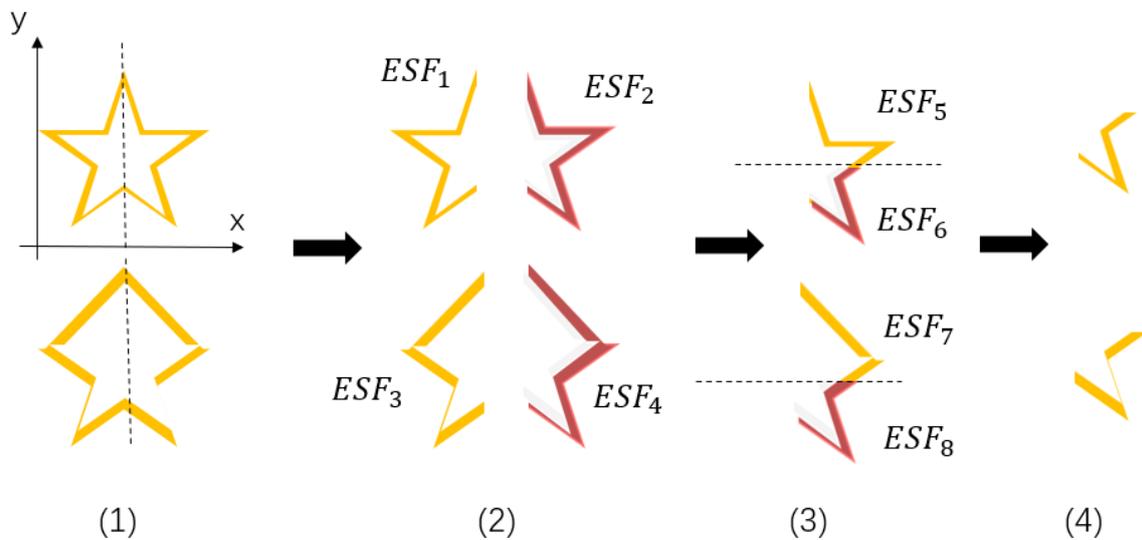
**Output:** Two overlapping regions  $S''$  and  $T''$

```

1: // Local overlapping regions extraction
2: for all axis in {x, y, z} do
3:   Get the variance of the two point clouds on x-axis,  $V_{axis}^{s'}$   $V_{axis}^{t'}$ .
4:   Set the tolerance of variance  $\epsilon_{axis}$ 
5:   while  $V_{axis}^{s'} - V_{axis}^{t'} \geq \epsilon_{axis}$  do
6:     Computing cutting line  $Axis_{cutting} = 1/2(axis_{sup} + axis_{inf})$  where  $axis_{sup}$  and  $axis_{inf}$  is the supremum and
       infimum of the point cloud with larger variance on the axis.
7:     Cut the point cloud with bigger variance
8:     Calculate the similarity between the two sub-point clouds after segmentation and the unsegmented point
       cloud,  $Sim_1$  and  $Sim_2$ 
9:     Keep sub-point clouds with highest similarity  $S'$ 
10:    if  $V_{axis}^{s''} \geq V_{axis}^{t''}$  then
11:       $axis_{inf} = Axis_{cutting}$ 
12:    else
13:       $axis_{sup} = Axis_{cutting}$ 
14:    end if
15:  end while
16: end for
17: // Global overlapping regions extraction
18: Set stop = False
19: Set point cloud similarity for global segmentation  $Sim_g$ 
20: Set point cloud scale constraints for global segmentation  $S_g$ 
21: while true do
22:   for all axis in x, y, z do
23:     Computing cutting lines  $axis_{cutting}^{s'} = 1/2(axis_{sup} + axis_{inf})$ ,  $axis_{cutting}^{t'} = 1/2(axis_{sup} + axis_{inf})$ 
24:     Cut two point clouds
25:     Calculate the similarity between the four point clouds after segmentation  $Sim_i$ 
26:     Keep sub-point clouds with highest similarity,  $S''$  and  $T''$ 
27:     if  $Sim_{st} > Sim_g$  and  $\frac{S_{s''}}{S_{s'}} < S_g$  and  $\frac{S_{t''}}{S_{t'}} < S_g$  then
28:       stop = True
29:       break
30:     end if
31:     if stop = True then
32:       break
33:     end if
34:   end for
35: end while

```

---



**Figure 6.** Global overlapping regions extraction. Here, two two-dimensional objects with different shapes are used to represent the source point cloud and the target point cloud. Segment two point clouds simultaneously, and keep the sub-point clouds with greater similarity.  $ESF_i$  in the figure represents the ESF feature of point clouds.

### 3.2.3. Acquisition of Corresponding Base (ACB)

After obtaining the overlapping regions of initial point clouds, we can easily select the appropriate registration base in the overlapping regions to get the optimal rigid transformation matrix. Theoretically, the following meaningful discussions for selecting corresponding registration base pairs can be obtained after summarizing the literature based on the 4PCS algorithm: First, the wide registration bases composed of four points away from each other are stable and desirable [49]. Next, the utilization of registration bases composed of four noncoplanar points can reduce the search space significantly [53]. Our bases selection strategy built on the above discussions.

In this step, we take the overlapping regions  $S''$  and  $T''$  obtained in ORE step as the input, and outputs the candidate registration base pairs set  $S_{base}$ . Algorithm 4 shows the framework of ACB. Two parameters need to be set in ACB:  $baseDis$  and  $fDis$ . The former is the distance error tolerance of the first three points in the registration base, and the latter is that of the fourth point. We summarize the ACB process as follows:

- (1) Calculate the centroids  $C$  and  $C'$  of two obtained overlapping regions  $S''$  and  $T''$ .
- (2) Search for the points farthest from centroids among the remaining points marked as  $A$  and  $A'$ .
- (3) Judge whether the length difference between  $AC$  and  $A'C'$  is less than  $baseDis$ .
- (4) If true,  $S_{A'} = \{A'\}$ ,  $S_A = \{A\}$  then skip to (6).
- (5) If false, keep the shorter one. Here take  $AC$  as example. Then re-find the point  $A'$  in  $T''$  so that the length difference between  $AC$  and  $A'C'$  is less than  $baseDis$ . The candidate set of point  $A'$  is  $S_{A'} = \{A' \mid \|AC - A'C'\| \leq baseDis\}$ ,  $S_A = \{A\}$
- (6) Search for the point  $B$  that is the farthest from the point  $A$  in  $S_A$  to form point pairs set  $S_{AB}$ .  $S_{AB} = \{(A, B) \mid A \in S_A, B \text{ is the farthest point from } A\}$ . Choose the largest distance among  $S_{AB}$  and mark it as  $AB$
- (7) Search for the point  $B'$  that is the farthest from the point  $A'$  in  $S_{A'}$  to form point pairs set  $S_{A'B'}$ .  $S_{A'B'} = \{(A', B') \mid A' \in S_{A'}, B' \text{ is the farthest point from } A'\}$ . Choose the largest distance among  $S_{A'B'}$  and mark it as  $A'B'$
- (8) Judge whether the length difference between  $AB$  and  $A'B'$  is less than  $baseDis$
- (9) If true,  $S_{A'} = \{A'\}$ ,  $S_A = \{A\}$ ,  $S_B = \{B\}$ ,  $S_{B'} = \{B'\}$ ,  $S_{AB} = \{AB\}$ ,  $S_{A'B'} = \{A'B'\}$  then skip to (15).
- (10) If false, keep the shorter one.

(11) If the shorter one is  $A'B'$ ,  $S_{A'} = \{A'\}$ ,  $S_{B'} = \{B'\}$ ,  $S_{A'B'} = \{A'B'\}$ . Then re-search for  $B$  in  $S''$  so that the length difference between  $AB$  and  $A'B'$  is less than  $baseDis$ . The candidate point set of  $B$  is  $S_B = \{B \mid \|AB - A'B'\| \leq baseDis\}$

(12) Update the  $S_B$  with the points in old  $S_B$  which meets the length differences between  $B'C'$  and  $BC$  are less than  $baseDis$ .  $S_{AB} = \{(A, B) \mid A \in S_{A'}, B \in S_B\}$  Skip to (15).

(13) If the shorter one is  $AB$ , search for the point pair  $A'B'$  in the set  $S_{A'B'}$  so that the length difference between  $A'B'$  and  $AB$  is less than  $baseDis$ . Update  $S_{A'B'}$  with the set of new point pairs  $A'B'$ ,  $S_{AB} = \{(A, B)\}$ ,  $S_{A'B'} = \{(A', B'_1), (A', B'_2), \dots\}$ .

(14) Choose  $A'B'$  in the new  $S_{A'B'}$  so that the length difference between  $B'C'$  and  $BC$  is less than  $baseDis$ . Update  $S_{A'B'}$ .

(15) Randomly select a point  $F$  in  $S''$  as the fourth base point.  $S_F = \{F\}$

(16) Search for  $F'$  in  $T''$  so that the length difference between  $AF$  and  $A'F'$ ,  $BF$  and  $B'F'$ ,  $CF$ , and  $C'F'$  are all less than  $fDis$ . The candidate set of point  $F'$  is  $S_{F'} = \{F' \mid \|AF - A'F'\| \leq fDis, \|BF - B'F'\| \leq fDis, \|CF - C'F'\| \leq fDis\}$

(17) Build registration base pairs:  $S_{base} = \{(b_1, b_2) \mid b_1 = (A, B, C, F), A, B \in S_{AB}, F \in S_F; b_2 = (A', B', C', F'), A', B' \in S_{A'B'}, F' \in S_{F'}\}$

---

#### Algorithm 4 Acquisition of Corresponding Bases(ACB)

---

**Input:** Two overlapping regions  $S''$  and  $T''$ ;

**Output:** Corresponding base set  $S_{base}$ ;

- 1: Get centroid  $C$  and  $C'$  of overlapping regions  $S''$  and  $T''$ ;
  - 2: Set the distance error tolerance of the first three base points:  $baseDis$
  - 3: Get the points  $A$  and  $A'$  that are farthest from the centroid  $C$  and  $C'$ ;
  - 4: **if**  $\frac{\|AC - A'C'\|}{AC} \leq baseDis$  **then**
  - 5:     **continue**
  - 6: **else**
  - 7:     Retain the smaller distance such as  $AC$ ;
  - 8:     Search again in  $T''$  for the candidate point set  $S_{A'}$ . The distance error between  $A'C'$  and  $AC$  is within the range of  $baseDis$ ;
  - 9:     **end if**
  - 10: Get the candidate point set  $S_{B'}$  and  $S_B$  whose point is the farthest from  $A$  and  $A'$  like the way to get  $S_{A'}$ .
  - 11: Set the distance error tolerance of the fourth point:  $fDis$
  - 12: Randomly select the fourth point  $F$  or  $F'$  to candidate set  $S_D$  and  $S_{D'}$  within the tolerance of  $fDis$
  - 13: Construct corresponding bases.
  - 14: **return base set**  $S_{base} = \{(b_1, b'_1), (b_2, b'_2), (b_3, b'_3), \dots\}$ ;
- 

Pay more attention to the selection of the fourth point that is based on the following facts. The cardinality of the point cloud has been dramatically reduced benefiting from BS and ORE. Besides, many candidate points have been excluded because of the first free points  $A, B, C$ . For these two reasons and the sake of simplicity, we randomly choose points  $F$  or  $F'$  as the fourth point. Then we can validate the candidate bases set with RMSE. The transformation matrix corresponding to the minimum RMSE is preserved as the final rigid transformation parameter.

### 3.3. Algorithm Details

#### 3.3.1. Computational Complexity

Firstly, the time complexity of the BS is  $O(kn)$ , where  $k$  is the number of neighborhood points around the reference point, and  $n$  is the bigger number of points in the initial point clouds  $S$  and  $T$ . Since the ORE is a binary segmentation secondly, the spent time is mainly concentrated on the first segmentation. Therefore, assuming that the time taken for the first segmentation of the boundary point cloud obtained by the BS is  $R_1$  and  $R_2$ , respectively, the time complexity of ORE can be expressed as  $O(R_1 + R_2)$ . As for ACB, we suppose that the time required to traverse the overlapping regions  $S''$  and  $T''$  one time is  $m_1$  and  $m_2$ , respectively. The proposed algorithm takes four traversals to find the most

suitable four base points, so the time complexity is  $O(3m_1 + 4m_2)$ , simplified to  $O(m)$  where  $m = \max\{m_1, m_2\}$ . In summary, the time complexity of the Super Edge 4PCS algorithm is  $O(n + R + c)$  where  $R = \max\{R_1, R_2\}$ ,  $m$  is omitted since it less than  $R$ , and  $c$  is the number of extracted registration base pairs.

### 3.3.2. Parameters in Super Edge 4PCS

The parameters used by the Super Edge 4PCS algorithm are shown in Table 1. A total of nine parameters need to be set.  $P_1$  is the search mode  $M$  in BS, which can be set to two values, R nearest neighbor search and K nearest neighbor search.  $P_2$  and  $P_3$  are the neighborhood range of normal vector and reference point, that is  $N_V$  and  $N_B$  in BS. We need to calculate the normal vector  $\vec{n}$  of the tangent plane of reference point  $P$  according to the  $N_V$  neighbor points. Besides,  $P_4$  is the angle threshold  $h$ . As long as the angle between vectors is greater than  $h$  in BS, the reference point  $P$  is reserved as a boundary point.  $P_5$  describes the distance tolerance of the variance of the two point clouds during local segmentation on each coordinate axis. If it is set to a small value, then local point cloud segmentation will be performed.  $P_6$  is the point cloud similarity constraint of global segmentation  $Sim_g$ . This parameter is measured by the distance between ESFs.  $P_7$  is the scale constraint of overlapping regions, which limit the scale of overlapping regions in ORE after global extraction. The value  $S_g$  is the ratio between the scale of overlapping regions to scale of original point clouds. We can stop cutting the point clouds only after the sub-point cloud satisfies the similarity constraint  $Sim_g$  and the scale constraint  $S_g$  simultaneously.  $P_8$  is the distance error tolerance  $baseDis$  of the first three base points in ACB. This value is calculated as  $P_8 = \text{abs}(AC-A'C')/\max(AC, A'C')$ .  $P_9$  is the distance error tolerance  $fDis$  of the fourth point in ACB. The value is calculated as follows:  $fDis = \text{abs}(AF-A'F')/\max(AF, A'F')$ . The above two parameters are utilized to construct candidate registration bases. The recommended value of these parameters will be given in Section 4.2.

**Table 1.** Algorithm parameters used in the Super Edge 4PCS algorithm.

Code	P1	P2	P3	P4	P5	P6	P7	P8	P9
parameters	$M$	$N_V$	$N_B$	$h$	$\epsilon_{axis}$	$Sim_g$	$S_g$	$baseDis$	$fDis$
Data type	bool	int	int	int	int	int	percentage	percentage	percentage

## 4. Results and Discussion

### 4.1. Data and Metrics

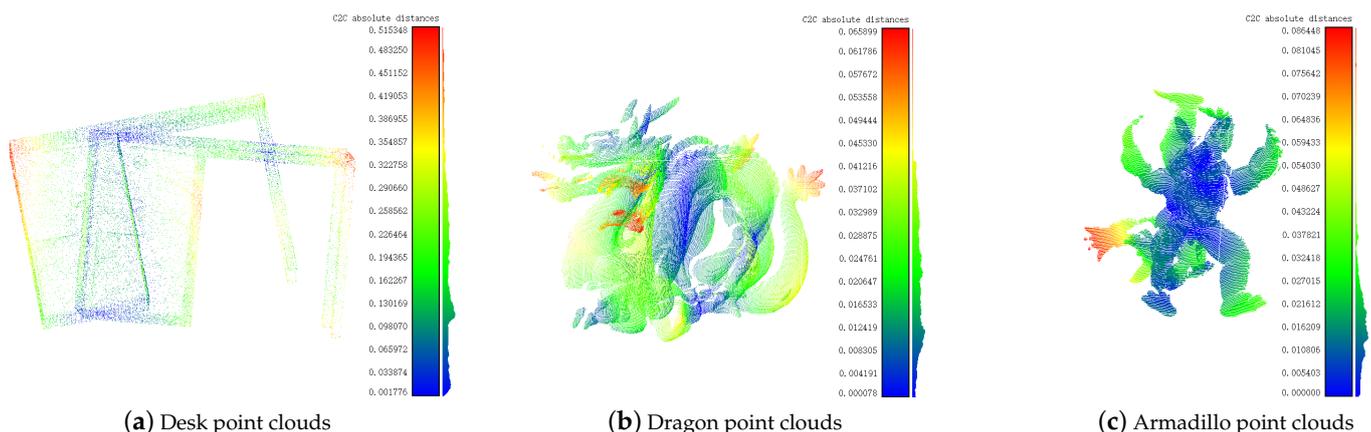
#### 4.1.1. Data

In this part, we test the Super Edge 4PCS algorithm and Super 4PCS algorithm on various data acquired with different geometric attributes, including varying overlap rates, scales, and also with different scanning environments, including natural scenes and manufactured objects. As for the baseline algorithm, the Super 4PCS benefiting from its advance (reduce the computational complexity to  $O(n)$  via smart indexing) and openness (the only open-source method) is selected as the comparison method [4,46]. We selected eleven point clouds whose cardinality and overlap rates were different from each other. The point number of original point clouds and the estimated overlap rate are shown in Table 2. The “num” in the table represents the number of points in the point cloud. View1 and View2 in the table represent the source point cloud and target point cloud. Dimension1, Dimension2, and Dimension3 in the table represent three dimensions of the bounding box which is the cuboid with minimum volume that wrap the point cloud. The scale of point clouds ranges from thousands (Flower) to hundreds of thousands (Bridge), and the overlap rates between point clouds varies greatly, ranging from 0.1 (Bridge) to 0.8 (Bubba). The abundant sample sizes and overlap rates can help us to perform a variable-controlling approach and research the influence of a single variable such as overlap rate

on the registration effect. We collected the first four point clouds from the Stanford open-source datasets [54]. The fifth was downloaded from the open-source data of the Super 4PCS literature [53]. The sixth and the following four groups were provided by the Autonomous Systems Laboratory [55] and Princeton University [56], respectively. The last point cloud(Bridge) was obtained by scanning the bridge with a Faro focus 3D  $\times$  330 scanner. In order to finish the registration between two point clouds with different integrity, we cut the tenth point cloud, flower model, and reserved the upper part as the target point cloud, while the source point cloud was the complete flower model. Parts of the original point clouds data are exhibited in Figure 7. We employed the deviation map to represent point clouds because this format can visually display the distance between points inside point clouds. The distance between the points in the deviation map is the distance between the reference point in the source point cloud and the point in the target point cloud that is closest to the reference point. Different colors in the deviation map represent different distances. Blue is the smallest, and red is the biggest.

**Table 2.** The descriptions of the experimental point clouds where View1 and View2 respectively represent the source point cloud and the target point cloud and  $Dimension_i$  is the length, width, and height of the bounding box.

Point Cloud	Number		Bounding Box Dimensions			Overlap Rates
	View1(num)	View2(num)	Dimension1(m)	Dimension2(m)	Dimension3(m)	
1. Dragon	34,836	41,841	0.05/0.07	0.16/0.16	0.17/0.18	0.5
2. Bunny	40,256	40,251	0.09/0.1	0.15/0.15	0.17/0.18	0.5
3. Bubba	59,692	65,438	0.07/0.06	0.08/0.08	0.10/0.10	0.8
4. Hippo	30,519	21,935	0.31/0.30	0.57/0.55	0.65/0.62	0.7
5. Hokuyo	189,202	190,063	17.41/17.66	27.40/28.33	32.46/33.39	0.7
6. Armadillo	27,678	26,623	0.17/0.11	0.19/0.19	0.19/0.21	0.3
7. Desk	10,000	10,000	0.79/0.79	1.62/1.62	1.80/1.80	0.4
8. Person	10,000	10,000	0.51/0.51	0.60/0.60	0.79/0.79	0.6
9. Toilet	10,000	10,000	1.02/1.02	1.39/1.39	1.72/1.72	0.8
10. Flower	6578	8000	0.97/1.29	1.45/1.47	1.71/1.71	0.4
11. Bridge	281,043	290,217	17.45/12.34	31.68/28.33	36.17/30.90	0.1



**Figure 7.** Poses of original point clouds. (a), (b) and (c) are original poses of Desk, Dragon and Armadillo, respectively. The different colors in the figure represent the distance between the closest points in the point cloud, and the histogram on the right represents the specific values of the distance in meters.

#### 4.1.2. Metrics

(1) **Accuracy Metric:** The metric to evaluate the registration accuracy is Root Mean Square Error(RMSE) and the difference between predicted transformation parameters and ground truth(DPG). The RMSE is adopted to estimate the mean distance between the

source point cloud and the target point cloud. Technically, the dimensions of bounding boxes are used as references with RMSE. If the RMSE is less than one-tenth of the bounding box or even less, then it can be considered that the registration error is small enough. The calculation method of RMSE is shown in Equation (4), where  $m$  is the number of points in the source point cloud, and  $(x'_i, y'_i, z'_i)$  is coordinate of the point in the target point cloud that is closest to  $(x_i, y_i, z_i)$  in the source point cloud. DPG is embodied in two refined aspects: angle error and distance error, which can be calculated according to Equations (5) and (6). The specific calculation methods of these two errors are as follows. The rigid transformation matrix  $T$  obtained from the registration base pairs is firstly decomposed into a rotation matrix  $R$  and a translation vector  $t$  according to Equation (5). Then Equation (6) is then employed to convert the rotation matrix  $R$  into three rotation angles  $\alpha, \beta, \gamma$  along three coordinate axis  $x, y, z$ , respectively. Finally, calculate the difference between the elements in the translation vector  $t$  and the ground-truth distances as distance error and the difference between the three predicted angles and the ground-truth rotation angles as the angle error.

(2) **Efficiency Metric:** The efficiency metric is expressed as the running time of algorithm. We accumulated 30-times running time of the Super 4PCS algorithms and Super Edge 4PCS algorithm, then reported the average value  $T_{avg}$  as the efficiency metric.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i^2 - x_i'^2) (y_i^2 - y_i'^2) (z_i^2 - z_i'^2)} \quad (4)$$

$$\text{if } T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ then } R = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} t = (t_{14}, t_{24}, t_{34})^{-1} \quad (5)$$

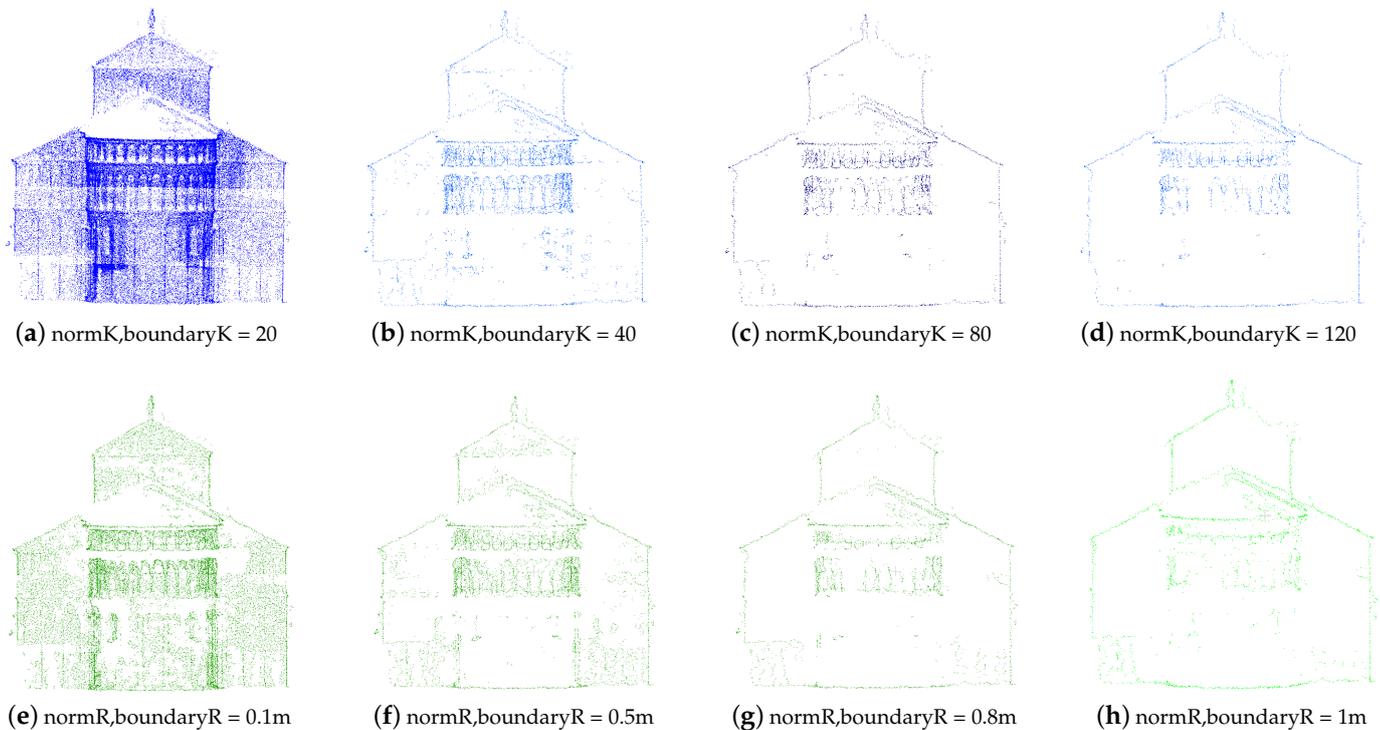
$$R(\gamma, \beta, \alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \quad \alpha = \text{Atan2}\left(\frac{r_{21}}{\cos(\beta)}, \frac{r_{11}}{\cos(\beta)}\right) \quad \gamma = \text{Atan2}\left(\frac{r_{32}}{\cos(\beta)}, \frac{r_{33}}{\cos(\beta)}\right) \quad (6)$$

#### 4.2. Parameters Test

This section describes the impact of different parameters used in the Super Edge 4PCS algorithm. All environment settings and experiments were run on one machine with Intel Core i7-7700CPU @ 3.60 GHZ and 8 GB RAM. The parameters used in the Super Edge 4PCS algorithm are shown in Table 1. First of all, we carried out experiments about the first four parameters: search mode  $M$ , neighborhood range of normal vector calculation  $N_V$ , neighborhood range of boundary calculation  $N_B$ , the threshold of vector angle  $h$ . The experimental data is required to have a rich geometric structure to analyze the results after boundary segmentation clearly. Consequently, the Pisa Cathedral point cloud provided in [10] that contained 1.08 million points is employed. The experimental results are shown in Figure 8. The normK and boundaryK in the figure are  $N_V$  and  $N_B$  described in Section 3.3.2. As can be seen from Figure 8, the cardinality of boundary point clouds decreases with the increase of  $K$  and  $R$ , where  $K$  and  $R$  are the neighborhood of K-search and R-search, respectively. With  $K$  varies from 120 to 40, the number of boundary points changes from 3324 to 22,580 and that is from 2681 to 24,698 when  $R$  ranges from 0.1 to 1. The reason can be inferred that the increase of neighborhood means more vector angles around the reference point  $P$ . Therefore, the maximum angle becomes smaller because the neighboring points become denser, and the reference point  $P$  is more difficult to be judged as a boundary point. In terms of two search modes, the density of point clouds affects the boundary segmentation under R-search mode because different point clouds correspond to the different number of points within the fixed search radius  $r$ . Therefore, we cannot provide a uniform reference value of the search radius. Instead, the K-search

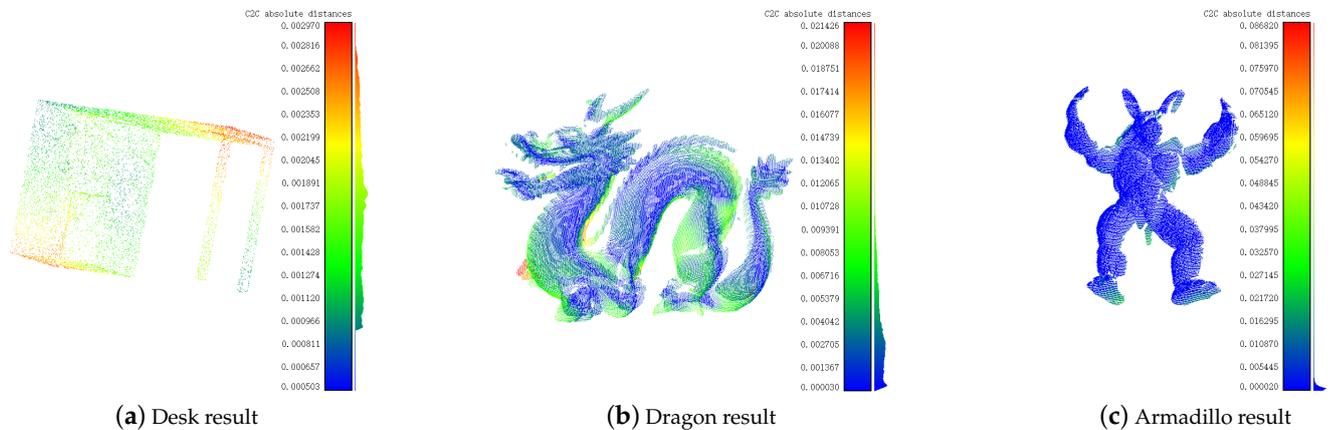
mode selects the neighborhood including fixed points, which is not affected by the density of point clouds. Specifically, we can get a significant result from Figure 8 that boundary segmentation can combine accuracy and the cardinality of boundary clouds when  $K$  is between 20 and 40. Therefore,  $K$ -search mode with  $K$  between 20 and 40 is recommended. The threshold  $h$  was set to  $\pi/2$  in later experiments according to the conclusion of [29] that  $\pi/2$  was a suitable tradeoff between time efficiency and contained information.



**Figure 8.** The effects of boundary extraction corresponding to different neighborhood ranges  $N_V$  and  $N_B$  are different under two modes  $M$ ,  $K$ -nearest neighbor search and  $R$ -nearest neighbor search.  $\text{norm}K$  and  $\text{norm}R$  are the neighborhood range  $N_V$  of normal vector calculation,  $\text{boundary}K$  and  $\text{boundary}R$  are the neighborhood range  $N_B$  of interest points.

The registration results of the Super Edge 4PCS algorithm are presented in Figure 9. Intuitively, three registration results all tend to be blue-green. Thus the distances between points are close and all lower than the biggest error significantly. Especially, the maximum distance errors of three point clouds are taken around 0.002 m, 0.017 m, and 0.016 m. These values are all less than one-tenth of the dimensions of bounding box of three point clouds. Therefore, we can preliminarily judge that the proposed algorithm has successfully completed the registration of the sample point cloud from Figure 9. The parameters used in these experiments are given in Table 3. The parameters  $P1(M)$  and  $P5(\epsilon_{axis})$  need to be manually set to determine the search mode of neighbor points and whether local segmentation is required. Firstly, TRUE in the first column of the Table 3 represents  $K$ -search and FALSE is  $R$ -search. We recommend using the  $K$ -search method to obtain the neighbor points of the reference point since  $K$ -search can directly obtain  $k$  nearest neighbors regardless of the density of point clouds. Secondly, if we need to perform local extraction, set  $P5$  to 10, or set it to 0.005. Besides, it can be seen from Table 3 that the parameters  $P2(N_V)$ ,  $P3(N_B)$ ,  $P4(h)$ ,  $P7(S_g)$ , and  $P9(fDis)$  maintain fixed values, so we can use these five values as the recommended values of the corresponding parameters. The meaning of parameter  $P6(Sim_g)$  is the similarity constraint between overlapping regions. The value of this parameter varies from point cloud to point cloud, so we cannot provide a fixed value. It can be concluded from Table 3 that the range of this parameter is 0.0005~0.05. The last parameter,  $P8(baseDis)$ , controls the distance tolerance between the point pairs. This parameter also varies with the distance between the point pairs, so only a range of

reference values 0~20% can be provided. It can be seen from Table 3 that most of the parameters have a fixed reference value, but the automatic nature of parameters P6 and P8 needs to be improved, which is a future improvement direction of the proposed algorithm.



**Figure 9.** The registration results of the Super Edge 4PCS algorithm on some samples. (a), (b) and (c) are the registration results of Desk, Dragon, Armadillo, respectively. Note that the distance errors of the three point clouds after registration are 0.003 m, 0.004 m, and 0.00002 m, which are smaller than tenth of the corresponding bounding box dimensions.

**Table 3.** Parameters for registration of different point clouds.

Point Clouds	P1 ( $M$ )	P2 ( $N_v$ ) (num)	P3 ( $N_B$ ) (num)	P4 ( $h$ ) (degrees)	P5 ( $\epsilon_{axis}$ )	P6 ( $Sim_g$ )	P7 ( $S_g$ )	P8 (baseDis)	P9 (fDis)
Dragon	TRUE	20	40	90°	10	0.005	15%	1%	3%
Bunny	TRUE	20	40	90°	10	0.0005	15%	1%	3%
Bubba	TRUE	20	40	90°	10	0.005	15%	1%	3%
Hippo	TRUE	20	40	90°	10	0.0007	15%	18.5%	3%
Hokuyo	TRUE	20	40	90°	10	0.05	15%	2.5%	3%
Armadillo	TRUE	20	40	90°	10	0.0005	15%	14.5%	3%
Desk	TRUE	20	40	90°	10	0.001	15%	1%	3%
Person	TRUE	20	40	90°	10	0.001	15%	1%	3%
Toilet	TRUE	20	40	90°	10	0.001	15%	1%	3%
Flower	TRUE	20	40	90°	0.005	0.001	15%	1%	3%

It is worth noting that the Super 4PCS algorithm also needs to manually set some parameters. We also need to set a suitable registration accuracy and sampling size in addition to estimating the overlap rate of the samples, which makes the Super 4PCS algorithm empirical. Therefore, the automation of parameter in the Super 4PCS algorithm also needs to be improved.

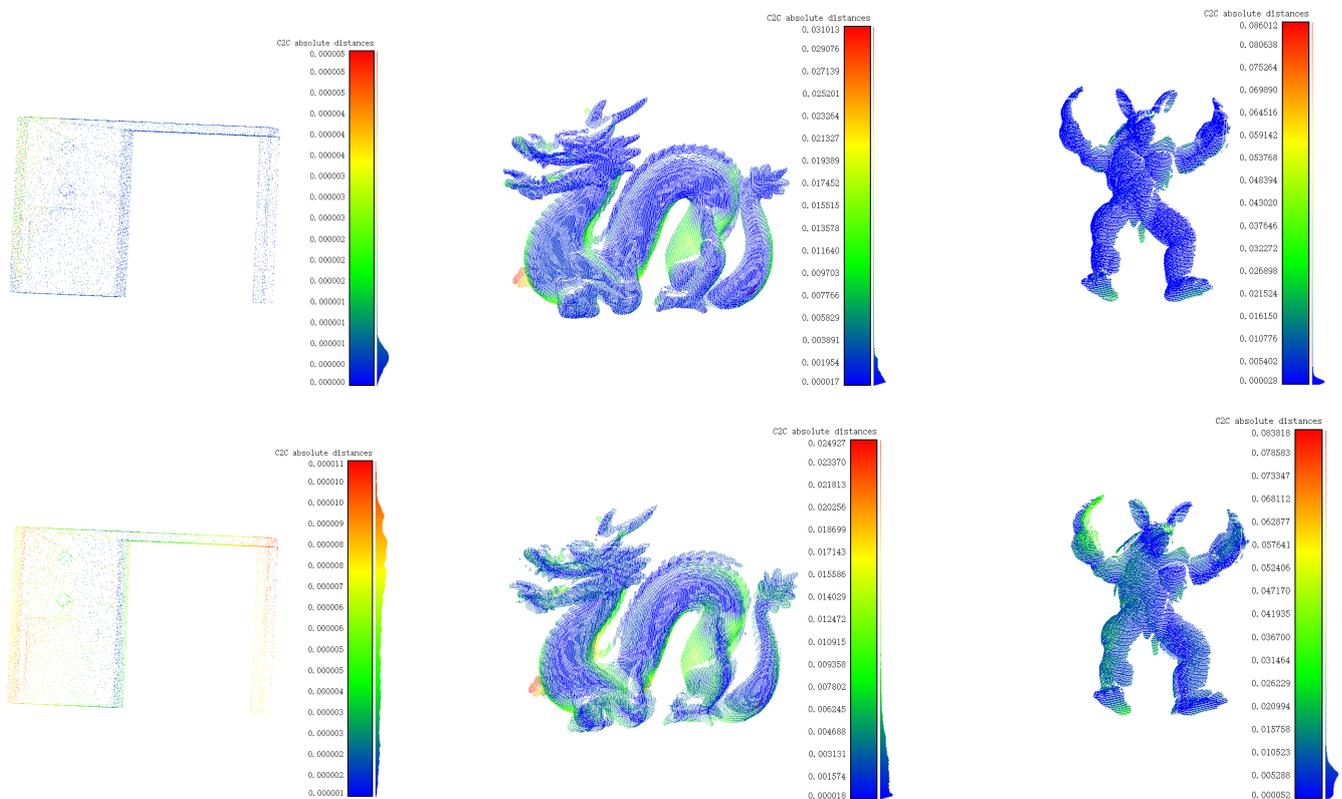
#### 4.3. Noise Elimination Test

Moreover, we tested the noise elimination effect of boundary segmentation(BS) steps. We first added the noise with the mean value of zero and the standard deviation of 0.1 to the Pisa Cathedral point clouds, which is used to extract the boundary. Then we performed boundary segmentation with different K neighbor numbers on the noise-added point cloud and the original point cloud to obtain two boundary point clouds. We tested the similarity between the boundary point clouds calculated from the noisy point clouds and the original point clouds under different K neighbor numbers, 20, 40, 80, 120. All the similarities are 0.9998. These values mean that the boundary point cloud with noise is similar to that without noise after boundary segmentation algorithms under different K-nearest neighbors, which proves that the boundary segmentation(BS) can greatly reduce the noise in the boundary point clouds.

#### 4.4. ICP Test

The results of registration with fine-tuning ICP are shown in Figure 10. The colors of three groups of deviation maps tend to be dark. Besides, with the attention to the legend

of the figure, we can observe that the distance error of most points within the result point clouds is less than 0.005m, which is smaller than the bounding box dimensions of point clouds. It can be concluded from Figure 10 that the registration effects of the two algorithms are close and excellent after the precise adjustment of the ICP algorithm. Therefore, we can infer that both algorithms can provide excellent initial values for the ICP algorithm. Detailed RMSEs between registered point clouds after the ICP algorithm are shown in Table 4. Dimension1, Dimension2, and Dimension3 in the table represent three dimensions of the bounding box where the point cloud is warped. These three dimensions are used as a reference for the magnitude of RMSE. The two algorithms achieve close and accurate registration results after fine-tuning of ICP algorithm. After comparing the RMSE of each point cloud with three dimensions of the bounding box, we can see that the RMSE is significantly smaller than the dimensions of bounding box. More accurately, the smallest gap between RMSE and dimensions of bounding box is 10 times (Dragon). The RMSE of the Desk point clouds through the proposed algorithm is  $4 \times 10^{-7}$  m that is much smaller than that of the Super 4PCS.



**Figure 10.** Results of ICP precise registration: the top is the results of the Super 4PCS algorithm, and the bottom is the results of Super Edge 4PCS algorithm. The main tones of three groups of point clouds toward bluey-green, which indicate that the distance error is relatively small. The specific values are reflected in the histogram on the right, and the unit is meter.

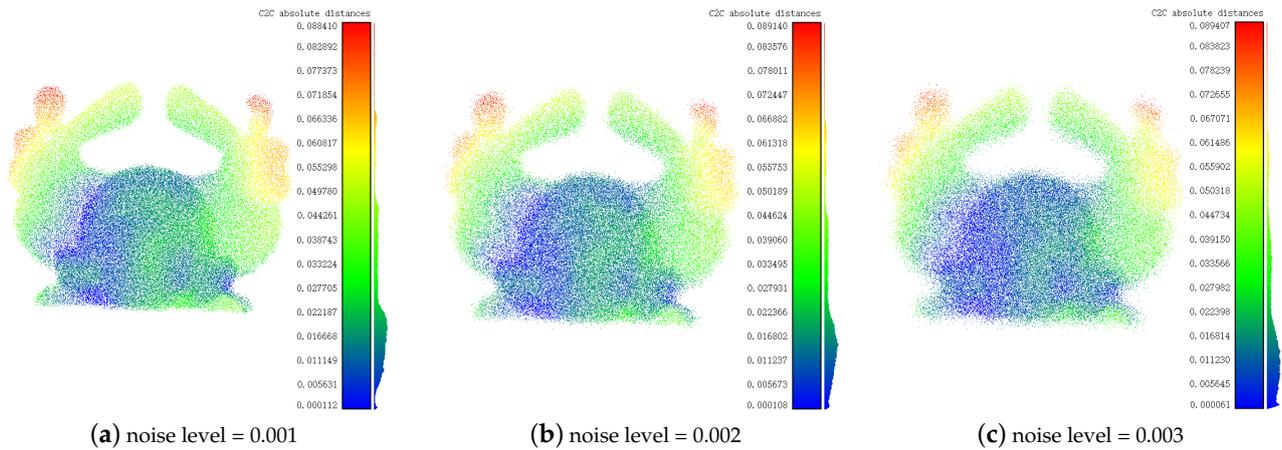
**Table 4.** RMSEs between point clouds after ICP algorithm. Dimension 1, 2, 3 represent the length, width, and height of the bounding box of each point cloud, respectively.

Point Cloud	Desk	Dragon	Armadillo
Super 4PCS (m)	0.001	0.004	0.004
Super Edge 4PCS (m)	$4 \times 10^{-7}$	0.005	0.006
Dimension1 (m)	0.79	0.05	0.17
Dimension2 (m)	1.62	0.16	0.19
Dimension3 (m)	1.80	0.17	0.19

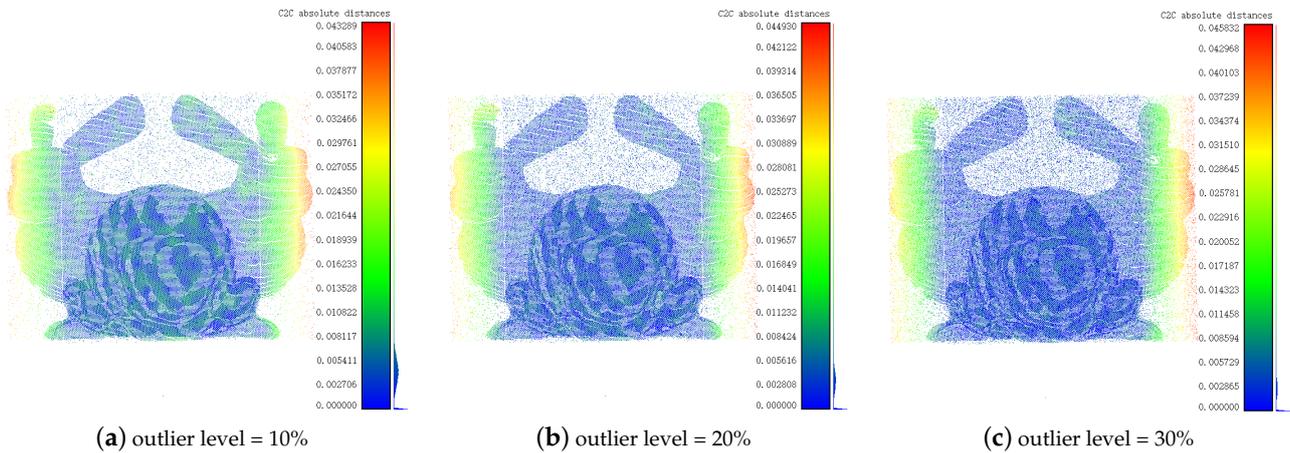
#### 4.5. Robustness Test

This section discusses the influence of different variables on the performance of the proposed algorithm and the Super 4PCS algorithm, including noise levels, outlier levels, overlap rates, and resolutions. There are many situations corresponding to the change of overlap rate, such as occlusion between point clouds. We integrated the analysis of these reasons into the experiment on the shift of overlap rate. We randomly chose two point clouds mentioned in Section 4.1 for this experiment. Here we took Bunny and Dragon as examples. In the following experiments, we always adhere to the principle of single variables, that is, only one variable is changed in each experiment, and the other variables remain unchanged. The original data with interference factors are shown in Figures 11–14. Firstly, we added zero-mean and different standard deviation Gaussian noise to the original data to form the noisy data. (a), (b) and (c) represent the point cloud after adding noise with standard deviations of 0.001, 0.002, 0.003. As can be seen from Figure 11, the point cloud gradually becomes more “blurred” with the increase of the standard deviation. Next, we randomly added different proportions of points to the bounding box, as shown in Figure 12. Figure 12a–c represent the point cloud with 10%, 20%, and 30% added outliers, respectively. Thirdly, the different overlap rates are achieved by manually setting the position of two point clouds, as shown in Figure 13. Figure 13a–c mean that the overlap rates between the source point cloud and the target point cloud are 20%, 30%, and 40%, respectively. Finally, we randomly reserved points of 20%, 40%, and 60% from the original point clouds following to the uniform distribution to set different resolutions as shown in Figure 14a–c. The point clouds with different resolutions shown in Figure 14 presented different sparsity.

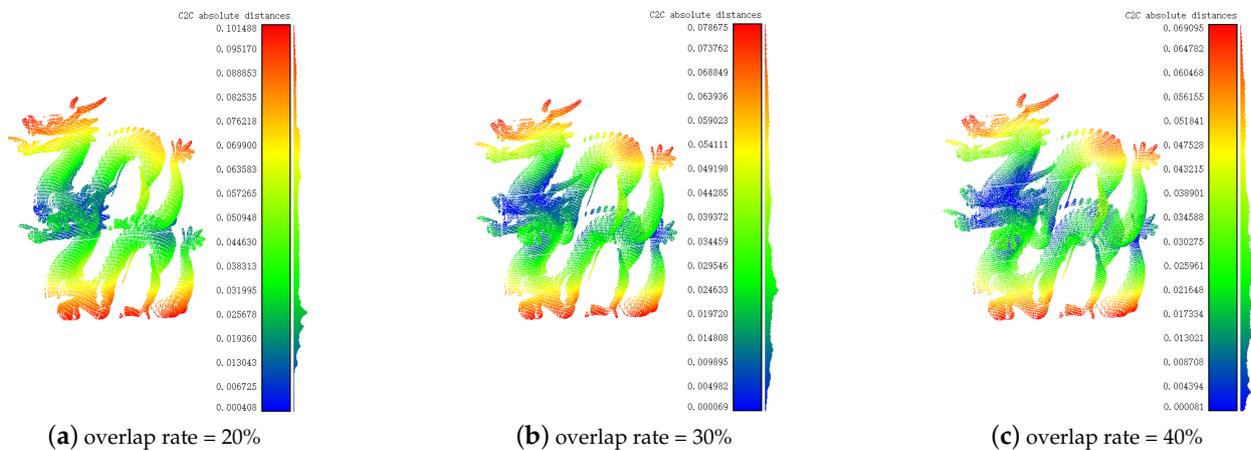
The results of the Super Edge 4PCS algorithm are visually shown in Figures 15–17. We can qualitatively analyze the registration results of our algorithm from these figures. First of all, it can be seen from Figure 15 that the colors of most points within result point clouds tend to be darker than inputs. The inside part of the deviation map of the bunny point cloud is red since the Bunny model is hollow, which is reasonable. When the noise level is 0.001 and 0.002, the edge of the Bunny is blue, and the distance between the point pairs is about 0.007 m, which is dramatically less than the dimensions of bounding box. The above observation indicates that the registration is mostly accomplished, and the error is low. The registration can still be performed basically completely even when the noise level is increased to 0.003. Therefore, we preliminarily derived that the algorithm in this paper has good robustness to noise. The same situation appears in Figure 16. The distance error is about 0.007 m when the outlier levels are 10% and 20%. Although there is misalignment when the outliers are as high as 30%, the registration is mainly completed. Finally, the situation has changed in Figure 17. The color of the point clouds does not vary with the decrease in the overlap rate. Instead, the color remains on the main tone of blue-green. The histogram on the right also shows that the distance error of the point cloud has not changed significantly. Therefore, we can infer that the registration effect of the proposed algorithm does not be primarily affected by the overlap ratio. Next, the point cloud registration under different influence factors is analyzed quantitatively.



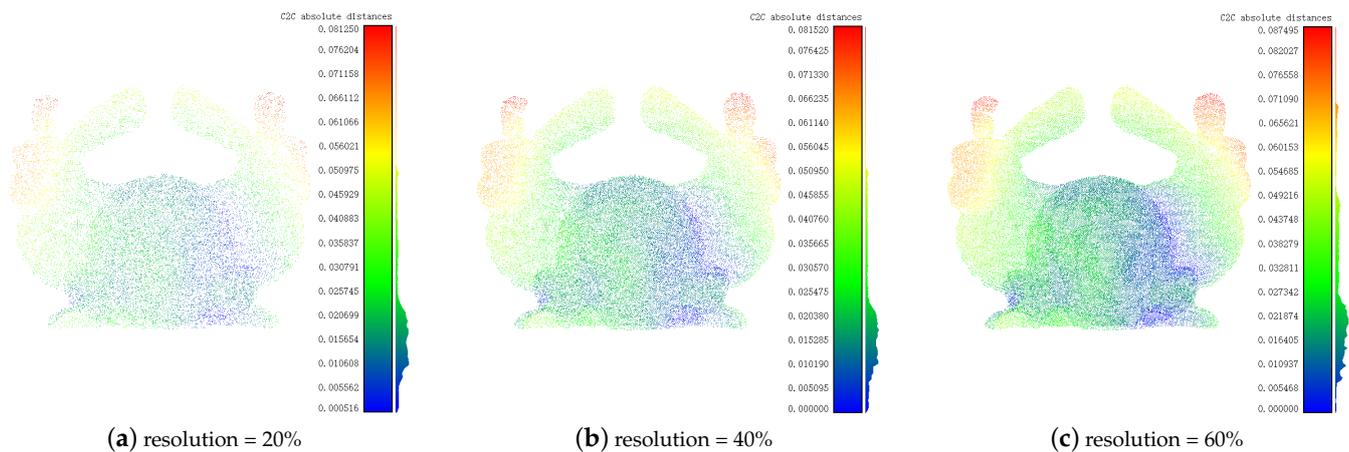
**Figure 11.** Point clouds affected by different noise levels. The noise level is measured by the different standard deviations of the added noise. (a), (b) and (c) represent the poses of source point cloud and target point cloud with noise level of 0.001, 0.002 and 0.003, respectively.



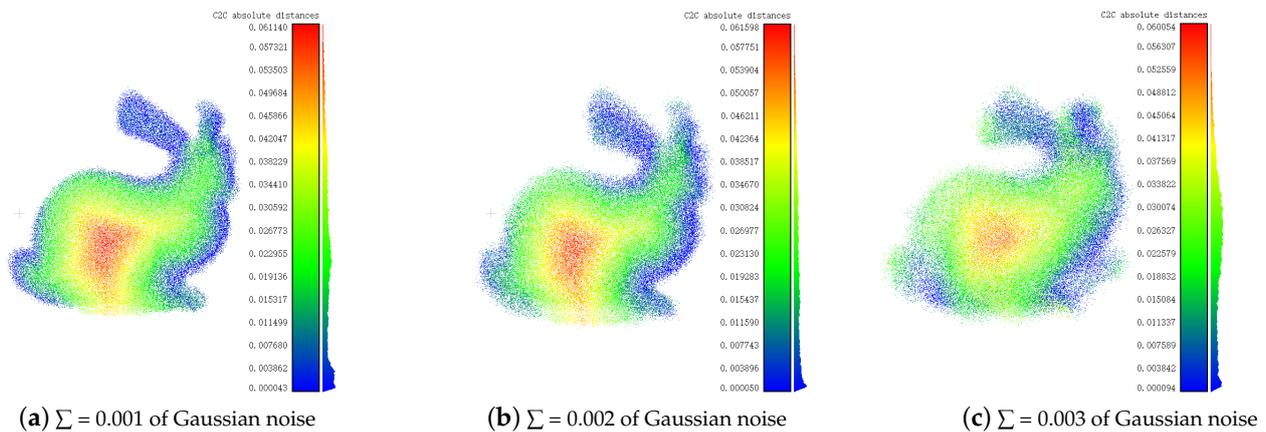
**Figure 12.** Point clouds affected by different outlier levels. The outlier level is measured by the ratio of the added outliers to the number of points in original point clouds. (a), (b) and (c) represent the poses of source point cloud and target point cloud with outlier level of 10%, 20% and 30%, respectively.



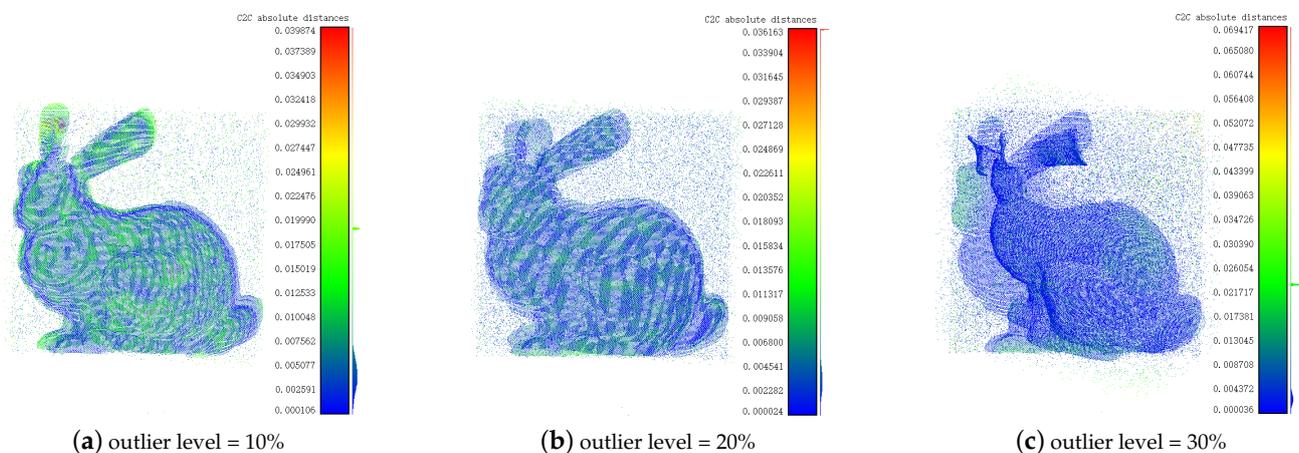
**Figure 13.** Point clouds affected by different overlap rates. The overlap rate is measured by the ratio of the number of points in the overlapping regions to the number of original point clouds. (a), (b) and (c) represent the poses of source point cloud and target point cloud with overlap rate of 20%, 30% and 40%, respectively.



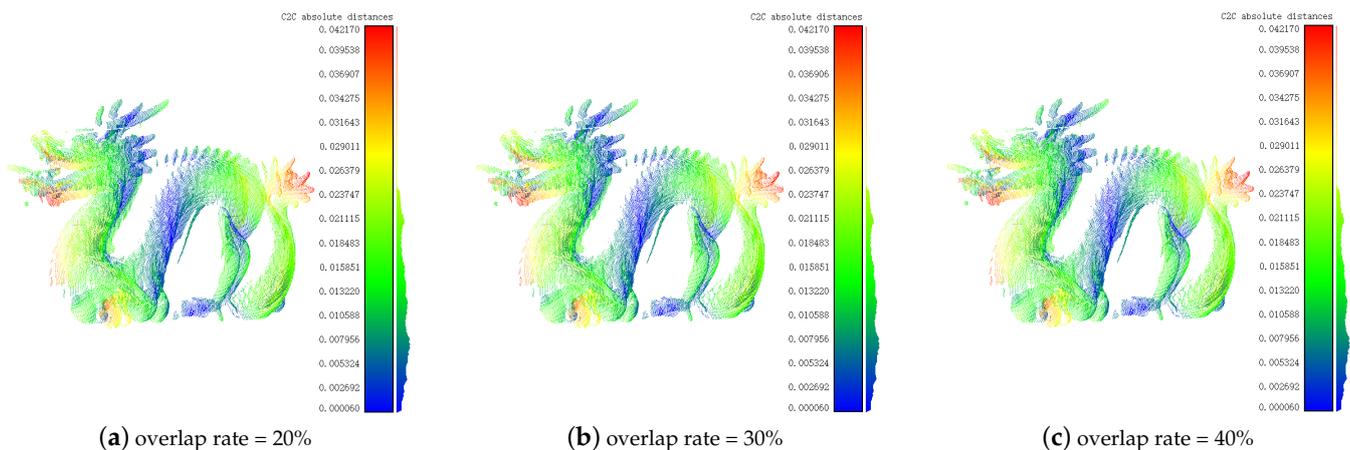
**Figure 14.** Point clouds affected by different resolutions. The resolution is measured by the ratio of the number of points in sparsed point cloud to the number of points in original point clouds. (a), (b) and (c) represent the poses of source point cloud and target point cloud with resolution of 20%, 40% and 60%, respectively.



**Figure 15.** Registration results of the Super Edge 4PCS algorithm under different noise levels.  $\Sigma$  represents the standard deviation of noise. The reason for the red in the center of the Bunny is that the source point cloud and the target point cloud hold hollow structures. (a), (b) and (c) represent the poses of registered point clouds under the noise level of 0.001, 0.002 and 0.003, respectively.



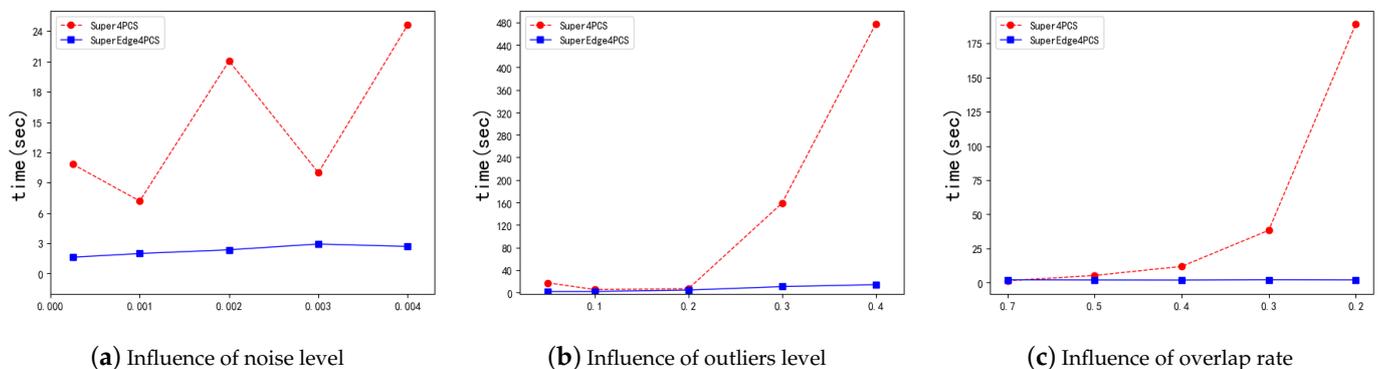
**Figure 16.** Registration results of the Super Edge 4PCS algorithm under different outlier levels. (a), (b) and (c) represent the poses of registered point clouds under the outlier level of 10%, 20% and 30%, respectively.



**Figure 17.** Registration results of the Super Edge 4PCS algorithm under different overlap rates. These three registered point clouds are close in terms of both color and the specific value of the histogram. (a), (b) and (c) represent the poses of registered point clouds under the overlap rate of 20%, 30% and 40%, respectively.

#### 4.5.1. Influence on Efficiency

The first three experiments compared the influence of different noise levels, outlier levels, and overlap rates on computational efficiency. We measured the time two algorithms took to make the RMSEs of registration results less than  $\epsilon$  like the description of efficiency metrics. The accuracy requirement  $\epsilon$  we chose here was 0.03 m, which means that the RMSE of registered clouds must be less than or equal to 0.03 m. The comparison results of two algorithms with varying overlap rates, noise levels, and outlier levels are shown in Figure 18.



**Figure 18.** Influence of noise level, outliers level and overlap rate on algorithm efficiency. The specific evaluation method is the time required for the algorithm to complete the registration under the influence of different factors. (a–c) correspond to different noise levels, outlier levels and overlap rates.

Firstly, we can obtain the influence of noise on the registration time from Figure 18a. In the overall analysis, the average and variance of registration time of the Super 4PCS algorithm under the influence of noise are 14.7 s and 61.41. Therefore, registration on noisy datasets makes the time of the registration unpredictable. However, we can find out that our proposed algorithm always maintains a stable level under different noise levels. The average and variance of registration time are 2.3 s and 0.2. Besides, as seen from this subgraph, the Super 4PCS algorithm (red line) has a significant fluctuation compared with the proposed algorithm (blue line). More accurately, the registration time of the Super 4PCS algorithm changes from 7.2 s to 21.1s when the noise level changes slightly from 0.001 to 0.002 and the changing amplitude is 13.9 s. The changing amplitude is 14.6 s when it varies from 0.003 to 0.004 and for the Super 4PCS algorithm. This vast difference means

that the proposed algorithm is more efficient and robust than the Super 4PCS algorithm to the change of noise. The reasons for this result can be summarized from the corresponding experiments in Section 4.3 Noisy Elimination test. The proposed algorithm eliminates a lot of noise in the process of boundary segmentation. However, the Super 4PCS algorithm has no special steps to remove noise, so the registration time is influenced by noise.

Next, Figure 18b shows the effect of different outlier levels on the registration results. It can be seen from the figure that the registration time of the Super 4PCS algorithm presents an increasing trend with the increase of outlier levels. The time of the proposed algorithm is always lower than the former, and remains stable all the time. Numerically, the average time gap between the two algorithms is 7.0 s when the outlier level is less than 0.3, while the time spent by the Super 4PCS algorithm rises sharply when the outlier level is greater than 0.2 where the average time gap is 305.7 s. Amazingly, the Super 4PCS algorithm took 477.0 s when the outlier level equaled 0.4. Therefore, the tolerance of the Super 4PCS algorithm to outliers is about 0.2. As long as the outlier level is higher than this value, the running time of the Super 4PCS algorithm will be affected by the outlier levels. Moreover, all the registration time of proposed algorithm remained below 14 s no matter how the outlier levels changed. Therefore, we can conclude that the algorithm proposed in this paper shows higher computational efficiency and stability than the Super 4PCS algorithm under the influence of different outlier levels. The reason for the above comparison results is similar to the analysis of noise. Compared with the Super 4PCS algorithm, the proposed algorithm reduces the redundant search space caused by a large number of outliers with boundary segmentation step, thus saving the registration time.

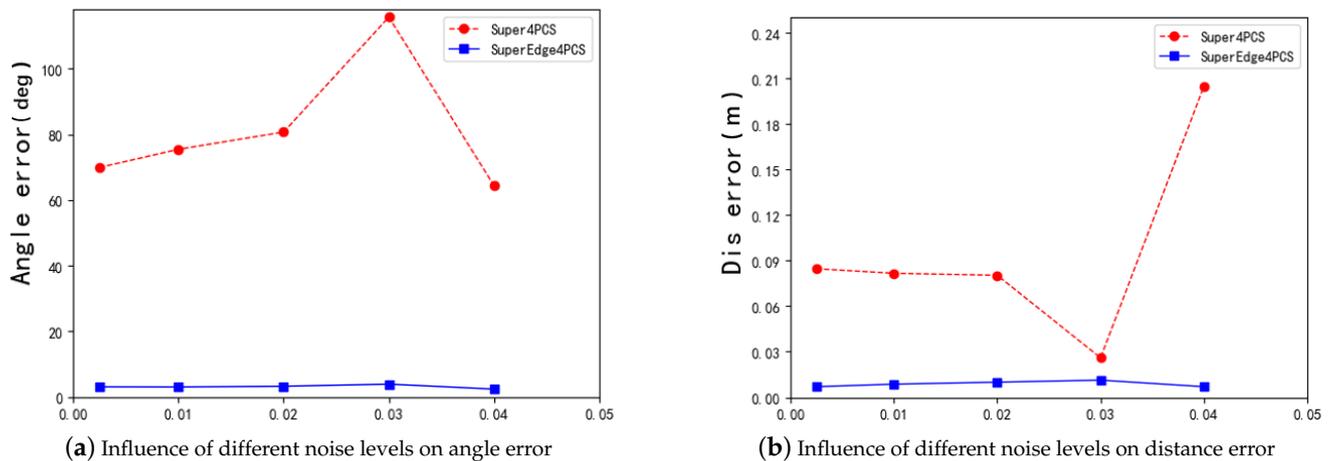
Finally, a similar situation occurs when we focus on the influence of overlap ratio. We can observe from Figure 18c that the time gap is 1.3 s when the overlap rate is greater than 0.4, while the gap becomes 78.0s when the overlap rate is less than 0.5. Specifically, the time is 189.4 s when the overlap rate is 0.2. Conversely, the registration time taken by proposed algorithm is always less than 2.0 s. Enough to prove that the proposed algorithm is less affected by the changing overlap rates than the Super 4PCS algorithm. Furthermore, the difference of computation time in restricted situations where the outlier level is 0.4 and the overlap ratio is 0.2 shows that the proposed algorithm possesses better applicability than the Super 4PCS algorithm when dealing with the point clouds with extreme conditions like few overlapping regions. Tracing the source, the decrease of overlap rate makes the search space of proper registration bases decrease. The separated overlapping regions extraction(ORE) arises at the moment, which can output the overlapping regions directly in exponential time. Thus the effect of small overlap rates on registration is weakened.

#### 4.5.2. Influence on Accuracy

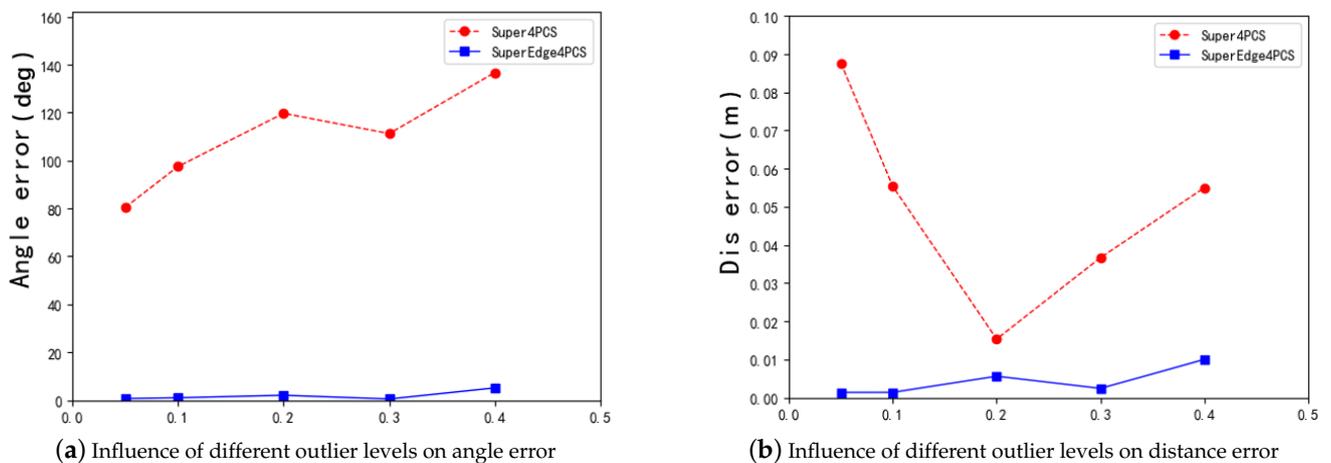
In the following three experiments, we compared the registration accuracy under the influence of noise, outliers, and resolution. The registration accuracy is reflected in the translation error and rotation error, which is calculated according to the method described in Section 4.1.2. The ground-truth registration parameters are obtained by manually selecting control points to construct registration bases for coarse registration and then performing the ICP algorithm for fine registration. Figures 19–21 show the changes of angle error and distance error of the two algorithms under the interference of three factors.

Firstly, we can obtain the registration accuracy of the two algorithms under the influence of noise from Figure 19a,b. We can preliminarily infer from the position of the two broken lines in these subgraphs that the proposed algorithm reaches higher registration accuracy than the Super 4PCS algorithm. Accurately, the angle errors calculated by the Super Edge 4PCS algorithm are less than  $5^\circ$  under the influence of different noise levels. In comparison, the angle errors of Super 4PCS algorithm are obviously large, which are all more than  $60^\circ$ . Especially the error is up to  $115.7^\circ$  when the noise level is 0.03. Therefore, we can conclude that the Super Edge 4PCS algorithm has higher rotation accuracy than the Super 4PCS algorithm under various noise conditions. Not only the angle error, but the distance accuracy of the Super Edge 4PCS algorithm also is more considerable than

that of the Super 4PCS algorithm from Figure 19b. The average distance error of Super 4PCS algorithm is 0.1 m, while that of the proposed algorithm is 0.04 m. Therefore, the proposed algorithm has advantages over Super 4PCS algorithm in both rotation and translation accuracy. The main reason for such disparity is that the elimination of noise in the boundary segmentation process saves the search time of the candidate registration base and removes a large number of wrong candidate points, thus reducing the translation and angle errors of registration.



**Figure 19.** Influence of different noise levels on registration accuracy (a,b) in the figure show the influence on the registration angle error and distance error. The units are degrees and meters.

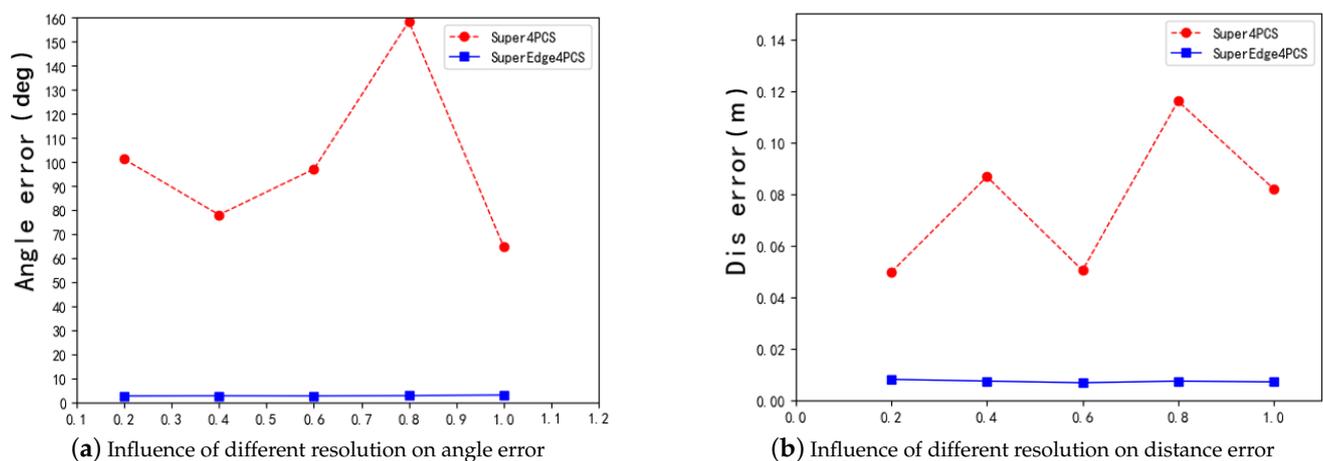


**Figure 20.** Influence of different outlier levels on registration accuracy. (a,b) in the figure show the influence on the registration angle error and distance error. The units are degrees and meters, respectively.

Next, we focus on the influence of outliers on the accuracy of registration results. The initial inference about accuracy difference between the two algorithms can still be obtained from the position of red and blue broken lines. Obviously, the proposed algorithm has a smaller registration error than the Super 4PCS algorithm under different outlier levels. After observing the data from Figure 20a, the average angle error of Super 4PCS algorithm is  $109.2^\circ$ , while that of the Super Edge 4PCS algorithm is  $1.9^\circ$ . Therefore, the proposed algorithm has better registration rotation accuracy than the Super 4PCS algorithm under the influence of outliers. Figure 20b shows the changes of distance error of the two algorithms under the interference of outliers. The average distance errors of the two algorithms are 0.05 m and 0.004 m, respectively, so the algorithm in this paper achieves better results in translation. Besides, the average fluctuations of distance error are 0.03 m and 0.004 m,

respectively. This fact also reflects that the robustness of proposed algorithm to the outliers is better than Super 4PCS algorithm. In summary, the proposed algorithm achieves higher accuracy and robustness than the Super 4PCS algorithm under the influence of different outlier levels, which benefits from separate boundary segmentation and overlapping regions extraction.

Last, we analyzed the influence of different resolutions on point cloud registration. It was evident from Figure 21a,b that the angle error and distance error of the proposed algorithm is less than Super 4PCS algorithm. Precisely, the differences of average of angle errors and distance errors are  $97.2^\circ$  and 0.07 m, respectively. Regarding stability, the standard deviations of angle error calculated by Super 4PCS algorithm and the proposed algorithm are 32.1 and 0.1, respectively, and that of distance error are 0.02 and 0.0004, respectively. Therefore, the proposed algorithm is more accurate and robust to the change of resolution than Super 4PCS. The main reason for this phenomenon is that point clouds where the candidate registration bases are selected by two algorithms are different. As a result of reduced resolution, the number of suitable candidate registration bases inside the point cloud is reduced. However, the algorithm in this paper only selects candidate registration bases on the boundary of the point cloud, so it is less affected by the reduction in resolution.



**Figure 21.** Influence of different noise levels on registration accuracy. (a,b) in the figure show the influence on the registration angle error and distance error. The units are degrees and meters.

#### 4.6. Computational Test

We separated the running time of each stage in Super 4PCS algorithm and Super Edge 4PCS algorithm. The statistical results were integrated into Table 5. The Super 4PCS algorithm needs to be set a parameter called sampling size. This parameter controls the sampling size of two input point clouds and finally perform registration for the sampled point cloud. It is necessary to comprehensively compare the Super 4PCS algorithm from two aspects: large sampling size and small sampling size. Therefore, we carried out the same experiment on Super 4PCS algorithm with different sampling sizes. “small” in Table 5 represents a small sampling size, and “large” is the opposite. Besides, CSE and CSV are abbreviations for congruent set extraction and congruent set validation, respectively. The former stands for the acquisition process of candidate registration base pairs. The latter is the verification process of the extracted registration base pairs depending on the RMSEs calculated from the registered point clouds. The base pair corresponding to the minimum RMSE is finally selected to obtain the optimal rigid transformation matrix. The validation time is the total time to verify all the extracted registration base pairs.

**Table 5.** The running time of sub stage of Super 4PCS and Super Edge 4PCS in different samples. CSE and CSV represent the congruent sets extraction and congruent sets verification, respectively.

Point Cloud	Sample Size (num)	CSE			CSV			Overlap Rate
		SE4PCS (s)	S4PCS (s) (Small)	S4PCS (s) (Large)	SE4PCS (s)	S4PCS (s) (Small)	S4PCS (s) (Large)	
Flower	7289	0.9	1.5	8.9	0.04	0.008	0.4	<b>0.4</b>
Desk	<b>10,000</b>	<b>0.4</b>	<b>9.6</b>	<b>35.3</b>	<b>0.06</b>	<b>0.3</b>	<b>2.9</b>	<b>0.4</b>
Person	<b>10,000</b>	<b>0.4</b>	<b>2.4</b>	<b>8.1</b>	<b>0.06</b>	<b>0.4</b>	<b>4.1</b>	<b>0.6</b>
Toilet	<b>10,000</b>	<b>0.4</b>	<b>0.5</b>	<b>1.5</b>	<b>0.06</b>	<b>0.02</b>	<b>0.3</b>	<b>0.8</b>
Bubba	62,565	2.0	0.6	1.5	0.4	0.3	4.4	0.8
Dragon	38,338	2.6	8.2	16.2	0.2	1.2	9.2	0.5
Bunny	40,253	1.1	2.2	7.0	0.2	0.1	1.8	0.5
Average	-	1.1	3.6	11.2	0.1	0.3	3.3	-

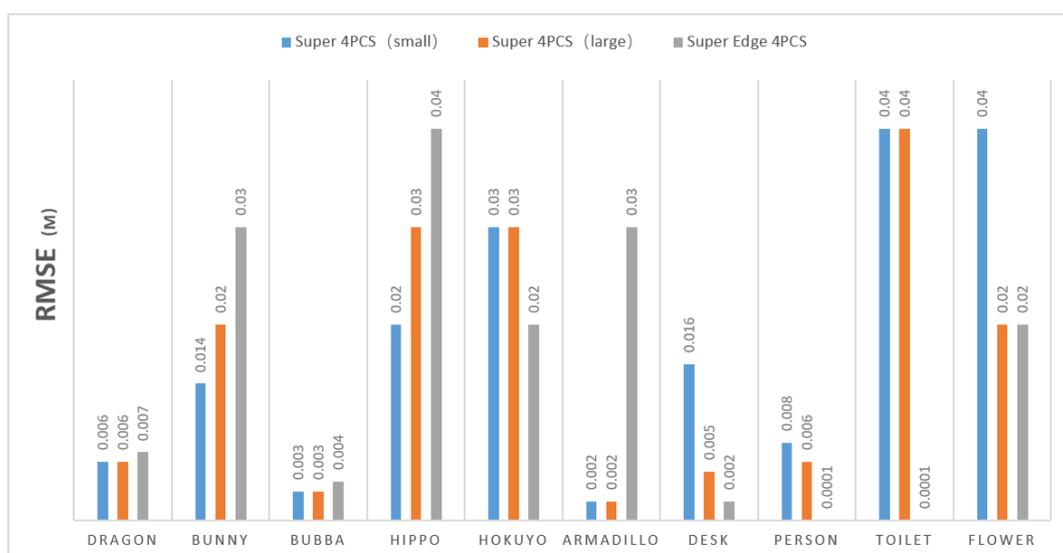
As shown in Table 5, the average time of the Super Edge 4PCS algorithm is 1.1 s in CSE, while that of the Super 4PCS algorithm are 3.6 s and 11.2 s. The differences are 2.5 s and 10.1 s, respectively. It can be inferred that the proposed algorithm improves the speed of the CSE. Besides, the sizes of candidate congruent set extracted from the point clouds shown in Table 5 are all less than 10 after CSE except for Bunny and Bubba point clouds, which are 64 and 21, respectively. These numbers are far less than that extracted by Super 4PCS algorithm ( $10^n, n \geq 2$ ). Correspondingly, the difference in the size of the candidate congruent set results in the disadvantage of Super 4PCS algorithm in the computation time of CSV. Specifically, the average time of Super Edge 4PCS spend on CSV is 0.1 s, while that is 0.3 s and 3.3 s in Super 4PCS algorithm. Therefore, the algorithm proposed in this paper can both improve the efficiency of CSE and CSV. The search space of the former is limited by ORE(overlapping regions extraction), thus reducing the search time. The latter thanks to the acquisition of the corresponding bases, which makes full use of the volumetric information of boundary point clouds. Especially for point clouds (Desk, Person, Toilet) with the same sample size but different overlap rates, the CSE and CSV of the proposed algorithm are 0.4 s and 0.6 s, respectively. The stability of the calculation time spend on CSE and CSV further illustrates the robustness to overlap rates. As the overlap rate decreases, the amplitudes of CSE in Super 4PCS is 9.1 s and 33.8 s, while the CSV is only 0.28 s and 2.6 s. Therefore, the difference in overlap rates mainly affects the CSE extraction process. The reason is that the reduction of the overlap rate reduces the search space of the Super 4PCS algorithm.

We also recorded the computing time and registration error of the two algorithms for ten point clouds. The results are shown in Table 6, where  $T_S$  and  $T_E$  are the average calculation time of Super 4PCS and Super Edge 4PCS spend in registration of each point cloud like the definition of efficiency metric in Section 4.1.2,  $T_S\% = (T_S - T_E)/T_S$ , which measures the improvement in calculation time of the proposed algorithm, and  $D_R = RMSE_E - RMSE_S$ , which measures the enhancement in accuracy. The penultimate column in Table 6 represents the percentage of the improvement in the computation time of our algorithm compared with Super 4PCS algorithm. The last column represents the improvement in RMSE. From the penultimate column of Table 6, we can see that the efficiencies of proposed algorithm are improved by 66.2% and 89.8%, respectively, compared with Super 4PCS algorithm. Specifically, the registration result of Desk point cloud shows that the proposed algorithm improves by 99.0%. Therefore, the proposed algorithm can dramatically improve the efficiency of point cloud registration. The RMSEs of all point cloud registrations are shown in Figure 22 and Table 6. Firstly, the RMSEs of Super Edge 4PCS registration for DRAGON, BUNNY, HIPPO, ARMADILLO, and BUBBA are higher than Super 4PCS algorithm. However, except for Armadillo, the average RMSE gap of all point clouds is only 0.005m. Armadillo point cloud has more detailed contour information, so the boundary segmentation and overlapping regions extraction produce

bigger errors. Especially, RMSE of the proposed algorithm is 0.003 m lower than the Super 4PCS algorithm with a large sampling size, while it is 0.005 m higher than the Super 4PCS algorithm with a small sampling size. Although the gaps are up and down, these RMSEs are all less than one-tenth of the dimensions of bounding boxes, so these values can be considered sufficiently small. Furthermore, the fine registration algorithm like ICP stitched after the Super Edge 4PCS algorithm can make up for these errors. In summary, the Super Edge 4PCS algorithm complete faster registration than the Super 4PCS algorithm under the guarantee of registration error. It is worth noting that the proposed algorithm has 99% efficiency improvement and 0.01m accuracy improvement in the registration results of large-scale point clouds, called Hokuyo point cloud, compared with Super 4PCS algorithm. Moreover, the registration speed of Armadillo point cloud with a slight overlap rate (0.3) is improved by 99%. Therefore, the Super Edge 4PCS algorithm performs better adaptability than Super 4PCS algorithm, especially in the scene of few overlapping regions and large-scale point clouds.

**Table 6.** Statistics of the running time and registration error of the two algorithms in ten point clouds. TS% and DR represent the improvement in calculation time and accuracy of the proposed algorithm compared to the Super 4PCS algorithm, respectively.

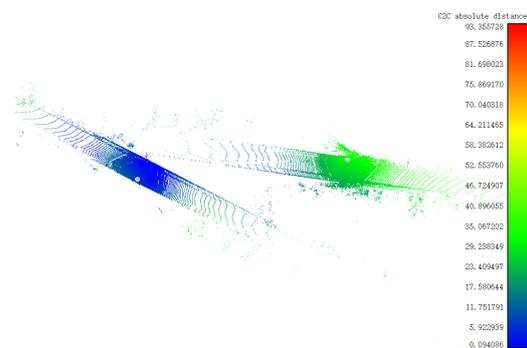
Model	Super 4PCS			Super Edge 4PCS		Comparison	
	Sampling Size (num)	$T_s$ (s)	$RMSE_S$ (m)	$T_E$ (s)	$RMSE_E$ (m)	$T_S\%$	$D_R$ (m)
Dragon	680/1080	11.2/26.4	0.006/0.006	1.3	0.007	88.3%/95.0%	−0.001/−0.001
Bunny	522/953	6.0/13.9	0.014/0.02	1.6	0.03	73.2%/88.6%	−0.01/−0.01
Bubba	732/1378	1.6/6.0	0.003/0.003	2.4	0.004	−51.3%/60.2%	−0.001/−0.001
Hippo	220/413	0.4/1.9	0.02/0.03	1.0	0.04	88.9%/97.9%	−0.02/−0.01
Hokuyo	1300/2000	93.5/770.1	0.03/0.03	6.0	0.02	93.7%/99.2%	0.01/0.01
Armadillo	634/1080	55.9/264.1	0.002/0.002	1.3	0.03	97.7%/99.5%	−0.02/−0.02
Desk	407/913	12.2/42.2	0.016/0.005	0.4	0.002	96.6%/99.0%	0.01/0.003
Person	513/871	3.6/11.9	0.008/0.006	0.4	$3.17 \times 10^{-7}$	88.2%/96.5%	0.008/0.006
Toilet	431/796	0.7/1.8	0.04/0.04	0.4	$3.6 \times 10^{-7}$	35.9%/75.7%	0.04/−0.04
Flower	213/527	3.2/11.4	0.04/0.02	1.6	0.02	51.2%/86.1%	0.03/0.001
Average	-	18.8/114.9	0.02/0.02	1.6	0.01	66.2%/89.8%	0.005/−0.003



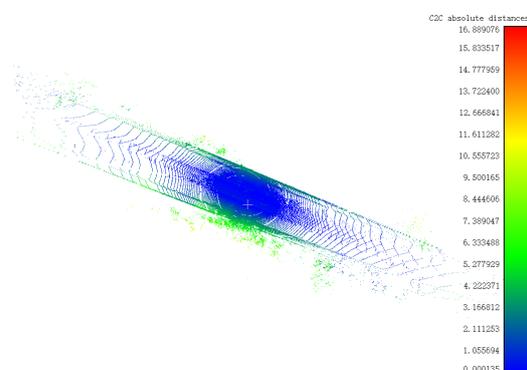
**Figure 22.** RMSEs of ten point clouds after the Super 4PCS algorithm with a large sampling size, a small sampling size and the Super Edge 4PCS algorithm. The corresponding values in the figure are compared with the bounding box dimensions of each point cloud in Table 2, then an objective evaluation can be obtained.

#### 4.7. Limitation Test

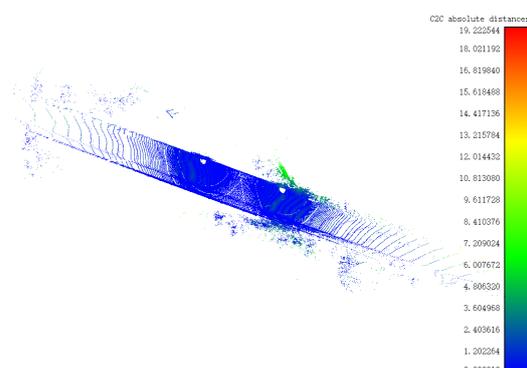
We finally demonstrated the limitation of the bridge point cloud to our algorithm in Figure 23. The scanner needs to be placed inside the bridge during the scanning process, which leads to the uneven density distribution at the boundary of the point cloud. The minimizing-RMSE principle promotes the correspondence of partial point clouds with high cloud density, which results in the wrong matching, as shown in Figure 24. However, the ground truth is shown in Figure 25. We can observe from the figure that the dense parts in the source and target point clouds are not matched, which is contrary to the registration result of proposed algorithm. Therefore, the large difference in point distribution between source point cloud and target point cloud is a challenge to our algorithm. The reason is that the different points distribution leads to a tremendous difference in centroid selection, which affects the selection of correct registration bases. Therefore, the application of proposed algorithm is limited to the registration scenes with the same point distribution.



**Figure 23.** The original poses of bridge scanned in natural scene. The source and target point cloud contain a lot of noise and have uneven point distribution.



**Figure 24.** Registration results of the Super Edge 4PCS algorithm. The proposed algorithm gives priority to the correspondence of the dense part, which is a wrong alignment.



**Figure 25.** Ground-truth bridge registration. The correct registration result is that the dense parts are symmetrically distributed.

#### 4.8. Discussion

It can be concluded that the Super Edge 4PCS algorithm is a more efficient and robust global point cloud registration algorithm than the Super 4PCS method from the above extensive experiments. Mainly, some meaningful discussions can be summarized as below:

##### 4.8.1. Parameters Test

The proposed algorithm takes the search mode  $M$  and the necessity to perform local boundary extraction  $\epsilon_{axis}$  as manual parameters. Although most parameters have fixed reference values, the similarity constraint  $Sim_g$  and the distance gap tolerance  $baseDis$  still vary with different point clouds. Similarly, the Super 4PCS algorithm also needs to be manually set the overlap rate,  $-o$ , sub-sampling size,  $-n$ , and registration accuracy,  $-d$ . Therefore, the automation of parameter of these two algorithms needs to be improved.

##### 4.8.2. Noise Elimination Test

The similarity between the boundary point clouds, which is extracted from point clouds with and without noise reached 0.9998, indicating that the boundary segmentation process can eliminate noise. Nevertheless, the Super 4PCS algorithm does not include noise reduction processing on the original data, so the resistance to noise is lower than our proposed method.

##### 4.8.3. ICP Test

After observing the results of the ICP registration algorithm, we can conclude that both algorithms can provide a good initial value for the ICP algorithm.

##### 4.8.4. Robustness Test

The proposed algorithm is more robust and efficient than the Super 4PCS algorithm under noise levels, outlier levels and overlap rates. Notably, the difference between these two methods is more evident in the restricted cases, such as few overlap regions and vast outlier levels. Therefore, the Super Edge 4PCS algorithm also shows a more comprehensive application than the Super 4PCS algorithm.

##### 4.8.5. Computational Test

After comparing the calculation time of CSE and CSV, we can conclude that the Super Edge 4PCS algorithm has improved calculation efficiency in both parts. Besides, the difference in calculation time and accuracy on various point clouds were summarized. The algorithm in this paper increases the calculation efficiencies by 66.2% and 89.8% compared with the Super 4PCS algorithm under the guarantee of adequately small registration errors.

##### 4.8.6. Limitation Test

It can be seen from the limitation test that limited application scenarios still exist, that is, the scene where the point distribution is quite different. In this case, the Super Edge 4PCS algorithm produces incorrect registration results.

These meaningful observations can provide us with helpful guidance when applying the Super Edge 4PCS algorithm to a new scene.

## 5. Conclusions

This paper proposed a new algorithm called Super Edge 4PCS for coarse registration. First of all, the algorithm in this paper achieves the same linear complexity  $O(n)$  as the Super 4PCS algorithm in theory. Besides, the proposed algorithm shows satisfactory performance in robustness, accuracy and efficiency, benefiting from the separated overlapping regions extraction (ORE) and acquisition of corresponding bases (ACB). Expressly, the accuracy of Super Edge 4PCS algorithm can still be guaranteed even in extreme cases like few overlapping regions. We tested datasets with different properties in experimental parts to compare the performance of the proposed method with the Super 4PCS algorithm. The

results demonstrate that our algorithm is more reliable than the Super 4PCS in efficiency, accuracy, and robustness.

Although the Super Edge 4PCS algorithm achieved surprising results in efficiency and accuracy, there are still two improvable directions for further research:

1. We will improve the automatic nature and intelligence of the parameters in future research.
2. We will find a new solution to tackle the challenge caused by the huge difference in point distribution.

**Author Contributions:** Conceptualization, S.L.; methodology, S.L. and L.G.; software, S.L. and R.L.; validation, S.L. and J.L.; investigation, S.L. and J.L.; resources, S.L. and R.L.; data curation, S.L. and L.G.; Writing, S.L. and R.L. supervision, L.G.; project administration, S.L. and L.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The public data set used in this article mainly includes four parts: Stanford University open source point cloud model data set: <http://graphics.stanford.edu/data/3Dscanrep/> (accessed on 20 April 2021). Princeton University open source point cloud model data set: <http://modelnet.cs.princeton.edu/> (accessed on 20 April 2021). Super 4PCS algorithm open source data set: <http://geometry.cs.ucl.ac.uk/projects/2014/super4PCS/> (accessed on 20 April 2021). Automatic system laboratory open source data set: <https://projects.asl.ethz.ch/datasets> (accessed on 20 April 2021).

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

## Abbreviations

The following abbreviations are used in this manuscript:

ICP	Iterative Closest Point
4PCS	4Points Congruent Sets
BS	Bounding segmentation
ORE	Overlapping Regions Extraction
ACB	Acquisition of Corresponding Bases
ESF	Ensemble of shape functions
DPG	Difference between calculated transformation parameters and ground truth

## Appendix A

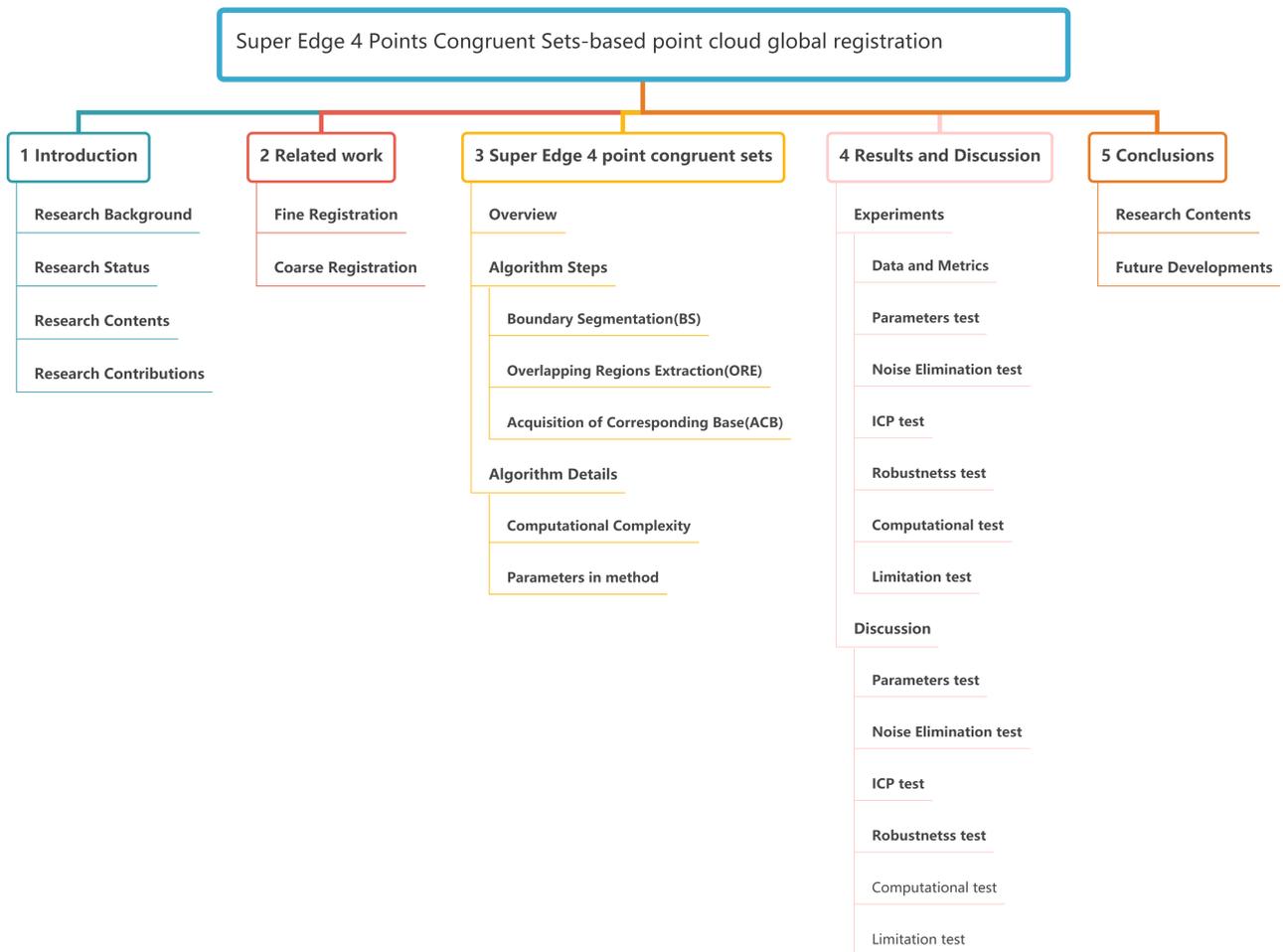


Figure A1. The sketch figure of our paper.

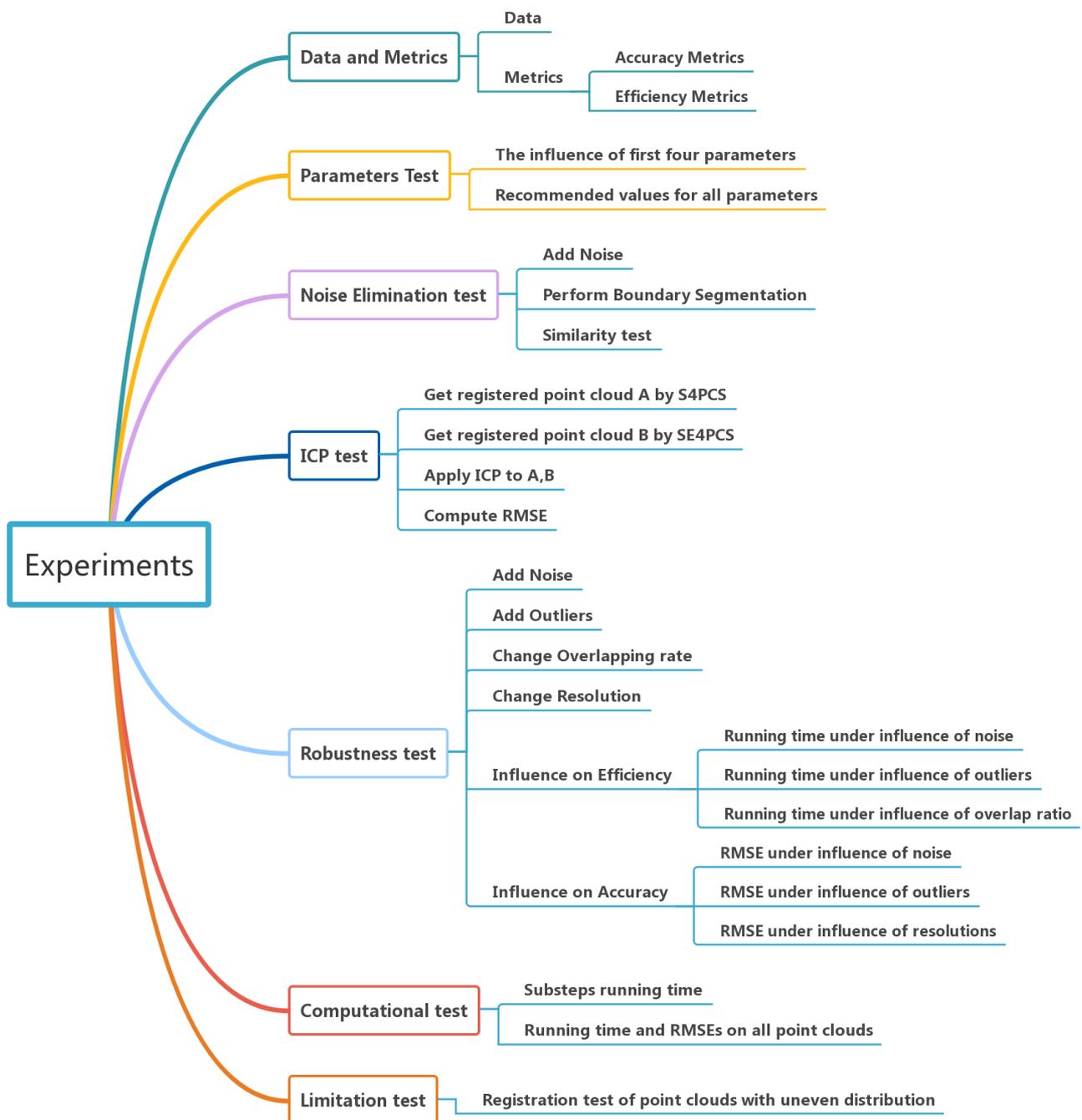


Figure A2. The sketch figure of experiments.

## References

1. Besl, P.J.; McKay, N.D. Method for registration of 3-d shapes. *Sensor Fusion IV: Control Paradigms and Data Structures. Int. Soc. Opt. Photonics* **1992**, *1611*, 586–606.
2. Aiger, D.; Mitra, N.J.; Cohen-Or, D. *4-Points Congruent Sets for Robust Pairwise Surface Registration*; ACM SIGGRAPH 2008 Papers; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1–10.
3. Nicolas, M.; Dror, A.; Niloy, J.M. Super 4pcs fast global pointcloud registration via smart indexing. *Comput. Graph. Forum* **2014**, *33*, 5.
4. Huang, J.; Kwok, T.; Zhou, C. V4pcs: Volumetric 4pcs algorithm for global registration. *J. Mech. Des.* **2017**, *139*, 11. [\[CrossRef\]](#)
5. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [\[CrossRef\]](#)
6. Miola, G.A.; Santos, D.R.D. A framework for registration of multiple point clouds derived from a static terrestrial laser scanner system. *Appl. Geomat.* **2020**, *12*, 409–425. [\[CrossRef\]](#)

7. Xu, Y.; Boerner, R.; Yao, W.; Hoegner, L.; Stilla, U. Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets. *ISPRS J. Photogramm. Remote Sens.* **2009**, *151*, 106–123. [[CrossRef](#)]
8. Li, P.; Wang, J.; Zhao, Y.; Wang, Y.; Yao, Y. Improved algorithm for point cloud registration based on fast point feature histograms. *J. Appl. Remote Sens.* **2016**, *10*, 045024. [[CrossRef](#)]
9. Lei, H.; Jiang, G.; Quan, L. Fast descriptors and correspondence propagation for robust global point cloud registration. *IEEE Trans. Image Process.* **2017**, *26*, 3614–3623. [[CrossRef](#)]
10. Huang, X.; Zhang, J.; Fan, L.; Wu, Q.; Yuan, C. A systematic approach for cross-source point cloud registration by preserving macro and micro structures. *IEEE Trans. Image Process.* **2017**, *26*, 3261–3276. [[CrossRef](#)]
11. Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. A regularized point cloud registration approach for orthogonal transformations. *J. Glob. Optim.* **2020**, 1–23. [[CrossRef](#)]
12. Rabbani, T.; Dijkman, S.; Heuvel, F.V.; Vosselman, G. An integrated approach for modelling and global registration of point clouds. *ISPRS J. Photogramm. Remote Sens.* **2007**, *61*, 355–370. [[CrossRef](#)]
13. Li, W.; Song, P. A modified icp algorithm based on dynamic adjustment factor for registration of point cloud and cad model. *Pattern Recognit. Lett.* **2015**, *65*, 88–94. [[CrossRef](#)]
14. Prokop, M.; Shaikh, S.A.; Kim, K. Low overlapping point cloud registration using line features detection. *Remote Sens.* **2020**, *12*, 61. [[CrossRef](#)]
15. Zhang, X.; Yang, B.; Li, Y.; Zuo, C.; Wang, X.; Zhang, W. A method of partially overlapping point clouds registration based on differential evolution algorithm. *PLoS ONE* **2018**, *13*, e0209227. [[CrossRef](#)] [[PubMed](#)]
16. Bae, K.H.; Lichti, D.D. A method for automated registration of unorganised point clouds. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 36–54. [[CrossRef](#)]
17. Fengguang, X.; Biao, D.; Wang, H.; Min, P.; Liqun, K.; Xie, H. A local feature descriptor based on rotational volume for pairwise registration of point clouds. *IEEE Access* **2020**, *8*, 100120–100134. [[CrossRef](#)]
18. Gressin, A.; Mallet, C.; Demantké, J.; David, N. Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 240–251. [[CrossRef](#)]
19. Kim, P.; Chen, J.; Cho, Y.K. Automated point cloud registration using visual and planar features for construction environments. *J. Comput. Civil Eng.* **2018**, *32*, 04017076. [[CrossRef](#)]
20. Kwon, S.; Lee, M.; Lee, M.; Lee, S.; Lee, J. Development of optimized point cloud merging algorithms for accurate processing to create earthwork site models. *Autom. Constr.* **2013**, *35*, 618–624. [[CrossRef](#)]
21. Kim, C.; Son, H.; Kim, C. Fully automated registration of 3d data to a 3d cad model for project progress monitoring. *Autom. Constr.* **2013**, *35*, 587–594. [[CrossRef](#)]
22. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–1552. [[CrossRef](#)]
23. Das, A.; Diu, M.; Mathew, N.; Scharfenberger, C.; Servos, J.; Wong, A.; Zelek, J.S.; Clausi, D.A.; Waslander, S.L. Mapping, planning, and sample detection strategies for autonomous exploration. *J. Field Robot.* **2014**, *31*, 75–106. [[CrossRef](#)]
24. Holz, D.; Ichim, A.E.; Tombari, F.; Rusu, R.B.; Behnke, S. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robot. Autom. Mag.* **2015**, *22*, 110–124. [[CrossRef](#)]
25. Segal, A.; Haehnel, D.; Thrun, S. Generalized-icp. *Robotics Sci. Syst.* **2009**, *2*, 435.
26. Agamennoni, G.; Fontana, S.; Siegwart, R.Y.; Sorrenti, D.G. Point clouds registration with probabilistic data association. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4092–4098.
27. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing icp variants on real-world data sets. *Auton. Robot.* **2013**, *34*, 133–148. [[CrossRef](#)]
28. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2241–2254. [[CrossRef](#)]
29. Stechschulte, J.; Ahmed, N.; Heckman, C. Robust low-overlap 3-d point cloud registration for outlier rejection. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 7143–7149.
30. Rusinkiewicz, S.; Levoy, M. Efficient variants of the icp algorithm. In Proceedings of the Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
31. Bouaziz, S.; Tagliasacchi, A.; Pauly, M. Sparse iterative closest point. In *Computer Graphics Forum*; Wiley Online Library: New York City, NY, USA, 2013; Volume 32, pp. 113–123.
32. Xu, Z.; Xu, E.; Zhang, Z.; Wu, L. Multiscale sparse features embedded 4-points congruent sets for global registration of tps point clouds. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 286–290. [[CrossRef](#)]
33. Liu, H.; Liu, T.; Li, Y.; Xi, M.; Li, T.; Wang, Y. Point cloud registration based on mcmc-sa icp algorithm. *IEEE Access* **2019**, *7*, 73637–73648. [[CrossRef](#)]
34. Habib, A.; Ghanma, M.; Morgan, M.; Al-Ruzouq, R. Photogrammetric and lidar data registration using linear features. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 699–707. [[CrossRef](#)]
35. Hänsch, R.; Weber, T.; Hellwich, O. Comparison of 3d interest point detectors and descriptors for point cloud fusion. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *2*, 57. [[CrossRef](#)]

36. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
37. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [[CrossRef](#)]
38. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
39. Yang, B.; Zang, Y. Automated registration of dense terrestrial laser-scanning point clouds using curves. *ISPRS J. Photogramm. Remote Sens.* **2014**, *95*, 109–121. [[CrossRef](#)]
40. Dold, C.; Brenner, C. Automatic matching of terrestrial scan data as a basis for the generation of detailed 3d city models. *Int. Arch. Photogramm. Remote Sens.* **2004**, *35 Pt B3*, 1091–1096.
41. Brenner, C.; Dold, C. Automatic relative orientation of terrestrial laser scans using planar structures and angle constraints. *ISPRS Workshop Laser Scanning* **2007**, *200*, 84–89.
42. Bosché, F. Plane-based registration of construction laser scans with 3d/4d building models. *Adv. Eng. Informatics* **2012**, *26*, 90–102. [[CrossRef](#)]
43. Dam, Y.E.; Mu Wook, P.; Sun Woong, K.; Jang Ryul, K.; Dong Yeob, H. Coregistration of terrestrial lidar points by adaptive scale-invariant feature transformation with constrained geometry. *Autom. Constr.* **2012**, *25*, 49–58.
44. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [[CrossRef](#)] [[PubMed](#)]
45. Bueno, M.; Bosché, F.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. 4-plane congruent sets for automatic registration of as-is 3d point clouds with 3d bim models. *Autom. Constr.* **2018**, *89*, 120–134. [[CrossRef](#)]
46. Sun, J.; Zhang, R.; Du, S.; Zhang, L.; Liu, Y. Global adaptive 4-points congruent sets registration for 3d indoor scenes with robust estimation. *IEEE Access* **2020**, *8*, 7539–7548. [[CrossRef](#)]
47. Mohamad, M.; Rappaport, D.; Greenspan, M. Generalized 4-points congruent sets for 3d registration. In *Proceedings of the 2014 2nd International Conference on 3D Vision*, Tokyo, Japan, 8–11 December 2014; Volume 1, pp. 83–90.
48. Mohamad, M.; Ahmed, M.T.; Rappaport, D.; Greenspan, M. Super generalized 4pcs for 3d registration. In *Proceedings of the 2015 International Conference on 3D Vision*, Lyon, France, 19–22 October 2015; pp. 598–606.
49. Theiler, P.W.; Wegner, J.D.; Schindler, K. Keypoint-based 4-points congruent sets—automated marker-less registration of laser scans. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 149–163. [[CrossRef](#)]
50. Ge, X. Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 344–357. [[CrossRef](#)]
51. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Matching 3d models with shape distributions. In *Proceedings of the Proceedings International Conference on Shape Modeling and Applications*, Genova, Italy, 7–11 May 2001; pp. 154–166.
52. Wohlkinger, W.; Vincze, M. Ensemble of shape functions for 3d object classification. In *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, Karon Beach, Thailand, 7–11 December 2011; pp. 2987–2992.
53. The Dataset of Super 4pcs Algorithm. Available online: <http://geometry.cs.ucl.ac.uk/projects/2014/super4PCS/> (accessed on 20 April 2021).
54. The Stanford 3d Scanning Repository. Available online: <http://graphics.stanford.edu/data/3Dscanrep/> (accessed on 20 April 2021).
55. Autonomous Systems Lab Datasets. Available online: <https://projects.asl.ethz.ch/datasets> (accessed on 20 April 2021).
56. The Datasets of Princeton University. Available online: <http://modelnet.cs.princeton.edu/> (accessed on 20 April 2021).