IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 April 2023, accepted 18 April 2023, date of publication 26 April 2023, date of current version 4 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270502

RESEARCH ARTICLE

Fast and High Accuracy 3D Point Cloud Registration for Automatic Reconstruction From Laser Scanning Data

ANRAN XU, LEI RAO, GUANGYU FAN, AND NIANSHENG CHEN, (Member, IEEE)
Laboratory of Robot and Intelligent Technology, Shanghai Dianji University, Shanghai 201306, China
College of Electronic Information, Shanghai Dianji University, Shanghai 201306, China

Corresponding author: Lei Rao (raol@sdju.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61702320.

ABSTRACT Point cloud registration from laser scanning data is a technique to establish the mapping relationship between source and target point clouds, which has been widely used in automatic 3D reconstruction, pose estimation, localization, and navigation. While algorithms like Super4PCS and MSSF-4PCS can achieve registration without initial poses, they are relatively slow, less accurate, and require iterations. To address these issues, we propose a 3D point cloud registration algorithm based on interval segmentation and multi-dimensional feature. Firstly, the source and target point clouds are segmented internally and the point cloud curvature is designed to narrow down the search range for the registration between the segmented point clouds. Secondly, the corresponding four-point sets in the segmented areas of the source and target point clouds are determined using affine invariance constraints. Finally, a multi-dimensional feature vector based on curvature features and fast point feature histogram is established to determine the unique corresponding four-point set pairs, and the rigid body transformation matrix is solved accordingly. Our algorithm is tested on publicly available 3D point cloud data models Bunny, Dino, Dragon, and Horse from Stanford University. Results showed that our algorithm improved registration accuracy by 24.39% and registration efficiency by 46.21% compared to the MSSF-4PCS point cloud registration algorithm. Multiple sets of experimental results confirmed the effectiveness of our algorithm. The proposed 3D point cloud registration is proved to be fast with high accuracy, which can be utilized for automatic segmentation, reconstruction, and modelling from Laser Scanning Data.

INDEX TERMS Point cloud registration, point cloud curvature, affine invariance, multi-dimensional features.

I. INTRODUCTION

With the advent of 3D imaging technology, 3D reconstruction has become increasingly prevalent in fields such as robotics, intelligent manufacturing, and heritage conservation. 3D LIDAR [1], which captures depth information and represents it as point clouds, is a widely used technology in 3D reconstruction [2]. However, due to device scanning angle limitations and object shape, acquiring the complete point cloud of an object often requires multiple views [3]. This necessitates the registration of corresponding point clouds to create a complete 3D point cloud model for subsequent reconstruction. The primary objective of point cloud registration is to determine the optimal mapping transformation between different point cloud frames. Point cloud registration algorithms have found widespread use in 3D reconstruction and robotics [4]. As computer vision technology matures [5], the demands for point cloud registration accuracy and real-time performance are increasing.

There are various point cloud registration algorithms [6] that can be broadly classified into three categories: global search-based, local feature-based, and deep learning-based. Global search-based algorithms involve traversing all point cloud data to find corresponding points, making them

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

computationally intensive and less real-time for large-scale point clouds. Local feature-based algorithms rely on feature points to determine point cloud correspondence, but may result in matching errors for complexly featured point clouds. Deep learning-based algorithms obtain point cloud transformation matrices through neural networks, but require high computational costs.

To address the accuracy and real-time limitations of current point cloud registration algorithms, an effective approach is to segment point clouds into subsets. In this regard, we propose a multi-dimensional feature-based point cloud registration algorithm that leverages interval segmentation. By utilizing interval segmentation and multidimensional features, we improve both the accuracy and efficiency of point cloud registration. In summary, the main contributions of this paper are as follows:

1) We propose an interval partitioning method to reduce the computational time required for point cloud registration. In the preliminary process, the point cloud is segmented into several subintervals based on distance. A histogram is then created based on point curvature, and Bhattacharyya distance is used to determine corresponding subintervals for point cloud registration. This significantly narrows down the search range of corresponding points and solves the problem of slow registration due to a large number of point clouds. As a result, the efficiency of point cloud registration is improved.
2) We convert the smallest unit of point cloud data into a four-point set composed of approximately coplanar data points. Then, we use the affine invariance constraint to find the corresponding four-point set pairs in the corresponding two point cloud subintervals, which improves the efficiency of point cloud registration.
3) We create multi-dimensional feature vectors for four-point sets by utilizing curvature features and fast point feature histograms. Then, we use a feature similarity function to identify unique corresponding pairs of four-point sets. This approach effectively addresses the issue of varying point cloud density, which previously resulted in multiple four-point set pairs and subsequently reduced registration accuracy. As a result, our method significantly improves the accuracy of point cloud registration.
4) We performed experimental validation on publicly available 3D point cloud data models including Bunny, Dino, Dragon, and Horse, which were provided by Stanford University. Our proposed algorithm resulted in a 24.39% increase in registration accuracy and a 46.21% improvement in registration efficiency compared to the MSSF-4PCS point cloud registration algorithm. Several sets of experimental results verify the effectiveness of our algorithm.

This paper is organized as follows. Section II provides a review of recent types of point cloud registration algorithms and related works. In Section III, we present the principles and procedures of our point cloud interval segmentation algorithms. Section IV focuses on multidimensional features.The results are discussed in Section V. The experimental results for validation are presented in Section VII, and finally, in Section VI, we draw conclusions to summarize the paper.

II. RELATED WORK

The global search-based point cloud registration algorithm determines the optimal transformation by imposing certain conditions in the global scope and iterating through the loop. The most widely used global search-based point cloud registration algorithm is the Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [7] in 1992. The principle of the ICP algorithm is to find the corresponding point set in the source and target point clouds and establish the correspondence between the two clusters of point clouds by continuously iterating the transformation matrix of the two clusters of point clouds. However, the ICP algorithm suffers from issues such as low computational efficiency, the tendency to fall into the local optimum, and high dependence on the initial poses. To overcome these challenges, several improvement algorithms have been proposed, such as GICP [8], NICP [9], and SACICP [10], [11]. Among these, SACICP divides the original ICP into two steps: coarse registration and fine registration. Firstly, the FPFH of the point cloud is computed by the Sample Consensus Initial Registration (SAC-IA) algorithm [11], which finds the corresponding points with similar FPFH features in the target point cloud and calculates the rigid body transformation matrix between the corresponding points to achieve coarse registration. The rough registration step of the SACICP algorithm partially addresses the computational efficiency and initial registration dependence issues of the ICP algorithm. However, for many point clouds or irregularly distributed point clouds, the FPFH computation of coarse registration is significant, which significantly impacts the real-time performance of the algorithm.

The point cloud registration algorithm based on local features [12] describes features based on point neighborhood information, establishes feature descriptors for the points, and determines correspondences between two frames of point clouds using these descriptors to complete registration. Widely used algorithms include the consistent four-point congruent sets (4PCS) method [13] and its derivative algorithms Super4PCS [14] and MSSF-4PCS [15]. The 4PCS algorithm uses a group of four coplanar points in the source point cloud as the base set and finds all four-point locations on the target point cloud that are approximately congruent to the base set. The 4PCS algorithm does not require point cloud denoising and has lower overlap requirements, but it suffers from problems such as misregistration of four-point sets and high computational effort. To address these issues, Super4PCS optimizes the quad point set identification method, streamlines the range of corresponding quad points, and determines related quad point sets using an optimized indexing method

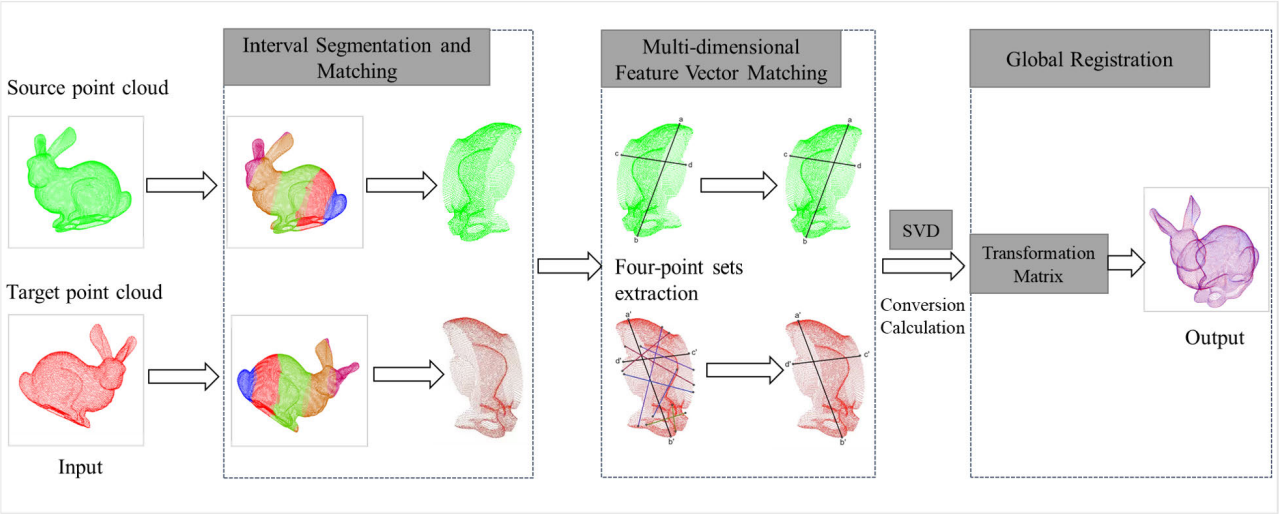


FIGURE 1. Algorithm framework diagram.

to solve the transformation matrix. This method improves the screening mechanism for the corresponding four-point set in the target point cloud, reducing the search range, but it does not solve the misregistration problem. MSSF-4PCS addresses this problem by incorporating sparse features to improve registration accuracy. However, the computation of light features increases significantly with large-scale point clouds, posing a significant challenge to the algorithm's real-time performance.

Deep learning-based point cloud registration algorithms [16] can be divided into two categories: those based on point cloud feature extraction [17] and those that are end-to-end deep learning [18] methods. The deep learning registration method based on point cloud feature extraction is a registration method that combines deep learning with optimization methods. This approach primarily uses deep neural networks to learn the geometric features of point clouds, and then iteratively applies optimization methods to obtain the transformation matrix. For example, Choy et al. [19] proposed a fully convolutional geometric feature network (FCGF) to extract geometric features of point clouds using a fully convolutional network. The method generates high-resolution features based on the sparsity of point clouds by inputting a sparse tensor using a unit cubic kernel. The end-to-end deep learning-based registration method takes two clusters of point clouds as input and directly outputs the transformation matrix through the neural network. For instance, Ao et al. [20] proposed SpinNet to extract rotation invariant features. SpinNet consists of a spatial point converter and a neural feature extractor, which learn local features for 3D point cloud registration. However, the end-to-end deep learning-based registration method requires a large graphics processing unit (GPU) memory and incurs a high computational cost.

This paper presents the principle framework of the algorithm, as shown in Figure 1. The input data source consists of two frames of point clouds to be aligned. Firstly, the point cloud is partitioned into intervals based on distance and aligned to the corresponding interval using the curvature histogram. Secondly, the complementary multiple four-point sets are determined based on the affine invariance constraint in the corresponding point cloud intervals. Finally, the rigid body transformation matrix is derived by obtaining the unique corresponding four-point set through multi-dimensional feature registration.

III. POINT CLOUD INTERVAL SEGMENTATION AND MATCHING

A. PROBLEM DESCRIPTION

The goal of point cloud registration is to find the optimal transformation matrix between the source point cloud P and the target point cloud Q. Specifically, for each point p_i in P, we aim to find the point q_i in Q with the shortest Euclidean distance. To achieve this, we iteratively calculate the transformation matrix using the corresponding set of point pairs (p_i, q_i), and use equation (1) as the target error function [7]. We repeat this process until the value of f is below a predetermined threshold, at which point we have obtained the optimal transformation matrix.

$$f(R, T) = \frac{1}{k} \sum_{i=1}^{k} \|q_i - (Rp_i - T)\|^2 \tag{1}$$

The rotation transformation matrix is denoted as R, the translation transformation matrix as T, and k represents the number of corresponding points. The iterative computation involved in solving the nearest issues using ICP can lead to significant computational problems and insufficient real-time performance. To overcome this challenge and reduce registration time, this algorithm partitions the source and target point clouds into several subintervals. It then identifies the matching relationship between these subintervals and aligns

the point clouds in the matched subintervals to reduce the search range for corresponding points.

B. INTERVAL SEGMENTATION

The source and target point clouds are segmented by using distance segmentation [21]. Firstly, determine the coordinates of the point pair $a(a_x, a_y, a_z)$, $b(b_x, b_y, b_z)$ with the maximum distance between two points in the point cloud, and also define the width d of the interval as shown in equation (2), where n represents the number of segmented intervals. Define the segmented source and target point clouds as $P[P_1, P_2..P_n]$ and $Q[Q_1, Q_2..Q_n]$, respectively, where $P_x(x \in n)$ and $Q_y(y \in n)$ represent the segmented source and target point cloud subintervals.

$$d = \frac{\sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}}{n} \tag{2}$$

Take n=5 as an example, take a as the sphere's center and $x \times d$ as the radius to make a sphere tangent (x=1, 2, 3, 4, 5), and the tangent divides the point cloud into five parts. The effect after the point cloud interval segmentation is shown in Figure 2. Figure 2 (a)-(d) shows the impact before the source point cloud segmentation, after the source point cloud segmentation, before the target point cloud segmentation, and after the target point cloud segmentation, the general procedure is described in Algorithm 1, Point cloud P as an example.

Algorithm 1 Interval Segmentation

Require: pointcloud P and n(n is the number of intervals, for example n=5).

Ensure: Interval point cloud after segmentation.

Define a, b to be the point pair with the largest distance in the point cloud P.
$D(a, b) = 0$
for $p_i \in P$ **do**
  for $p_j \in P$ **do**
    if $d(p_i, p_j) > d(a, b)$ **then**
      $a \leftarrow p_i, b \leftarrow q_i$
      $i++, j++$
    end if
  end for
end for
Define interval point cloud P_1, P_2, P_3, P_4 and P_5.
for $p_i \in P$ **do**
  if $D(a, p_i) > D(a, b)/n$ **then**
    insert p_i into interval point cloud P_1
  end if
  if $D(a, p_i) > D(a, b)/n$ && $D(a, p_i) < 2D(a, b)/n$ **then**
    insert p_i into interval point cloud P_2
  end if
  if $D(a, p_i) > 2D(a, b)/n$ && $D(a, p_i) < 3D(a, b)/n$ **then**
    insert p_i into interval point cloud P_3
  end if
  if $D(a, p_i) > 3D(a, b)/n$ && $D(a, p_i) < 4D(a, b)/n$ **then**
    insert p_i into interval point cloud P_4
  end if
  if $D(a, p_i) > 4D(a, b)/n$ && $D(a, p_i) < 5D(a, b)/n$ **then**
    insert p_i into interval point cloud P_5
  end if
end for
return P_1, P_2, P_3, P_4 and P_5.

C. POINT CLOUD SUBINTERVAL MATCHING

To find the correspondence between the segmentation point clouds P_x and Q_y in the source point cloud $P[P_1, P_2..P_n]$ and the target point cloud $Q[Q_1, Q_2..Q_n]$, the curvature of all segmentation point clouds P_x and Q_y is first calculated to obtain the curvature histogram, and then the correspondence between the segmentation point clouds P_x and Q_y is found by the similarity of the histogram based on the grayscale histogram matching principle [22]. Thus realizing the subinterval matching between the source and target point clouds.

1) CURVATURE CALCULATION

Point cloud surface information primarily consists of the coordinate information of points. While the coordinate information changes with the position of the point cloud, the relative position information of points and their neighbors remains constant. The curvature of a point cloud is an inherent geometric feature that can describe the geometric information of a point and its neighbors. However, surface fitting methods for calculating curvature can produce significant errors due to the presence of outliers and noise points. Therefore, this algorithm characterizes point cloud features by calculating the normal curvature [23].

For any point p in the point cloud, let the average unit vector of p be $\overrightarrow{N}$ and construct a local coordinate system $L\{p, X, Y, \overrightarrow{N}\}$ as shown in Figure 3(a), where $\overrightarrow{N}$ denotes the average vector of point p and X and Y are orthogonal unit vectors. Suppose there are m nearest neighbors near point p and $l_i(i \in m)$ is the i-th nearest neighbor of point p. The average vector of l_i is $\overrightarrow{M}_i$, and the normal curvature of point p concerning point l_i is defined as k_{l_i}.

Let the coordinates of p be (0, 0, 0), the coordinates of l_i be (x_i, y_i, z_i), and the coordinates of $\overrightarrow{M}_i$ be $(n_{x,i}, n_{y,i}, n_{z,i})$. As shown in Figure 3(b), a triangle pOl_i is obtained by making a close circle with points p, l_i, and O.

Then the average curvature k_{l_i} of the point p concerning the point l_i can be obtained by equation (3) and equation (4).

$$k_{l_i} = -\frac{\sin(\beta)}{|pq_i|\sin(\alpha)} \approx -\frac{n_{xy}}{\sqrt{n_{xy}^2 + n_z^2}\sqrt{x_i^2 + y_i^2}} \tag{3}$$

$$n_{xy} = \frac{x_i n_{x,i} + y_i n_{y,i}}{\sqrt{x_i^2 + y_i^2}}, \quad n_z = n_{z,i} \tag{4}$$

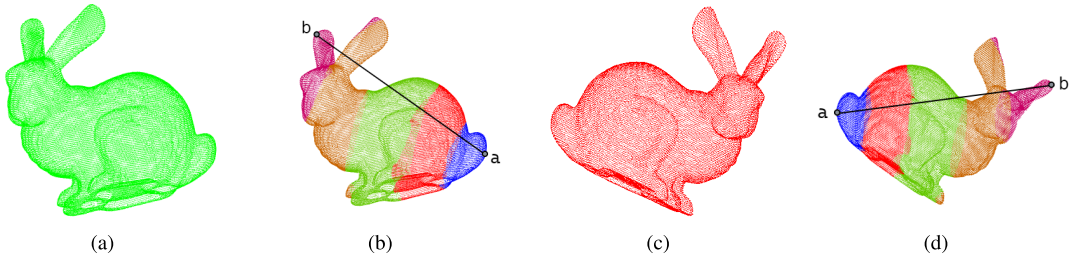
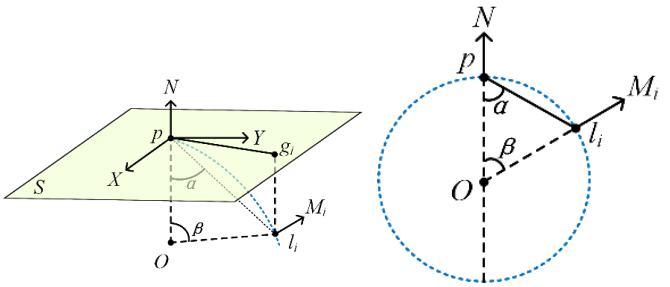


FIGURE 2. Point cloud segmentation diagram.



(a) Local coordinate system L

(b) Triangles $pOli$ is obtained from close circles, neighboring points, and normal vectors

FIGURE 3. Schematic diagram of the chord standard vector method.

Where α is the angle between $-\overrightarrow{N}$ and $\overrightarrow{pl_i}$, β is the angle between $-\overrightarrow{N}$ and $-\overrightarrow{M}_i$. From this, the normal curvature of each point in the point cloud can be calculated.

2) HISTOGRAM SIMILARITY

As shown in Figure 4(a) and 4(b), the distribution of characteristic curvature points of the source point cloud P and its curvature characteristic histogram are shown, from which it can be found that the points of different curvatures are distributed in the corresponding curvature intervals, for example, the points in the interval with more significant curvature (>0.006) are mainly distributed in the folds and edges of the bunny point cloud.

Based on the above method, the curvature histograms of the segmented subintervals P_x and Q_y can be obtained. And the similarity between the curvature histograms of P_x and Q_y can be obtained by using the Bhattacharyya distance [24].The formula for calculating the Bhattacharyya distance is shown in equation (5).

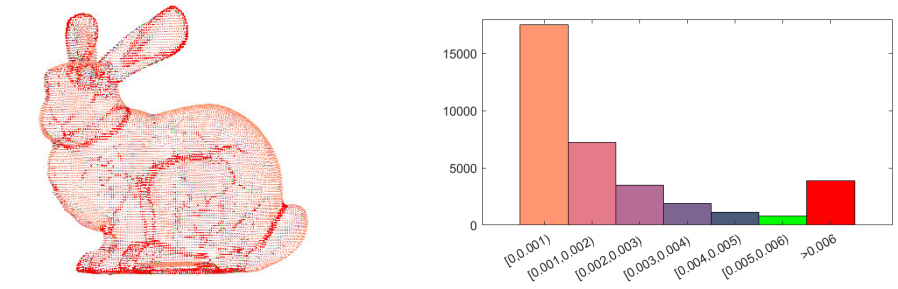
$$d\left(H_{P_x}, H_{Q_y}\right) = \sqrt{1 - \frac{1}{\sqrt{\overline{H_{P_x}}\overline{H_{Q_y}}n^2}}\sum_{m=1}^{n}\sqrt{H_{P_x(m)} \cdot H_{Q_y(m)}}} \tag{5}$$

where $\overline{H_{P_x}} = \frac{1}{n}\sum_{m=1}^{n} H_{P_x(m)}$, H_{P_x} and H_{Q_y} are the curvature histograms of P_x and Q_y. The shorter the Bhattacharyya distance between the segmented point cloud interval P_x and Q_y, the higher the similarity between P_x and Q_y, from which the matching relationship between the segmented point clouds can be obtained, thus realizing the subinterval matching between the source and target point clouds, and the subsequent point cloud registration and transformation matrix calculation only needs to be performed within the subinterval with the corresponding relationship. For a better illustration, a brief pseudo-code is stated below as Algorithm 2. Compared with the traditional global search for corresponding points, this algorithm significantly reduces the search range of point cloud registration, and the number of point clouds to be aligned is reduced to $1/n^2$, which greatly improves the efficiency of point cloud search.

IV. MULTI-DIMENSIONAL FEATURE MATCHING

A. AFFINE INVARIANCE CONSTRAINT

Once point cloud subintervals are aligned to their corresponding counterparts, point cloud registration can be performed on each subinterval. Existing algorithms for point cloud registration with a large number of point clouds typically use global search-based point cloud registration or registration methods based on geometric feature descriptions. However, these algorithms often suffer from extensive computation, low efficiency, and low accuracy. To address these challenges, this paper presents a subinterval point cloud registration method based on the 4-point all-equal set registration algorithm. Figure 5 shows a schematic diagram of 4PCS, where the left side displays the segmented source point cloud interval P_x and the right side displays the corresponding segmented



(a) Curvature feature point distribution (b) Histogram of curvature characteristics

FIGURE 4. Curvature feature points and feature histogram.

Algorithm 2 Point Cloud Interval Matching

Require: $P_1, P_2, \ldots P_n, Q_1, Q_2, \ldots Q_n$ and Bhattacharyya distance threshold d_B
Ensure: Correspondence between P_i and Q_i.
1: P_1, P_2,$\ldots P_n$ and Q_1, Q_2,$\ldots Q_n$ are the interval point cloud obtained by algorithm 1.
2: **for** $P_i \in P$, $Q_i \in Q$ **do**
3: **for** $p_j \in P_i$, $q_j \in Q_i$ **do**
4: Calculate average curvature according to Eq.(3) and Eq.(4)
5: **end for**
6: Get histogram H_{P_1}, H_{P_2},$\ldots$, H_{P_n} and H_{Q_1}, H_{Q_2},$\ldots$, H_{Q_n} by corresponding curvature.
7: **end for**
8: **for** $P_i \in P$ **do**
9: **for** $Q_i \in Q$ **do**
10: Calculate the Bhattacharyya distance $d(H_{P_i}, H_{Q_i})$ between H_{P_i} and H_{Q_i} by Eq.(5)
11: **if** $d(H_{P_i}, H_{Q_i}) < d_B$ **then**
12: Q_i is the corresponding interval point cloud of P_i
13: **end if**
14: **end for**
15: **end for**
16: **return** Correspondence between P_1, P_2,$\ldots P_n$ and Q_1, Q_2,$\ldots Q_n$.

interval Q_y for the target point cloud. The 4PCS algorithm is based on affine invariance, which involves finding four approximately coplanar point pairs (i.e., an approximately congruent four-point set) in the source and target point clouds that have similar intersection points and distances before and after rigid body transformation. To obtain the transformation matrix, multiple sets of corresponding four-point sets are selected iteratively using the RANSAC algorithm framework, followed by the least squares method.

First, select the set of four points $B = \{a, b, c, d\}$ in the subinterval P_x of the source point cloud as shown in Figure 6(a), first randomly select the three points a, b, c that form the triangle with the largest possible area, and after determining the three points, traverse all the points in the source point cloud interval and select the fourth point d that is as coplanar as possible with the plane composed of these three points.

The distance constraint invariants d_1, d_2 are calculated as shown in equation (6). The ratio constraint invariants r_1, r_2 are shown in equation (7). For the target interval point cloud, any pair of points $\{q1, q2\}$ is taken to form a line segment. For each pair of points, q_1 and q_2 in the target point cloud subinterval Q_y, the intersection position can be calculated by equation (8), and four possible intersection points are obtained as shown in Figure 6(b).

$$d_1 = |b - a|, \quad d_2 = |c - d| \tag{6}$$

$$r_1 = \frac{\|a - e\|}{\|a - b\|}, \quad r_2 = \frac{\|c - e\|}{\|c - d\|} \tag{7}$$

$$\begin{cases} e_1 = q_1 + r_1 (q_2 - q_1) \\ e_2 = q_1 + r_2 (q_2 - q_1) \end{cases} \tag{8}$$

For any pair of points, if their intersection points are the same, then they are the four-point sets that meet the requirements. From the affine invariance, we can get the four-point set pairs in the target point cloud subinterval Q_y that are approximately equal to the four-point set in the source point cloud subinterval P_x.

B. MULTI-DIMENSIONAL FEATURE VECTOR

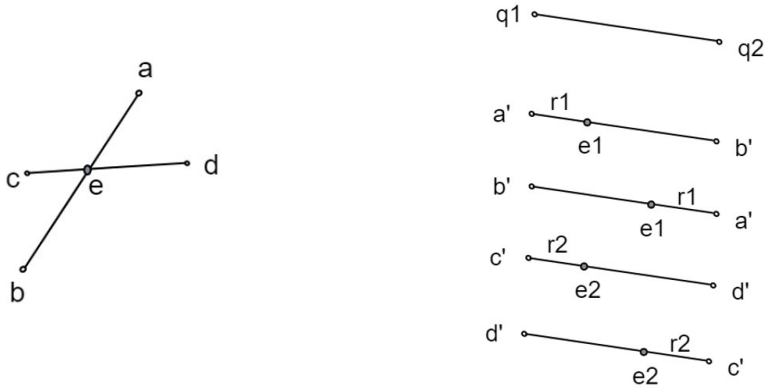
Due to the non-uniform density of the point cloud, there may be multiple pairs of four-point set pairs between the four-point set points in the target point cloud subinterval Q_y and the four-point set points in the source point cloud subinterval P_x. To solve the problem, two features, the curvature of the points [23] and Fast Point Feature Histograms (FPFH) [10], are extracted to construct a multi-dimensional feature vector as shown in equation (9).

$$\mathrm{X}_i = [q_1, q_2, q_3, q_4, f_1, f_2, f_3, f_4] \tag{9}$$

where denotes the curvature feature of the four-point concentration points, and represents the FPFH feature of the four-point concentration points. Where the FPFH feature is obtained by Rusu et al. [10] based on their proposed PFH feature, and the formula for calculating the FPFH feature of

(a) Source point cloud subinterval P_x (b) Target point cloud corresponds to the subinterval Q_y

FIGURE 5. 4PCS schematic diagram.



(a) Source point cloud interval four-point set (b) Possible intersections of line segments

FIGURE 6. Four-point set search graph.

point pi is as in equation (10).

$$\begin{cases} \text{FPFH}\,(p_i) = \text{SPFH}\,(p_i) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\,\text{SPFH}\,(p_k) \\ \omega_k = ||p_i : |p_i - p_k| < r|| \end{cases} \tag{10}$$

where *SPFH* is the simplified point feature histogram of a point, p_k is a neighborhood point of p_i, r denotes the neighborhood radius of point p_i, and k denotes the number of neighborhood points.

The correspondence of the four-point set is estimated by matching the correlation of the multidimensional feature vectors. Let X and Y be the multi-dimensional vectors of the source point cloud interval four-point set and the corresponding target point cloud interval four-point set, respectively. The feature similarity coefficient r_{XY} of X, Y can be calculated using equation (11).

$$r_{XY} = \frac{\sum(X-\bar{X})\times(Y-\bar{Y})}{\sqrt{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\times\sqrt{\sum_{i=1}^{n}\left(Y_i-\bar{Y}\right)^2}} \tag{11}$$

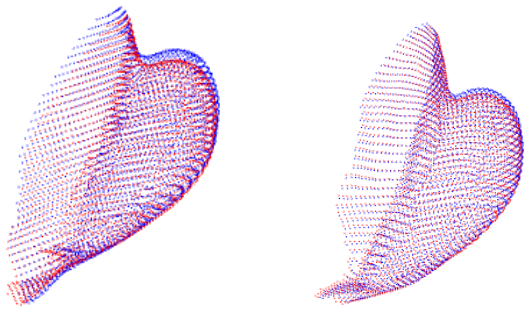
where $\overline{X}$ and $\overline{Y}$ denote the average of the multidimensional feature vectors X and Y. If $r_{XY} > T$ (T is a predefined threshold), the four-point set of the target point cloud subinterval Q_y is updated to be identical to the four-point set in the source point cloud subinterval P_x to obtain the corresponding best four-point set pair, the application flow is reported in Algorithm 3. After receiving the unique corresponding four-point set in the source point cloud subinterval P_x and the target point cloud subinterval Q_y, the transformation matrix is calculated by the least squares method.

Algorithm 3 Multi-Dimensional Feature Vector

Require: T, four-point set B, and the corresponding four-point set $\{B_1, B_2, \ldots, B_n\}$.

1: **for** $B_i \in \{B, B_1, B_2, \ldots, B_n\}$ **do**
2: Calculate FPFH features by Eq.(10).
3: Construct multi-dimensional feature vectors $X_B, X_{B_1}, X_{B_2}, \ldots, X_{B_n}$ by Eq.(9).
4: Calculate feature similarity coefficient $r_{X_B X_{B_i}}$ by Eq.(11).
5: **if** $r_{X_B X_{B_i}} > T$ **then**
6: B_i is the corresponding four-point set of B.
7: **end if**
8: **end for**
9: **return** B of the corresponding four-point set B_i.

As shown in Figure 7(a) and 7(b), the subinterval registration effects obtained by using point curvature feature-based [25] and the point cloud registration algorithm based on multidimensional feature vectors in this paper are shown, respectively. From the plots, it can be seen that the multidimensional features make the point cloud overlap more, and the effect is better than the registration effect of using point curvature features alone.



(a) Point cloud registration based on point curvature features (b) Point cloud registration based on Multi-dimensional features

FIGURE 7. Comparison before and after adding multidimensional features.

V. EXPERIMENTS AND ANALYSIS OF RESULTS

A. EXPERIMENTAL DESIGN

1) EXPERIMENTAL SCENARIO

To verify the effectiveness of the proposed algorithm, the algorithm is tested experimentally. The experiments are built on Ubuntu 18 operating system with processor Intel Core i5-6300HQ CPU@ 2.30 GHz memory 8 GB, and the environment of PCL (Point Cloud Library) [26] is used. The comparison algorithms are ICP [7], SACICP [11]], Teaser [27], Super4PCS [14], and MSSF-4PCS [15].

2) DATASETS

To test the performance of the registration algorithm under various positional deviations, the original point cloud is first transformed with a certain degree of angular deflection and translation. Then the original point cloud is aligned with the transformed point cloud by the algorithm in this paper to test the performance of the algorithm. The algorithm's performance is then tested by aligning the original point cloud with the transformed one. The angular deflections and translations of the original point cloud and the transformed point cloud are shown in Table 1 (taking Bunny as an example).

3) ALGORITHM EVALUATION METRICS

In the experimental part, the root means square error (RMSE) [28] is used as the evaluation index of the registration accuracy. The calculation formula is shown in equation (12), where n is the number of corresponding point pairs, X_i is the Euclidean distance between corresponding points after registration, and $\hat{X}$ is the actual value of the Euclidean distance between corresponding points. Ideally, the distance between the corresponding points is 0 after complete registration, so the smaller RMSE indicates the higher accuracy of the algorithm registration. The registration time is used as the evaluation index for the real-time performance of the algorithm.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \hat{X}_i\right)^2}{n}} \tag{12}$$

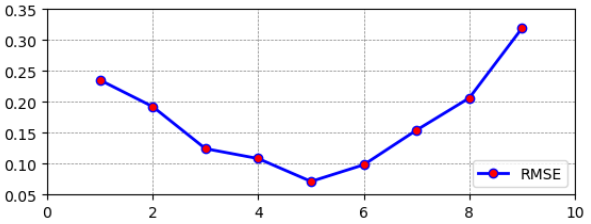


FIGURE 8. Algorithm accuracy when n takes different values.

While increasing the number of segmentation regions can decrease the search time of the algorithm, it can also lead to a decrease in the accuracy of point cloud registration due to insufficient sub-interval point clouds. The number of segmentation regions is found to be related to the point cloud density, and the dataset used in this paper was acquired using a 16-line LiDAR, resulting in a point cloud density of approximately $1315/cm^2$.

4) PARAMETER SELECTION

The primary step of the algorithm in this article is to segment the source point cloud and target point cloud. Therefore, the relationship between the number of segmentation intervals and the experimental results is closely related. Table 2 shows the comparison of algorithm running time for different numbers of segmentation intervals n, using the Bunny point cloud model as an example. Figures 8 shows the comparison of algorithm accuracy.

When n<5, the number of feature four point sets is large, the calculation amount is large, and the algorithm runs for a long time. When n>5, the number of intervals is large, the number of data points within the interval is small, and the matching rate of error points increases, resulting in lower matching accuracy of the algorithm. Therefore, the value of n is selected as 5. For the dataset used in this article, $n = 5$ is the most appropriate. When using other datasets, such as larger point cloud models, n will need to be adjusted accordingly.

B. REGISTRATION ACCURACY, REAL-TIME EXPERIMENTS

1) REGISTRATION ACCURACY

We used several algorithms, namely ICP, SACICP, Teaser [27], Super4PCS, MSSF-4PCS, and the algorithm proposed in this paper, for conducting registration experiments. The final result was obtained by taking the average of 50 experimental results. Figure 9 shows the root-mean-square error of each algorithm's registration at different angles for Bunny, Dino, Dragon, and Horse point clouds. It can be observed that several algorithms exhibit similar trends of RMSE error under different point clouds. For instance, consider the Bunny point cloud registration shown in Fig 9(a). When the angular deflection and translation of Bunny are minimal, the ICP algorithm yields a root mean square error of about 0.1m, indicating good registration performance. However, as the angular deflection and translation increase, the performance of ICP gradually deteriorates. This is because

TABLE 1. Description of the data set.

Transformed point cloud	X-axis deflection angle(°)	Y-axis deflection angle(°)	Z-axis deflection angle(°)	X-direction translation amount(m)	Y-direction translation amount(m)	Z-direction translation amount(m)
Bunny1	20	20	20	0.2	0.2	0.2
Bunny2	60	60	60	0.2	0.2	0.2
Bunny3	100	100	100	0.2	0.2	0.2
Bunny4	20	20	20	0.5	0.5	0.5
Bunny5	60	60	60	0.5	0.5	0.5
Bunny6	100	100	100	0.5	0.5	0.5

TABLE 2. Algorithm time when n takes different values.

n	1	2	3	4	5	6	7	8	9
time	4.01	2.81	1.98	1.54	**1.32**	1.72	2.09	2.41	2.68

the iterative nearest point method of ICP can achieve better registration in the slight deflection and translation range. ICP heavily relies on the initial value of the point cloud and is therefore unsuitable for aligning point clouds with large positional deviations.

SACICP is an improvement over ICP and helps to decrease the reliance on the initial value, which, in turn, reduces the registration error caused by an increase in initial attitude deviation to some extent. The refined registration stage of SACICP optimizes the error function through iterative methods to determine the optimal positional transformation. However, achieving high-accuracy registration with a limited number of iterations is a challenging task. Therefore, the registration error for Bunny4, Bunny5, and Bunny6 point cloud models, which have significant initial attitude deviation, is higher.

Super4PCS and MSSF-4PCS are advanced versions of the 4PCS algorithm, and they differ from the registration principle of ICP. Both algorithms utilize affine invariance of the corresponding four-point set in the same plane to identify the corresponding two sets of points in the target point cloud and the point cloud to be aligned. They then choose the optimal transformation matrix from several candidate matrices, which is less dependent on the initial value of the positional attitude. As shown in Figure 9, the root mean square error of Super4PCS and MSSF-4PCS changes smoothly with an increase in the initial attitude deviation, and the registration accuracy does not abruptly change with the initial attitude. This suggests that these algorithms are more robust and accurate than ICP when it comes to aligning point clouds with a large positional deviation.

However, Super4PCS filters the four-point set by limiting the distance range and angle, while MSSF-4PCS filters the four-point set by introducing sparse features. Based on this filtering mechanism, irregular surface shapes of the point cloud models may result in missing or misaligned four-point sets, thus affecting the accuracy of the algorithm. As shown in Figure 9(b) and 9(c), the errors of the Super4PCS and MSSF-4PCS algorithms are higher for Dino and Dragon models as compared to other models. This is because the point clouds of Dino and Dragon models are irregular, having more folds on the surface, leading to the missing or misaligned four-point set. However, the accuracy of this algorithm has been improved by 24.39% compared to the MSSF-4PCS point cloud registration algorithm.

From Figure 9, it can be observed that the algorithm proposed in this paper has the lowest root mean square error for different point cloud models and initial attitude deviations, and the error does not significantly increase with an increase in initial attitude deviation. Moreover, there is no significant difference in the registration error when comparing the irregularly shaped Dino and Dragon models. Hence, the algorithm proposed in this paper is highly robust for point clouds with large initial pose deviations and irregular surface shapes.

In order to compare the performance of each algorithm, Fig. 10 shows several algorithms' point cloud model registration results at the same angle. In Fig. 10, the red point cloud is the target point cloud, the green point cloud is the point cloud to be aligned, and the blue point cloud is the aligned point cloud, and the higher overlap between the blue and red point clouds indicates the better registration effect [29]. The number of iterations of ICP and SACICP algorithms is set to 30, and it can be seen from Fig. 10 that ICP has a significant registration error. SACICP has a lower error than ICP, but the degree of point cloud overlap is still lacking. super4PCS and MSSF-4PCS algorithms successfully achieve the registration, but there is a local part of the model surface where the point clouds do not overlap. The algorithm in this paper has the highest degree of point cloud coincidence, i.e., the best registration accuracy.

2) REAL TIME

Table 3 shows the registration times for each algorithm, and the results in the table are averaged over 50 runs. The

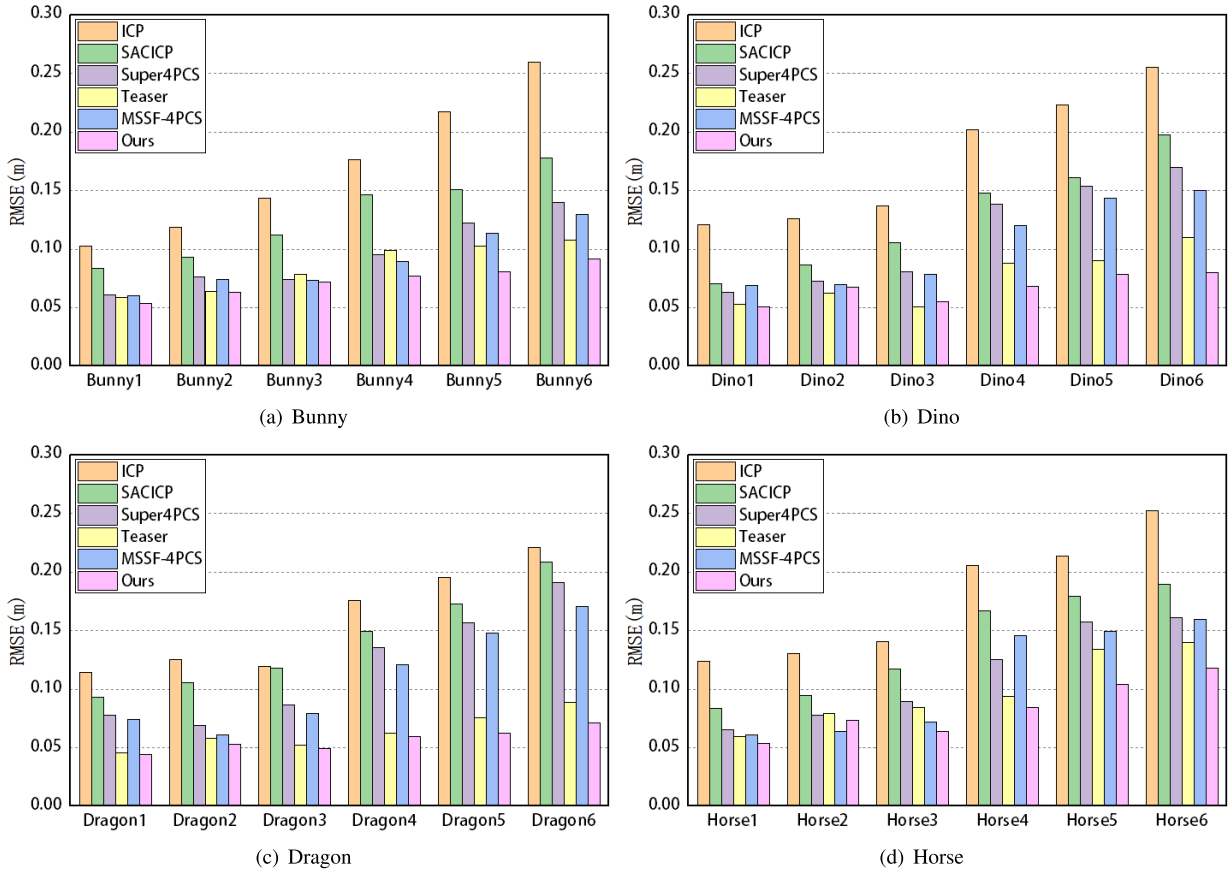


FIGURE 9. Comparison of Algorithm registration Errors.

TABLE 3. Registration time (s).

	ICP [7]	SACICP [11]	Super4PCS [14]	Teaser [27]	MSSF-4PCS [15]	Ours
Bunny	3.02331	10.9711	1.732829	1.09576	1.546107	**1.036441**
Dino	2.897	9.784	1.18636	1.00932	1.097825	**0.91202**
Dragon	2.92826	8.734	2.08908	0.70582	1.865301	**0.51922**
Horse	3.0993	9.4389	1.15109	0.58103	0.997462	**0.46099**

Teaser (truncated least squares estimation and semidefinite relaxation) algorithm is slower and more computationally expensive when performing large-scale point cloud registration. Compared with the Super4PCS and MSSF-4PCS algorithms, the algorithm in this paper has higher computational efficiency, with a 46.21% improvement over the MSSF-4PCS registration efficiency. On the one hand, the interval partitioning of the point cloud reduces the search range of the point cloud corresponding to the four-point set. On the other hand, the multi-dimensional feature filtering mechanism is used in the four-point set screening stage, which can effectively filter out a large number of invalid and wrong four-point set pairs and reduce the four-point set registration time. In summary, the algorithm in this paper takes the least registration time and has the best real-time performance.

VI. DISCUSSION

The results of Section V show that the algorithm proposed in this paper can achieve the goals stated in Section I and make contributions. Firstly, by segmenting the point cloud and reducing the range for searching corresponding points, the computational complexity is greatly reduced. In contrast, the ICP algorithm determines corresponding points by comparing the Euclidean distances of all points in the point cloud, which is computationally expensive. The SAC-ICP algorithm, as an improved algorithm of ICP, only solves the problem of high dependence on initial values to a great extent, but does not address the problem of high computational complexity. Secondly, based on the principle of 4PCS algorithm, the proposed algorithm seeks approximately coplanar four-point sets through affine invariance and constructs

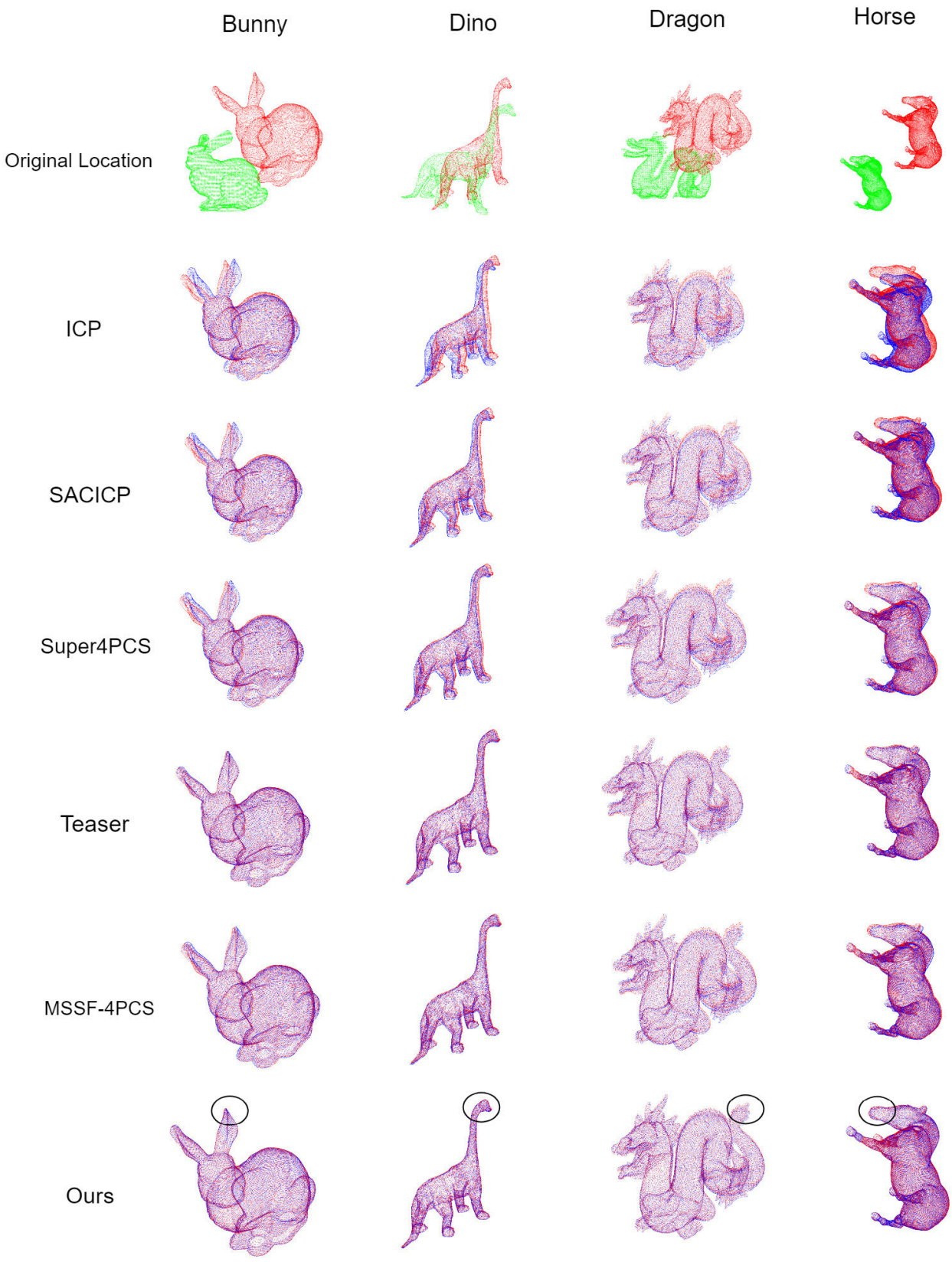


FIGURE 10. Comparison of algorithm registration effects.

multi-dimensional feature vector to constrain the features of the four-point sets, which improves the registration accuracy and speed. In contrast, the Super4PCS and MSSF-4PCS algorithms, as improved algorithms of 4PCS, use distance and angle constraints and sparse feature selection to match corresponding four-point sets. However, when the point cloud model is rough or the point cloud features are complex, the four-point sets may be mismatched, which may affect the global registration accuracy of the point cloud. In summary, compared with the algorithms mentioned in Section V, the proposed algorithm in this paper has higher registration efficiency and accuracy.

VII. CONCLUSION

Due to the currently insufficient registration accuracy and efficiency of existing point cloud registration algorithms, we propose a novel multi-dimensional point cloud registration algorithm based on interval segmentation to enhance registration accuracy and ensure real-time performance. Our algorithm utilizes interval partitioning and curvature histogram extraction of the source and target point clouds to narrow down the search range. Furthermore, we optimize the four-point set filtering mechanism by embedding curvature and FPFH features into the multi-dimensional feature vector, resulting in improved registration accuracy for the traditional affine invariant registration algorithm. We compared and analyzed the registration performance of ICP, SACICP, Super4PCS, Teaser, and MSSF-4PCS on the Stanford point cloud model. The results demonstrate that our algorithm achieves higher registration accuracy for large initial positional deviations and better robustness, while maintaining registration efficiency.

Although our algorithm achieves superior registration results in object registration scenarios, there is still room for performance improvement in other scenarios. For instance, the calculation of curvature may not be optimal in noisy point clouds, and there may be a challenge of low overlap in large-scale 3D reconstruction. As a result, the performance of our algorithm may suffer to some extent in such scenarios. Thus, future research [30] should aim to achieve robust registration in noisy and large-scale scenes with low overlap.

REFERENCES

[1] B. Biström, "Comparative analysis of properties of LiDAR-based point clouds versus camera-based point clouds for 3D reconstruction using SLAM algorithms," Tech. Rep., 2019.
[2] Y. Wu, Y. Liu, M. Gong, P. Gong, H. Li, Z. Tang, Q. Miao, and W. Ma, "Multi-view point cloud registration based on evolutionary multitasking with bi-channel knowledge sharing mechanism," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 2, pp. 357–374, Apr. 2023.
[3] X. Gu, X. Wang, and Y. Guo, "A review of research on point cloud registration methods," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 782, no. 2, Mar. 2020, Art. no. 022070.
[4] P. Kim, J. Chen, and Y. K. Cho, "SLAM-driven robotic mapping and registration of 3D point clouds," *Automat. Construct.*, vol. 89, pp. 38–48, May 2018.
[5] Z. Kang, J. Yang, Z. Yang, and S. Cheng, "A review of techniques for 3D reconstruction of indoor environments," *Int. J. Geo-Inf.*, vol. 9, no. 5, pp. 1–31, 2020.
[6] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," 2021, *arXiv:2103.02690*.
[7] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.
[8] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot. Sci., Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.
[9] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 742–749.
[10] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.
[11] Z. Xie, P. Liang, J. Tao, L. Zeng, Z. Zhao, X. Cheng, J. Zhang, and C. Zhang, "An improved supervoxel clustering algorithm of 3D point clouds for the localization of industrial robots," *Electronics*, vol. 11, no. 10, p. 1612, May 2022.
[12] Y. Liu, D. Kong, D. Zhao, X. Gong, and G. Han, "A point cloud registration algorithm based on feature extraction and matching," *Math. Problems Eng.*, vol. 2018, pp. 1–9, Dec. 2018.
[13] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, Aug. 2008.
[14] N. J. Mitra, N. Mellado, and D. Aiger, "Super 4PCS fast global pointcloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, Aug. 2014.
[15] Z. Xu, E. Xu, Z. Zhang, and L. Wu, "Multiscale sparse features embedded 4-points congruent sets for global registration of TLS point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 286–290, Feb. 2019.
[16] Z. Zhang, Y. Dai, and J. Sun, "Deep learning based point cloud registration: An overview," *Virtual Reality Intell. Hardw.*, vol. 2, no. 3, pp. 222–246, Jun. 2020.
[17] S. Yan, S. Pathak, and K. Umeda, "PointpartNet: 3D point-cloud registration via deep part-based feature extraction," *Adv. Robot.*, vol. 36, no. 15, pp. 724–734, Aug. 2022.
[18] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 12–21.
[19] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8958–8966.
[20] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "SpinNet: Learning a general surface descriptor for 3D point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11753–11762.
[21] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, p. 339, Apr. 2017.
[22] P. Tan, X.-F. Li, J.-M. Xu, J.-E. Ma, F.-J. Wang, J. Ding, Y.-T. Fang, and Y. Ning, "Catenary insulator defect detection based on contour features and gray similarity matching," *J. Zhejiang Univ.-Sci. A*, vol. 21, no. 1, pp. 64–73, Jan. 2020.
[23] Z. Yao, Q. Zhao, X. Li, and Q. Bi, "Point cloud registration algorithm based on curvature feature similarity," *Measurement*, vol. 177, Jun. 2021, Art. no. 109274.
[24] X. Zhang, H. Li, Z. Cheng, and Y. Zhang, "Robust curvature estimation and geometry analysis of 3D point cloud surfaces," *J. Inf. Comput. Sci.*, vol. 6, no. 5, pp. 1983–1990, 2009.
[25] N. Wei, K. Y. Gao, R. Ji, and P. Chen, "Surface saliency detection based on curvature co-occurrence histograms," *IEEE Access*, vol. 6, pp. 54536–54541, 2018.
[26] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.
[27] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2020.
[28] P. Li, R. Wang, Y. Wang, and W. Tao, "Evaluation of the ICP algorithm in 3D point cloud registration," *IEEE Access*, vol. 8, pp. 68030–68048, 2020.
[29] G. Li, Y. Cui, L. Wang, and L. Meng, "Automatic registration algorithm for the point clouds based on the optimized RANSAC and IWOA algorithms for robotic manufacturing," *Appl. Sci.*, vol. 12, no. 19, p. 9461, Sep. 2022.
[30] H. Si, J. Qiu, and Y. Li, "A review of point cloud registration algorithms for laser scanners: Applications in large-scale aircraft measurement," *Appl. Sci.*, vol. 12, no. 20, p. 10247, Oct. 2022.

ANRAN XU received the B.E. degree in software engineering from Anhui Polytechnic University, in 2021. She is currently pursuing the master's degree with the School of Electronic Information, Shanghai Dianji University. Her main research interests include 3D reconstruction and computer vision applications.

LEI RAO was born in 1985. She received the B.S. and Ph.D. degrees in electronic science and technology from Zhejiang University, Hangzhou, in 2007 and 2012, respectively. She is currently an Assistant Professor with Shanghai Dianji University. Her research interests include mobile robot navigation technology and 3D reconstruction.

GUANGYU FAN received the Ph.D. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2015. He is currently an Associate Professor with the College of Electronic Information, Shanghai Dianji University. His current research interest includes integrated navigation systems.

NIANSHENG CHEN (Member, IEEE) received the bachelor's degree in physics from Hubei Normal University, in 1989, and the master's and Ph.D. degrees in computer applied technology from the Wuhan University of Technology, in 1999 and 2007, respectively. He was the Vice Dean and the Dean of the School of Computer Science, Hubei Normal University, and was promoted to Professor, in 2006. Since 2012, he has been with Shanghai Dianji University, where he was the Director of the Institute of Electronic Information and the Director of the Institute of Internet of Things Technology Application.