Point cloud registration algorithm based on NPFC neighborhood feature descriptor

MINGDE ZHANG,[1] XINGYU CHEN,[1,*] LE XIE[1,2]

[1]Chongqing University of Technology, Chongqing, 400054, China

[2]Chongqing Miao Qi Technology Co., Ltd., Chongqing 400054, China

*1803196745@qq.com

Abstract: Point cloud registration technology is widely used in machine vision, reverse engineering, and other neighborhood applications, and its efficiency and accuracy have an important impact on product data models. Aiming at the problems of the traditional ICP registration algorithm, such as slow convergence speed and high requirements on the initial point cloud position, this paper proposes a point cloud registration algorithm based on the neighborhood point feature covariance matrix descriptors. First, the key points of the point cloud data are extracted by combining the average angle of the neighborhood normal vector and ISS algorithms; then, the NPFC descriptors of the key points are calculated, and the two-way nearest neighbor feature matching is performed according to the similarity of the NPFC to obtain the initial correspondence; for the initial correspondence, the RANSAC algorithm is used to reject the mismatches to obtain the final correspondence pairs, and the initial registration parameters are calculated by using the final correspondence; finally, the iterative nearest point algorithm is used to perform the fine registration. Experiments on public datasets show that NPFC descriptor has high descriptiveness and robustness in the face of noise. The registration results also confirm the superiority of our registration method in terms of accuracy and efficiency.

1. Introduction

With the rise of 3D imaging technology and the rapid development of 3D scanning technology [1-2], 3D point cloud registration techniques are widely used in many fields, including robot navigation [3], shape recognition [4], and reverse engineering [5]. In the process of scanning and measuring, the object must be scanned several times at different angles due to the influence of object occlusion, environmental factors, or the error of the measuring tool. The registration algorithm is then used to register several groups of scanned data together to form a complete point cloud representation of the object.

As the most commonly used registration algorithm, the iterative nearest point algorithm [6] (ICP) is simple in concept and high in accuracy, but the speed and convergence of the algorithm are highly dependent on the initial point cloud poses, and the objective function also tends to fall into the local optimum. For this reason, a common registration system is the "coarse to fine" strategy. Coarse registration provides a good initial position, then ICP algorithms are used for fast and accurate registration. Coarse registration using feature descriptors is currently a common method. The feature descriptor can be defined as a mapping from a 3D object space to a high-dimensional vector space. Due to the irregular nature of 3D point clouds, it is challenging to design a feature descriptor with high overall performance.

According to the relative range of object information, the existing feature descriptors can be divided into global feature descriptors and local feature descriptors. The global feature descriptor characterizes the object by extracting the full model features and shape information of the object. The local feature descriptor implements the feature coding according to the neighborhood spatial distribution information. In practical situations, the obtained point cloud information is often incomplete. The global feature descriptors are very sensitive to the information, so this paper investigates and analyzes for the local feature descriptors. According to the implementation principle of rotation invariance, feature descriptors can be divided into two categories: the 3D local description method based on rotation invariance and the local feature description method based on the local reference system (LRF). For the first type of feature descriptor: Johnson and Heber proposed the spin image [7] (SI) descriptor. SI first uses the normal vectors of the key points as reference axes, then projects the local neighborhood points in the horizontal and vertical directions of the two-dimensional plane, and finally encodes the information using histogram statistics. Therefore, it is more robust and efficient against noise, but histogram statistics cause a loss of effective information, which in turn limits the descriptive subaccuracy. Frome et al. proposed the 3D Shape Context [8] (3DSC) descriptor. The 3DSC descriptor divides the 3D spherical space into multiple containers according to azimuth, elevation, and radius, transforms them into frequency domain signals using the harmonic transform, and finally realizes the rotational invariance of the descriptor by the rotational invariance of the frequency domain amplitude. This method reduces the loss of valid information and improves the accuracy of the descriptor, but frequency domain coding also increases the complexity of the descriptor. Rusu et al. constructed Point Feature Histograms [9] (PFH) and Fast Point Feature Histograms [10] (FPFH) from histograms of the angles between vectors in local neighborhoods. The descriptor can effectively overcome the problem of missing local information about the object in a single view, but it still faces the challenges of noise interference, point density variation, and other factors. For the local reference system-based feature description method: the most widely used is an LRF-based Signature of Histograms of Orien Tations [11] (SHOT) descriptor proposed by Tombari et al. The method builds the LRA and then fixes the data set on the LRA and divides it into different blocks. The normal offset angles in each block are summed, and all histograms are stitched together to obtain SHOT. The experimental results show that the method has high descriptiveness, robustness, and computational efficiency.

The above methods have problems such as low robustness, poor descriptiveness, or high computational complexity. Therefore, this paper proposes a fast and robust local 3D feature descriptor, NPFC. NPFC can efficiently complete the coarse registration processes such as point cloud feature description, feature matching, and transformation matrix solving. The point cloud registration method based on the neighborhood point pair feature covariance matrix descriptor (NPFC) can achieve high registration results.

2. NPFC descriptor calculation

The local feature-based 3D registration method uses point cloud correspondence for registration. Correspondence is usually obtained by matching feature descriptors. The point cloud feature descriptor is a collection of parameters used to describe the geometric features of a point cloud surface that can sufficiently reflect the neighborhood information of the sampled points and has rotational translation invariance [12]. In this section, we propose a local feature descriptor that encodes the surface geometric features of a sampled point and its neighbors as a covariance matrix, called the neighborhood point-pair feature covariance matrix descriptor (NPFC).

2.1 Local point-pair feature

The choice of feature parameters is the key to constructing covariance feature descriptors. For the construction of the point cloud feature descriptors, elements with good robustness and representativeness should be selected. Geometric properties such as normal vectors, distance parameters, and curvature variation reflect the most basic geometric features of point clouds [13]. They are rotation-translation invariants and easy to compute. Classical point-pair features were originally used for hemispherical 3D object recognition [9]. It uses four parameters to describe the relative position and direction of two directed points. Based on this, we add curvature change features to enhance the descriptor's feature description capability. As shown in Fig. 1, for two random points p and p_1 with their normal vectors n and n_1 and curvatures c and c_1, let $d = p_1 - p$. Define characteristics:

$$F(p,p_1) = (\|d\|, \angle(n,n_1), \angle(p,d_1), \angle(p_1,d), c - c_1) \tag{1}$$

where $\angle(a,b) \in [0,\pi]$ denotes the angle between two vectors. FPH constructs global feature descriptions by computing point-pair features for all points in the point cloud, making it computationally inefficient and therefore not suitable for point cloud registration. We have studied and improved the PPF[14] from the perspective of local description, combining the point-pair features of domain points to improve the feature description capability of the feature descriptors in this paper.

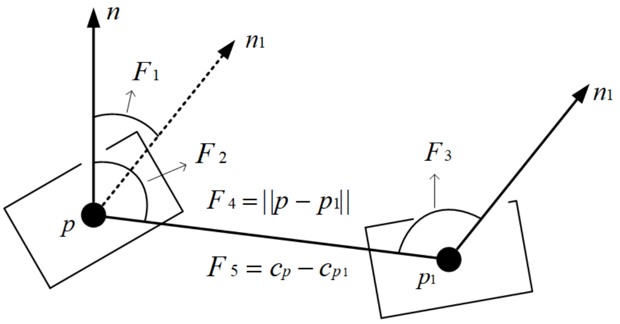
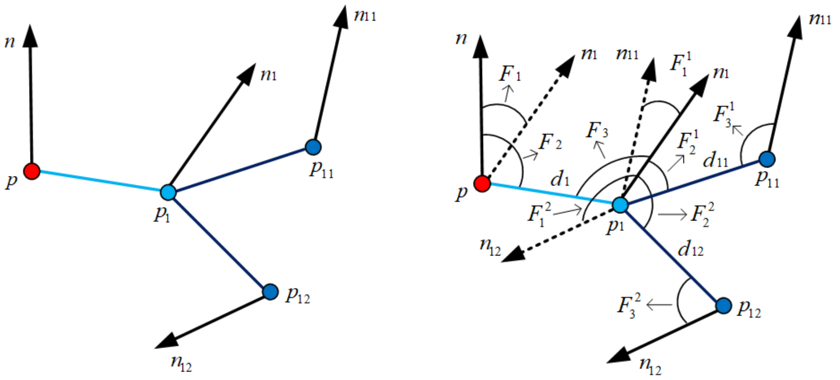


Fig. 1. Point-pair feature of two oriented points



(a) Four oriented points

(b) Feature parameter set

Fig. 2. Four oriented points neighbor feature parameter set

2.2 *Neighborhood point-pair feature parameters*

Generally, the characterization of the sampled points in the specified neighborhood does not fully reflect their characteristics. Based on the local point-pair feature described above, the neighborhood point-pair feature parameters (NPFP) are proposed by combining the geometric feature parameters of the neighboring points. As shown in Fig. 2(a), for any point p and its neighbor p_1, and two neighboring points p_{11} and p_{12} of p_1. Neighboring points are defined as points that are less than a threshold distance r from the sampled points, and the normal vectors of p, p_1, p_{11}, and p_{12} are n, n_1, n_{11}, and n_{12}. The neighborhood point-pair feature parameter (NPFP) is defined as:

$$NPFP(p,p_1) = (F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9) \tag{2}$$

$$\begin{cases} F_1 = \|p - p_1\| \\ F_2 = \angle(n, n_1) \\ F_3 = \angle(n, d_1) \\ F_4 = \angle(n_1, d_1) \\ F_5 = c_p - c_{p_1} \\ F_6 = \frac{1}{2}(F_2^1 + F_2^2) = \frac{1}{2}[\angle(n_1, n_{11}) + \angle(n_1, n_{12})] \\ F_7 = \frac{1}{2}(F_3^1 + F_3^2) = \frac{1}{2}[\angle(n_1, d_{11}) + \angle(n_1, d_{12})] \\ F_8 = \frac{1}{2}(F_4^1 + F_4^2) = \frac{1}{2}[\angle(n_{11}, d_{11}) + \angle(n_{11}, d_{12})] \\ F_9 = \frac{1}{2}(F_5^1 + F_5^2) = \frac{1}{2}\left[\left(c_{p_1} - c_{p_{11}}\right) + \left(c_{p_1} - c_{p_{12}}\right)\right] \end{cases} \tag{3}$$

where $d_1 = p_1 - p$, $d_{11} = p_{11} - p_1$, and $d_{12} = p_{12} - p_1$ are vector differences; c_p, c_{p_1}, $c_{p_{11}}$, and $c_{p_{12}}$ are the curvatures of p, p_1, p_{11}, and p_{12}. The neighborhood feature parameters of the four directed points are shown in Fig. 2(b).

Extend the neighborhood point-pair feature parameter (NPFP) into point cloud space. For any point p in the point cloud, given the neighborhood radius r, the neighborhood point $q_i(i = 1,2,\cdots m)$ within the space sphere can be determined. Using the same threshold radius r, m spatial spheres can be determined using the neighborhood point as the center $q_i(i = 1,2,\cdots m)$. Let the neighborhood point with radius q_i be $q_{ij}(i = 1,2,\cdots m)$. Define NPFP as:

$$NPFP(p,q_1) = (F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9) \tag{4}$$

$$\begin{cases} F_1 = \|p - q_i\| \\ F_2 = \angle(n, n_i) \\ F_3 = \angle(n, d_i) \\ F_4 = \angle(n_i, d_i) \\ F_5 = c_p - c_{q_1} \\ F_6 = \frac{1}{n}\sum_{j=1}^{n} F_2^j = \frac{1}{n}\sum_{j=1}^{n} \angle\left(n_i, n_{ij}\right) \\ F_7 = \frac{1}{n}\sum_{j=1}^{n} F_3^j = \frac{1}{n}\sum_{j=1}^{n} \angle\left(n_i, d_{ij}\right) \\ F_8 = \frac{1}{n}\sum_{j=1}^{n} F_4^j = \frac{1}{n}\sum_{j=1}^{n} \angle\left(n_{ij}, d_{ij}\right) \\ F_9 = \frac{1}{n}\sum_{j=1}^{n} F_5^j = \frac{1}{n}\sum_{j=1}^{n} \angle\left(c_{q_i}, c_{q_{ij}}\right) \end{cases} \tag{5}$$

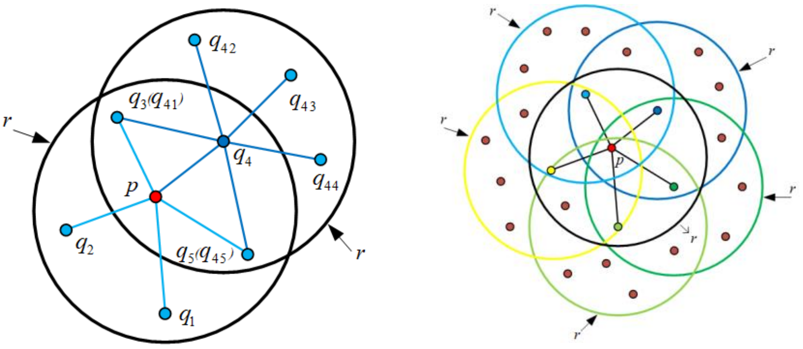
where $d_i = p - q_i$ and $d_{ij} = q_i - q_{ij}$ are the vector differences; n_i and n_{ij} are the normal vectors of q_i and q_{ij}; and c_{q_i} and $c_{q_{ij}}$ are the curvatures of q_i and q_{ij}. The NPFP contain the geometric features of the neighboring point-pair and the mean value of the neighboring point-pair features. It is rotationally and translationally invariant and has stronger feature description capabilities compared to local point-pair features.

2.3 Neighborhood point-pair feature covariance matrix (NPFC) descriptor

By calculating all $NPFP(p,q_i)$, m sets of neighborhood point-pair feature parameters can be obtained, and each set of neighborhood point-pair feature parameters contains nine elements. When $i=1$, the influence range of $NPFP(p,q_i)$ is shown in Fig. 3(a). After obtaining the feature vector $NPFP(p,q_i)$, define the neighborhood point-pair feature covariance matrix descriptor as:

$$NPFC(p)=\frac{1}{m-1}\sum_{i=1}^{m}\left[NPFP(p,q_i)-\overline{NPFP(p,q_\iota)}\right]\left[NPFP(p,q_i)-\overline{NPFP(p,q_\iota)}\right]^T \qquad (6)$$

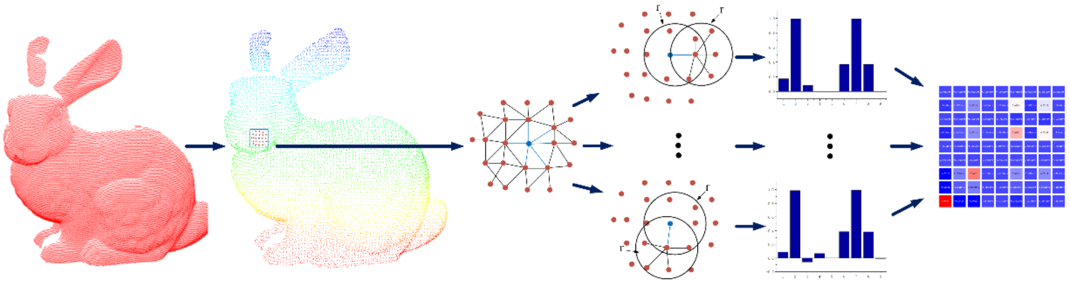
where m is the number of neighboring points p, $\overline{NPFP(p,q_\iota)}$ is the mean value of the neighboring points of p on the feature parameters, and the influence region of $NPFC(p)$ is shown in Fig. 3(b).



(a)The Influence region of $NPFP(p,q_i)$ (b)The Influence region of $NPFC(p)$

Fig. 3. The Influence region of descriptor

In probability theory and statistics, covariance is a measure of the degree of linear correlation of the joint distribution of two random variables [15]. In image processing, covariance has been introduced as a descriptor [16,17]. The covariance matrix descriptor can be viewed as an abstract description of the information about the neighborhood of the sampled points. It samples the collected features as joint distributions, weakens the spatial distribution information of neighboring points, and has spatial rotation invariance. By the central limit theorem, the feature distribution can be correctly represented as long as the obtained features are meaningful enough, which is more robust than with the histogram descriptors [18]. The NPFC descriptor in this paper is compared to the multiscale covariance matrix descriptor [19], where the NPFC descriptor uses neighborhood mean parameters such as $F_{6\sim9}$. The robustness of the descriptors for noise can be improved. Fig. 4 shows an illustration of the descriptors for a given key point.



(a)Source point cloud (b) Downsampling (c) Local points (d) Division (e) NPFP (f) NPFC

Fig. 4. Key point descriptor generation procedure. (a) Rabbit_bun000 point cloud. (b) Downsampling. (c) Local points determined by the neighborthood radius r. (d) Take different points to calculate parameters. (e) Neighborhood point-pair feature parameter. (f) NPFC descriptor.

3. NPFC-based point cloud registration algorithm

The NFPC-based point cloud registration process includes data pre-processing, feature point extraction, feature matching, coarse registration, and ICP fine registration. The registration process is shown in Fig .5.

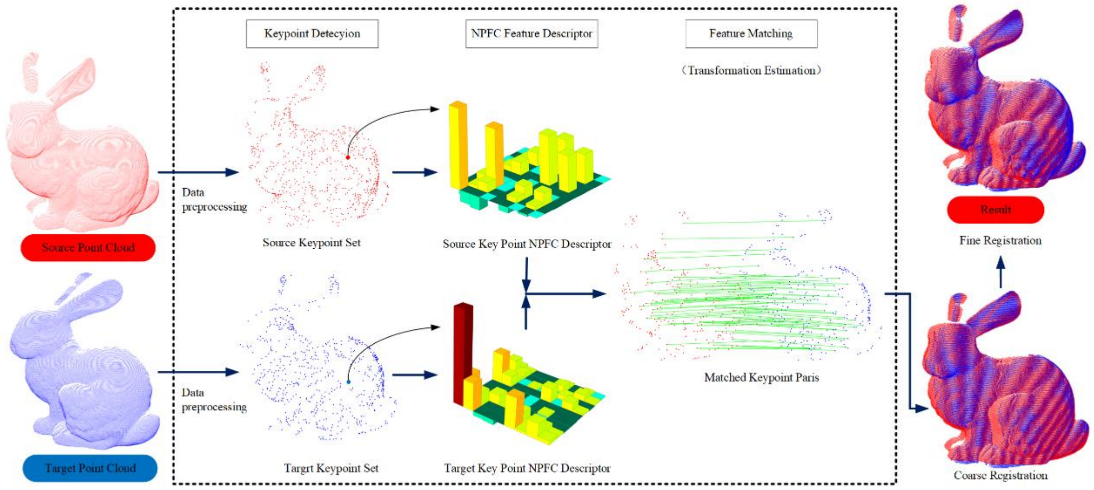


Fig. 5. Registration pipeline of point cloud.

3.1 Data pre-processing

As shown in Fig. 6(a), the point cloud is first downsampled for voxels to reduce the number of points while preserving the shape features of the point cloud. The point cloud data after downsampling is shown in Fig. 6(b). Principal component analysis is then used to calculate the normal vector and curvature of each point in the downsampled point cloud data, which are used as input parameters for key point extraction and NPFC descriptor construction. The principal component analysis algorithm [20] is a common method for computing curvature and normal vectors. The analysis of covariance is performed on p_i and its neighboring points p_{ij} in its radius neighborhood for any point p_i within the point cloud P.

$$\overline{p_i} = \frac{1}{k}(p_{i1} + p_{i2} + \cdots + p_{ik}) \tag{7}$$

$$C_i = \frac{1}{k}\begin{pmatrix} p_{i1} - \overline{p_i} \\ \vdots \\ p_{ik} - \overline{p_i} \end{pmatrix}^T \begin{pmatrix} p_{i1} - \overline{p_i} \\ \vdots \\ p_{ik} - \overline{p_i} \end{pmatrix} \tag{8}$$

$$C_i \times {v_i}^{(j)} = {\lambda_i}^{(j)} \times {v_i}^{(j)}, j = \{1,2,3\} \tag{9}$$

where $\overline{p_i}$ is the center of gravity of the neighborhood point cloud, C_i is a 3×3 covariance matrix, and k is the number of neighborhood points at p_i. ${\lambda_i}^{(j)}$ and ${v_i}^{(j)}$ denote the eigenvalues and eigenvectors of C_i. If the eigenvectors satisfy ${\lambda_i}^{(1)} < {\lambda_i}^{(2)} < {\lambda_i}^{(3)}$, then ${\lambda_i}^{(1)}$ is the normal vector of p_i, and the curvature of p_i is:

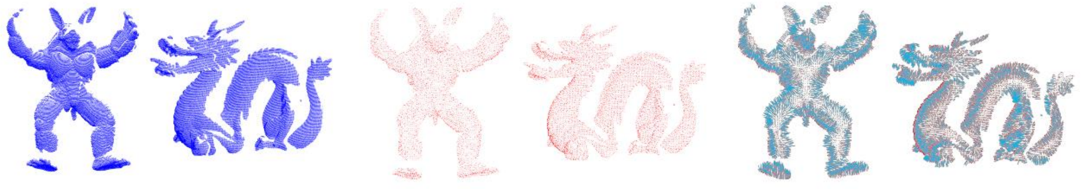
$$c_i(p_i) = \frac{{\lambda_i}^{(1)}}{{\lambda_i}^{(1)} + {\lambda_i}^{(2)} + {\lambda_i}^{(3)}} \tag{10}$$

where $i = 1,2,\cdots,N$, N is the number of downsampling points.

The normal vector estimated by the principal component analysis method has no direction, so the direction of the normal vector must be redefined. The direction of the normal vector is defined as:

$$n_i = \begin{cases} n_i, n_i \cdot p_i - \overline{p_i} > 0 \\ -n_i, n_i \cdot p_i - \overline{p_i} \leq 0 \end{cases} \tag{11}$$

Both the source and target point clouds must redefine the direction of the normal vector. The normal vector is shown in Fig. 6(c).



(a)The original point cloud

(b)Downsampling

(c)Normal of the sampled cloud

Fig. 6. Data preprocessing. (a) The point cloud from left to right are ArmadilloOnHeadMutiple_0 and dragonStandRight_0. (b) Downsampling. (c) The point is red and the sky blue line represents the normal.

3.2 Key Points Extraction

The key point of the point cloud is to obtain a set of points with stability and differentiation by defining detection criteria. The key points provide more pronounced geometric information than other point sets and are used as input point sets in subsequent coarse registration work. Extracting key points from a single feature can result in incomplete feature information and redundant calculations. In this paper, the key point extraction strategy is to first extract the key points using the average angle of the neighborhood normal vector and then use the ISS algorithm [21] for secondary key point extraction.

1) Average angle of the neighborhood normal vector

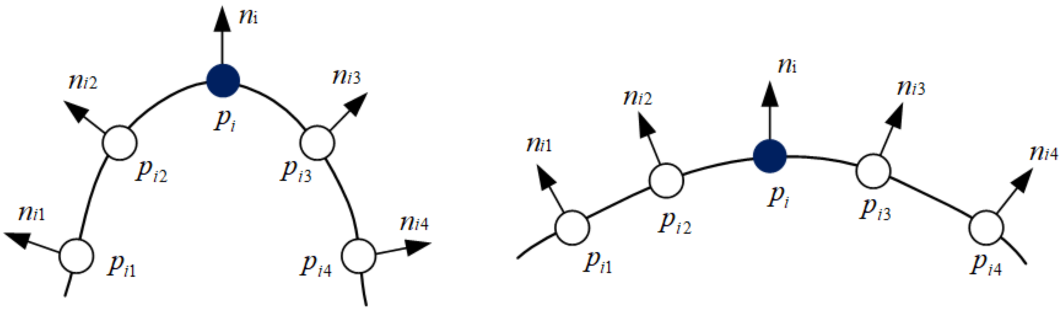
The normal vector is an important geometric feature in three-dimensional space with rotation-translation invariance. In point cloud processing, the normal vector angle is an important property to describe the geometric properties of the point cloud. The key point extraction algorithm selects key points by comparing the mean value of the normal vector angle of the sampled points with the mean value of the normal vector angle of the neighboring points in the radius neighborhood. For any point p_i in the point cloud, let the normal vector of p_i be n_i, the neighboring points of p_i be p_{ij}, and the normal vector of p_{ij} be n_{ij}. The mean values of the normal vector entrainment angles in the neighborhood of p_i is:

$$angle(p_i) = \frac{1}{k}\sum_{j=1}^{k} \angle\left(n_i, n_{ij}\right) \tag{12}$$

where k is the number of neighboring points of p_i, $j = 1,2,\cdots,k$, $i = 1,2,\cdots,N$, and N is the number of points in the point cloud. The larger $angle(p_i)$ is, the greater the curvature of the point cloud surface, as shown in Fig. 7(a). The smaller $angle(p_i)$ is, the smaller the curvature of the point cloud surface, as shown in Fig. 7(b). The global average angle of the neighborhood normal vector is used as the screening threshold, then the key point screening condition is:

$$\begin{cases} angle(p_i) \geq \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} angle(p_i),\ p_i\ is\ the\ key\ point; \\ angle(p_i) < \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} angle(p_i),\ p_i\ is\ not\ the\ key\ point; \end{cases} \tag{13}$$

The points with more prominent geometric characteristics on the point cloud surface can be quickly extracted as key points by calculating the average angle of the neighborhood normal vector.



(a) Large angle

(b) Small Angle

Fig. 7. Average angle of neighborhood normal vector. (a) Large angle. (b) Small angle.

2) ISS Key Points

Secondary extraction of key points using the ISS algorithm [21]: the ISS key point extraction algorithm is a method for filtering key points based on neighborhood information. For any point p_i in the point cloud, perform covariance analysis on p_i and its neighboring point p_{ij} in its neighborhood:

$$cov(p_i) = \frac{\sum_{j=1}^{k} \omega_{ij} (p_i - p_{ij})(p_i - p_{ij})^T}{\sum_{j=1}^{k} \omega_{ij}} \tag{14}$$

$$\omega_{ij} = \frac{1}{\|p_i - p_{ij}\|} \tag{15}$$

where k is the number of neighbors and ω_{ij} is the weight value. Calculation of all characteristic values $\{\lambda_i^3, \lambda_i^2, \lambda_i^1\}(\lambda_i^1 \leq \lambda_i^2 \leq \lambda_i^3)$ of $cov(p_i)$. Given thresholds ε_1 and ε_2, satisfying [Eq. (1)] is the key points of the ISS. The key points detected by combining average angle of the neighborhood normal vector and the ISS algorithm are shown in Fig. 8.

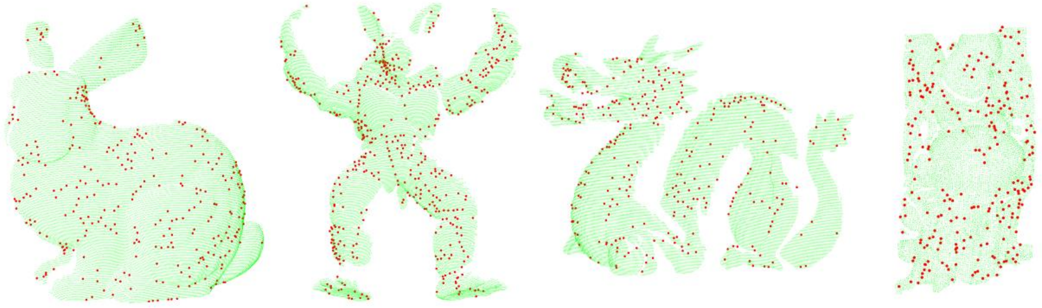


Fig. 8. Key points. (The red are the key points. The Point from left to right are bun000, ArmadilloOnHeadMutiple_0, dragonStandRight_0, happySideRight_0).

3.3 Feature Matching

Feature matching refers to establishing a set of feature correspondences between two sets of neighborhood covariance matrix feature descriptors, thus determining the correspondence between key points. The difficulty lies in computing the similarity of the descriptors and finding the correspondence based on the similarity of the descriptors.

1) Initial Feature Matching

The similarity of NPFC descriptors can be measured by the similarity of the covariance matrices: the smaller the distance, the greater the similarity between the descriptors. The covariance matrix does not depend on Euclidean space, so it is not possible to compute the similarity between covariance matrix descriptors using the distance formula commonly used in machine learning. The covariance matrix belongs to a type of Riemannian manifold, and the similarity of the covariance matrix descriptors can be measured by calculating the geodesic distance [22]. To simplify the calculation, the logarithmic eigenvalue method [23] is used in this paper to calculate the similarity of NPFC descriptors. Calculating the eigenvalues starts with calculating the generalized covariance matrix. The generalized covariance matrix is defined as follows:

$$C(X,Y) = Y^{-1}X \tag{16}$$

where, X and Y are the NPFC descriptors of the key points in the source point cloud and target point cloud. The covariance matrix is a positive definite matrix, so Y must be an invertible matrix. Calculate the similarity of the covariance matrix descriptors using the eigenvalue method.

$$sim\big(NPFC(p), NPFC(q)\big) = \frac{1}{N}\sum_{I=1}^{N} log_2^2(\lambda_i) \tag{17}$$

where N is the dimension of the covariance matrix, which is 9 in this paper, is the eigenvalue of the generalized covariance matrix, and the subscript 2 of the logarithmic function denotes the base and the superscript 2 denotes the computed square.

The similarity of descriptors is calculated according to the NFPC of key points in the source and target point clouds according to Equation 18, and then the feature matching is performed for the bidirectional nearest neighbor distance. For the source and target point clouds, the correspondence is found in another set of point clouds according to the nearest neighbor distance according to the similarity of the NFPC, respectively, and we can get two sets of correspondences. If there are identical pairs of correspondence points in the set of these two correspondences, the correspondence is considered to be the correct correspondence. This completes the initial feature matching.

2) Matching relationship screening

Point cloud data scanned from different viewpoints has some non-overlapping areas that may be mismatched during initial feature matching. And the local point cloud feature descriptor only needs to traverse the points in the specified range, improving computational efficiency while sacrificing some accuracy. Therefore, to ensure the correct match, the initial match must be mismatched and discarded to improve the coarse registration accuracy. In this paper, we use the RANSAC algorithm [24,25] for mismatch rejection.

The RANSAC algorithm selects the optimal rotation and translation parameters using the least-cost function:

$$J = \sum_{d\in D} L\big(E(d;M)\big) \tag{18}$$

where M is the estimated rotation and translation matrix parameters, D is the matching relationship dataset, d is the sample set including three matched point pairs, E is the error function, and L is the loss function. RANSAC algorithm based on 0-1 Loss Function:

$$L_r(e) = \begin{cases} 0, |e| \le t \\ 1, |e| > t \end{cases} \tag{19}$$

where e is the alignment error and t is the error threshold for determining the matching relationship. The number of samples required by the RANSAC algorithm should be large enough to ensure that at least three pairs of correct matching relations are selected with a certain confidence probability p.

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{20}$$

where s is the number of matching relationships and ε is the proportion of mismatches. Equation 20 shows that by reducing the proportion of mismatches and appropriately reducing the number of matching relationships used to estimate the rotation translation matrix, the sampling number of the RANSAC algorithm can be reduced, and the efficiency and accuracy of the RANSAC algorithm can be improved. The key point matching relationship is shown in Fig. 9

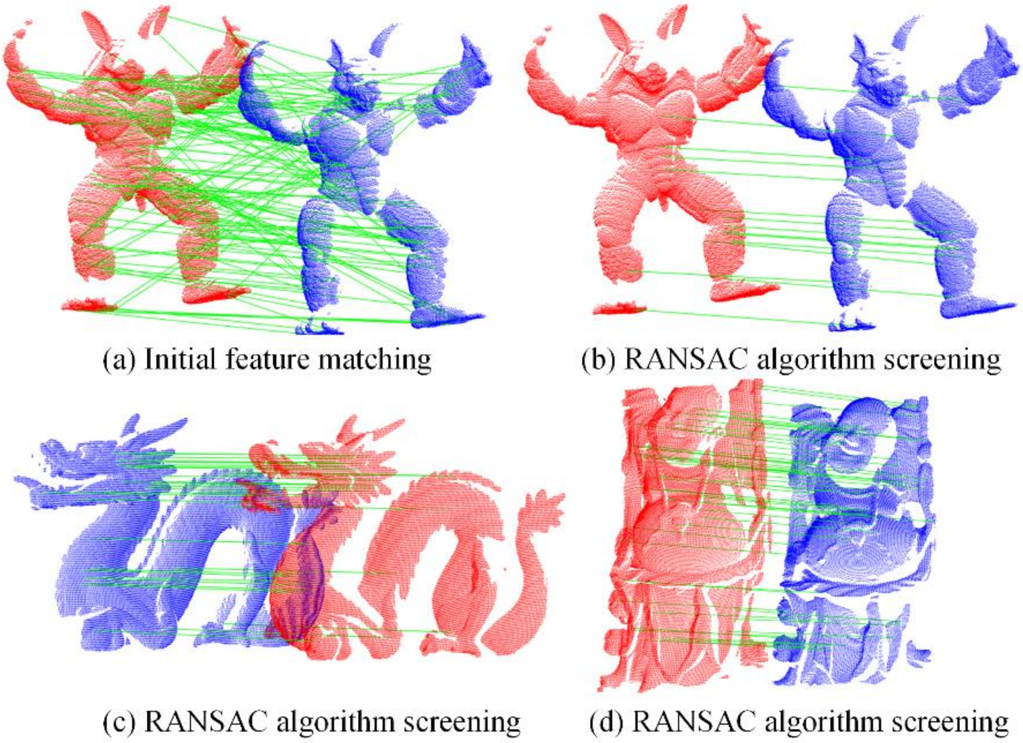

(a) Initial feature matching
(b) RANSAC algorithm screening
(c) RANSAC algorithm screening
(d) RANSAC algorithm screening

Fig. 9. Feature matching. The point cloud used by (a), (c), and (d) are ArmadilloOnHeadMultiple_0&30, dragonStandRight_0&24, and happySideRight_0&24 from the Stanford 3D dataset.

3.4 ICP algorithm

To improve the registration accuracy of point clouds with different viewpoints, the point clouds should be fine-registered after coarse registration. In this paper, the exact registration of the point cloud is performed using the Iterative Closest Point [6] (ICP) algorithm. The main idea is to iteratively compute the optimal rotation matrix R and translation vector T of the corresponding point set by minimizing the distance between the source point cloud $P\{p_i\}_{i=1}^{n}$ and the target point cloud $Q\{q_i\}_{i=1}^{m}$. The minimum objective function is:

$$\min_{R,T}\left(\sum_{i=1}^{N}\|Rp_i + T - q_i\|_2^2\right) \tag{21}$$

The ICP algorithm requires a high initial position and slow convergence of iterations, and it is easy to fall into a local optimum in the case of noise interference. The initial transformation matrix is the key to obtaining good registration accuracy. The above initial registration process can be used to obtain the initial rotation and translation matrix to achieve coarse registration of the point cloud. This provides a good initial position for the ICP algorithm to perform fine registration.

4. Experimental results and analysis

In this section, we perform three types of experiments, namely the descriptor performance experiment, the descriptor coarse registration experiment, and the Gaussian noise registration experiment. The descriptor performance experiments verify the robustness of NPFC descriptors under Gaussian noise. The coarse descriptor registration experiments demonstrate the superior accuracy and efficiency of the NPFC coarse descriptor registration method. The Gaussian noise registration experiment verifies that the algorithm in this paper can guarantee high efficiency and accuracy even in the case of Gaussian noise. The point cloud datasets used in the experiments were the Stanford Bunny, Armadillo, and Dragon datasets [26]. PFH, 3DSC, SHOT, FPH, and the algorithms in this paper were implemented in Visual Studio 2017 and PCL 1.8.1. The experiments were performed on a computer configured with an Intel Core i5 2.5 GHz CPU, 8 GB of RAM, and the Windows 10 operating system.

4.1 Descriptor Performance Experiment

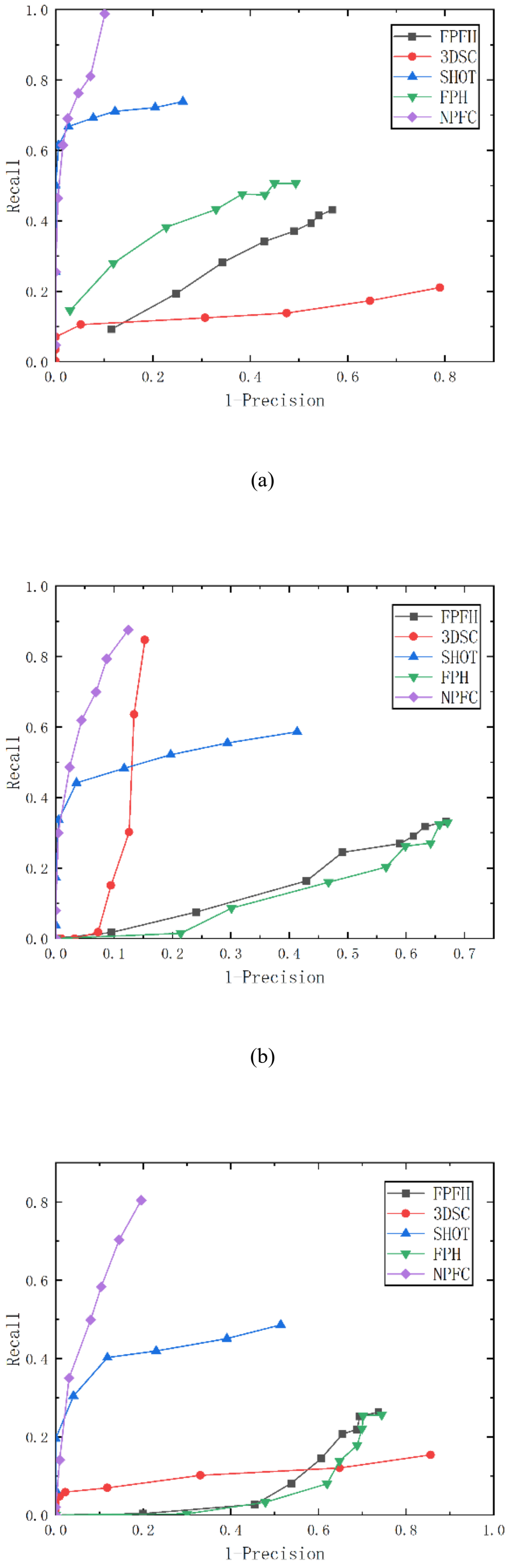
Recall and 1-Precision Curves [27,28] are the metrics used to evaluate the descriptors. In this part of the experiment, the Recall and 1-Precision Curves (Recall vs. 1-Precision Curve, RP Curve) is used to compare the performance of different descriptors. The RP Curve is generated as follows: Given the field point cloud, the model point cloud, and the truth transformation matrix between them, the distance between each feature descriptor in the field point cloud and the feature descriptors of the full model point cloud is calculated to find the nearest and second-nearest feature descriptors. If the ratio between the closest distance and the next closest distance is less than a set threshold, the feature descriptors of the site point cloud are considered to form a matching point pair with the feature descriptors in that closest model point cloud. From this pair of feature descriptors, the corresponding feature point pairs are found, and the Euclidean distances between the feature points in the model point cloud and the feature points in the corresponding field point cloud are transformed by the true transformation matrix. If this distance is small enough, the matching point pair is considered a correct matching point pair, and vice versa. Recall and precision are defined as:

$$Recall = \frac{the\ amount\ of\ true\ matchs}{tatal\ amount\ of\ corresponding\ feature} \tag{22}$$

$$1 - Precision = \frac{the\ amount\ of\ false\ matchs}{tatal\ amount\ of\ matchs} \tag{23}$$

Multiple recalls and accuracies can be obtained by varying the closest distance and next closest distance ratios to generate RP curves. Ideally, if the descriptor achieves both precision and recall, the RP curve will appear in the upper left corner of the graph. In our experiments, we use the ISS algorithm to compute the key points in the point cloud and obtain the corresponding key points in the scene through the truth transformation matrix to remove the effect of the key point localization error

The same Gaussian noise, key point set, and matching conditions are used to match features in the Stanford Common dataset using five feature descriptors, and the matching results are analyzed for comparison. Add Gaussian noise with standard deviations of 0.1 pr, 0.3 pr, and 0.5 pr to the point cloud data (pr is the point cloud resolution) and compare FPFH, FPH, SHOT, 3DSC, and NPFC. The experimentally generated RP curves are shown in Fig. 10. It can be seen from Fig. 10 that the NPFC descriptors outperform all other feature descriptors for different Gaussian noises.



(a)

(b)

(c)

Fig.10. (a) Set $\sigma = 0.1\,pr$ standard Gaussian noise. (b) Set $\sigma = 0.3\,pr$ standard Gaussian noise. (c) Set $\sigma = 0.5\,pr$ standard Gaussian noise.

4.2 Descriptor coarse registration experiment

To verify the superiority of the method in this paper, the key point extraction method proposed in this paper is used to perform coarse alignment experiments on two point cloud datasets from different viewpoints using FPFH, 3DSC, SHOT, FPH feature descriptions, and sampling consistency algorithms. The registration experiments are performed using three common datasets for different descriptors, and the experimental results are compared and analyzed with the registration results of the method described in the paper. The metric used to evaluate registration accuracy in text is the root mean square error (RMSE) [29], defined as:

$$RMSE = \sqrt{\frac{\sum_i^N \|Tp_i - q_i\|}{N}} \tag{24}$$

where point q_i in the target point cloud is the closest point to p_i in the source point cloud under the estimated pose.

Fig. 11 shows the results of coarse registration of three point cloud datasets using different methods, and table 1 shows the registration errors and time consumed. From Fig. 11, it can be seen that the FPFH and FPH combined with the sampling consistency method and the method in this paper achieve a better coarse registration process, but the coarse registration result of the method in this paper is the best. Table 1 quantifies the coarse registration of the five methods, and the proposed method is optimal in terms of alignment error and operational efficiency. The method in this paper reduces the registration error by 46.7%, 53.4%, and 49.4% on average, and the registration speed is better than other methods. In conclusion, the NPFC-based coarse registration method proposed in this paper can achieve better registration results.

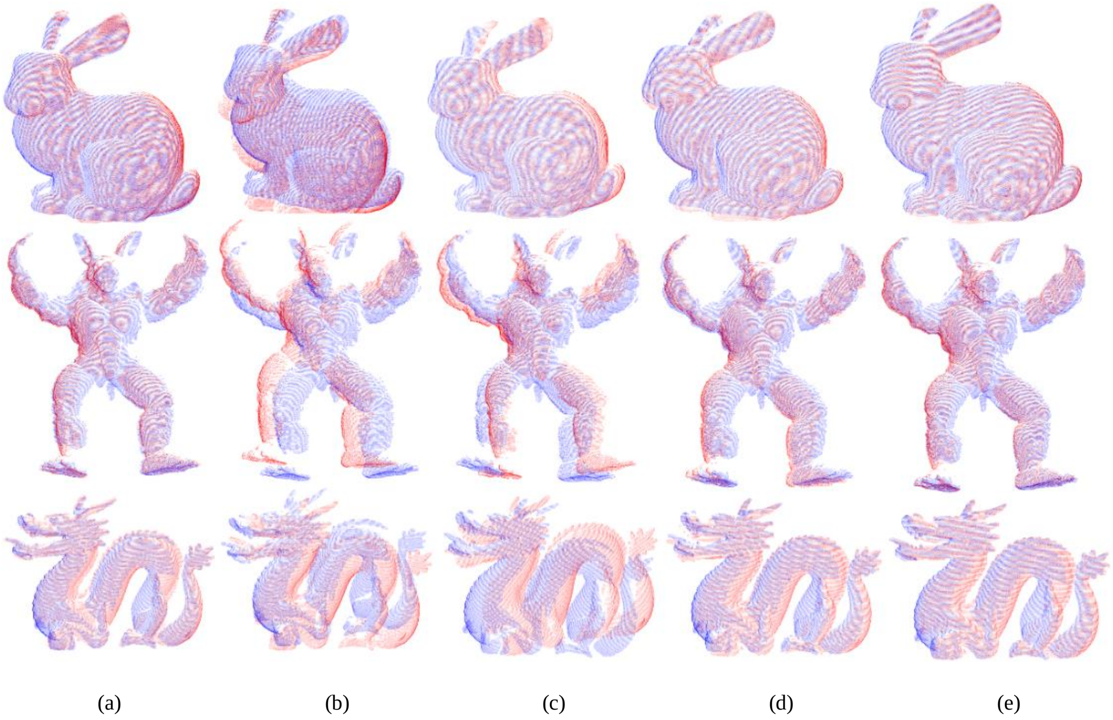


Fig. 11. Coarse registration experiment. The experimental dataset are bun000&bun045, ArmadilloOnHeadMultiple_0&30, and dragonStandRight_0&24. (a) SAC-IA + FPFH. (b) SAC-IA + 3DSC. (c) SAC-IA + SHOT. (d) SAC-IA + FPH. (e) Proposed algorithm.

Table 1. Point cloud coarse registration accuracy and time

Dataset	Algorithm	Error/10-6m	Time-consuming/s
bun000&045	SAC_IA + FPFH	2.572	24.626
	SAC_IA + 3DSC	6.574	80.117
	SAC_IA + SHOT	4.906	30.79
	SAC_IA + FPH	4.155	26.995
	Proposed	2.165	2.063
ArmadilloOnHead-Multiple_0&30	SAC_IA + FPFH	1.736	19.371
	SAC_IA + 3DSC	7.675	67.608
	SAC_IA + SHOT	7.5019	27.152
	SAC_IA + FPH	4.255	21.036
	Proposed	1.732	2.171
dragonStandRight_0&24	SAC_IA + FPFH	4.347	22.509
	SAC_IA + 3DSC	7.762	81.949
	SAC_IA + SHOT	7.998	28.1
	SAC_IA + FPH	6.993	22.579
	Proposed	3.866	1.891

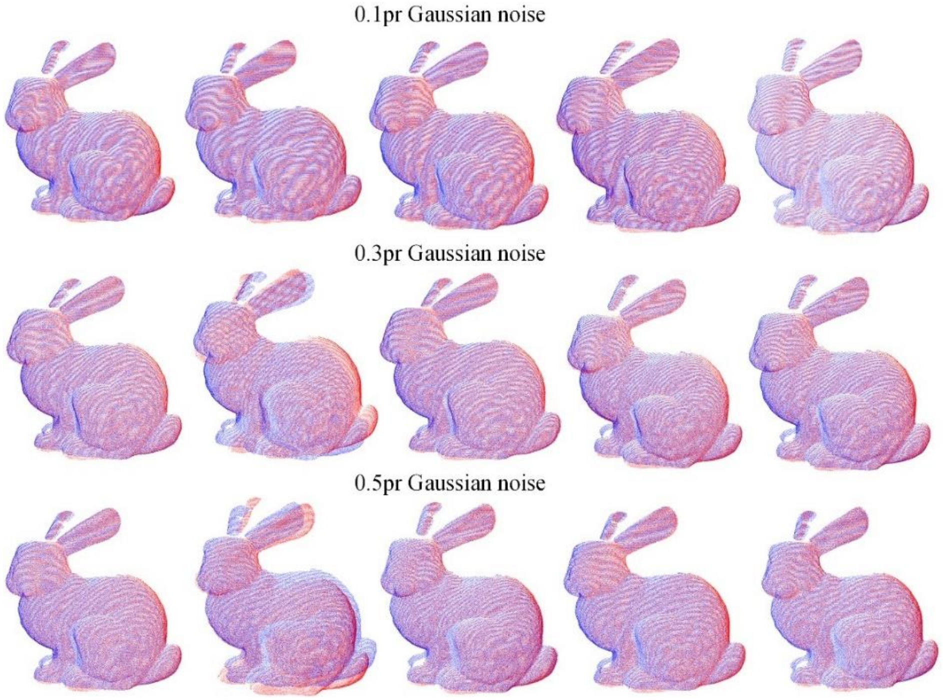


Fig. 12. Gaussian noise registration result, the experimental dataset are bun000&045. (a) SAC-IA + FPFH + ICP. (b)

SAC-IA + 3DSC + ICP. (c) SAC-IA + SHOT + ICP. (a) SAC-IA + FPH + ICP. (d) Proposed algorithm.

Table 2. Gaussian noise registration accuracy and time

Item	Algorithm	Gaussian noise threshold		
		0.1 pr	0.3 pr	0.5 pr
Time-consuming/s	SAC-I + FPFH + ICP	29.971	34.114	44.945
	SAC-I + 3DSC + ICP	97.682	138.822	150.765
	SAC-I + SHOT + ICP	41.835	55.135	62.923
	SAC-I + FPH + ICP	34.174	38.582	43.542
	Proposed	7.334	8.220	8.588
Error/10-6m	SAC-I + FPFH + ICP	2.318	2.179	2.180
	SAC-I + 3DSC + ICP	2.202	4.292	5.022
	SAC-I + SHOT + ICP	2.388	2.179	2.180
	SAC-I + FPH + ICP	2.165	2.202	2.326
	Proposed	2.161	2.169	2.171

4.3 Descriptor coarse registration experiment

The actual acquisition point cloud will be noisy. To further verify the robustness of the algorithm in the paper, Gaussian noise registration experiments are performed using the Bunny dataset from Stanford University. Gaussian noise of 0.1 pr, 0.3 pr, and 0.5 pr is added to the source and target point clouds. The SAC-IA coarse registration and ICP fine registration are performed using different descriptors and then compared with the algorithm in this paper. The registration elapsed time is the total time for coarse and fine registration, and the registration error is the RMSE mentioned above. The plot of the Bunny dataset registration results and registration data is shown in Fig. 12 and table 2.

It can be seen from Fig. 12 that the 3DSC descriptor registration results get worse as the Gaussian noise threshold increases, while all other methods can achieve good registration results. Table 2 shows that the operating efficiency of the algorithm in this paper is significantly better than that of the registration methods for other descriptors at different Gaussian noises levels. With the addition of Gaussian noise of 0.1 pr, 0.3 pr, and 0.5pr, the registration error of the algorithm in this paper is reduced by 4.58%, 12.97%, and 16.308% on average. Through the analysis of Gaussian noise registration results, the algorithm in this paper has high registration efficiency and accuracy, and also has good robustness.

5. Conclusions

In this paper, a three-dimensional feature descriptor (NPFC) for base and neighbor point pair features is proposed. The NPFC descriptor is a covariance matrix feature descriptor formed by computing the local neighborhood information of the point cloud. The experimental data show that NPFC has better robustness in describing 3D local surface information compared to FPFH, FPH, 3DSC, and SHOT. Based on the NPFC, we propose a point cloud registration method. The registration method can be divided into two stages: coarse registration and fine registration. The coarse registration process includes point cloud pre-processing, key point extraction, key point feature description, and feature matching. Point cloud preprocessing involves downsampling the point cloud and calculating the normal vector and curvature of the downsampled point cloud. The key point extraction is a set of points proposed by combining the average angle of the neighborhood normal vector with the ISS algorithm to extract geometric features. The key point feature description is the NPFC feature descriptor used to calculate the key point. Feature matching is done by using NPFC descriptors, using descriptor similarity and the RANSAC algorithm to find and filter the correspondence, and calculating the transformation matrix to complete the coarse registration. In the fine registration stage, the transformation matrix obtained from the coarse registration is used as the initial value, and then the ICP algorithm is used to complete the final registration of the point cloud. Compared with other feature descriptors, the coarse registration method using NPFC has better registration accuracy and provides better initial values for the ICP algorithm, which can effectively improve the registration accuracy of the ICP algorithm. The method in this paper has better accuracy, efficiency, and robustness to noise.

Funding. This project is supported by the University Innovation Research Group (No. CXQT20022), Research on key technologies and software for digital design and manufacturing of high-performance bevel gears based on cloud (No. cstc2022ycjh-bgzxm0218).

Disclosures. The authors declare that there are no conflicts of interest related to this article.

Data availability. Data underlying the results presented in this paper are available in Ref. [5].

References

1. S. Orts-Escolano, J. Garcia-Rodriguez, V. Morell, M. Cazorla, J. A. S. Perez, and A. Garcia-Garcia, "3D Surface Reconstruction of Noisy Point Clouds Using Growing Neural Gas: 3D Object/Scene Reconstruction," Neural Process Lett 43, 401–423 (2016).
2. B. Li, Y. Zhang, B. Zhao, and H. Shao, "3D-ReConstnet: A Single-View 3D-Object Point Cloud Reconstruction Network," IEEE Access 8, 83782–83790 (2020).
3. P. Kim, J. Park, Y. K. Cho, and J. Kang, "UAV-assisted autonomous mobile robot navigation for as-is 3D data collection and registration in cluttered environments," Automation in Construction 106, 102918 (2019).
4. Y. Ye, H. Chen, C. Zhang, X. Hao, and Z. Zhang, "SARPNET: Shape attention regional proposal network for liDAR-based 3D object detection," Neurocomputing 379, 53–63 (2020).
5. M. Shoukry and G. Gad, "Reverse Engineering: Investigation of Optimization Techniques in Point Clouds Registration," PAMM (2020).
6. P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 239–256 (1992).
7. A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 433–449 (1999).
8. A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," in Computer Vision - ECCV 2004, T. Pajdla and J. Matas, eds. (Springer Berlin Heidelberg, 2004), pp. 224–237.
9. R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (2008), pp. 3384–3391.
10. R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in 2009 IEEE International Conference on Robotics and Automation (2009), pp. 3212–3217.
11. S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," Computer Vision and Image Understanding 125, 251–264 (2014).

12. Y. Zou, X. Wang, T. Zhang, B. Liang, J. Song, and H. Liu, "BRoPH: An efficient and compact binary descriptor for 3D point clouds," Pattern Recognition 76, 522–536 (2018).

13. G. Vosselman, B. Gorte, G. Sithole, and T. B, "Recognising structure in laser scanner point clouds," Inter. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 46, (2003).

14. L. Kiforenko, B. Drost, F. Tombari, N. Krüger, and A. Glent Buch, "A performance evaluation of point pair features," Computer Vision and Image Understanding 166, 66–80 (2018).

15. T. Agami Reddy, "Mathematical Models and Data Analysis," in Applied Data Analysis and Modeling for Energy Engineers and Scientists, T. A. Reddy, ed. (Springer US, 2011), pp. 1–25.

16. O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," in Computer Vision – ECCV 2006, A. Leonardis, H. Bischof, and A. Pinz, eds. (Springer Berlin Heidelberg, 2006), pp. 589–600.

17. Y.-Y. Zhang, Z.-P. Wang, and X.-D. Lv, "Saliency Detection via Sparse Reconstruction Errors of Covariance Descriptors on Riemannian Manifolds," Circuits, Systems, and Signal Processing 35, 4372–4389 (2016).

18. P. Cirujeda, Y. Dicente Cid, X. Mateo, and X. Binefa, "A 3D Scene Registration Method via Covariance Descriptors and an Evolutionary Stable Strategy Game Theory Solver," International Journal of Computer Vision 115, 306–329 (2015).

19. X. Fengguang and H. Xie, "A 3D Surface Matching Method Using Keypoint- Based Covariance Matrix Descriptors," IEEE Access 5, 14204–14220 (2017).

20. Dr. V. Naidu and J. R. Raol, "Pixel-level Image Fusion using Wavelets and Principal Component Analysis," Defence Science Journal 58, (2008).

21. W. Liu, W. Sun, S. Wang, and Y. Liu, "Coarse registration of point clouds with low overlap rate on feature regions," Signal Processing: Image Communication 98, 116428 (2021).

22. W.Förstner and B.Moonen. A Metric for Covariance Matrices, 3rd ed. (Springer, 1996).

23. J. N. L. Brummer and L. R. Strydom, "An Euclidean distance measure between covariance matrices of speech cepstra for text-independent speaker recognition," in Proceedings of the 1997 South African Symposium on Communications and Signal Processing. COMSIG '97 (1997), pp. 167–172.

24. S. Quan and J. Yang, "Compatibility-Guided Sampling Consensus for 3-D Point Cloud Registration," IEEE Transactions on Geoscience and Remote Sensing 58, 7380–7392 (2020).

25. Y.-H. Jin and W.-H. Lee, "Fast Cylinder Shape Matching Using Random Sample Consensus in Large Scale Point Cloud," Applied Sciences 9, 974 (2019).

26. http://graphics.stanford.edu/data/3Dscanrep/

27. C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors," International Journal of Computer Vision 37, 151–172 (2000).

28. T. Hou and H. Qin, "Efficient Computation of Scale-Space Features for Deformable Shape Correspondences," in Computer Vision – ECCV 2010, K. Daniilidis, P. Maragos, and N. Paragios, eds. (Springer Berlin Heidelberg, 2010), pp. 384–397.

X. Yue, Z. Liu, J. Zhu, X. Gao, B. Yang, and Y. Tian, "Coarse-fine point cloud registration based on local point-pair features and the iterative closest point algorithm," Applied Intelligence 52, 12569–12583 (2022).