Journal of Information and Computing Science
J. Info. Comput. Sci., Vol. 19, No. 2, pp. 171-190

DOI: 10.4208/JICS-2024-010
December 2024

MFIPC: Point Cloud Registration Algorithm via Multi-feature Fusion and Interval Pairing Consistency

Xin Wei[1], Bing Tu[2,*], Siyuan Chen[3] and Jiadong Zhou[1]

[1] *School of Mechanical Engineering, Hunan Institute of Science and Technology, Yueyang 414000, China*
[2] *Institute of Optics and Electronics, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[3] *School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414000, China*

Abstract. Inspired by the Fast point Feature Histogram (FPFH) feature extraction algorithm, this paper proposes a new 3D point cloud registration method, MFIPC (Multi-feature Fusion and Interval Pairing Consistency). The method uses feature fusion and interval pairwise consistency to improve the registration accuracy. In the MFIPC framework, the point cloud is first downsampled to optimize computational efficiency and expand the analysis domain. Then, clustering algorithm using local directional centrality (CDC) classification algorithm is used to calculate the DCM (directional centrality measure) value of each point. The Gaussian curvature values of the points are calculated at the same time, and these eigenvalues are fused. To further refine the registration process, the range between the minimum and maximum eigenvalues is divided into several equal intervals and sorted in ascending order. A sorting algorithm is used to assign each eigenvalue to a corresponding interval. For the global point cloud computing step, after the operation is completed, the number of points in each interval and its proportion are calculated. The program processes both point clouds in order to analyze their interval percentage. This algorithm significantly improves the robustness of MFIPC in establishing point correspondence. To verify the effectiveness of MFIPC for 3D point cloud registration, we conducted extensive testing on various datasets, including 3DMatch, RESSO, ModelNet40, Stanford Rabbit, and Dragon. The experimental results show that the algorithm has high efficiency, good consistency of point cloud, significantly reduced registration errors, low error and high registration accuracy.

AMS subject classifications: 68W40, 68W05

Key words: Point cloud registration method, Multi-feature fusion, Interval pairing consistency

[*] Corresponding author. *Email addresses:* `vision911@163.com` (X. Wei), `tubing@nuist.edu.cn` (B. Tu), `siyuan@hnist.edu.cn` (S. Chen), `zhoujdwork@qq.com` (J. Zhou)

1 Introduction

Traditional 2D image sensors can only obtain two dimensional image information, for the detection of complex scenes, such as the number of people passing through the intersection statistics, semantic segmentation [1], the need for complex algorithmic processing of two-dimensional image data, which will take up most of the CPU's computing resources, and the real-time accuracy is low, so the threedimensional depth camera has become a hot spot in the current research of machine vision, which is more than the two dimensional image, which has more depth information [2] [3]. The extraction of three-dimensional information can complete the object recognition, classification [4], motion tracking and other aspects of complex applications. Two kinds of binocular stereo vision matching algorithms are available: Partial and total [5]. The method of partial has the advantages of simple implementation and high performance for image regions with rich texture, but it suffers from low accuracy for photo regions with poor or repetitive configuration [6–10]. Compared to the method of partial, the total one performs better on matching low and repetitive substance, but it has higher computational complexity, more dependency on the object in the scene, and weak compatibility with on-line addressing. Both techniques have difficulties in solving the occlusion region problem, and they are not robust enough to deal with different measurement scenes [11].

Provided a pair of point clouds, the registration tries to find the best tele-transform alignment with the smallest possible error for multiple point clouds to obtain a complete 3D scene [12] [13].As spending on 3D scanning equipment decreases, point cloud processing technology continues to evolve. Point cloud registration is often applied in AR/VR, autonomous driving [14], robotics and other fields. The registration quality affects the purpose and effect of image registration, such as unmanned driving, pattern recognition, and 3D reconstruction [15]. The achievement of high registration quality depends heavily on the choice of registration methods, the adjustment of registration parameters, and the evaluation criteria used. Given the diverse range of application scenarios and needs, different registration qualities may be required.

Point cloud registration algorithms have been researched in the 1970s, we often use the ICP algorithm [16] (which is an iterative nearest-point based point cloud fine registration method, and it aims to go and locate a rigid transformation including rotational and translational transformations) to maximize the registration between two point clouds. The algorithm is simple and light to come true, applicable to various types of point cloud data; The disadvantage is that it is easy to trap in a local optimum, the prerequisite for implementation is the need for a good initial transform, which is sensitive to noise and occlusion.

RANSAC algorithm is a robust registration method based on the consistency of random samples, which aims at the same as the ICP algorithm, so as to minimise the error between

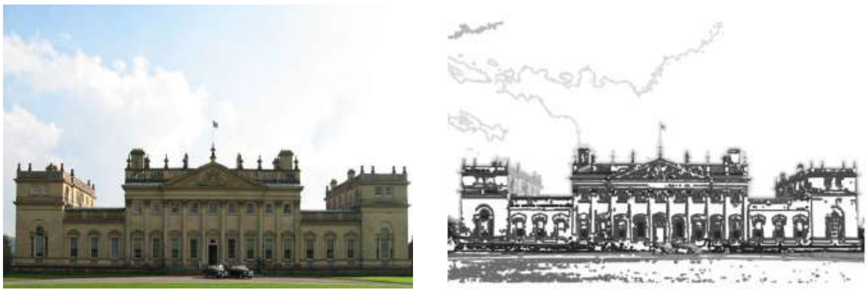


Figure 1: Extraction contour: the building drawing is on the left and the contour extraction drawing is on the right

the two point clouds [15]. The advantages of the RANSAC algorithm is that it can resist a large amount of noise and outliers and is appropriate for miscellaneous types of point cloud data, the disadvantage is that it needs to set several parameters, such as minimum subset size, threshold, preset ratio, etc. and it takes a long time to run. NDT (normal distributions transform) is a traditional method for fine registration, which aligns the source points with the target points by optimizing the scores of source points on the normal distributions probability density, which is computed after voxelizing the target points. [17]. ICP, NDT, and their variations need the point cloud to be pre-aligned, or to have previous poses from sources like IMU (inertial measurement unit), GNSS (global navigation satellite system), and odometer. If not, rough registration is necessary to give it a starting posture [18].

In the existing point cloud feature extraction methods, there have been a variety of registration methods such as point features, line features and surface features, etc.The corresponding point cloud registration method is robust and accurate [19] [20]. The 4PCS algorithm is a coarse registration method, which does not need to provide the initial position, and can offer a fine starting position for the great registration method [21]. The 4PCS algorithm can deal with the noise and shelter of the point cloud, even for a small amount of outliers "contamination" of the point cloud data, there is no need for filtering and denoising, and the quality of the registration is generally better. 4PCS algorithm's disadvantage is that the running time is longer, for the feature extraction and the error computation is time-consuming.

The traditional FPFH method only considers the nearest neighbour message of every point in the point cloud, and the algorithm focuses on the computation of each point, with less consideration of global information, resulting in the algorithm not being robust enough [22] [23]. If there is noise or occlusion in the point cloud, the stability of the algorithm decreases, resulting in poor registration accuracy. This paper presents a point cloud registration algorithm that incorporates CDC and FPFH feature extraction to tackle the intricate issue of registration fusion between theoretical and authentic original shapes, building upon the aforementioned background research. The CDC algorithm [24] en-

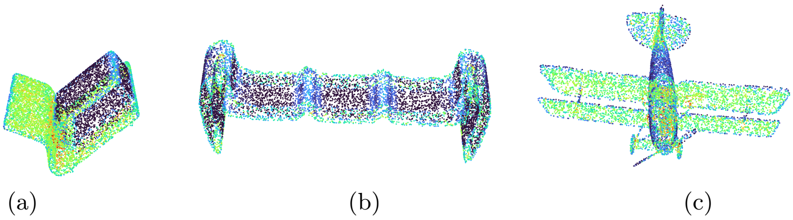


Figure 2: Figures(a), (b), and (c) show the point clouds of the sofa, bench, and aircraft in the ModelNet40 dataset derived using the DCM method. Using the DCM method allows the contours of the objects to be more clearly visible

sures precise extraction of feature point ranges, the introduction of boundary points can significantly improve the feature expression capability and alignment robustness of the algorithm, because the high curvature region corresponding to boundary points is often highly correlated with the key geometric structure of the target object, which can be used as sparse but highly distinguishable feature descriptors to reduce the probability of mismatching in the fuzzy region of the point cloud; furthermore, by extracting the boundary points, we can significantly reduce the size of the point cloud involved in the alignment, and in combination with the sampling strategy based on boundary features, we can significantly reduce the computational complexity while maintaining the alignment accuracy. Combined with the sampling strategy based on boundary features, the computational complexity can be significantly reduced while maintaining the alignment accuracy. while the FPFH algorithm guarantees accurate alignment of features for coarse registration of point pairs. The algorithm's registration precision, error analysis, and computational cost are validated through rigorous comparative testing. Currently, 3D point cloud registration techniques remain an active research area, with numerous individuals pursuing new advancements that continue to revolutionize the field. The method introduced in this paper builds upon and enhances the classical FPFH feature algorithm, delivering comparatively excellent results. The main contributions of this paper are concluded below:

- An efficient and precise point cloud registration algorithm is proposed, which explores an effective feature range and learns point features that are invariant to rotation and translation of the point cloud.
- Through a series of experiments on indoor point cloud scans as well as synthetic and real models, we prove the capability of the designed features in handling geometric registration tasks.
- By fusing Gaussian curvature and DCM values to construct multi-feature descriptors, the weighted fusion strategy enhances the comprehensiveness of feature expression. The paper conducted an experimental verification of this method. The internal points in the point cloud can be clearly distinguished from the contour points, and the error accuracy after alignment is much smaller than that of other methods.

The core idea of innovative points in this paper is to distinguish the edge points and

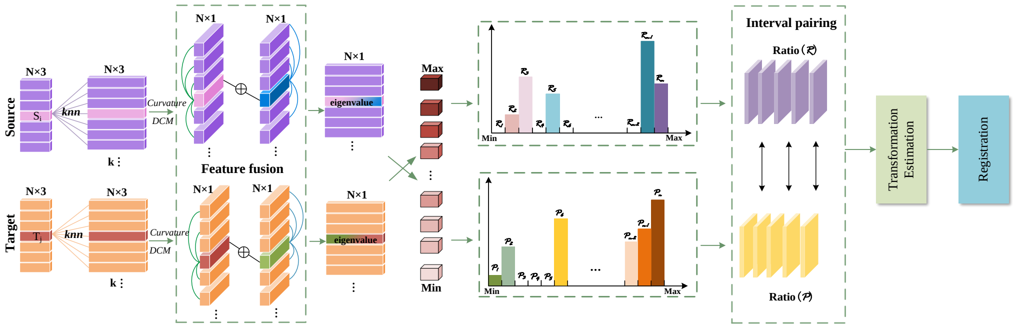


Figure 3: System flow chart: The system first inputs the point cloud, then calculates the DCM value and curvature value, the feature fusion calculation, and finds the maximum and minimum number of the feature value, then performs the interval proportion calculation, interval correspondence calculation, and finally the transformation estimation and registration

internal points in point cloud based on KNN distribution. The calculated DCM value can well distinguish the boundary point from the internal point, and can outline the object outline, as shown in Fig.2. The robustness of the points is strengthened by the Gaussian curvature value of the points. Multi-feature fusion makes the local features of the description unique and enhances the accuracy of point cloud registration [25] [26].

2 Related Work

2.1 Registration based on conventional local optimum algorithms

The ICP algorithm is an iterative nearest-point based point cloud fine registration way that is usually used in conjunction with other coarse registration methods [27]. The goal is to get a stiffness transformation (including rotations and translations) that maximises the telescope between two point clouds.The basic steps of the ICP algorithm are: given an starting transformation, the origin and target point clouds are transformed so that they roughly overlap. Then, for each point in the origin point cloud, the closest point in the target point cloud is found as its matching point, and unreasonable pairs of matching points are eliminated. Then, based on these corresponding pairs, an optimal transformation is solved using the least squares method to minimise the sum of squares of the errors between the source and target point clouds. Finally, the position of the source point cloud is updated with this optimal transform, and the above steps are repeated until the transform converges or reaches a preset number of iterations or an error threshold [28].

The advantages of the ICP algorithm are that it is simple and easy to implement, does not require segmentation and feature extraction of the point cloud, and is applicable to various types of point cloud data. The disadvantages are that it is easy to trap in a local optimum, needs a better initial transformation, and is sensitive to noise and occlusion, only considers the distance between points and points, and lacks the use of the whole architecture of the point cloud.

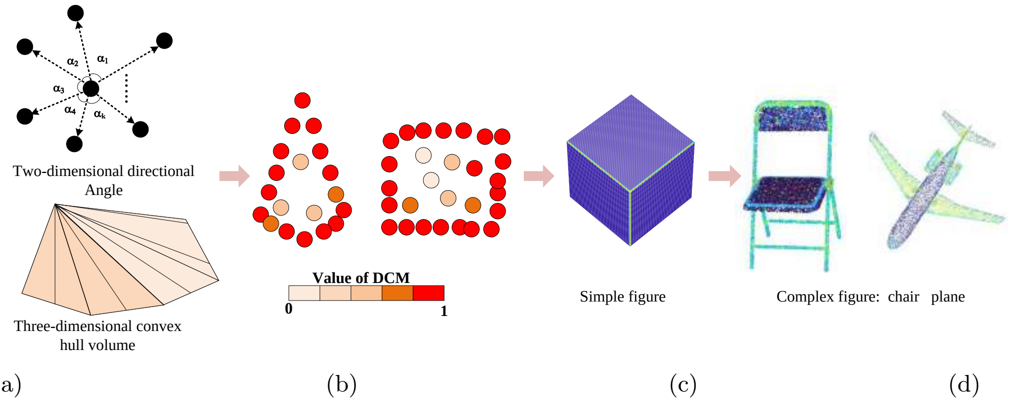

α1
α2
α3
α4
αk
Two-dimensional directional Angle
Three-dimensional convex hull volume
Value of DCM
0
1
Simple figure
Complex figure: chair plane
(a)
(b)
(c)
(d)

Figure 4: In Figure (a), DCM is the Angle between the calculated directions in two-dimensional space and the convex hull volume in three-dimensional space; Figure (b) is the calculation result of the sample data of DCM, and the DCM value of the boundary region is close to 1, and the internal point is close to 0. Figure (c) is a schematic diagram of the DCM calculation results of a simple three-dimensional graph, and the junction points are highlighted above the plane points. Figure (d) is a schematic diagram of the DCM calculation results of complex three-dimensional graphics, and the junction points are highlighted above the plane points

2.2 Feature-based point cloud registration method

Some common point cloud features include: point features: extracting key points by farthest point sampling method [29], normal space sampling method [30] , and applying covariance matrix [31], line features: extracting curves by dividing planar blocks [32], extracting point cloud pole-like points, intersections, and vertices, and connecting them into three types of vertical feature lines [33], extracting curves, calculating curvature, and dividing curvature to fit it as the extraction of line features; face features: calculating planar features, dividing the point cloud into voxels, calculating the curvature of the points within the voxels, and combining the planar blocks that satisfy the planar requirements into planar features [34] [35]. Using the feature-based point cloud regiatration method, the overall steps are firstly, to extract the corresponding point features, line features or surface characterisations from the two point clouds, which can reflect the local geometric information of each point or region in the point cloud, for example normal vector, curvature, angle, etc. Then, based on the analogy between these features, the corresponding features in the two point clouds are searched, and these correspondences can be used as initial matching point pairs for subsequent registration. Finally, based on these matched point pairs, an optimisation method is utilised to settle a rigid transformation (including rotation and translation) that minimizes the inaccuracy between the two point clouds.

2.3 Point cloud registration approach based on statistical analysis

The features detected by the above method may not be fully applicable to feature matching, because it will be affected by noise, occlusion and other factors, so the step of a feature descriptor is needed to compare and match features more accurately [36]. Typical

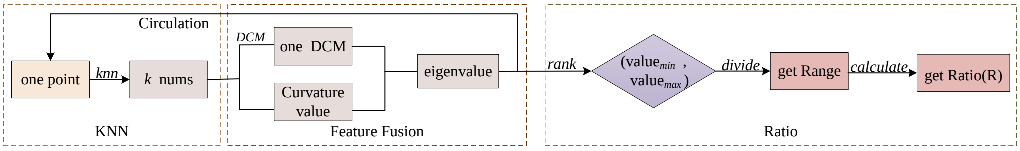


Figure 5: Innovative point flow chart: Firstly, K-Nearest Neighbors (*KNN*) algorithm is used to locate k points around a given point, and the DCM value and curvature value of each point are calculated. The global eigenvalue is obtained by iterating this process, and then the values are sorted and divided into equal intervals. Calculate the proportion of each interval. This calculation is performed on both point clouds to assess whether the ratio of the corresponding interval is the same

descriptors are feature signatures and feature histograms. FPFH algorithm belongs to category feature histogram, which represents point cloud information as histogram. It is a simplified version of PFH (Point Feature Histogram) algorithm to improve computational efficiency. The principle of the two algorithms is the same, both are based on statistical analysis to draw the geometric features of the point cloud, and construct the corresponding histogram to obtain the statistical features.

In the FPFH algorithm, a fixed neighborhood is first defined and then a series of eigenvalues are computed in this neighborhood and a histogram of the eigenvalues is built to obtain the statistical features. By comparing the histograms of features between point clouds, matching and registration can be performed, enabling applications such as point cloud registration and regeneration. This has the advantage of taking into account the geometric characteristics of the points as well as the assignment of features in their neighboring regions, avoiding situations where anomalous points are incorrectly identified. In point cloud registration, the FPFH algorithm can be used for accurate registration [37]. The method proposed in this paper combines the methods of point features and statistical analysis, which effectively solves the limitation problems of parameter sensitivity and noise robustness of the existing traditional algorithms, significantly improves the accuracy of point cloud feature matching, and provides a high-precision solution for point cloud alignment in complex scenes.

3 Innovative Method

The boundary points effectively depict the contour shape of an image, greatly enhancing image recognition and segmentation capabilities. By diminishing the informational content and intricacy of the image, they simplify subsequent processing tasks. Notably, despite occupying only a fraction of the overall image, the boundary points encapsulate the majority of pertinent information. The regularity and continuity exhibited by these points permit the use of a limited dataset to represent the entire image contour, thereby achieving image compression (as Fig.1). Compressed images not only conserve storage space but also boost the efficiency of image processing.

Due to the complexity of the network structure, the boundary point algorithm can incur high computational costs for large-scale image datasets. The performance of the algorithm relies on the correct selection of parameters, such as neighborhood size and threshold choice. In cases where there are too many or too few feature points, further optimization is required to select the most meaningful eigenvalues. In response, we further introduce the curvature feature of points and consider integrating Gaussian curvature with the boundary point algorithm to form a multi-scale feature descriptor for point clouds. By describing the features of point cloud data at different scales, we can construct the feature vector of the point cloud.

3.1 DCM value calculation

The basic concepts of CDC classification algorithm lies in distinguishing edge points from interior points within a cluster, primarily based on the distribution of K-Nearest Neighbors (KNNs). This method involves outlining the object roughly by highlighting boundary points and creating containers to regulate the connectivity of interior points. To assess the variance in direction allocation, we utilize the angular deviation created by the KNN in two dimensions as local direction-centred measure (DCM).

The principle of DCM calculation in 2D space is as follows: first get the X, Y coordinates of the input points, a total of n points, then get a 2D array of n rows and 2 columns. Then take the KNN algorithm: determine the point coordinates of a point (X_i, Y_i), use the KNN algorithm to find its surrounding proximity to the k points, write down the coordinates (X_k, Y_k), calculate the difference between the point and the angle of the surrounding k points, each point performs the above operation, and finally get an $n*k$ dimensional array, each row of the array can be calculated a DCM value.

After all the calculations, n DCM values are obtained, i.e., an array of n rows and 1 column; then the obtained DCM values are sorted, and a threshold is set to distinguish the edge points from the inner points (The larger the event, the closer to the boundary, the lower the event, the closer to the center).

DCM value calculation method: calculate the difference between two adjacent numbers in a row, the last number and the first number for the difference calculation, in this way, will be able to get n angular differences, these differences in the range of 0 to 2π, using the formula

$$DCM = \frac{1}{n}\sum_{i=1}^{n}\left(pa_i - \frac{2\pi}{n}\right)^2. \tag{3.1}$$

The DCM value for the row can be derived.From this, we derive the three-dimensional space: the knn are used to the unit sphere. They link neighbouring points to create a ball-shaped triangle, and the DCM is spreaded to the real angle variance of the triangle. For hyperspherical subdivision in higher dimensional space, the qhull algorithm is used to construct convex complexes of KNNs [38]. In three-dimensional space, a convex complex form is a convex polyhedron consisting of a number of tetrahedra, and each of its faces is a two-dimensional simplex (a simplex denotes the simplest diagram), is a triangle. Since all knn points are mapped onto the hypersphere, they are the vertices of the convex polytope.

In N dimensional space, each face of a convex polytope is an $N-1$ dimensional simplex, corresponding to a subdivision unit, which is subdivided to measure generalised angles. These angles correspond to the volume of the counterpart subdividing cell(typically the arc length in a two-dimensional circle and the area of a spherical triangle in a three-dimensional sphere).Computing the volume of subdivision units in higher dimensional spaces is challenging because of the computational complexity of multiple integrals. Hence, we have to estimate the volume of each simplex and allocate the global volume incorrect evenly to per partition cell for an approximation.Despite the error between the real and calculated volumes of the partition cells, since the volume of the subdivision cells monotonically increases with the matching simplex, the DCM sorting order based on these two volumes is the same. Therefore, DCM thresholds can be set to efficiently identify interior and border points. Points on intersecting surfaces will be brighter than interior points, allowing for better depiction of object shapes.

Suppose that a simplex in 3D space is consisted of k KNN points n_1, n_2, ... , n_k which have been reflected onto a hyperball, where $n_i = (x_i^1,\ x_i^2,\ ...,\ x_i^k)$, and the vectors $\overrightarrow{n_k p_1}$, $\overrightarrow{n_k n_2}$, - - - , $\overrightarrow{n_k n_{k-1}}$, n_k with the same origin can be determined as a two-dimensional parallel hexahedron n in vector space. Let $\xi_i{=}\overrightarrow{n_k n_1}{=}(x_i^1\text{-}x_i^1,\ x_i^2\text{-}x_i^2,\ ...\ ,\ x_i^k\text{-}x_k^k)$ and $G{=}(\xi_1,\ \xi_2,\ ...,\ \xi_{k-1})$, then we have the volume of n ([39] [40]) as

$$vol(n) = \sqrt{\det(\mathbf{G}\mathbf{G}^\top)}. \tag{3.2}$$

A simplex Y is a hypertetrahedron embedded in a parallel hexahedron n that shares $k-1$ edges $n_k n_1$, $n_k n_2$, ... , $n_k n_{k-1}$. Therefore, the volume of Y is [41]

$$vol(y) = \frac{vol(n)}{\Gamma(k-1)} = \frac{\sqrt{\det(G\mathbf{G}^\top)}}{(k-1)!}. \tag{3.3}$$

We compute the difference between the sum of simplex volumes and the surface area of the hyper-sphere after measuring volume of a simple form. Then we divide overall volume tolerance equally among per subdivided cell. This ensures that the volume sum of subdivided cells under various partitions is invariant in the same dimension. k-dimensional unit sphere has a generalised surface area of

$$y = \frac{2\pi^{\frac{k}{2}}}{\Gamma(\frac{k}{2})}. \tag{3.4}$$

We assume that the convex complex consists of j simplexes y_1, y_2, ... , y_j, and accordingly subdivide the hypersphere into j units v_1, v_2, ... , v_j. The volume of the subdivided cell v_i can be solved for ([24] [42])

$$vol(v_i) = vol(y_i) + \frac{Y - \sum_{i=1}^{j} vol(y_i)}{j}. \tag{3.5}$$

DCM measures the difference in volume of all subdivided cells [24]

$$DCM=\frac{1}{j}\sum_{i=1}^{j}\left\{vol(vi)-\frac{y}{j}\right\}^{2}. \tag{3.6}$$

Algorithm 1 gives an outline of the DCM calculation process. Lines 1 and 2 use the KNN algorithm to find out the proximity points and calculate the coordinate difference between its own point and the surrounding proximity points. In lines 3 to 6, dimensionality reduction is calculated using the map function, and when the calculated dimension is 2, the DCM value is directly taken as 0. Lines 7 to 9 calculate the individual convex packet volume using the convhulln function, and line 10 starts calculating the global convex packet volume using a for loop, and then calculates the volume variance to derive the DCM using the variance function, ending the DCM algorithm.

Algorithm 1 Outline of the DCM algorithm

```
Function [DCM] = DCMs(n, X, get knn)
**Input:** point cloud A
 1: **for** 1:n **do**
 2:    Calculate the difference between the x-coordinates of the k points in the KNN
       neighborhood around the point .
 3:    Dimensionality :
 4:       $k=rank(map_x)$;
 5:          $ifk==2$
 6:             $DCM(i)=0$;
 7:    Compute the convex envelope function.
 8:    Count the number of convex polygonal surfaces:
 9:     $simplex_num=length(convex(:,1))$;
10:    Dim-pack volume:
11:  $simplex_vol=zeros(simplex_num,1)$;
12:    The for loop calculates the convex packet volume:
13:     $forj=1:simplex_num$;
14:    Calculate the volume variance to find the DCM value.
15:    $DCM(i)=var(simplex_vol)$;
16: **end for**
End Function
```

3.2 Calculated gaussian curvature

Gaussian curvature is a quantity used in differential geometry to describe the degree of curvature of a surface. It is defined as the product of the principal curvatures k_1 and k_2 at a point on the surface. Mathematically, the Gaussian curvature K is defined as K = k_1

$\times$ k_2. Alternatively, it can also be calculated using the radii of curvature R_1 and R_2 at that point, which is K $= 1/(R_1 \times R_2)$. The Gaussian curvature of the point and the local centrality metric value are fused to obtain new eigenvalues.

3.3 Calculation of the interval ratio

The calculation of eigenvalues was executed for both point clouds. For one of these clouds, the eigenvalues were systematically categorized into distinct intervals, and the relative proportion of each interval's eigenvalues was determined, considering the total count of eigenvalues.

Given the range of eigenvalues spanning from 0 to 1, this range was partitioned into ten equal segments, each segment possessing a length of 0.1. A counting-like process was then initiated to identify the specific interval corresponding to each eigenvalue. By systematically incrementing the count in the respective interval, we arrived at the final tally of eigenvalues in each interval. For example, in the interval spanning from 0 to 0.1, there were precisely 20 eigenvalues. Assuming a total count of 1000 eigenvalues, the percentage representation of this interval would be 2%. Additionally, the interval between 0.8 and 0.9 had no recorded eigenvalues, indicating the absence of points with eigenvalues within this specified range. With a total count of 1000 eigenvalues, the percentage representation of this interval would be 0%. These interval ratios were calculated for both the source and target point clouds.

3.4 Interval Correspondence Calculations

Based on the traditional statistical analysis method, this paper makes an innovation and pairs the interval ratio values of two point clouds to determine whether the corresponding interval ratio values are the same. The rotation matrix and translation matrix of the generated point cloud are calculated with the Groundtruth value of the original data, so as to achieve the purpose of registration.

4 Experimental Results and Parameter Analysis

In this paper, experiments and method comparison experiments were conducted on ModelNet40 datasets, 3DMatch datasets, RESSO datasets and Stanford datasets and the results of the experiments are analysed below.The experimental results show that the algorithm proposed in this paper is much better than the FPFH algorithm in terms of rotation error(R_err), translation error(T_err), root-mean-square error(RMSE), and mean absolute error(MAE), and exhibits remarkable advantages.

In the parameter setting of this paper, careful considerations are made regarding the value of the downsampling parameter. When the parameter value is set too high, the resulting point cloud data is significantly reduced, making it impossible to achieve effective and precise alignment, thereby affecting the final matching effect. Conversely, if the

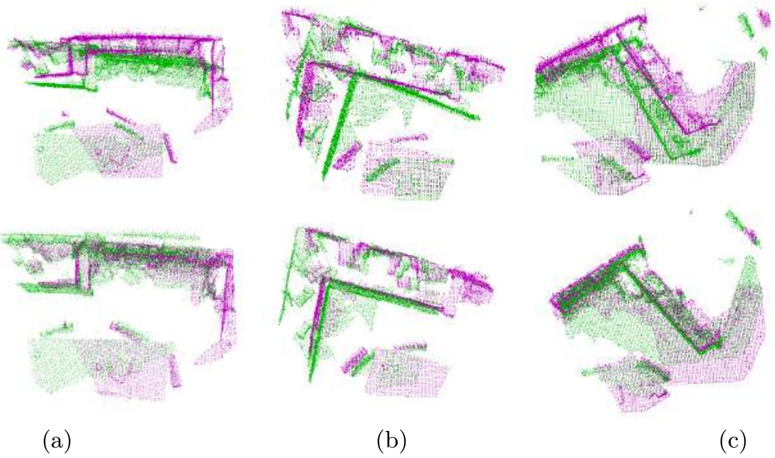


Figure 6: 3Dmatch dataset Kitchen scene: Three scenarios(a), (b)and(c), the first row is pre-registration, the second row is post registration

parameter value is set too low, the point cloud data becomes excessively dense, greatly increasing the processing time and placing higher demands on the operating equipment, posing numerous challenges in practical applications. Therefore, after comprehensive considerations, we have chosen an intermediate range of values for the downsampling parameter to balance alignment accuracy, processing efficiency, and equipment requirements.

4.1 Indoor Point Cloud Registration

We train and count the newly put forward method in this paper with interior point cloud sweep data from the 3DMatch dataset. The dataset includes various indoor scenes such as bedrooms, kitchens, offices, laboratories and hotels. We select two different locations within the kitchen scenes in the dataset for reading the point clouds. As in Fig.6, the first column on the left is the point cloud registration for the two scenes Cloud_bin8 and 9 of the kitchen; the middle column is the point cloud registration for the two scenes Cloud_bin11 and 12 of the kitchen; and the rightmost column is the point cloud registration for the two scenes Cloud_bin22 and 23 of the kitchen, with the downsampling threshold taken as 0.03, and the KNN taken as 30.

The GroundTruth value comprises of a rotation matrix and a translation matrix, we calculated the error between the evaluation matrix generated after the registration and the GroundTruth matrix (including the rotation error and translation error), and the error accuracy is much smaller than that of 3D-SIFT algorithm [43], and the experiments will be done to compare the other scenes needing to be aligned with the 3D-SIFT method, and the table of the error accuracies is shown in Table 1.

The table of the comparison between the method of this paper and the 3D-SIFT algorithm under different parameters and different registration environments is shown in Table 1. 3D-SIFT algorithm in different parameters and different registration environments to

Table 1: The 3D point cloud registration performance of the proposed method (Ours) and the 3D-SIFT algorithm is evaluated through comparative experiments at different sampling rates (sub=0.04 and sub=0.06) and different alignment scenarios, with the main metrics including rotation error (R_err), translation error (T_err), root mean square error (RMSE) and mean absolute error (MAE). As shown in the results in the table, the error of the proposed method is reduced by 73.5% in RMSE, 12.9% in R_err and 77.7% in MAE

Parameter	Scene	Ours				3D-SIFT			
		R_err	T_err	RMSE	MAE	R_err	T_err	RMSE	MAE
sub=0.04	02 and 12	35.6067	1.1215	0.2780	0.1979	40.5900	2.7650	0.8151	0.6562
	02 and 40	39.5486	0.3923	**0.0353**	**0.0251**	57.8700	0.3905	0.1334	0.1125
	02 and 41	46.6357	**0.3911**	0.2590	0.1819	55.1540	0.9154	0.2890	0.2415
sub=0.06	02 and 12	36.9505	1.1266	0.3115	0.2638	42.5860	2.3181	0.8814	0.6544
	02 and 40	16.4478	0.7157	0.2734	0.1935	36.7511	2.1586	0.7130	0.5403
	02 and 41	**14.7876**	1.3079	0.4021	0.2811	16.9801	1.9733	0.8154	0.7052

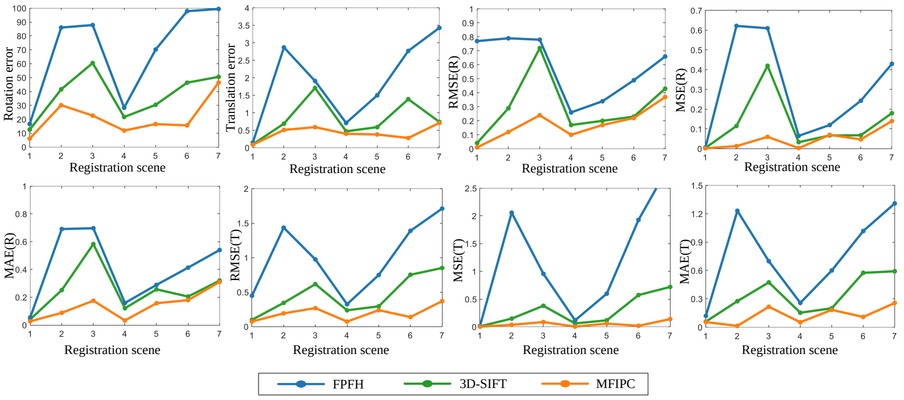
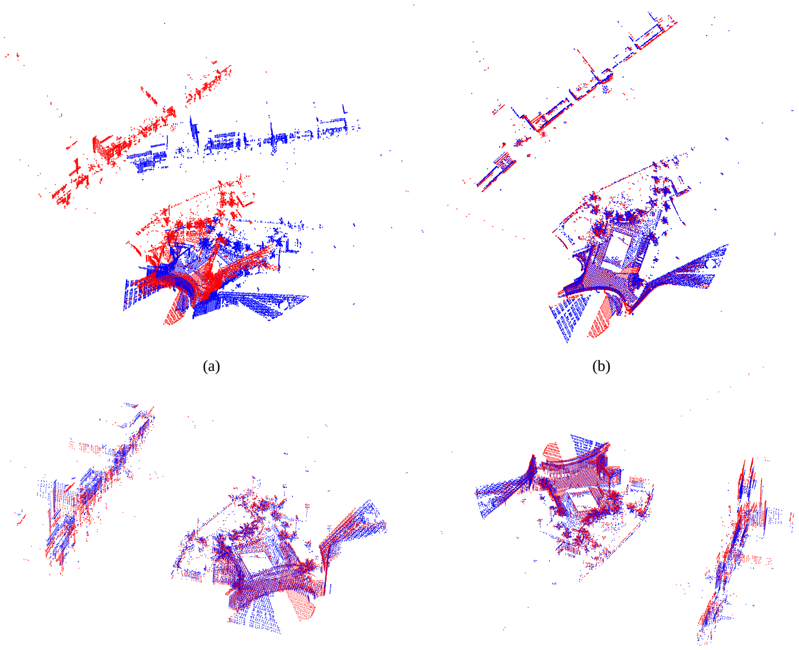


Figure 7: Three methods of registration error line chart

do a comparison, when the downsampling parameter is set to 0.04, will be the same two scenes (such as Scene2, 12) for the registration, one with this paper's method, the other with the 3D-SIFT algorithm, it can be clearly seen that this paper's method of R_err and T_err are much lower than 3D-SIFT algorithm. The same parameters then Scene2, 40 and Scene2, 41 for registration, this paper's method of the two errors are much smaller.

Overall, the proposed method has significant advantages in reducing RMSE and MAE, and especially performs more stably at low sampling rates. The experimental results show that the method can effectively adapt to different point cloud densities and scene complexities, and improve the robustness of the algorithm while ensuring the alignment accuracy.

In Fig.7, we can intuitively see that the error accuracy obtained by the multi-feature fusion registration algorithm is the smallest in eight different scales. The downsampling value is 0.09, the blue curve represents the classical registration method FPFH, the green curve represents the 3D-SIFT method, and the orange curve represents the multi-feature fusion method proposed in this chapter. The number on the horizontal axis represents the scene number, representing the registration between scene 0 and the scene.



(a)

(b)

(c)

(d)

Figure 8: RESSO dataset figure7D part0, 1, downsampling is taken as 0.9, and knn is taken as 30. figure (a) is the point cloud before registration, and figures (b) (c) (d) are the top, side, and rear views, respectively, of the point cloud after registration

The vertical axis from left to right from top to bottom is: rotation error, translation error, rotation matrix root mean square error; mean square error of rotation matrix, average absolute error of rotation matrix, translation matrix root mean square error, Translation matrix mean square error, translation matrix mean absolute error.

4.2 RESSO Datasets

The RESSO dataset boasts an array of point clouds that meticulously capture the intricate details of both urban and castle landscapes. For the purposes of our experiment, we diligently chose two point clouds from each of these scenarios. The results of our method, as it was applied to these point clouds, are comprehensively outlined in Table 2. Notably, the downsampling values utilized were 0.5, 0.9, 1.5, and 2.0. The precision of each error achieved is notably superior to that of the 3D-SIFT algorithm. Our method demonstrates higher registration accuracies than the 3D-SIFT algorithm across diverse point cloud segments within the same outdoor scene. To delve deeper into the registration outcomes, we zoomed in on the partial point clouds. By closely examining the alignment results of the 3D point cloud presented in Fig.8, along with the point clouds captured from various viewpoints (illustrated in Figs.7c and 7d), it becomes abundantly clear that the point clouds for both scenarios have been successfully registered. This enhanced understanding

Table 2: The alignment performance of the proposed method (Ours) is compared with the 3D-SIFT algorithm for different downsampling values (sub=0.5, 0.9, 1.5, 2.0) under the RESSO dataset (Scene 0 and Scene 1), and the evaluation metrics include rotational error (R_err), translation error (T_err), root-mean-square error (RMSE) and mean absolute error (MAE) . The experimental results show that the alignment error values of the proposed method are smaller, with the rotation error reduced by 10.4% and the translation error reduced by 82.9%

Dataset	Parameter	Ours				3D-SIFT			
	para	R_err	T_err	RMSE	MAE	R_err	T_err	RMSE	MAE
RESSO part 0/1	sub=0.5	62.9408	**0.3755**	**0.0108**	0.0091	66.4482	2.1990	0.0031	0.0026
	sub=0.9	63.4642	0.6282	0.0128	0.0113	66.5934	2.0740	0.0068	0.0040
	sub=1.5	**60.3010**	1.1687	0.0129	**0.0083**	67.3677	3.0300	0.0062	0.0053
	sub=2.0	62.7329	2.3683	0.0345	0.0257	66.7735	5.6630	0.1373	0.0908

of the registration process offers valuable insights into the effectiveness of our method.

4.3 Stanford datasets

We tested textual algorithm on a 3D point cloud from the Stanford Rabbit dataset [44–46].The registration results were taken when the downsampling thresholds were taken as 0.005 and 0.001, respectively, and the experiments showed that the registration results were better with the method, and that the registration was better when the downsampling values were lower. We compared MFIPC with other ways and the outcomes are shown in Table 3.

Table 3: The performance of different 3D point cloud registration methods is compared on the Stanford dataset, and the evaluation metrics include rotation error (MSE(r), RMSE(r), MAE(r)), translation error (MSE(t), RMSE(t), MAE(t)), and running time (TIME). Experiments show that the proposed method (Ours) significantly outperforms the existing methods in both rotational and translational errors. However, the running time is relatively the longest and the efficiency of the algorithm still needs to be optimised

Approach	MSE (r)	RMSE (r)	MAE (r)	MSE (t)	RMSE (t)	MAE (t)	TIME
ICP [47]	168.59	14.01	10.88	0.0024	0.0502	0.0242	1.7002
Go-ICP [48]	161.38	12.88	5.03	0.0018	0.0418	0.0282	1.2990
R-PointHop [49]	2.19	1.39	0.97	0.0013	0.0378	0.0259	0.9755
FGR [50]	4.01	2.01	1.32	0.0406	0.1989	0.0661	3.4621
PointNetLK [51]	39.89	5.99	4.81	0.0016	0.0406	0.0369	0.2764
Ours	**0.46**	**0.68**	**0.58**	**0.0012**	**0.0331**	**0.0205**	7.3000

Table 3 shows that the algorithm proposed in this paper has lower error accuracy, but the registration time is slightly longer [45]. The reason for this phenomenon is that when calculating the eigenvalue of the point, it needs to be calculated to all the points, so the calculation time is long. Later we will consider how to reduce the time complexity and running time.

We conducted registration tests for rabbit and dragon with different parameter values, as shown in Fig.9 and 10. In Fig.9, from left to right, each column represents a subsample

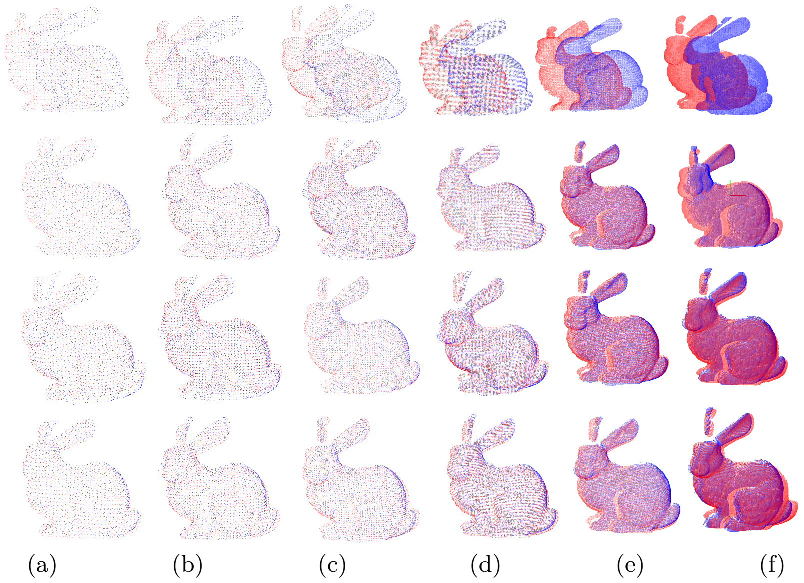


Figure 9: The registration experiment was carried out on the Stanford rabbit dataset, and the experiment was carried out on the same two rabbits, and the variables were different downsampling parameters

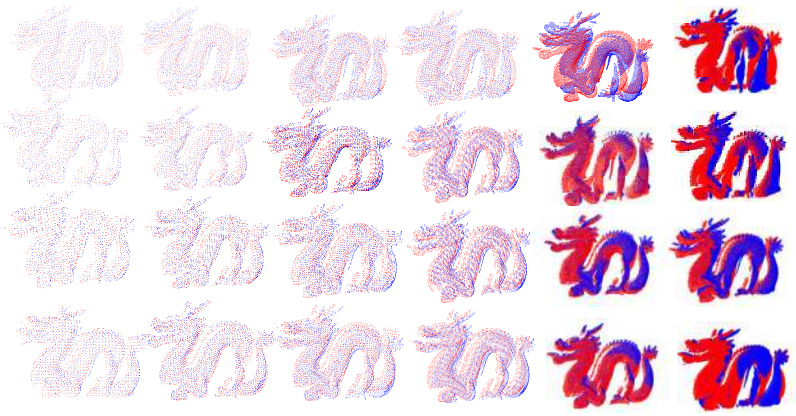
parameter with values of 0.005, 0.004, 0.003, 0.002, 0.001, and 0.0007. Column a has a drop sample value of 0.005, the first row shows the pre-registration status, and the second through fourth rows show the after-registration results. Each subsequent column is the same as column a, except that the subsampling parameters are different.

In Fig.10, from left to right, each column represents a subsample parameter with values of 0.005, 0.004, 0.003, 0.002, 0.001, and 0.0005. Column a has a drop sample value of 0.005, the first row shows the pre-registration status, and the second through fourth rows show the after-registration results. Each subsequent column is the same as column a, except that the subsampling parameters are different.

The test findings indicate that reducing sampling parameters in different ways can lead to improved registration results. It was observed that a downsampling value of 0.001 did not yield satisfactory registration outcomes. Consequently, the parameters were adjusted to further decrease the downsampling value. It was subsequently discovered that a downsampling value of 0.0005 provided more satisfactory registration results.

4.4 Ablation experiment

Table 4 shows the experimental errors of the three data sets under different modules, and 8 evaluation indicators are selected. The results show that the error of experimental results is the largest when no module is added. Error reduction when adding a module; When the two modules are combined, the error is minimum and the accuracy is highest.



(a) (b) (c) (d) (e) (f)

Figure 10: The registration experiment was carried out on the Stanford long dataset, and the experiment was carried out on the same two longs, and the variables were different downsampling parameters

5 Conclusions

In this paper, we introduce a new point cloud registration algorithm and propose an improved registration framework, MFIPC, specifically tailored for point cloud networks and integrated optimization techniques. We delve into the intricacies of their design, outlining each component in detail. The framework includes four stages: eigenvalue calculation, feature fusion calculation, interval proportion calculation and interval correspondence calculation, and the global calculation is realized by iteration.

Table 4: Ablation experiments on modules

Dataset	Baseline	DCM	Curvature	Evaluation metrics							
				R_err	T_err	RMSE	MSE	MAE	rmse	mse	mae
ModelNet40	✓			16.9423	0.1454	0.0691	0.0480	0.0552	**0.0727**	0.0022	0.2701
	✓	✓		10.6658	0.4855	0.0344	0.0107	0.0618	0.1642	0.0270	0.1032
	✓		✓	9.3193	0.1857	0.0096	0.0128	0.0347	0.1725	0.0298	0.2035
	✓	✓	✓	**4.5760**	**0.2578**	0.0446	**0.0020**	0.0279	0.1290	0.0166	0.1048
3DMatch	✓			56.4819	1.2681	0.3422	0.2429	0.2983	1.3886	1.9278	1.0175
	✓	✓		35.6067	1.1215	0.2780	**0.0099**	0.1979	0.8747	0.7652	0.5952
	✓		✓	30.7384	0.6946	0.1227	0.0553	0.0885	0.2389	0.0571	0.2035
	✓	✓	✓	5.9696	0.9407	0.4894	0.2966	0.3757	0.4704	0.2212	0.3705
RESSO	✓			66.4482	5.0300	**0.0031**	0.6199	**0.0026**	1.0341	1.0693	0.8094
	✓	✓		60.3010	0.6282	0.0108	0.2648	0.0091	0.1991	0.0396	0.1237
	✓		✓	62.7329	0.3755	0.0128	0.2966	**0.0083**	0.1961	0.0385	0.1436
	✓	✓	✓	9.8408	0.4418	0.0446	0.0472	0.0279	0.0820	**0.0067**	**0.0705**

Experimental evidence highlights the excellent accuracy, efficiency and robustness of the framework. It is worth noting that the error rate of the text method is significantly lower than that of similar point cloud registration methods, while maintaining similar classification performance. Our work provides a practical solution for point cloud registration that can be seamlessly integrated into a variety of deep learning platforms. Looking ahead, our future efforts will focus on expanding the range of downsampled values for each data set in the algorithm and improving the algorithm run time. At present, the algorithm has the problem of slow speed or excessive memory usage when processing small subsampled data. To solve this problem, we aim to improve the algorithm's efficiency, expand the range it can handle, and optimize the framework's performance on real-world environmental data.

References

[1] C. Yang, Y. Chen, C. Wang, and J. Li, Foreground-background segmentation of sequential point clouds, in: *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2022: 7503–7506.

[2] W. Yu and G. Zheng, 2D-3D regularized deformable B-spline registration: Application to the proximal femur, in: *Proc. IEEE Int. Symp. Biomed. Imaging (ISBI)*, 2015: 829–832.

[3] L. Ren, H. Chang, C. Liu, S. Chen, L. Zhao, T. Yang, W. Zhang, and L. Wang, A calibration algorithm of 3-D point cloud acquisition system based on KMPE cost function, *IEEE Trans. Instrum. Meas.*, 2024, 73: 1–11.

[4] J. Zhang, L. Chen, B. Ouyang, B. Liu, J. Zhu, Y. Chen, Y. Meng, and D. Wu, Pointcutmix: Regularization strategy for point cloud classification, *Neurocomputing*, 2022, 505: 58–67.

[5] Y. Huang, L. Zhang, Y. Bi, and Y. Liu, Point cloud classification based on local fusion and dense connection structure, in: *Proc. Int. Conf. Neural Netw., Inf. Commun. Eng. (NNICE)*, 2024: 546–549.

[6] H. Xiao, W. Kang, Y. Li, and H. Xu, Text-free controllable 3-D point cloud generation, *IEEE Trans. Instrum. Meas.*, 2024, 73: 1–12.

[7] Y. Ji, Y. Li, X. Sun, S. Yan, and N. Guo, Stereo matching algorithm based on binocular vision, in: *Proc. Int. Forum Electr. Eng. Autom. (IFEEA)*, 2020.

[8] K. Zhang, J. Lu, and G. Lafruit, Cross-based local stereo matching using orthogonal integral images, *IEEE Trans. Circuits Syst. Video Technol.*, 2009, 19(7): 1073–1079.

[9] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, Local stereo matching using geodesic support weights, in: *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2009: 2093–2096.

[10] Z.-F. Wang and Z.-G. Zheng, A region based stereo matching algorithm using cooperative optimization, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2008: 1–8.

[11] L. Cheng, S. Chen, X. Liu, H. Xu, Y. Wu, M. Li, and Y. Chen, Registration of laser scanning point clouds: A review, *Sensors*, 2018, 18(5): 1641.

[12] F. Tombari, S. Salti, and L. Di Stefano, Performance evaluation of 3D keypoint detectors, *Int. J. Comput. Vis.*, 2013, 102(1-3): 198–220.

[13] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, A comprehensive performance evaluation of 3D local feature descriptors, *Int. J. Comput. Vis.*, 2016, 116: 66–89.

[14] W. Chen, P. Li, and H. Zhao, MSL3D: 3D object detection from monocular, stereo and point cloud for autonomous driving, *Neurocomputing*, 2022, 494: 23–32.

[15] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, 1981, 24(6): 381–395.

[16] F. Pomerleau, F. Colas, and R. Siegwart, A review of point cloud registration algorithms for mobile robotics, *Found. Trends Robot.*, 2015, 4(1): 1–104.

[17] M. Magnusson, A. Lilienthal, and T. Duckett, Scan registration for autonomous mining vehicles using 3D-NDT, *J. Field Robot.*, 2007, 24(10): 803–827.

[18] V. Ilci and C. Toth, High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation, *Sensors*, 2020, 20(3): 899.

[19] J. Yang, Z. Huang, S. Quan, Z. Qi, and Y. Zhang, SAC-COT: Sample consensus by sampling compatibility triangles in graphs for 3-D point cloud registration, *IEEE Trans. Geosci. Remote Sens.*, 2021, 60: 1–15.

[20] H. Deng, T. Birdal, and S. Ilic, PPFNet: Global context aware local features for robust 3D point matching, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018: 195–205.

[21] D. Aiger, N. J. Mitra, and D. Cohen-Or, 4-points congruent sets for robust pairwise surface registration, in: *Proc. ACM SIGGRAPH*, 2008: 1–10.

[22] R. B. Rusu, N. Blodow, and M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009: 3212–3217.

[23] Y. Fu, Z. Li, F. Xiong, H. He, W. Wang, and Y. Deng, Pairwise coarse registration of point clouds by traversing voxel-based 2-plane bases, *Int. J. Remote Sens.*, 2022, 43(13): 5100–5123.

[24] D. Peng, Z. Gui, D. Wang, Y. Ma, Z. Huang, Y. Zhou, and H. Wu, Clustering by measuring local direction centrality for data with heterogeneous density and weak connectivity, *Nat. Commun.*, 2022, 13(1): 5455.

[25] J. Yang, K. Xian, P. Wang, and Y. Zhang, A performance evaluation of correspondence grouping methods for 3D rigid data matching, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, 43(6): 1859–1874.

[26] X. Wang, B. Xu, S. Zhao, and X. Li, Automatic registration of optical image and airborne LiDAR data based on centers and corner points of building boundaries, *Int. J. Remote Sens.*, 2022, 43(18): 6644–6668.

[27] J. Yu, C. Yu, and C. Lin, Improved iterative closest point (ICP) point cloud registration algorithm based on matching point pair quadratic filtering, in: *Proc. Int. Conf. Comput., Internet Things Control Eng. (CITCE)*, 2021: 1–5.

[28] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, Geometrically stable sampling for the ICP algorithm, in: *Proc. Int. Conf. 3-D Digit. Imaging Model. (3DIM)*, 2003: 260–267.

[29] P. Kamousi, S. Lazard, A. Maheshwari, and S. Wuhrer, Analysis of farthest point sampling for approximating geodesics in a graph, *Comput. Geom.*, 2016, 57: 1–7.

[30] S. Rusinkiewicz and M. Levoy, Efficient variants of the ICP algorithm, in: *Proc. Int. Conf. 3-D Digit. Imaging Model. (3DIM)*, 2001: 145–152.

[31] Y. Zhong, Intrinsic shape signatures: A shape descriptor for 3D object recognition, in: *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2009: 689–696.

[32] I. Stamos and P. K. Allen, Geometry and texture recovery of scenes of large scale, *Comput. Vis. Image Underst.*, 2002, 88(2): 94–118.

[33] B. Yang, Z. Dong, F. Liang, and Y. Liu, Automatic registration of large-scale urban scene point clouds based on semantic feature points, *ISPRS J. Photogramm. Remote Sens.*, 2016, 113: 43–58.

[34] Y. Xu, R. Boerner, W. Yao, L. Hoegner, and U. Stilla, Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets, *ISPRS J. Photogramm. Remote Sens.*, 2019, 151: 106–123.

[35] Z. Dong, F. Liang, B. Yang, Y. Xu, Y. Zang, J. Li, Y. Wang, W. Dai, H. Fan, and J. Hyyppä, Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark, *ISPRS J. Photogramm. Remote Sens.*, 2020, 163: 327–342.

[36] C. Ru, F. Wang, T. Li, B. Ren, and X. Yan, Outline viewpoint feature histogram: An improved point cloud descriptor for recognition and grasping of workpieces, *Rev. Sci. Instrum.*, 2021, 92(2): 025102.

[37] W. Dai and P. Chen, A point cloud registration network combined with graph convolution and FPFH for feature extraction, in: *Proc. China Autom. Congr. (CAC)*, 2022: 2280–2284.

[38] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, The quickhull algorithm for convex hulls, *ACM Trans. Math. Softw.*, 1996, 22(4): 469–483.

[39] B. Peng, The determinant: A means to calculate volume, *Am. Math. Mon.*, 2007, 21: 22.

[40] Z. Gui, D. Peng, H. Wu, and X. Long, MSGC: Multi-scale grid clustering by fusing analytical granularity and visual cognition for detecting hierarchical spatial patterns, *Future Gener. Comput. Syst.*, 2020, 112: 1038–1056.

[41] C.-I. Chang, Real-time recursive hyperspectral sample and band processing: Algorithm architecture and implementation, *Springer*, 2017.

[42] D. U. Pizzagalli, S. F. Gonzalez, and R. Krause, A trainable clustering algorithm based on shortest paths from density peaks, *Sci. Adv.*, 2019, 5(10): eaax3770.

[43] X. Lian and Y. Bai, 3D-SIFT point cloud registration method integrating curvature information, in: *Proc. Int. Conf. Commun. Technol. Inf. Technol. (ICCTIT)*, 2023: 181–186.

[44] G. Turk and M. Levoy, Zippered polygon meshes from range images, in: *Proc. ACM SIGGRAPH*, 1994: 311–318.

[45] B. Curless and M. Levoy, A volumetric method for building complex models from range images, in: *Proc. ACM SIGGRAPH*, 1996: 303–312.

[46] V. Krishnamurthy and M. Levoy, Fitting smooth surfaces to dense polygon meshes, in: *Proc. ACM SIGGRAPH*, 1996: 313–324.

[47] G. C. Sharp, S. W. Lee, and D. K. Wehe, ICP registration using invariant features, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, 24(1): 90–102.

[48] J. Yang, H. Li, D. Campbell, and Y. Jia, Go-ICP: A globally optimal solution to 3D ICP point-set registration, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015, 38(11): 2241–2254.

[49] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, R-PointHop: A green, accurate, and unsupervised point cloud registration method, *IEEE Trans. Image Process.*, 2022, 31: 2710–2725.

[50] Q.-Y. Zhou, J. Park, and V. Koltun, Fast global registration, in: *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016: 766–782.

[51] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, PointNetLK: Robust & efficient point cloud registration using PointNet, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019: 7163–7172.

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