

Cross-source point cloud registration network with low overlap based on multi-scale features and attention mechanisms

ZHONG Xingjian, WANG Peng*, LI Yue, LI Lin, FU Luhua, SUN Changku

State Key Laboratory of Precision Measurement Technology and Instruments, Tianjin University, Tianjin 300072, China

*Corresponding author: WANG Peng (wang_peng@tju.edu.cn)

Received: February 1, 2025

Revised: March 10, 2025

Accepted: May 1, 2025

Abstract: Machine vision-based detection methods have been widely applied in the detection of aircraft skin damage. During drone inspection processes, a key step is to spatially locate high-resolution detailed images of aircraft skin from multiple angles onto a three-dimensional point cloud model of the aircraft. This relies on the rigid registration of image center position coordinate point cloud with the aircraft 3D point cloud. To address the issues of low accuracy and poor robustness encountered by existing registration algorithms when dealing with heterogeneous point clouds with significant differences in density and low overlap, this paper presents a novel cross-source point cloud registration network. The network integrates multi-scale information from the point cloud and employs an attention mechanism to identify representative overlapping points. First, the network achieves initial correspondences using the multi-scale geometric features and positional information of the point cloud. Then, an overlapping feature guidance module predicts the overlapping score of the point cloud. By utilizing information interaction through the attention mechanism, the network combines point overlapping scores with fused features to filter out representative overlapping points, achieving precise correspondences in the point cloud. The network employs weighted singular value decomposition (SVD) to estimate two sets of transformation matrices, yielding the relative pose parameters of the point cloud. Experiments were conducted in an unsupervised manner. The experimental results on the ModelNet40 dataset and the aero object dataset aircraft measurement data showed that, compared to other existing traditional and learning-based methods, this approach demonstrated excellent performance in terms of registration accuracy and robustness.

Key words: visual inspection; point cloud registration; spatial localization; machine learning; cross-source registration; unsupervised learning

0 Introduction

Aircraft are widely utilized in national defense and military operations, as well as in transportation, where safety and stability are of paramount importance^[1]. Damage to aircraft skin is one of the critical factors affecting structural safety. The skin surface is subject to atmospheric erosion, bird strikes, lightning impacts, and the stresses experienced during flight, leading to defects such as cracks, dents, and corrosion. These issues can significantly reduce the overall lifespan of the aircraft, highlighting the importance of routine maintenance and quality inspection of the skin. However, traditional manual visual inspection methods are inefficient, time-consuming, and yield inconsistent results, failing to meet the modern aerospace industry's demands for precision and efficiency. Machine vision-based inspection methods offer enhanced accuracy and speed, while leveraging the advantages of drones in terms of efficiency^[2], safety, and path planning, making

them an ideal platform for skin inspection.

Rapid and accurate detection of aircraft skin is a critical focus of research. Computer vision^[3] methods based on neural networks demonstrate significant advantages in detection speed and accuracy. By utilizing drone platforms to scan high-definition detailed images of the aircraft skin and integrating two-dimensional pixel data with three-dimensional spatial localization algorithms and machine vision target detection algorithms, it is possible to achieve the localization, identification, and marking of skin defects. The three-dimensional spatial localization of detailed images is a key step in the inspection process. The coordinates of the center point of the detailed image, obtained in the camera coordinate system during the capture of panoramic images, serve as the source point cloud in the point cloud registration task, while the three-dimensional point cloud model of the aircraft under inspection acts as the target point cloud. Since the source point cloud is derived from the depth information of the sensors and the rotational

angles of the gimbal, while the target point cloud represents the true three-dimensional model of the aircraft, both belong to heterogeneous point clouds captured by different sensors and imaging mechanisms. Consequently, there are significant differences in resolution, scale, field of view, and overlap, rendering the registration problem more complex.

In the rigid registration of point clouds, traditional methods such as the iterative closest point (ICP) [4] algorithm are commonly used. This approach optimizes the transformation estimation by iteratively refining the matching pairs of closest points within the point cloud. However, due to its non-convex nature, the ICP method is highly dependent on initial values and is prone to converge to local optima. Various enhanced algorithms, such as PLICP [5], utilize point-line matching to update correspondences, while Go-ICP [6] employs branch-and-bound (BnB) algorithms to guide the process and avoid local minima. Nonetheless, these methods are significantly slower than ICP and demonstrate poor robustness in complex scenarios. Fast global registration (FGR) [7] is an efficient global point cloud registration method that leverages local fast point feature histograms (FPFH) for feature extraction and global optimization. However, FGR is sensitive to noise and outliers and performs inadequately with low-resolution point clouds. Traditional registration algorithms often encounter issues such as initial value dependency, local optima, noise sensitivity, and computational inefficiency when dealing with heterogeneous point clouds that exhibit substantial differences in density and low overlap.

Compared to traditional point cloud registration algorithms, deep learning methods integrate differentiable pose estimation into the learning pipeline, demonstrating significant advantages in registration accuracy, speed, and robustness. Existing deep learning approaches can be categorized into two main types: those based on global features and those based on point correspondences. Methods leveraging global features, such as PointNetLK [8] and PCRNet [9], adopt different frameworks. PointNetLK combines a global feature descriptor based on PointNet [10] with the Lucas & Kanade (LK) optimization algorithm [11], iteratively estimating the relative pose transformation. On the other hand, PCRNet integrates PointNet with a multilayer perceptron (MLP) [12] within a Siamese network structure, enhancing robustness against noisy points. However, these methods rely on global information for matching, and the absence of information in non-overlapping regions can lead to inaccurate transformations. Therefore, when dealing with partial-to-global point

clouds exhibiting low overlap, they may not capture sufficient feature information, resulting in challenges for the registration process.

Methods based on point correspondences establish correspondences through local features. DCP [13] employs a DGCNN network [14] to extract local features and compute soft correspondences. However, it struggles to mitigate negative impacts when matching point clouds of different resolutions. PRNet [15] predicts the correspondences of key points using Gumbel-Softmax [16], effectively addressing the partial-to-partial registration problem, but its high computational overhead makes it less suitable for practical applications. DBDNet [17] proposes a dual-branch structure to decouple rotation and translation, enhancing the accuracy of key point matching in overlapping regions. OMNet [18] combines overlap region prediction and adaptive feature fusion to reduce the mismatching of overlapping region points. Nevertheless, these methods tend to produce mismatches when dealing with low-overlap heterogeneous point clouds, thereby reducing registration accuracy. Overall, while methods based on point correspondences can partially address the registration issues between partial and global point clouds with low overlap, their susceptibility to mismatches between point clouds of different resolutions [19] and their high sensitivity to the structure of heterogeneous point clouds continue to impact registration accuracy and robustness.

In summary, this paper aims to achieve the registration of heterogeneous point clouds with significant sparsity-densification differences and low overlap using deep learning methods. The network integrates multi-scale spherical geometric features from point clouds, combining feature class information and positional information to compute the initial transformation matrix for the point clouds. Subsequently, self-attention and cross-attention mechanisms are employed to filter representative overlapping points in the pre-matching region, facilitating the calculation of an accurate transformation matrix and enabling correspondence-based registration. The method presented in this paper effectively addresses the accuracy issues associated with the registration of point clouds that exhibit large sparsity-densification differences and low overlap, maintaining robustness even in noisy environments and heterogeneous conditions.

1 Method

1.1 Acquisition of multi-angle detailed images and their representative coordinate points

The data acquisition system established for the

experiments in this paper includes a high-precision digital turntable, a laser rangefinder, and a zoom camera. The data acquisition system employs a distributed measurement and collection strategy, as illustrated in Fig.1.

This paper employs the method described in reference^[20] to obtain detailed images of the aircraft from multiple angles along with their representative coordinate points. The flow

of the acquisition method is illustrated in Fig.2.

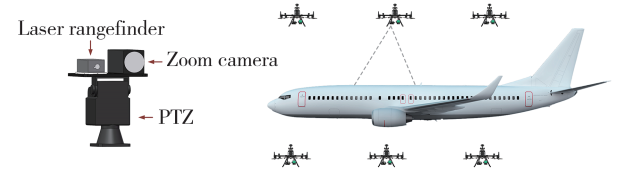


Fig. 1 Schematic diagram of data acquisition system and distributed measurement point locations

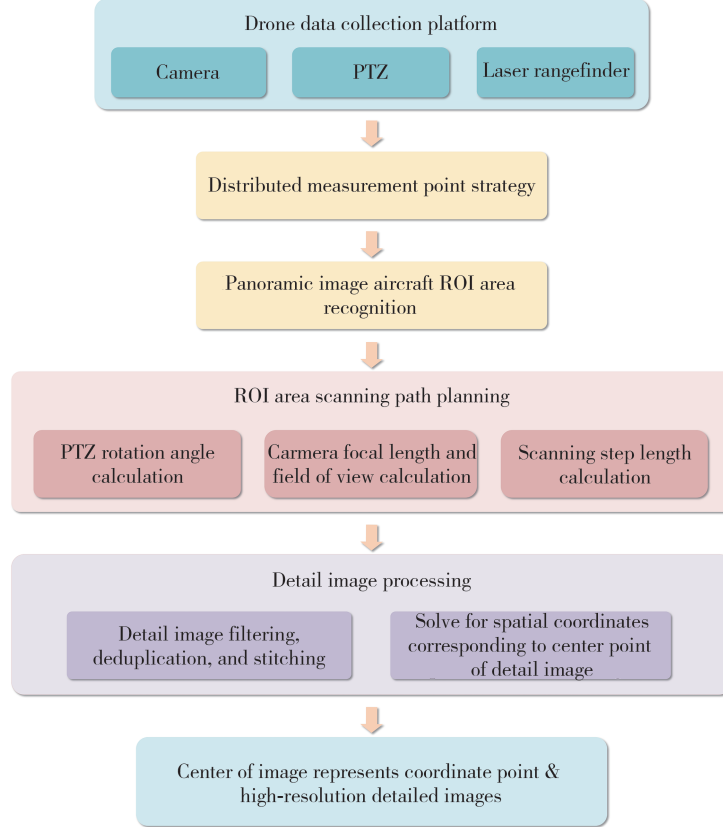


Fig. 2 Flowchart for acquisition of detailed images and their representative coordinate points

1.2 Three-dimensional spatial localization of detailed images

In the registration task, the source point cloud consists of the coordinates of the center point of the detailed image in the camera coordinate system when capturing the

panoramic image, while the target point cloud is the 3D point cloud model of the aircraft being tested. An overview of the network architecture is shown in Fig.3. The input point cloud consists of the red point cloud representing the 3D point cloud model of the aircraft, serving as the target point cloud.

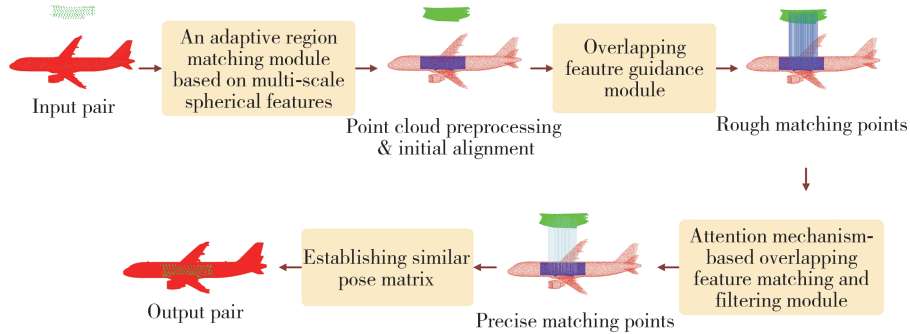


Fig. 3 Overview of network architecture

The green point cloud represents the spatial coordinates of the center point of the detailed image in

the camera coordinate system during the capture of the panoramic image, serving as the source point cloud for

the point cloud registration task. The purple point cloud indicates the initial corresponding region.

1.2.1 Problem statement

The problem of rigid point cloud registration from a partial point cloud to a complete point cloud can be stated as follows. Given two input point clouds X and Y , our objective is to determine a rigid transformation T , which includes parameters $R \in \text{SO}(3)$ and $t \in \mathbb{R}^3$, to best align the source point cloud X with the target point cloud Y in the least squares sense. We use the same notation to represent the sets of point clouds and their corresponding matrices $X \in \mathbb{R}^{N \times 3}$ and $Y \in \mathbb{R}^{M \times 3}$. We apply the true rigid transformation $\bar{T}_X^Y$ to determine the optimal alignment by minimizing the distance between the precise matching points of the transformed source point cloud and the precise matching points of the target point cloud as

$$\min_{R,t} \sum_{i=1}^K \left\| \text{NN}(\bar{T}_X^Y(x_i), Y) - y_i \right\|_2, \quad (1)$$

where K is the number of precise matching points, $\text{NN}(\bar{T}_X^Y(x_i), Y)$ denotes the nearest neighbor point of the transformed source point x_i in the target point cloud Y , x_i and y_i represent the matching points in the source point cloud and target point cloud, respectively.

1.2.2 Adaptive region matching module based on multi-scale spherical features

In order to achieve the initial correspondence between the source point cloud and the target point cloud, this paper proposes an adaptive region matching module based on multi-scale spherical features^[21], as illustrated in Fig. 4. Specifically, it involves extracting the multi-scale spherical geometric features of the point cloud, integrating class information and positional information, and then calculating the similarity of point cloud patch features to establish the initial correspondence of the point clouds.

Given the input point cloud X , we construct local patches at different scales. Specifically, we define the number of scales η and utilize the farthest point sampling to select central points, ensuring that the sampled points are uniformly distributed across the surface of the point cloud. The set of selected central points is denoted as $CT^\eta = \text{FPX}(X)$, where $\text{FPX}(\cdot)$ indicates that the farthest point sampling method is adopted for the point cloud, and S^η represents the number of patches at scale η . Given the central points, we generate the corresponding local patches foreach central point using the K -nearest neighbors (KNN) method. The representation of the patches at the η th scale is denoted as

$$P^\eta = \text{KNN}(K^\eta, X, CT^\eta), \quad (2)$$

where the feature dimension is $S^\eta \times K^\eta \times 3$, $\text{KNN}(\cdot, \cdot, \cdot)$ represents the K -nearest neighbors operation, and K^η denotes the size of the patches, which is the number of points contained in each patch. This size is adjusted based on the specific datasets to capture features at different spatial resolutions. In our experiments, we set 4 scales $\eta = 1, 2, 3, 4$, with patch sizes $K^\eta = 16, 32, 64, 128$ and the corresponding number of patches $S^\eta = 64, 32, 16, 8$, and the total number of patches is $S = \sum_{\eta=1}^4 S^\eta = 120$.

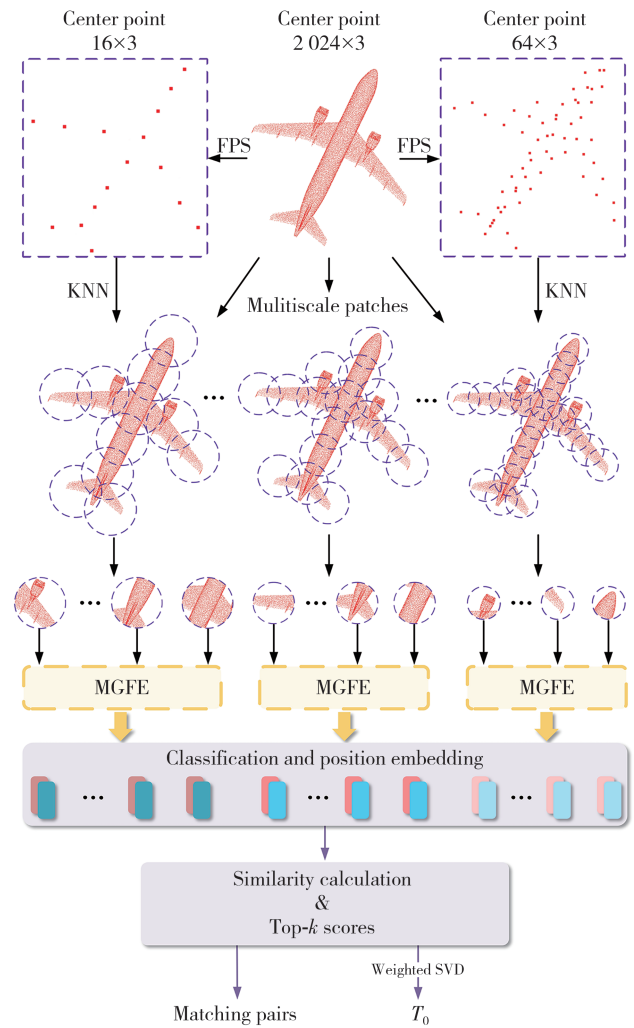


Fig. 4 Adaptive region matching module based on multi-scale spherical features

Due to the significant non-Euclidean geometric characteristics of point clouds, accurately capturing their geometric features becomes challenging. However, mapping point clouds to spherical space can address this issue. The improved shared multi-scale spherical geometric feature extractor (MGFE) is illustrated in Fig. 5. MGFE generates fixed-length local geometric feature representations by constructing local spherical mappings of

the point clouds and combining them with a multi-scale patch extraction mechanism.

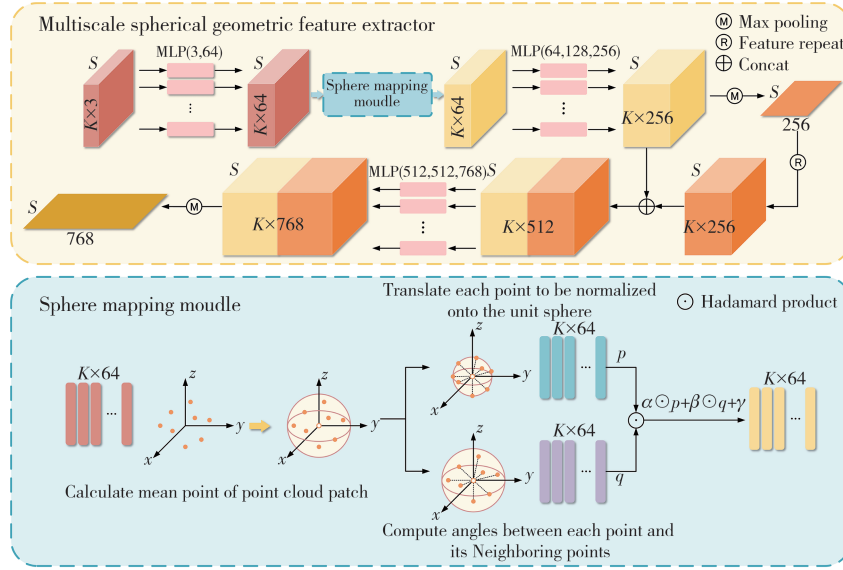


Fig. 5 Multi-scale spherical geometric feature extractor

We denote each neighboring point of $P_{i=1,\dots,S^\eta}$ as $\{P_{i,j}^\eta\}_{j=1,\dots,K^\eta}$, where the feature dimensions are $K^\eta \times d$. We set the neighborhood point vector $d=64$. The neighboring points are mapped onto the unit sphere as

$$\bar{P}_i^\eta = \frac{1}{K^\eta} \sum_{j=1}^{K^\eta} P_{i,j}^\eta, \quad (3)$$

$$\begin{aligned} \{P_{i,j}^\eta\} = & \alpha^\eta \odot \frac{P_{i,j}^\eta - \bar{P}_i^\eta}{|P_{i,j}^\eta - \bar{P}_i^\eta| + \delta} + \\ & \frac{\beta^\eta}{K^\eta} \odot \sum_{s=1}^{K^\eta} \frac{(P_{i,j}^\eta - \bar{P}_i^\eta)(P_{i,s}^\eta - \bar{P}_i^\eta)}{|P_{i,j}^\eta - \bar{P}_i^\eta| \cdot |P_{i,s}^\eta - \bar{P}_i^\eta| + \delta} + \gamma^\eta, \end{aligned} \quad (4)$$

where α^η , β^η , and γ^η are learnable parameters, $\odot$ denotes the Hadamard product, and $\delta = e^{-5}$ is a set stability coefficient to prevent instability in calculations caused by a zero denominator. The first part of Eq. (4) normalizes each point in the patch to the unit sphere, the second part computes the angular relationships between the points in the patch and the neighboring points, applying weighted normalization, and the third part, γ^η , represents a bias term used for translation.

The point cloud patch features $F_{p_i}^\eta$ outputted through spherical mapping have a feature dimension of $S^\eta \times K^\eta \times d$. After applying the max pooling operation, we obtain the multi-scale global features $F_{p_i}^\eta = \text{maxpool}(F_{p_i}^\eta)$ of the point cloud patch. $\text{Maxpool}(\cdot)$ represents the max pooling operation. After feature concatenation, a multilayer perceptron (MLP) processes the features to derive the multi-scale embedding local feature representation $F_{p_i}^\eta$ of the point cloud patch, where the feature dimension is $S^\eta \times K^\eta \times d_{\text{out}}$.

$$F_{p_i}^\eta = \text{MLP}(\text{Concat}[F_{p_i}^\eta, F_{p_i}^\eta]), \quad (5)$$

where d_{out} represents the output feature dimension, $\text{Concat}[\cdot, \cdot]$ represents the feature concatenation operation, and $\text{MLP}(\cdot)$ represents the multi-layer perceptron. Subsequently, $F_{p_i}^\eta$ undergoes another max pooling operation to yield the multi-scale embedding global feature representation $F_{p_i}^\eta = \text{Maxpool}(F_{p_i}^\eta)$ of the point cloud patch.

The η scales of point cloud patch features are fused into a unified multi-scale feature representation.

$$F_p = \text{Concat}[F_{p_i}^1, \dots, F_{p_i}^\eta], \quad (6)$$

where the feature dimension is $S \times d_{\text{out}}$. We initialize the class and position encoding of the point cloud patch embeddings randomly, expanding the patch embeddings to include both global category information and local positional relationships. First, we randomly initialize a learnable class label vector C_{cls} and represent the positional information of each point cloud patch using the centroid coordinates CT_i . After concatenating CT_i with C_{cls} , we map this combined information to the embedding dimension d_{out} using a learned MLP, resulting in the class and positional encoding.

$$F_{\text{pos},i} = \text{MLP}(\text{Concat}[CT_i, C_{\text{cls}}]), \quad (7)$$

where the feature dimension is $S \times 2d_{\text{out}}$. Based on the class and positional information of the embedding from the source point cloud and the target point cloud, we denote the total feature sequences of the point cloud patches as $F_{p_z,x}$ and $F_{p_z,y}$ for convenience, using F_x and F_y to represent them, respectively. We then compute the similarity matrix of the point cloud patch features

between the source and target point clouds to achieve a coarse matching between patches, thereby establishing the initial correspondence between the source and target point clouds. The calculation of the similarity $\text{Sim}(F_{Y,j}, F_{X,j})$ between the target point cloud patch features $F_{Y,j}$ and the source point cloud patch features $F_{X,j}$ is defined as

$$\text{Sim}(F_{Y,j}, F_{X,j}) = \frac{F_{Y,j} \cdot F_{X,j}}{\|F_{Y,j}\|_2 \|F_{X,j}\|_2}, \quad (8)$$

where $\text{Sim}(\cdot)$ denote the calculation of the cosine similarity, which is then normalized to construct the similarity matrix $S \in \mathbb{R}^{s_y \times s_x}$. From the score matrix, we select the top- k most descriptive matching patches to form the matching patch set K_{XY} . Utilizing the positional information of the patches, we compute the initial transformation matrix $[R_0, t_0]$, using a weighted singular value decomposition (SVD). SVD decomposes any matrix into the product of two orthogonal matrices and a diagonal matrix (the singular value matrix). In pose estimation, it involves decomposing the covariance matrix of two sets of point clouds to extract the orthogonal basis corresponding to the largest singular value, obtaining the optimal rotation matrix, and calculating the translation vector by combining centroid offsets, thereby minimizing registration error.

Based on the aforementioned content, we have established the initial correspondence between the source point cloud and the target point cloud, providing initial conditions for precise point cloud alignment for the prediction of overlapping points within the corresponding

regions of the point clouds.

1.2.3 Overlapping feature guidance module

We achieve precise point cloud alignment based on the prediction of overlapping points in the initial correspondence regions between the transformed source point cloud and the target point cloud. We design an overlapping point feature guidance module based on point cloud fusion features, as shown in Fig.6, to predict the overlap score of the points.

We utilize a modified PointNet network as the shared encoder of the overlapping point feature guidance module to obtain the local features $F_{CX'}$ and F_{CY} corresponding to the source and target point cloud regions within the patch set K_{XY} , with feature dimensions of $N \times d$ and $M \times d$. We consider the prediction of overlapping points as a binary classification problem, and the calculation of the overlap point score is expressed as

$$O_{X'} = g_\psi \text{softmax} \left(\text{Concat} \begin{bmatrix} F_{CX'}, R(F_{CX'}^g), R(F_{CY}^g) \\ R(F_{CX'}^g - F_{CY}^g) \end{bmatrix} \right), \quad (9)$$

$$O_Y = g_\psi \text{softmax} \left(\text{Concat} \begin{bmatrix} F_{CY}, R(F_{CY}^g), R(F_{CX'}^g) \\ R(F_{CY}^g - F_{CX'}^g) \end{bmatrix} \right), \quad (10)$$

where g_ψ is the shared overlap decoder, followed by a softmax function, $F_{CX'}^g$ and F_{CY}^g represent the global features obtained after the max pooling operation, $R(\cdot)$ denotes the expanded feature dimension and feature replication, while $O_{X'}$ and O_Y represent the point overlap scores for the source point cloud X' and the target point cloud Y .

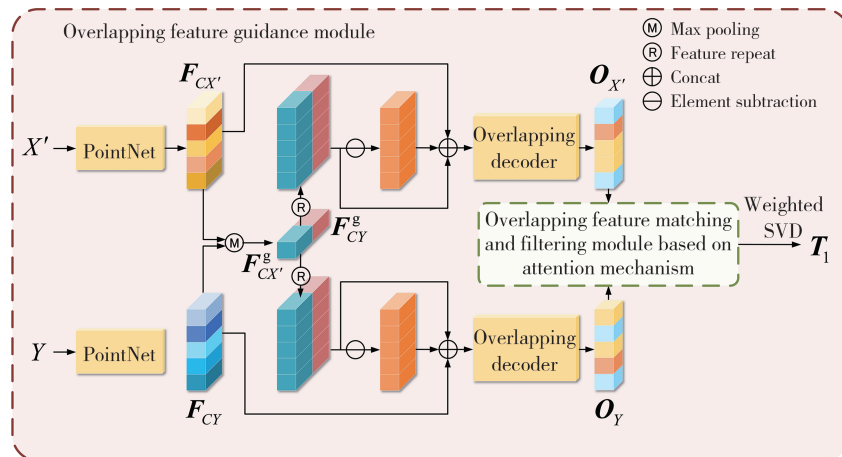


Fig. 6 Overlapping point feature guidance module

1.2.4 Attention mechanism-based overlapping feature matching filtering module

To filter out more representative overlapping point pairs and improve alignment accuracy, we design an attention mechanism-based overlapping feature

matching filtering module, as shown in Fig.7. The specific steps involve enhancing the point pair features, which undergoes information interaction through the attention mechanism, followed by filtering using a similarity matrix.

For each point $x \in X'$, the k nearest neighbors U_k are obtained within a predefined radius using the K -nearest neighbors' method. For each point $u_i \in U_k$, the augmented feature representation $F_{u_i}^a$ of point u_i is derived by linking the spatial coordinates and point pair features PPF_i .

$$F_{u_i}^a = \text{Concat}[u_i, u_i - x, \text{PPF}_i]. \quad (11)$$

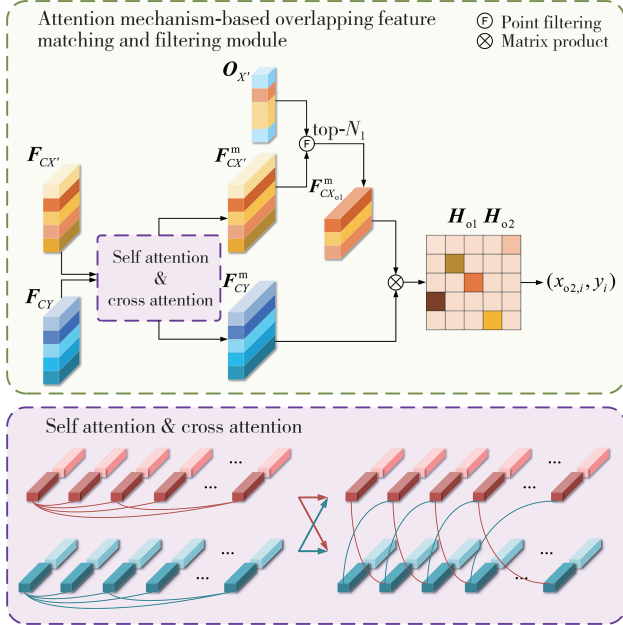


Fig. 7 Attention mechanism-based overlapping feature matching filtering module

We obtain the augmented feature representation $F_{C_x}^e$ of point x , with a feature dimension of d_p

$$F_{C_x}^e = \text{MaxPool}(\mu_\theta(F_{u_i}^a)), \quad (12)$$

where μ_θ consists of multiple one-dimensional convolutional layers. The local augmented features of the point cloud are fed into the attention layer.

The attention layer includes a self-attention layer and a cross-attention layer. The input to the self-attention layer comes from multi-scale features of the same point cloud, while the input to the cross-attention layer is derived from multi-scale features of two different point clouds. This results in the feature representations $F_{C_X'}^m$ and $F_{C_Y}^m$ after the information interaction based on the attention mechanism

$$F_{C_X'}^m = \varphi_{\text{cross}}(\varphi_{\text{self}}(F_{C_X'}, F_{C_X'}), \varphi_{\text{self}}(F_{C_Y}, F_{C_Y})), \quad (13)$$

$$F_{C_Y}^m = \varphi_{\text{cross}}(\varphi_{\text{self}}(F_{C_Y}, F_{C_Y}), \varphi_{\text{self}}(F_{C_X'}, F_{C_X'})), \quad (14)$$

where $\varphi(\cdot)$ is a transformer^[22] encoder layer, consisting of a multi-layer perceptron (MLP), a layer normalization (LN), and a multi-head attention operation (MA) that executes H attention functions in parallel. Each attention head generates a query vector Q , a key vector K , and a

value vector V .

$$\begin{cases} Q = FW^Q, \\ K = FW^K, \\ V = FW^V, \end{cases} \quad (15)$$

where W^Q , W^K , and W^V are learned projection matrices. Each H attention function computes the attention score matrix through a dot product, which is then multiplied by the value vector V to output the attention features.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{head}}}}\right)V, \quad (16)$$

where the number of attention heads is H , and $d_{\text{head}} = d/H$. The $\text{softmax}(\cdot)$ function represents the normalization operation. The multi-head attention operation can be expressed as

$$\text{head}_h = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \quad (17)$$

$$F^m = \text{Concat}[\text{head}_1, \text{head}_2, \dots, \text{head}_H]W^O, \quad (18)$$

where W^O is a learnable projection matrix. The self-attention layer enhances the feature representations of $F_{C_X'}$ and F_{C_Y} , while the cross-attention layer enables the interaction of feature information, resulting in the fused and enhanced point cloud features $F_{C_X'}^m$ and $F_{C_Y}^m$. The feature dimensions are $N \times 4d_p$ and $M \times 4d_p$, respectively.

We then perform representative overlapping point selection. First, based on the overlap score O_X , we select the top- N_1 points from the source point cloud X' as the initial overlapping points X_{o1} . Using the similarity matrix $H_{o1} = F_{C_{X_{o1}}}^m \cdot (F_{C_Y}^m)^T$ for feature matching, we then filter to obtain a more representative set of overlapping points.

$$X_{o2} = X_{o1} \left[\arg \max_{\text{top-prob}} \left(\max_j \left(F_{X_{o1}}^m \cdot (F_{C_Y}^m)^T \right)_{ij} \right) \right], \quad (19)$$

where top-prob refers to selecting points that achieve the highest similarity score in the probability distribution of the attention mechanism, and $\max_j$ denotes the features

$F_{y_j}^m$ of the target points that are the most similar to the points $x_{o1,i} \in X_{o1}$. Therefore, the similarity matrix $H_{o2} = F_{X_{o2}}^m \cdot (F_{C_Y}^m)^T$ for the representative overlapping point set X_{o2} is obtained. And for each point $x_{o2,i} \in X_{o2}$, we search for its corresponding target point $y_i \in Y$, following the matching rules.

$$J = \arg \max_{\text{top-}k} H_{o2,i}, \quad (20)$$

$$w_{ij} = \begin{cases} H_{o2,ij}, & j \in J, \\ 0, & j \notin J, \end{cases} \quad (21)$$

$$y_i = \frac{w_{i,i}}{\sum_{j \in J} w_{i,j}} \cdot Y, \quad (22)$$

where J represents the indices of the top- k points in point cloud Y that have the highest similarity scores for each point $x_{o2,i}$. Based on the matched point pairs $(x_{o2,i}, y_i)$ and their corresponding overlap weight scores $O_{x_{o2,i}}$, we utilize weighted SVD to compute the overlap-based transformation matrix $R_1 \in \text{SO}(3)$ and translation vector $t_1 \in \mathbb{R}^3$.

Based on the initial correspondences and the overlap-based transformation matrix, we obtain the final precise transformation matrix $R = R_1 R_0$ and translation vector $t = R_1 t_0 + t_1$.

1.2.5 Loss function

The prediction of overlapping fractions is approached as a binary classification task, where we consider utilizing the cross-entropy loss function to supervise the training process. We define the overlapping loss function L_{ol} as

$$L_{ol} = \frac{1}{2N} \sum_i \sum_j (\hat{O}_X)_{ij} \cdot \log(O_X)_{ij} + \frac{1}{2N} \sum_i \sum_j (\hat{O}_Y)_{ij} \cdot \log(O_Y)_{ij}, \quad (23)$$

$$\Phi_X = \left\{ X_i \mid NN(\hat{R} \cdot X_i + \hat{t}, Y) < d, \forall i \right\}, \quad (24)$$

where $\hat{O}_X$ and $\hat{O}_Y$ represent the true overlapping scores of point clouds X and Y , which are obtained from the mapping Φ_X based on Eq. (19). $NN(X, Y)$ denotes the nearest neighbor of point X found within Y , where mapping values below the threshold d are set to 1, while those above this threshold are set to 0.

To enhance training efficiency, we calculate the differences between the model's predicted values (R and t) and their corresponding true values ($\hat{R}$ and $\hat{t}$) as an auxiliary loss function.

$$L_{\Pi} = \|X \cdot R^T - X \cdot \hat{R}^T\|_1 + \|t - \hat{t}\|_1. \quad (25)$$

The total loss function of the network is expressed as

$$L = L_{ol} + \lambda_t L_{\Pi}. \quad (26)$$

2 Experiments

The training of the network was implemented in PyTorch. All experiments were conducted on an Intel i9-13900K CPU and an NVIDIA GeForce RTX 3090 Ti GPU, with the CPU operating at 3 GHz. For the ModelNet40 dataset, the model was trained for 800 epochs using the Adam optimizer^[23], with an initial learning rate of 1×10^{-4} . The learning rate was adjusted using a cosine

annealing schedule, and the batch size was set to 32, with the network iterating twice over the dataset. Unless explicitly stated otherwise in the experimental studies, we input 1 024 (1 K) points.

2.1 Dataset

2.1.1 ModelNet40 dataset

The ModelNet40^[24] dataset is widely used for point cloud registration, containing 40 categories with a total of 12 311 mesh CAD models, including 9 843 training models and 2 468 testing models. For each data model, 1 024 points were uniformly sampled from the surface of the mesh, and the coordinates were normalized for network input. During the training process, three rotation angles were randomly generated within the range of $[-45^\circ, 45^\circ]$, and translations were applied along the three axes within the range of $[-0.5, 0.5]$.

2.1.2 Aircraft dataset

Due to the current lack of large-scale optical measurement labeled databases, the network proposed in this paper employed an unsupervised training method. A large-scale point cloud dataset named the aero object dataset has been created based on the CAD models of the aircraft to be tested. Initially, aircraft point cloud models are randomly cropped with a clipping ratio of 0.5 to generate a subset of point clouds. From this subset, 2 048 points were randomly sampled and combined to form the final dataset. The resulting synthetic training set contained 30 000 point cloud samples, while the testing set includes 8 000 point cloud samples, both of which were utilized for practical training. The data processing methods were consistent with those used for the ModelNet40 dataset.

2.2 Benchmarks and evaluation indicators

To evaluate the performance of point cloud registration, we adopt the isotropic rotation and translation errors proposed in RPMNet^[25].

$$\text{Error}(R) = \arccos\left(\frac{\text{tr}(\hat{R}^{-1}R) - 1}{2}\right), \quad (27)$$

$$\text{Error}(t) = \|\hat{R}^{-1}t - \hat{t}\|_1, \quad (28)$$

where $\hat{R}$ and $\hat{t}$ represent the ground truth and predicted transformations, respectively, while $\text{tr}(\cdot)$ denotes the trace of a matrix. Additionally, we evaluate the root mean square errors for rotation and translation, denoted as $\text{RMSE}(R)$ and $\text{RMSE}(t)$, as well as the mean absolute errors, $\text{MAE}(R)$ and $\text{MAE}(t)$, as utilized in OMNet.

2.3 Contrast experiment

To validate the effectiveness of the proposed method, we conducted comparisons with mainstream traditional registration algorithms (ICP, FGR) as well as deep learning-based methods (DCP, PCRNet, DBDNet, OMNet). The implementations of ICP and FGR were sourced from Open3D^[26], while we utilized the code provided by the authors for the other algorithms. The performance metrics on the complete dataset, without any additional processing, are presented in Table 1 for the ModelNet40 dataset.

The experimental results presented in Table 1 indicate that the traditional algorithm ICP performs poorly due to significant initial differences in point clouds. In contrast, FGR outperforms certain learning-based methods, as it utilizes additional data. Compared to mainstream learning-based approaches, our method achieves optimal rotation and translation estimates, demonstrating leading precision metrics on the ModelNet40 dataset.

Table 1 Results on ModelNet40 for unseen categories

Method	Error(<i>R</i>)	Error(<i>t</i>)	RMSE(<i>R</i>)	RMSE(<i>t</i>)	MAE(<i>R</i>)	MAE(<i>t</i>)
ICP	35.901 2	0.283 8	28.251 5	0.255 4	19.495 6	0.208 6
FGR	14.885 2	0.165 5	10.571 3	0.100 9	7.712 7	0.077 3
DCP	17.431 6	0.178 5	12.490 6	0.119 3	8.770 6	0.082 8
PCRNet	18.521 0	0.199 0	14.591 1	0.147 3	9.367 5	0.096 5
DBDNet	4.203 2	0.080 3	5.381 1	0.081 8	2.027 1	0.037 5
OMNet	1.266 5	0.015 9	2.455 8	0.021 0	1.002 1	0.008 2
Ours	1.200 1	0.009 1	2.156 0	0.014 8	0.593 3	0.004 2

We also assessed the robustness of the model in the presence of noise by adding Gaussian noise with a mean of 0 and a variance of 0.02. The results are presented in Table 2. The metrics indicate that the traditional algorithm ICP is highly susceptible to noise, while the error for FGR increases dramatically, demonstrating its sensitivity to noise. In contrast, the learning-based methods exhibit less sensitivity to noise, maintaining a certain level of robustness. Our method, along with OMNet, continues to demonstrate high accuracy and superior performance, thereby confirming the strong robustness of our network.

Table 2 Results on ModelNet40 for unseen categories with added noise

Method	Error(<i>R</i>)	Error(<i>t</i>)	RMSE(<i>R</i>)	RMSE(<i>t</i>)	MAE(<i>R</i>)	MAE(<i>t</i>)
ICP	40.201 2	0.355 0	30.235 5	0.300 2	21.127 4	0.254 3
FGR	28.315 8	0.276 6	26.197 5	0.251 0	16.375 3	0.179 9
DCP	20.431 6	0.204 8	13.490 6	0.134 2	9.152 8	0.092 7
PCRNet	21.521 0	0.231 5	15.591 1	0.164 9	9.876 1	0.115 6
DBDNet	5.203 2	0.105 9	5.881 1	0.093 4	2.534 6	0.048 2
OMNet	1.866 5	0.029 5	2.755 8	0.032 7	1.267 9	0.011 8
Ours	1.541 0	0.012 3	2.279 3	0.015 8	0.774 0	0.005 7

To evaluate the model's performance on the registration task of partially overlapping point clouds to complete point clouds, we further processed the ModelNet40 dataset by cropping the source point clouds to a ratio of 0.3, while the target point clouds remained untrimmed, with both sets augmented by Gaussian noise. The performance of various methods on the ModelNet40 dataset is presented in Table 3. The experimental results indicate that traditional algorithms do not perform well in handling non-overlapping regions in the partial-to-whole registration task. While the overall accuracy of learning-based methods also declines, our method consistently achieves optimal performance across all metrics.

Table 3 Results on ModelNet40 for unseen categories with 0.30 cropping and added noise

Method	Error(<i>R</i>)	Error(<i>t</i>)	RMSE(<i>R</i>)	RMSE(<i>t</i>)	MAE(<i>R</i>)	MAE(<i>t</i>)
ICP	55.117 5	0.364 2	45.105 8	0.317 8	36.621 0	0.310 0
FGR	40.275 0	0.282 5	38.045 3	0.256 8	29.488 7	0.215 6
DCP	25.600 3	0.251 7	17.765 9	0.161 0	13.764 5	0.144 0
PCRNet	24.653 7	0.248 8	19.845 0	0.181 5	13.989 0	0.152 2
DBDNet	8.612 5	0.122 0	9.563 7	0.101 2	5.733 6	0.055 9
OMNet	3.607 4	0.045 7	4.522 5	0.049 6	2.323 1	0.016 7
Ours	2.972 8	0.027 7	3.635 3	0.035 1	1.529 7	0.013 1

We also conducted a comparison of the inference efficiency of the model, as shown in Table 4. The analysis of inference time indicates that our model demonstrates high efficiency in the point cloud registration task, with inference times significantly lower than traditional algorithms, although slightly inferior to some learning-based algorithms. However, our model achieves a good balance between efficiency and registration accuracy, maintaining a reasonable inference speed while delivering optimal performance.

Table 4 Inference time results on ModelNet40

Method	Time/s
ICP	0.485 5
FGR	0.733 6
DCP	0.016 8
PCRNet	0.009 1
DBDNet	0.012 0
OMNet	0.008 2
Ours	0.014 7

To further validate the robustness and generalization ability of our model in low-overlap point cloud scenarios, we designed and conducted experiments under low overlap conditions (0.05–0.30). The experiments cover two datasets: ModelNet40 and the aero object dataset. We cropped the source point clouds at ratios of 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30, while the target point clouds were not cropped and noise was added. The results on

ModelNet40 are shown in Table 5, and the results on the aero object dataset are shown in Table 6.

Table 5 Results on ModelNet40 for unseen categories with different cropping ratios

Ratio	Error(R)	Error(t)	RMSE(R)	RMSE(t)	MAE(R)	MAE(t)
0.05	7.507 7	0.075 4	10.811 2	0.089 8	3.930 7	0.036 6
0.10	5.149 2	0.052 6	6.985 7	0.063 9	2.676 4	0.025 2
0.15	3.914 1	0.038 5	5.094 5	0.046 0	2.035 6	0.018 4
0.20	3.646 6	0.037 2	4.828 5	0.048 2	1.889 4	0.017 4
0.25	3.342 7	0.031 1	4.234 3	0.038 7	1.710 2	0.014 8
0.30	2.972 8	0.027 7	3.635 3	0.035 1	1.529 7	0.013 1

Table 6 Results on aero object dataset with different cropping ratios

Ratio	Error(R)	Error(t)	RMSE(R)	RMSE(t)	MAE(R)	MAE(t)
0.05	5.980 3	0.081 1	11.440 0	0.135 7	3.276 9	0.038 2
0.10	3.412 9	0.036 5	4.898 3	0.052 4	1.761 9	0.016 7
0.15	2.196 8	0.024 9	1.769 1	0.018 3	1.155 8	0.011 9
0.20	1.215 3	0.014 0	2.593 9	0.028 0	0.652 3	0.006 8
0.25	0.966 2	0.009 8	1.340 2	0.011 0	0.516 6	0.004 7
0.30	0.670 2	0.007 3	0.599 7	0.006 0	0.366 8	0.003 6

The experimental results for low-overlap point cloud scenarios indicate that our model demonstrates strong robustness and generalization ability when the overlap ratio ranges from 0.10 to 0.30. Despite the differing complexities of the ModelNet40 and aero object dataset, our model maintains strong performance. Notably, the

model's performance significantly declines at extremely low overlap rates (such as 0.05), accompanied by considerable fluctuations in error. This suggests that registration tasks under extreme low-overlap conditions, where point cloud information is severely lacking, remain challenging.

We conducted a visualization of the algorithm's performance. The registration performance of our model on the ModelNet40 dataset for fully overlapping and partially overlapping cases with an overlap ratio of 0.30 is shown in Fig.8. The performance of all algorithms on the ModelNet40 dataset at an overlap ratio of 0.30 is illustrated in Fig.9.

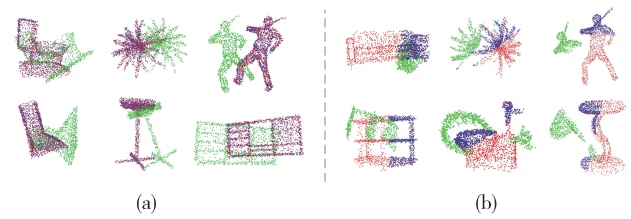


Fig. 8 Visualization of registration on ModelNet40 dataset with full overlap and an overlap ratio of 0.30. (a) Full overlap; (b) Overlap ratio of 0.30. Red, green, and blue represent target point cloud, source point cloud, and transformed source point cloud, respectively

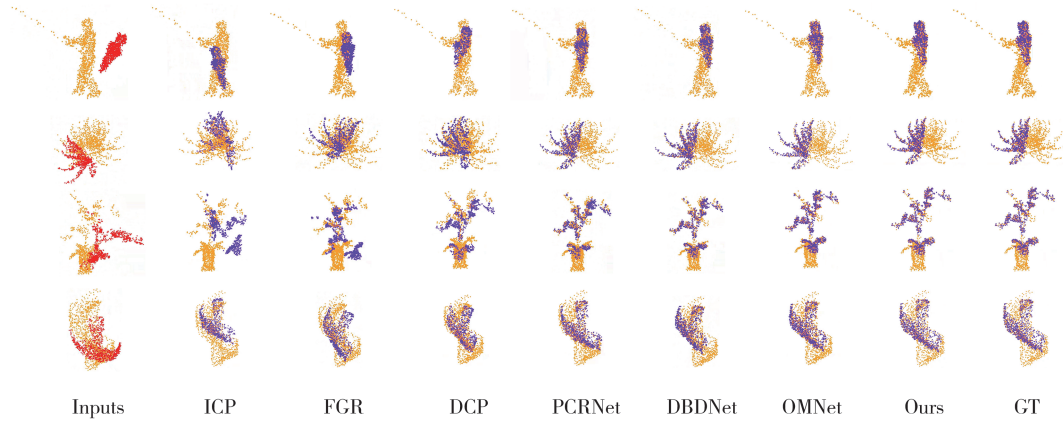


Fig. 9 Visualization of registration on ModelNet40 dataset with an overlap ratio of 0.30. Red, yellow, and purple colors represent source point cloud, target point cloud, and transformed source point cloud, respectively

Additionally, we evaluated the performance of different algorithms on the aero object dataset, where we again cropped the source point clouds to a ratio of 0.30. The results are presented in Table 7. It can be seen that all methods show an improvement in accuracy; however, our method maintains the highest accuracy and robustness.

Representative center coordinates of detailed images were enriched through an interpolation method based on Delaunay triangulation. The actual training registration effects with the three-dimensional point cloud model of

the aircraft are demonstrated in Fig.10.

Table 7 Results on aero object dataset for cropped portion with a ratio of 0.30 to overall data

Method	Error(R)	Error(t)	RMSE(R)	RMSE(t)	MAE(R)	MAE(t)
ICP	13.326 4	0.151 3	11.435 0	0.140 4	10.100 9	0.125 5
FGR	8.036 6	0.108 5	7.512 3	0.084 5	6.628 8	0.069 1
DCP	8.745 9	0.114 9	7.980 4	0.083 8	5.570 2	0.059 0
PCRNet	5.489 1	0.051 3	4.880 2	0.042 7	3.978 5	0.037 1
DBDNet	3.273 0	0.034 9	2.943 1	0.030 3	2.362 4	0.023 9
OMNet	0.835 8	0.099 7	0.740 5	0.008 2	0.535 1	0.006 5
Ours	0.670 2	0.007 3	0.599 7	0.006 0	0.366 8	0.003 6

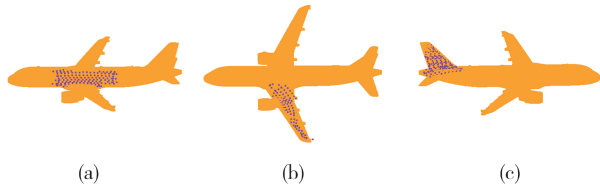


Fig. 10 Schematic diagram of registration effect between representative center coordinate points of detailed images and three-dimensional point cloud model of aircraft. (a) Middle of left fuselage; (b) Left wing; (c) Right tail wing. Purple and yellow colors represent representative center coordinate points of detailed images and three-dimensional point cloud model of aircraft, respectively

2.4 Ablation experiment

To demonstrate the effectiveness of the network modules, we conducted a series of ablation experiments on the ModelNet40 dataset with an overlap ratio of 0.30. This section focuses on investigating the roles of the multi-scale spherical feature extractor (MGFE), the overlapping feature matching filtering module (FMF), and the fused attention (SCA) layer. The results are presented in Table 8. The removal of any individual module leads to a degradation in network performance, highlighting the importance of each component in the overall architecture.

Table 8 Ablation studies of different components

MGFE	FMF	SCA	Error(R)	Error(t)
	✓	✓	6.259 3	0.102
✓		✓	4.663 6	0.059
✓	✓		4.797 8	0.062
✓	✓	✓	2.972 8	0.027 7

3 Conclusions

In the task of spatial localization registration from detailed images to the three-dimensional point cloud model of an aircraft, this paper presents a cross-source point cloud registration network that integrates multi-scale information and employs an attention mechanism to discern representative overlapping points. This network is capable of addressing the challenges posed by sparsity variations and low overlap conditions in partial-to-whole point cloud registration. An initial correspondence of the point clouds was established through an adaptive region matching module based on multi-scale spherical features. Furthermore, an overlapping feature guidance module was utilized to predict the overlap scores of the point clouds, while the overlapping feature matching filtering module selectively identified representative overlapping points, achieving precise correspondence between the point clouds.

Experimental results demonstrated that our method exhibited strong robustness and high accuracy in the presence of noise interference and significant sparsity variations during partial-to-whole point cloud registration, thus validating the effectiveness of the proposed approach.

Acknowledgement

I would like to express my gratitude to the reviewers and editors.

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

- [1] YANDOUZI M, GAYDOS S, GUO D, et al. Aircraft skin restoration and evaluation. *Journal of Thermal Spray Technology*, 2014, 23 (8): 1281-1290.
- [2] LI D R, LI M. Research advance and application prospect of unmanned aerial vehicle remote sensing system. *Geomatics and Information Science of Wuhan University*, 2014, 39 (5): 505-513.
- [3] LU R S, WU A, ZHANG T D, et al. Review on automated optical (visual) inspection and its applications in defect detection. *Acta Optica Sinica*, 2018, 38 (8): 0815002.
- [4] BESL P J, MCKAY N D. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell*, 1992, 14: 239-256.
- [5] CENSI A. An ICP variant using a point-to-line metric//2008 IEEE International Conference on Robotics and Automation, May 19-23, 2008, Pasadena, CA, USA. New York: IEEE, 2008: 19-25.
- [6] YANG J L, LI H D, CAMPBELL D, et al. Go-ICP: a globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38 (11): 2241-2254.
- [7] ZHOU Q Y, PARK J, KOLTUN V. Fast global registration//Computer Vision-ECCV 2016. Cham: Springer International Publishing, 2016: 766-782.
- [8] AOKI Y, GOFORTH H, SRIVATSAN R A, et al. PointNetLK: robust & efficient point cloud registration using PointNet//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2020: 7156-7165.
- [9] SARODE V, LI X Q, GOFORTH H, et al. PCRNNet: point cloud registration network using PointNet encoding. 2019: arXiv: 1908. 07906.
- [10] CHARLES R Q, HAO S, MO K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 77-85.

- [11] LUCAS BD, KANADE T. An iterative image registration technique with an application to stereo vision. Morgan Kaufmann Publishers Inc, 1981: 674-679.
- [12] CHOE J, PARK C, RAMEAU F, et al. PointMixer: MLP-mixer for Point cloud understanding. Computer Vision-ECCV 2022. Cham: Springer, 2022: 620-640.
- [13] WANG Y, SOLOMON J. Deep closest point: learning representations for point cloud registration//2019 IEEE/CVF International Conference on Computer Vision, October 27-November 2, 2019. Seoul, Korea. New York: IEEE, 2019: 3522-3531.
- [14] WANG Y, SUN Y B, LIU Z W, et al. Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics, 2019, 38(5): 1-12.
- [15] WANG Y, SOLOMON J. Prnet: Self-supervised learning for partial-to-partial registration//33rd Conference on Neural Information Processing Systems (NeurIPS), December 8-14, 2019, Vancouver, CANADA. La Jolla: NIPS, 2019.
- [16] JANG E, GU S S, POOLE B. Categorical reparameterization with gumbel-softmax. ArXiv, 2016, abs/1611.01144.
- [17] MA J Y, PENG C L, TIAN X, et al. DBDnet: a deep boosting strategy for image denoising. IEEE Transactions on Multimedia, 2022, 24: 3157-3168.
- [18] XU H, LIU S C, WANG G F, et al. OMNet: learning overlapping mask for partial-to-partial point cloud registration//2021 IEEE/CVF International Conference on Computer Vision, October 10-17, 2021, Montreal, QC, Canada. New York: IEEE, 2022: 3112-3121.
- [19] FU L H, SUN Y J, SUN C K, et al. Pose measurement without marked points based on prediction of overlapping area with hybrid features. Journal of Measurement Science and Instrumentation, 2023, 14(3): 253-262.
- [20] WANG P, ZHAO M H, SUN C K, et al. Research on image scanning and positioning system of aircraft skin. Chinese Journal of Sensors and Actuators, 2023, 36(11): 1706-1713.
- [21] WEI X, WANG M Y, LIN S J, et al. Multi-scale geometry-aware transformer for 3D point cloud classification. 2023: arXiv: 2304.05694.
- [22] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need//31st Annual Conference on Neural Information Processing Systems (NIPS), December 4-9, 2017, Long Beach, CA. La Jolla: NIPS, 2017: 6000-6010.
- [23] KINGMA DP, BA J. Adam: A method for stochastic optimization. Computing research repository, 2014, abs/1412.6980.
- [24] WU Z R, SONG S R, KHOSLA A, et al. 3D ShapeNets: a deep representation for volumetric shapes//2015 IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, Boston, MA, USA. New York: IEEE, 2015: 1912-1920.
- [25] YEW Z J, LEE G H. RPM-net: robust point matching using learned features//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 13-19, 2020. Seattle, WA, USA. New York: IEEE, 2020: 11821-11830.
- [26] ZHOU Q Y, PARK J, KOLTUN V. Open3d: A modern library for 3d data processing. ArXiv, 2018, abs/1801.09847.

基于多尺度特征和注意力机制的低重叠度跨源点云配准网络

钟行建, 王 鹏*, 李 岳, 李 林, 付鲁华, 孙长库

天津大学 精密测试技术及仪器全国重点实验室, 天津 300072

摘 要: 基于机器视觉的检测方法被广泛应用于飞机蒙皮损伤检测中。无人机巡检过程中, 一个关键步骤是将多角度飞机蒙皮高清细节图像数据进行飞机三维点云模型上的空间定位。这依赖于图像中心坐标点与点云的刚性配准。针对现有配准算法在处理疏密差异大且低重叠度的部分到整体的异源点云时存在精度低、鲁棒性差的问题, 本文提出了一种新型的跨源点云配准网络。该网络融合点云多尺度信息, 通过注意力机制判别代表性重叠点。首先, 网络通过点云多尺度几何特征和位置信息实现初始对应。然后, 通过重叠特征引导模块预测点云重叠分数。利用注意力机制信息交互, 结合点重叠分数和融合特征匹配筛选出代表性的重叠点, 实现点云精确对应。网络使用加权SVD估计两组变换矩阵, 得到了点云的相对位姿参数。实验采用无监督方式引导学习, 在ModelNet40数据集和Aero Object Dataset飞机测量数据集的实验结果表明, 与其他方法相比, 该方法在配准精度和鲁棒性方面具有优异的性能。

关键词: 视觉检测; 点云配准; 空间定位; 机器学习; 跨源配准; 无监督学习

引用格式: ZHONG Xingjian, WANG Peng, LI Yue, et al. Cross-source point cloud registration network with low overlap based on multi-scale features and attention mechanisms. Journal of Measurement Science and Instrumentation, 2026, 17(2): 183-194. DOI: 10.62756/jmsi.1674-8042.2026016