# DMS: Low-overlap Registration of 3D Point Clouds with Double-layer Multi-scale Star-graph

Hualong Cao, Yongcai Wang, *Member, IEEE,* and Deying Li

**Abstract**—Registering 3D point clouds with low overlap is challenging in 3D computer vision, primarily due to difficulties in identifying small overlap regions and removing correspondence outliers. We observe that the neighborhood similarity can be utilized to detect point correspondence, and the consistent neighborhood correspondence can be used as a criterion to detect robust overlapping regions. So that a Double-layer Multi-scale Star-graph (DMS) structure is proposed to detect robust correspondences using two different types of multi-scale star-graphs. The first-layer *Multi-scale Neighbor Feature Star-graphs* (MNFS) takes each point as the center and its multi-scale nearest neighbors as the leaves. The MNFS enables to establish the initial correspondence candidate set between the two point clouds based on multi-scale neighborhood topology and feature similarity. Subsequently, each pair of corresponding points find their nearest neighbors within the correspondence sets to construct a Multi-scale Matching Star-graphs (MMS) on each side, so the mutual correspondence relationships between the MMS vertices are identified. These identified mutual correspondences are treated as vertices to construct the *Multi-scale Correspondence Star-graphs (MCS)* , that indicate the relationships among the correspondences. We design edge weight and vertex weight criterion in MCS to detect only the robust correspondence set that has strong neighborhood consistency, so as to reject the outliers. Finally, the point cloud registration is conducted based on the detected robust correspondence. The experimental results demonstrate clearly that the proposed DMS method exhibits superior robustness when compared to existing state-of-the-art registration algorithms. The code of this study will be available at https://github.com/HualongCao/DMS.

**Index Terms**—point cloud registration, low overlap, Double-layer Multi-scale Star-graph, Multi-scale Neighbor Feature Star-graphs, Multi-scale Correspondence Star-graphs, Multi-scale Matching Star-graphs, superior robustness

✦

## 1 INTRODUCTION

3D Point clouds registration (PCR) is to accurately register two point clouds by estimating an accurate pose transformation, which is a fundamental task in various fields [1], [2], [3] such as computer graphics [4], [5], [6], [7], computer vision [8], [9], [10], [11], and visualization [12], [13], [14], [15]. Current PCR algorithms can be categorized into traditional methods [16], [17] and learning-based methods [18], [19], [20], [21]. For traditional methods, the iterative closest point (ICP) algorithm [16] and its variants [22], [23], [24], [25] are undoubtedly the most well-known methods for PCR. Another traditional approach is based on random sample consensus (RANSAC) [26] and its variants [27], [28], [29], [30]. However, in recent years, learning-based methods [31], [32], [33], [34] have gained increasing attention in PCR especially with deep learning approaches [35], [36], [37], [38], [39].

While PCR has made significant advancements, the problem of Low overlapped Registration (LoR) of 3D point clouds remains

● *Hualong Cao, Yongcai Wang and Deying Li are with the School of Information Renmin University of China, Intelligent Network and Optimization Laboratory, Renmin University of China, Beijing 100872 , China.*
*E-mail: caohualong@ruc.edu.cn; ycw@ruc.edu.cn; deyingli@ruc.edu.cn.*
●
● *Corresponding author: Yongcai Wang*

challenging, and the low overlap regime is highly relevant in practical applications. This is because that ensuring a high degree of overlap is often challenging in many practical scenarios, e.g., when traversing narrow and long tunnels, or in situations of obstruction or limited field of view. Meanwhile, data acquisition often involves high costs, leading practitioners to aim for minimal scanning and capturing to obtain only the necessary overlaps.

For its importance, LoR has recently garnered significant research attentions. The first well-known work is PREDATOR [35], which leveraged an attention mechanism to effectively identify salient points within overlapping regions. Building upon this foundation, subsequent research endeavors have been undertaken. CoFiNet [36] introduced a coarse-to-fine method for hierarchical correspondence extraction, which proposed super-point. This process initially matches the super-points at a coarse scale, followed by the extension to patches at finer scales to enhance point correspondence refinement. Furthermore, GeoTransformer [37] contributed to the field by focusing on the acquisition of point cloud geometric features during the initial rough matching stage, building upon the work of CoFiNet [36]. Recently, PEAL [38] introduced an explicit attention learning model incorporating prior embeddings, inspired by the insights gleaned from GeoTransformer [37], aiming to enhance the robustness of the super-point matching within the LoR framework. From these methods, we can observe that superpoint matching has proven to be very effective in solving LoR.

While LoR has been the subject of extensive research, there remain certain issues that current methods fail to address. We delineate three pivotal **challenges** within the domain of LoR, which motivate the design of DMS.
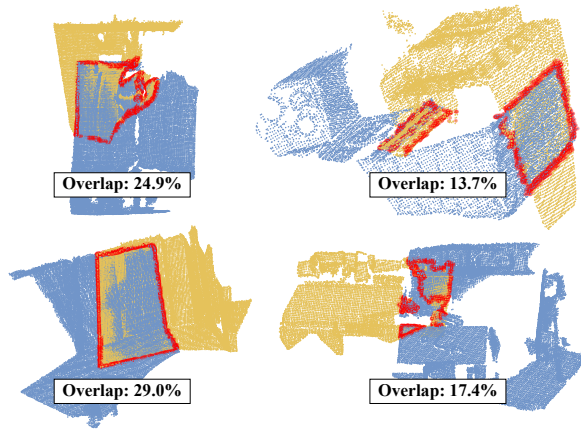
Fig. 1. Illustration of Peripheral Points at Overlap Boundaries in PCR, the red highlighted points represent peripheral points, which located at the overlap boundaries between point clouds, are key for accurate alignment, especially in low-overlap scenarios.

1) **Limited Usage of the Overlapped Peripheral Points**: In PCR, "peripheral points" refer to those points that lie at the edges or boundaries of the overlapping regions between two point clouds, as shown in Fig. 1. These points offer critical geometric information that is not available for points in the central overlap zone. This information includes clues about the shape and extent of the objects or scenes represented by the point clouds, which are invaluable for aligning the clouds more accurately, especially in scenarios of low overlap. However, many existing registration methods [35], [36], [37], [38], [40] primarily rely on the internal points within the overlapping regions, with limited consideration for the peripheral points. Nevertheless, as the degree of overlap decreases, the proportion of the peripheral points increases. Underscoring them loses useful information in LoR scenarios. Moreover, these peripheral points within the overlapping areas harbor valuable insights regarding the orientation and position of the point clouds.

2) **Ambiguity Issue in Correspondences**: Current methods [35], [36], [37], [38] primarily rely on the feature consistency for registration. Nonetheless, as the degree of overlapping decreases, capturing critical features in the low-overlap regions becomes challenging, and the detrimental influence of the features in the non-overlapping regions amplifies. Consequently, multiple candidate correspondences exhibit similarities, introducing ambiguity and diminishing the availability of meaningful correspondences.

3) **Outliers and Noise Issue**: Low-overlap regions exhibit heightened vulnerability to the adverse effects of outliers and noise. Existing algorithms [35], [36], [37], [41] struggle to distinguish outliers from valid correspondences. This inability to effectively differentiate between erroneous and meaningful associations can exert a pronounced impact on the overall accuracy of the registration, potentially leading to misalignments and suboptimal results.

To overcome above challenges, the crucial problem is how to discover and best utilize the overlapped points, and how to eliminate the correspondence ambiguity and the outliers. For this purpose, this study propose a novel *Double-layer Multi-scale Star-graph (DMS)* strategy to efficiently obtain robust correspondences in low-overlapping point clouds.

The DMS was designed primarily because of the star-graph's unique topology and properties. Compared to other graph structures like fully connected graphs, ring graphs, or arbitrary topology graphs, the star-graph offers a balance between computational efficiency and the ability to effectively capture and process local contextual information, ideal for LoR. In star-graphs, a central node connects with all others, enabling efficient information aggregation and rapid context understanding. This structure is particularly adept at highlighting local features and geometries relative to a central point, thereby enhancing model performance. Additionally, star-graphs fit well into multi-scale frameworks, allowing for hierarchical processing and global insight generation through the combination of multiple star-graphs.

Given the super-point matching has proven to be very effective in solving LoR [36], [37], [38], DMS is a multi-scale star graphs based on the super-points, which can quickly screen out a small number of possible matching candidates, reducing the computational complexity of fine matching. To enhance the usage of the peripheral super-points, we design multi-scale star graphs, so that the peripheral points show similarity in the small scale graphs and will not impact the similarity of inner points on other scales.

Based on this idea, we firstly construct the first-layer *Multi-scale Neighbor Feature Star-graphs (MNFS)* on both the source point cloud $\mathcal{P}$ and the target point cloud $\mathcal{Q}$. In the MNFS, each super-point is the center of a star graph and the leaf nodes are constructed based on the multi-scale nearest neighbors. The point features are integrated into the star graph, so that similarity between star graphs are calculated by the feature-based star graphs at the same scale, resulting in the establishment of the initial correspondence. However, the initial correspondences contain many ambiguities and outliers. To refine and extract the correct correspondences, we construct a second-layer *Multi-scale Correspondence Star-graphs (MCS)* based on the initial correspondences. To construct the MCS, we firstly construct a multi-scale matching star (MMS) graph for each corresponding point by identifying the nearest neighbors in the corresponding set. Then the mutual correspondences that exist simultaneously between these MMS vertices are found. So that the multi-scale MCS is constructed using these mutual correspondences as vertices, which indicate the relationship among the correspondence. We design edge weights and vertex weights to evaluate the neighborhood consistency, i.e., the neighborhood consistency among the neighboring correspondences. As a result, the incorrect correspondences are filtered out, leaving a substantial number of accurate correspondences for the subsequent point registration. The main **contributions** of our work are summarized below.

- **DMS:** DMS handles the registration problem of low-overlapping point clouds by constructing a two-level star graph. The first layer is MNFS, which selects similar points based on multi-scale neighborhood topology and feature similarity to establish initial correspondence. The second layer is MCS, which is used to evaluate the detected correspondences to reserve only robust correspondences with strong neighborhood consistency. DMS provides a novel approach to improve the accuracy and robustness of low-overlap point cloud registration.
- **MNFS:** In order to enhance the neighborhood and fea-

ture consistency of super-points, this study designed the MNFS method, which improves the accuracy of the overall registration process while maintaining the consistency of information between super-points.

- **MCS:** In the MCS stage, we construct the MCS by developing MMS for each point pair and identify mutual correspondences between the MMS graphs. Then the mutual correspondences are treated as vertices to generate MCS graphs, which are used to identify correspondences that meet neighborhood consistency criteria. This process ultimately filters and refines the correspondences, leading to an accurate and efficient point registration within the MCS framework.

The remainder of this article is organized as follows: Section 2 reviews the related work, providing a background and situating our study within the existing body of research. Section 3 provides a detailed description of the DMS method, including its theoretical foundations and implementation details. Section 4 evaluates the DMS method, including a comparison with state-of-the-art methods, an ablation study, and a visual experiment. Finally, Section 5 concludes the paper with a summary of our findings and discusses potential avenues for future research.

## 2 RELATED WORK

### 2.1 Traditional PCR Methods

For traditional PCR methods, there are well-known methods such as ICP [16], RANSAC [26] and their derivatives. ICP [16] is one of the most classic PCR methods. It iteratively finds the optimal rigid body transformation between two point clouds to minimize the distance between corresponding points. However, the ICP algorithm is highly sensitive to the accuracy of the initial alignment, and the number of iterations needs to be adjusted based on the specific scenario. As a result, subsequent advancements have been made to improve ICP-based PCR methods. For instance, GICP [22] introduces a more expressive probability model and integrates it into the ICP framework to filter out incorrect correspondences. GoICP [23] incorporates ICP into the BnB scheme to ensure global optimization while accelerating the registration process.

RANSAC [26] is an iterative algorithm that samples correspondences and generates geometric estimations until a satisfactory solution is achieved. However, evaluating these estimations can be time-consuming and sensitive to noise. To address these challenges, researcher have proposed various improvements. For example, SAC-IA [42] samples correspondences across the point cloud and uses the Huber penalty for evaluation. GC-RANSAC [27] applies the graph-cut algorithm during local optimization, while CG-SAC [28] incorporates normal information. SAC-COT [43] ranks and samples ternary loops from the compatibility graph. Despite these efforts, these methods still face challenges in terms of time efficiency and accuracy when dealing with high outlier rates.

Apart from the mentioned traditional methods, there exist other approaches. EMTR-SSC [17] enhances PCR with evolutionary multi-task optimization, solution space cutting, and introduces M-estimator Chamfer Distance for precision and overlap handling. Recently, MAC [40] constructs a compatibility graph using the correspondence information and obtains a robust correspondence relationship by searching for a maximal clique within the graph, thereby achieving optimal registration.

### 2.2 Learning-Based LoR Methods

Deep learning-based algorithms have gained widespread popularity in the field of LoR. These algorithms leverage the power of neural networks to learn feature representations and registration models for point clouds. They encompass various approaches, including extracting high-dimensional feature vectors using encoder networks and end-to-end deep learning registration networks. These techniques have garnered significant interest among researchers working in the field.

In the field of LoR, several notable methods based on deep learning have been developed. FCGF [44] utilizes a fully convolutional neural network to compute features for point clouds without the need for keypoint detection. D3Feat [45] employs a fully convolutional network to extract local information from point clouds and incorporates a joint learning framework for detecting and describing 3D local features. Predator [35] integrates an attention mechanism to extract salient points in overlapping regions, enabling robust registration even with low overlap rates. Spinnet [41] focuses on extracting rotationally invariant and informative local features to achieve precise registration.

Additionally, other methods focus on effectively distinguishing inliers and outliers in correspondences. Deep global registration (DGR) [46] and 3DRegNet [47] utilize end-to-end neural networks with operators such as sparse convolution and point-by-point MLP to classify correspondences. PointDSC [48] explores spatial consistency to remove outlier correspondences, while Fu et al. [49] proposed a deep graph matching framework (RGM) for robust and accurate point-to-point correspondences. Recent advancements include detection-free methods, which estimate transformations in an end-to-end manner. CoFiNet [36] extracts correspondences without the need for keypoint detection, and GeoTransformer [37] learns geometric features for robust super-point matching, even in scenarios with low overlap, while remaining invariant to rigid transformations. PEAL [38] is a prior-embedded explicit attention learning model that addresses feature ambiguity in LoR by incorporating prior knowledge and explicitly learning one-way attention with assumed overlap points.

## 3 METHOD

### 3.1 Problem Explanation and Method Process

The purpose of LoR is to align two or more point cloud data sets so that their corresponding points are as close or coincident as possible after being transformed into the same coordinate system. It is necessary to find a transformation matrix or transformation function to map the source point cloud $\mathcal{P} = \{\boldsymbol{p}_m\}_{m=1}^{M}$ to the coordinate system of the target point cloud $\mathcal{Q} = \{\boldsymbol{q}_n\}_{n=1}^{N}$, where $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ represent the location information of points in point cloud $\mathcal{P}$ and $\mathcal{Q}$, respectively, and both of which are 3-dimensional (3D). Given a correspondence set $\mathcal{C} = (\mathcal{P}^C, \mathcal{Q}^C)$ which indicates $|\mathcal{C}|$ pairs of one-to-one point correspondence, we denote $\mathbf{T}(\mathcal{C})$ the optimal transformation that can best register $\mathcal{P}^C$ and $\mathcal{Q}^C$, which is a function of $\mathcal{C}$ and can be carried out by ICP [16] etc..

$$\mathbf{T}(\mathcal{C}) = \arg \min_{\mathbf{T}} \sum_{\boldsymbol{p}_i \in \mathcal{P}^C, \boldsymbol{q}_i \in \mathcal{Q}^C} \|\mathbf{T} \cdot \boldsymbol{p}_i - \boldsymbol{q}_i\|_2^2 \qquad (1)$$

Our goal is to find the optimal correspondence $\mathcal{C}^\star$, so that the obtained transformation $\mathbf{T}(\mathcal{C}^\star)$ is as close as the ground truth $\mathbf{T}^G$ of the two point clouds $\mathcal{P}$ and $\mathcal{Q}$.

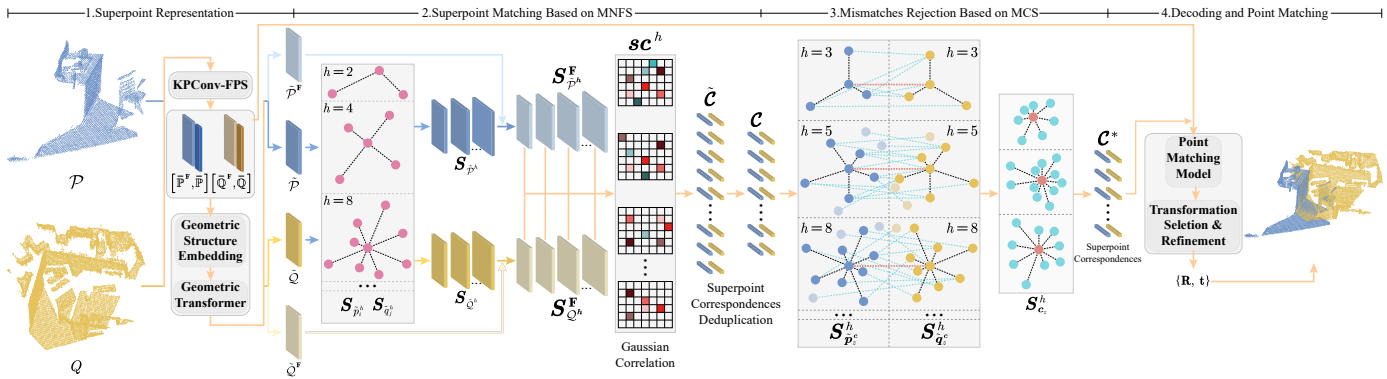$$\mathcal{C}^\star = \arg \min_{\mathcal{C}} \|\mathbf{T}^G \cdot \mathbf{T}^{-1}(\mathcal{C})\|_2^2 \qquad (2)$$

Fig. 2. The overall pipeline of our method. KPConv downsamples the input cloud and uses GeoTransformer for super-point feature extraction. MNFS utilizes the consistency of spatial structure and features to extract high-quality initial super-point correspondences between $\tilde{\mathcal{P}}$ and $\tilde{\mathcal{Q}}$. MCS solves correspondence relationships ambiguity and mismatching through neighborhood consistency. The super-point correspondences are then propagated to dense points $\bar{\mathbb{P}}$ and $\bar{\mathbb{Q}}$ through the point matching module, and finally, the transformation is calculated.

Note that in LoR, $\mathcal{P}^C$ and $\mathcal{Q}^C$ generally take very small proportions in $\mathcal{P}$ and $\mathcal{Q}$. To detect $\mathcal{P}^C$ and $\mathcal{Q}^C$ correctly, this work adopts a coarse-to-fine method for hierarchical correspondence extraction. We initially utilize KPConv-FPN to downsample the input point cloud and generate super-points. The super-point features are obtained based on GeoTransformer [37] (Section 3.2). We build MNFS based on these super-points. In constructing the MNFS, we integrate the feature information of super-points into each node, and then perform super-point matching to obtain the initial correspondence between the super-points (Section 3.3). Then, the MCS is constructed based on the established correspondence, and the neighborhood consistency is used as a screening criterion to exclude those star diagrams with poor consistency in the neighborhood, thereby retaining more reliable and robust correspondences (Section 3.4). At last, the points on each pair of super-point correspondences are registered, and then the optimal transformation matrix is recovered (Section 3.5). The pipeline is illustrated in Fig. 2.

To facilitate a better understanding of the symbolic expressions and analyses presented in this paper, Table 1 (Notation List) enumerates the primary mathematical symbols and variables employed, accompanied by brief definitions or descriptions. To avoid redundancy, similar symbols are not individually listed.

TABLE 1
Notation List

| | | | |
|---|---|---|---|
| $\mathcal{P}$ | Input source point cloud | $M$ | Number of points in $\mathcal{P}$ |
| $\tilde{\mathcal{P}}$ | Super points in $\mathcal{P}$ | $\tilde{M}$ | Number of super-points in $\tilde{\mathcal{P}}$ |
| $\tilde{\mathcal{P}}^{\mathbf{F}}$ | Features of $\tilde{\mathcal{P}}$ | $\bar{\mathbb{P}}$ | Downsampled point in $\mathcal{P}$ |
| $\bar{\mathbb{P}}^{\mathbf{F}}$ | Features of $\bar{\mathbb{P}}$ | $\bar{M}$ | Number of in $\bar{\mathbb{P}}$ |
| $\tilde{p}_i$ | $i$th super-point in $\tilde{\mathcal{P}}$ | $\boldsymbol{S}_{\tilde{p}_i^h}$ | $h$-star graph centered at $\tilde{p}_i$ |
| $M_1$ | Scales of MNFS | $M_2$ | Scales of MCS |
| $\boldsymbol{S}_{\tilde{p}_i^h}^{\mathbf{F}}$ | feature vector of $\boldsymbol{S}_{\tilde{p}_i^h}$ | $\tilde{sc}_{i,j}^h$ | Similarity of $\boldsymbol{S}_{\tilde{p}_i^h}^{\mathbf{F}}$ and $\boldsymbol{S}_{\tilde{q}_j^h}^{\mathbf{F}}$ |
| $\tilde{\mathcal{C}}$ | Initial correspondences set | $\mathcal{C}$ | $\tilde{\mathcal{C}}$ after deduplication |
| $\boldsymbol{c}_z$ | A correspond in $\mathcal{C}$ | $\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}$ | Points in $\mathcal{P}$ within $\boldsymbol{c}_z$ |
| $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ | $h$-star graph centered at $\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}$ | $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ | $h$-star graph centered at $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ |
| $\boldsymbol{c}_{\tilde{z}_l}$ | Leaf node of $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ | $\tilde{\boldsymbol{p}}_{z_l}^{\boldsymbol{c}}$ | Points in $\mathcal{P}$ within $\boldsymbol{c}_{\tilde{z}_l}$ |
| $\boldsymbol{M}_{\boldsymbol{c}_z}$ | Set of $\boldsymbol{c}_{\tilde{z}_l}$ | $N_{zh}$ | Number of leafs in $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ |
| $w_{zl}$ | Weight of the edge of $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ | $\mathcal{W}_{\boldsymbol{c}_z^h}$ | Weights of $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ |

## 3.2 Superpoint Representation and Feature Extraction

Following GeoTransformer [37], the KPConv-FPS backbone [50], [51] is first adopted to downsample the input point cloud into multi-level super-points and learn the corresponding features. In KPConv-FPS backbone, the original point cloud $\mathcal{P}$ and $\mathcal{Q}$ are aggregated into downsampled points $\bar{\mathbb{P}} \in \mathbb{R}^{\bar{M} \times 3}$ and $\bar{\mathbb{Q}} \in \mathbb{R}^{\bar{N} \times 3}$ and their features $\bar{\mathbb{P}}^{\mathbf{F}} \in \mathbb{R}^{\bar{M} \times \bar{d}}$ and $\bar{\mathbb{Q}}^{\mathbf{F}} \in \mathbb{R}^{\bar{N} \times \bar{d}}$ from the first level are also extracted, which for utilization in point matching after DMS. And then the downsampled points $\bar{\mathbb{P}}$ and $\bar{\mathbb{Q}}$ and their features $\bar{\mathbb{P}}^{\mathbf{F}}$ and $\bar{\mathbb{Q}}^{\mathbf{F}}$ also through a series of ResNet-like block and GeoTransformer, and then the super-points $\tilde{\mathcal{P}} \in \mathbb{R}^{\tilde{M} \times 3}$ and $\tilde{\mathcal{Q}} \in \mathbb{R}^{\tilde{N} \times 3}$ and their features $\tilde{\mathcal{P}}^{\mathbf{F}} \in \mathbb{R}^{\tilde{M} \times d}$ and $\tilde{\mathcal{Q}}^{\mathbf{F}} \in \mathbb{R}^{\tilde{N} \times d}$ are extracted separately.

## 3.3 Superpoint Matching Based on MNFS

The motivation behind MNFS is to significantly enhance point cloud registration by integrating multi-scale neighbor feature similarity and star graph structures, addressing the inherent limitations of traditional super-point representations and improving both accuracy and efficiency in capturing complex spatial relationships. And the current super-point representation process exhibits poor compatibility with surrounding neighbor information, with the neighbor information being of utmost value for registration. Moreover, overlapping peripheral super-points are often designated as outliers by existing methods. To address these challenges, this study introduce the MNFS, specifically designed to enhance both neighbor and feature consistency among super points.

### 3.3.1 Multi-scale Star Graph Construction

The motivation for utilizing multi-scale star graph stems from the imperative to intricately capture complex spatial relationships and feature dynamics across scales. This step aims to elevate matching precision by enhancing discriminatory power, diminishing the impact of noise and outliers, and optimizing the engagement of peripheral points in point cloud registration tasks.

Star graph can aggregate the information of internal points (ie. central point) and leaf nodes (ie. neighborhood points), which has been proven to be important in vision. We construct a $h$-star graph $\boldsymbol{S}_{\tilde{\mathcal{P}}^h} = \{\boldsymbol{S}_{\tilde{p}_i^h}\}_{i=1}^{\tilde{M}}$ for each node $\tilde{p}_i$, and for each node $\tilde{q}_j$ as well. We construct each $h$-star by picking the $h$ nearest neighbors, so a $h$-star $\boldsymbol{S}_{\tilde{p}_i^h}$ can be defined as $\boldsymbol{S}_{\tilde{p}_i^h} = \{V_{\tilde{p}_i^h}, E_{\tilde{p}_i^h}\}$,

where $V_{\tilde{p}_i^h} = \{\tilde{p}_i, \tilde{p}_i^1, \cdots, \tilde{p}_i^h\}$ are $h$ nearest neighbors of $\tilde{p}_i$ and $E_{\tilde{p}_i^h} = \{(\tilde{p}_i, \tilde{p}_i^1), \cdots, (\tilde{p}_i, \tilde{p}_i^h)\}$. So $\boldsymbol{S}_{\tilde{p}_i^h}$ is a graph with one internal node and $h$ leaves. The same process applies to $\tilde{q}_j$. The star graph construction process of MNFS is shown in the Fig. 3. However, it is evident that using a fixed $h$ value
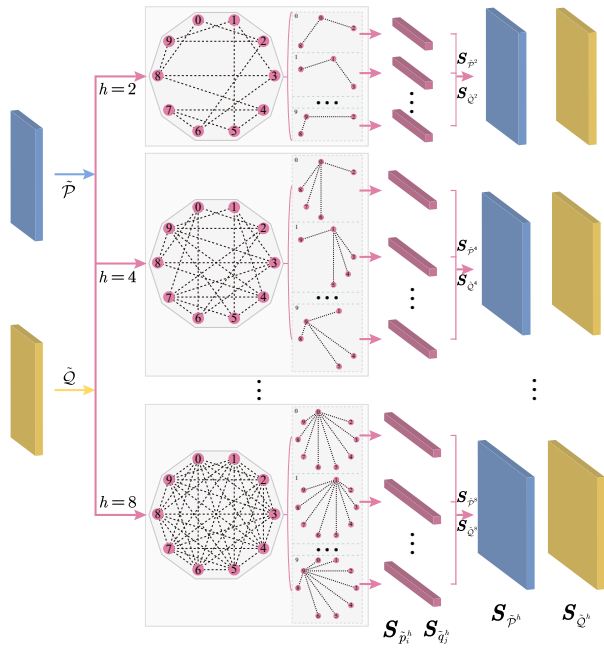


Fig. 3. Illustration of the MNFS star graph construction process, demonstrating the systematic aggregation of neighborhood points into multi-scale star graphs with varying levels of $h$ values.

for selecting edges in the graph could lead to several issues: 1) The fixed-scale star graph fails to capture varying structures and patterns at different scales, resulting in less comprehensive and accurate analysis outcomes; 2) The fixed-scale star graph does not adapt well to varying density distributions, leading to suboptimal performance with non-uniform data. To address the mentioned issues, a multi-scale strategy is employed, so that the peripheral points show similarity in the small scale graphs and will not impact the similarity of inner points on other scales. Specifically, we take the parameter $h$ on multiple values, i.e., $h = \{h_m\}_{m=1}^{M_1}$, where $M_1$ stands for the number of scales for the MNFS graph. So for each $\tilde{p}_i$, $M_1$ star graphs $\boldsymbol{S}_{\tilde{p}_i^h}, h = h_1, \cdots, h_{M_1}$ with different sizes are constructed for it.

By analyzing features at multi-scales, can differentiate between features that may appear similar at a single scale but diverge at another. For instance, some features might be indistinct at a lower scale but become highly distinctive at a higher scale, and vice versa. This multi-scale analysis provides a richer feature representation, enhancing the discrimination power of our matching algorithm. It allows us to identify correspondences with greater precision by considering how feature relevance changes across scales. In addition, sensitivity to noise and outliers can also be reduced by utilizing multi-scale information, as features that might be ambiguous or misleading at one scale can be clarified or corroborated at another.

### 3.3.2 Fusing Features on the $\boldsymbol{h}$-stars

The motivation for integrating features on the $\boldsymbol{h}$-stars is to augment the star graph with a detailed layer of contextual information

from various scales, crucial for improving superpoint matching accuracy. This approach combines the characteristics of the central superpoint with those of its neighboring points, creating enriched, multi-scale representations. Such representations are instrumental in achieving precise matching by capturing the nuanced variations in feature relationships at different scales.

For each $\boldsymbol{S}_{\tilde{p}_i^h}$, it is evident that it comprises one central point $\tilde{p}_i$ and $h$ neighboring points, solely containing spatial location information. To further enrich the information in the star graph, the feature information $\tilde{\mathcal{P}}^{\mathbf{F}}$ and $\tilde{\mathcal{Q}}^{\mathbf{F}}$ are merged into the star graph. By integrating features from both the central superpoint and its neighbors, the methodology encapsulates a harmonious blend of local and global contextual information, pivotal for the accuracy of superpoint matching. Fig. 4 provides a graphical explanation of feature fusion. Take $\boldsymbol{S}_{\tilde{p}_i^h}$ as an example, $\boldsymbol{S}_{\tilde{q}_i^h}$ is the same. We
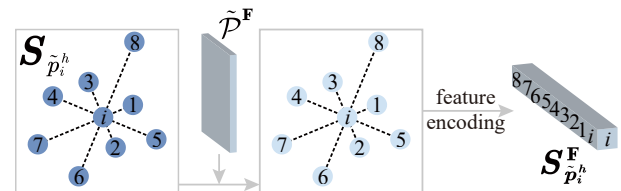


Fig. 4. Illustration of the process of fusing features within a MNFS star graphs. Starting from a central point $\tilde{p}_i$ and incorporating $h$ neighboring points, feature vectors are sequentially added based on their distance, creating a composite feature vector that combines both local geometry and associated features.

start from the center point $\tilde{p}_i$, and sequentially extract features for the leaf nodes based on their increasing order of distances. These feature vectors are concatenated with the feature vector of the central node to form a $(d \cdot h + d)$-dimensional composite feature vector, represented as $\boldsymbol{S}_{\tilde{p}_i^h}^{\mathbf{F}} \in \mathbb{R}^{1 \times (d+d \cdot h)}$. Therefore, the amalgamation of these individual composite vectors across all superpoints in $\mathcal{P}$ yields a global feature set, represented as $\boldsymbol{S}_{\tilde{\mathcal{P}}^h}^{\mathbf{F}} \in \mathbb{R}^{\tilde{M} \times (d+d \cdot h)}$.

It is important to note that prior to add features to the star graph, the feature vector of the considered vertex itself is put at the first place, i.e., $\tilde{\mathcal{P}}^{\mathbf{F}}$ and $\tilde{\mathcal{Q}}^{\mathbf{F}}$. It can also be said that it is feature when $h = 0$. Then, as we consider different scales of $h$, features are sequentially incorporated into the star graph. Therefore, $h$ is redefined as $h = \{0, h_1, \cdots, h_{M_1}\}$.

### 3.3.3 Superpoint Matching

The primary motivation for employing MNFS in super-point matching lies in its ability to precisely match points by leveraging the inherent geometric and feature-based similarities across point clouds, thereby enhancing accuracy and robustness against ambiguities and noise in complex point cloud registration tasks.

The proposed method for finding super-point correspondences utilizes feature matching based on MNFS. The process involves performing similarity calculations on the features of each superpoint obtained from the same-scale star graphs. Cosine similarity is employed for the calculation:

$$\tilde{sc}_{i,j}^h = \frac{\boldsymbol{S}_{\tilde{p}_i^h}^{\mathbf{F}} \cdot \boldsymbol{S}_{\tilde{q}_j^h}^{\mathbf{F}}}{\max(\|\boldsymbol{S}_{\tilde{p}_i^h}^{\mathbf{F}}\|_2 \cdot \|\boldsymbol{S}_{\tilde{q}_j^h}^{\mathbf{F}}\|_2, \epsilon)} \qquad (3)$$

where $i = \{1, \cdots, \tilde{M}\}$, $j = \{1, \cdots, \tilde{N}\}$, $\epsilon = 1\mathrm{e} - 8$, $\epsilon$ is a small value to avoid division by zero. According to Eq. (3), we can get $M_1 + 1$ similarity matrices $\tilde{sc}^h$ about $\boldsymbol{S}_{\tilde{\mathcal{P}}^h}^{\mathbf{F}}$ and $\boldsymbol{S}_{\tilde{\mathcal{Q}}^h}^{\mathbf{F}}$.

However, there are cases where a single point cloud contains a large area with similar characteristics, and the another point cloud may have multiple similar areas as well. To enhance global robustness and minimize the potential for ambiguous matches, a dual-normalization operation [52], [53] is applied to $\tilde{\boldsymbol{sc}}^h$ during the process:

$$sc_{i,j}^h = \frac{\tilde{sc}_{i,j}^h}{\sum_{k=1}^{\tilde{M}} \tilde{sc}_{i,k}^h} \cdot \frac{\tilde{sc}_{i,j}^h}{\sum_{k=1}^{\tilde{N}} \tilde{sc}_{k,j}^h} \qquad (4)$$

Therefore, the similarity matrix $\tilde{\boldsymbol{sc}}^h$ becomes $\boldsymbol{sc}^h$, and then the top-$K_h$ super-point correspondences are selected from each similarity matrices $\boldsymbol{sc}^h$. Therefore, totally $\tilde{\mathcal{Z}} = \sum_{h=0}^{M_1} K_h$ super-point correspondences are detected. We denote the index of the detected correspondences by $\tilde{z} \in [1, \tilde{\mathcal{Z}}]$ and denote the initial correspondence set by $\tilde{\boldsymbol{\mathcal{C}}}$. So that there are:

$$\tilde{\boldsymbol{\mathcal{C}}} = \{\boldsymbol{c}_{\tilde{z}}\}_{\tilde{z}=1}^{\tilde{\mathcal{Z}}}, \boldsymbol{c}_{\tilde{z}} = (\tilde{\boldsymbol{p}}_{\tilde{z}}^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_{\tilde{z}}^{\boldsymbol{c}}) \qquad (5)$$

where $\boldsymbol{c}_{\tilde{z}}$ denotes a correspondence, i.e., $\tilde{\boldsymbol{p}}_{\tilde{z}}^{\boldsymbol{c}} \in \tilde{\mathcal{P}}, \tilde{\boldsymbol{q}}_{\tilde{z}}^{\boldsymbol{c}} \in \tilde{\mathcal{Q}}$.

But note that these correspondences may contain duplicates and require deduplication. After deduplication, the number of super-point correspondences is denoted as $\mathcal{Z}$, and $\mathcal{Z} \leq \tilde{\mathcal{Z}}$. Therefore, $\mathcal{C}$ is redefined as:

$$\boldsymbol{\mathcal{C}} = \{\boldsymbol{c}_z\}_{z=1}^{\mathcal{Z}}, \boldsymbol{c}_z = (\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}) \qquad (6)$$

where $\tilde{\mathcal{P}}^{\boldsymbol{c}} = \{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}\}_{z=1}^{\mathcal{Z}}, \tilde{\mathcal{Q}}^{\boldsymbol{c}} = \{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}\}_{z=1}^{\mathcal{Z}}$, and $\tilde{\mathcal{P}}^{\boldsymbol{c}} \in \tilde{\mathcal{P}}, \tilde{\mathcal{Q}}^{\boldsymbol{c}} \in \tilde{\mathcal{Q}}$.

## 3.4 Mismatches Rejection Based on MCS

The motivation for implementing MCS for mismatches rejection is to refine and optimize the correspondence selection process by utilizing detailed neighborhood consistency and mutual relationships among correspondences, thus overcoming the limitations of the top-$K$ method in low-overlap scenarios, and effectively reducing incorrect matches to significantly enhance the reliability of point cloud registration outcomes.

Incorporating feature star graphs to establish super-point correspondences ensures the inclusion of the majority of points in the overlapping area. However, in low-overlap scenarios, the top-$K$ approach might select a few incorrect correspondences due to the challenge of setting an optimal $K$ value. Variations in the input point clouds may lead to occasional erroneous correspondences. Hence, outlier rejection techniques are crucial to mitigate ambiguity and reduce inaccuracies in super-point correspondences.

The graph space can more accurately depict the affinity relationship between correspondences than the Euclidean space [40]. Therefore, based on the correspondences obtained above, this study proposed MCS, which is a star graph constructed for each pair of correspondences. Then the star graphs satisfying the acceptation condition are extracted, which correspond to the correct correspondences. The MCS is constructed by firstly constructing the Multi-scale Matching Star (MMS) graphs, which help to identify the mutual correspondence relationships among the MMS vertices. Then the mutual correspondence relationships are used as vertices to construct MCS to evaluate the neighborhood consistency of the correspondences. The detailed MCS construction process is as following:

### 3.4.1 MCS Construction

The motivation for the MCS construction is to refine correspondence accuracy by leveraging mutual relationships and neighborhood consistency among Multi-scale Matching Star (MMS) graphs.

The schematic diagram of the MCS construction is shown in Fig. 5 and Fig. 6. After obtaining the correspondence sets $\tilde{\mathcal{P}}^{\boldsymbol{c}}$ and $\tilde{\mathcal{Q}}^{\boldsymbol{c}}$, Multi-scale Matching Star (MMS) graphs are constructed for each point in $\tilde{\mathcal{P}}^{\boldsymbol{c}}$ and $\tilde{\mathcal{Q}}^{\boldsymbol{c}}$. Then the set of mutual correspondences among the MMS vertices are identified. Then each mutual correspondence will be treated as vertices to finally generate the MCS graph. So the MCS construction follows three steps.
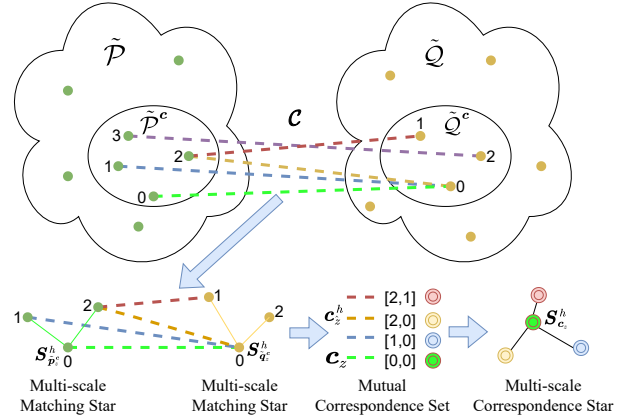


Fig. 5. The brief process of constructing MMS and MCS from input correspondences $\mathcal{C}$.

(1) **Construct MMS for each matching point**

For a correspondence $\boldsymbol{c}_z = (\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_z^{\boldsymbol{c}})$, $\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}$ finds $h$-nearest neighbors in $\tilde{\mathcal{P}}^{\boldsymbol{c}}$ to form a MMS graph, which is called $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$. The purpose of MMS is to obtain the neighborhood information around the corresponding $\boldsymbol{c}_z$ to ensure the development of neighborhood consistency screening work. Formally, $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ can be defined as $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h = (V_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h, E_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h)$, where $V_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h = \{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{p}}_{z_1}^{\boldsymbol{c}}, \cdots, \tilde{\boldsymbol{p}}_{z_h}^{\boldsymbol{c}}\}$ are $h$ nearest neighbors of $\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}$ found in $\tilde{\mathcal{P}}^{\boldsymbol{c}}$ and $E_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h = \{(\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{p}}_{z_1}^{\boldsymbol{c}}), \cdots, (\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{p}}_{z_h}^{\boldsymbol{c}})\}$ are edges from $\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}$ to these neighbors. Similarly, the $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h$ be defined as $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h = (V_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h, E_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h)$, where $V_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h = \{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_{z_1}^{\boldsymbol{c}}, \cdots, \tilde{\boldsymbol{q}}_{z_h}^{\boldsymbol{c}}\}$ and $E_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h = \{(\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_{z_1}^{\boldsymbol{c}}), \cdots, (\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_{z_h}^{\boldsymbol{c}})\}$.

Note that, the MMS $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ and $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h$ construction also allows for control over the scale size. By adjusting the parameter $h$ at different values, the number of leaf nodes in $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ and $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h$ varies. Hence, this aligns with the emphasis on the importance of scale size in the MNFS construction. The parameter $h$ in this context is also designed to be multi-scale $h = \{h_m\}_{m=1}^{M_2}$, where $M_2$ is the number of scales in the MCS construction.

(2) **Find mutual correspondence between $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ and $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h$**

After gathering neighborhood information for the correspondence point $\boldsymbol{c}_z$ within the star-graphs $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$ and $\boldsymbol{S}_{\tilde{\boldsymbol{q}}_z^{\boldsymbol{c}}}^h$, we focus on identifying consistent correspondences. This is done by selecting mutual correspondences, termed $\boldsymbol{c}_{\hat{z}_l} = (\tilde{\boldsymbol{p}}_{z_l}^{\boldsymbol{c}}, \tilde{\boldsymbol{q}}_{z_l}^{\boldsymbol{c}})$, from these star-graphs. These mutual correspondences must align with the original correspondence $\boldsymbol{c}_z$.

Specifically, we ensure that one part of the mutual correspondence, $\tilde{\boldsymbol{p}}_{z_l}^{\boldsymbol{c}}$ (a component of $\boldsymbol{c}_{\hat{z}_l}$), is located within the neighborhood of $\boldsymbol{c}_z$ in $\boldsymbol{S}_{\tilde{\boldsymbol{p}}_z^{\boldsymbol{c}}}^h$. Similarly, the corresponding point $\tilde{\boldsymbol{q}}_{z_l}^{\boldsymbol{c}}$, which aligns with $\tilde{\boldsymbol{p}}_{z_l}^{\boldsymbol{c}}$ in $\boldsymbol{c}_{\hat{z}_l}$, is also found within the neighborhood of
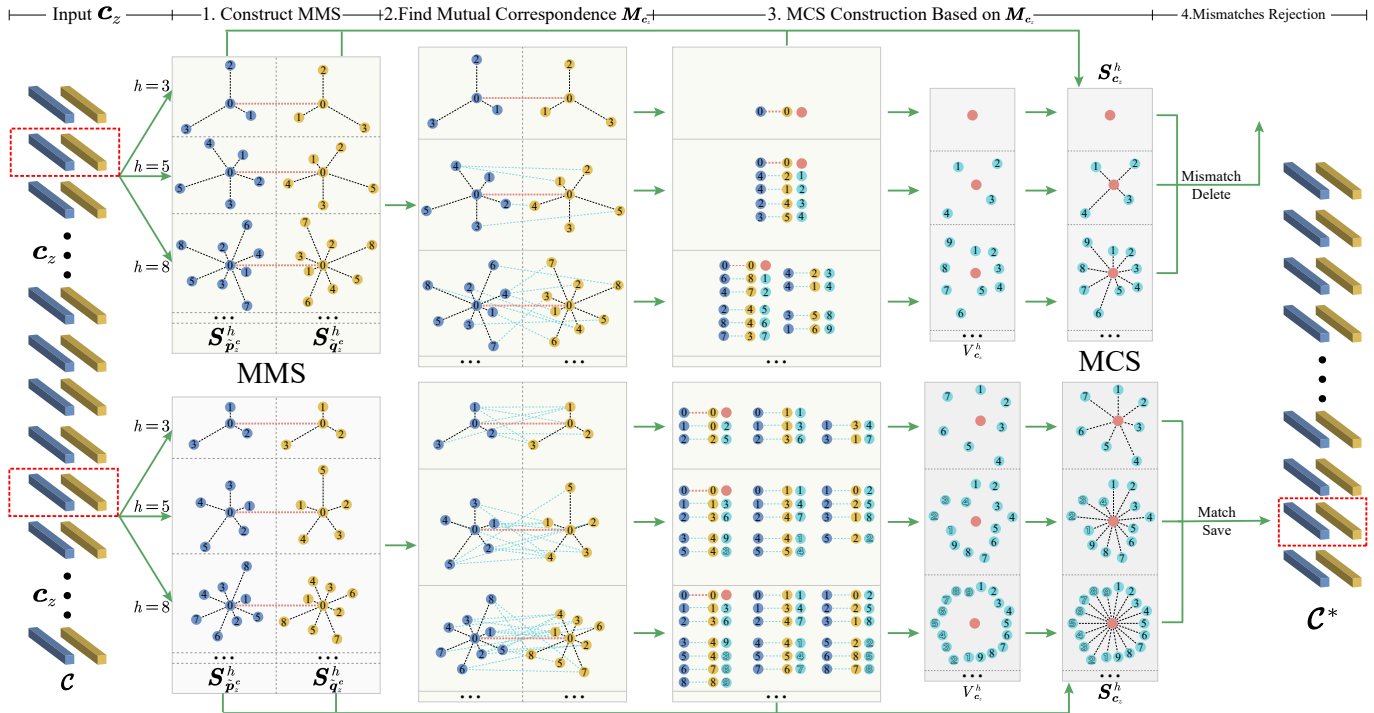
Fig. 6. Schematic diagram of the MCS process. The dots in $S_{\tilde{p}_z^c}^h$ and $S_{\tilde{q}_z^c}^h$ both belong to $\tilde{p}_z^c$ and $\tilde{q}_z^c$ in $\mathcal{C}$. The green point and dotted line in the figure represent the one-to-one mutual correspondence $c_{\hat{z}_l}$ in $\mathcal{C}$, and the dots at both ends of the dotted line represent $\tilde{p}_z^c$ and $\tilde{q}_z^c$ in $\mathcal{C}$. The red dots and dotted lines indicate the corresponding relationships that require evaluation to determine if the conditions are met.

$c_z$ in $S_{\tilde{q}_z^c}^h$. This step ensures that the mutual correspondences $c_{\hat{z}_l}$ are consistent with the original correspondence $c_z$.

Therefor, we find mutual correspondences $c_{\hat{z}_l}$ between $S_{\tilde{p}_z^c}^h$ and $S_{\tilde{q}_z^c}^h$ within $\mathcal{C}$. In particularly, we find the correspondences $c_{\hat{z}_l}$ that satisfy Eq. (7).

$$M_{c_z} = \{(\tilde{p}_{z_l}^c, \tilde{q}_{z_l}^c) | (\tilde{p}_{z_l}^c \in V_{\tilde{p}_z^c}^h) \cap (\tilde{q}_{z_l}^c \in V_{\tilde{q}_z^c}^h)\} \quad (7)$$

where $\cap$ represents the relationship of 'AND'. We denoted the extracted mutual correspondences set as $M_{c_z} = \{c_{\hat{z}_l}\}$, which contains the correspondences $c_{\hat{z}_l} = (\tilde{p}_{z_l}^c, \tilde{q}_{z_l}^c)$, where both conditions $\tilde{p}_{z_l}^c$ belongs to $V_{\tilde{p}_z^c}^h$ and $\tilde{q}_{z_l}^c$ belongs to $V_{\tilde{q}_z^c}^h$ are satisfied simultaneously. Therefore, it can be seen that $M_{c_z} \in \mathcal{C}$, which in Eq. (6)

(3) **MCS construction based on $M_{c_z}$**

Then, based on the extracted mutual correspondence set $M_{c_z}$, each correspondence is treated as a vertex to form a new star graph, i.e., the Multiple-scale Correspondence Star (MCS). In particular, we designate $c_z$ as the internal point of the MCS, and other correspondences in $c_{\hat{z}_l} \in M_{c_z}$ as the leaf nodes. Nevertheless, the leaf nodes are currently disconnected from the internal point, so it is essential to establish connections between these leaf nodes and the internal point. As mentioned earlier, the Euclidean distance within the graph space carries crucial information for visual tasks. Hence, the Euclidean distance between the leaf node $c_{\hat{z}_l}$ and the internal point $c_z$ is utilized as the weight for the edge connecting the two:

$$w_{zl} = \frac{\left\| \|\tilde{p}_{z_l}^c - \tilde{p}_z^c\| - \|\tilde{q}_{z_l}^c - \tilde{q}_z^c\| \right\|}{\max\left(\|\tilde{p}_{z_l}^c - \tilde{p}_z^c\|, \|\tilde{q}_{z_l}^c - \tilde{q}_z^c\|, \epsilon\right)} \quad (8)$$

This weight evaluates the similarity of the two correspondences, i.e., $c_{z_l} = (\tilde{p}_{\hat{z}_l}^c, \tilde{q}_{z_l}^c)$ and $c_z = (\tilde{p}_z^c, \tilde{q}_z^c)$. Smaller edge weights will be given for the correspondence pair $c_{\hat{z}_l}, c_z$ if they are geometrically similar. After assigning weights, the MCS $S_{c_z}^h$ can be defined as $S_{c_z}^h = (V_{c_z}^h, E_{c_{z_l}}^h, W_{c_{z_l}}^h)$, where $V_{c_z}^h = \{c_z, c_{\hat{z}_l} \in M_{c_z}\}$, $E_{c_{z_l}}^h = \{(c_{\hat{z}_l}, c_z)\}$, $W_{c_{z_l}}^h = \{w_{zl}\}$ for $c_{\hat{z}_l} \in M_{c_z}$. After these processes, we have successfully obtained the star graphs set $S_c = \{S_{c_z}^h\}$, where $z = \{1, \cdots, \mathcal{Z}\}$, $h = \{h_1, \cdots, h_{M_2}\}$.

### 3.4.2 Mismatches Rejection

The motivation for mismatches rejection is to utilize neighborhood consistency within star graphs for identifying true correspondences, effectively filtering out inaccuracies and enhancing registration precision.

The neighborhood consistency among true correspondences is particularly pronounced, as they often appear in groups. Taking inspiration from this observation, our approach leverages the construction of the star graphs to vividly illustrate the neighborhood consistency between the corresponding relationships. This enables us to effectively filter out corresponding relationships that demonstrate strong neighborhood consistency. In this context, this study determines the correctness of the correspondence based on weight considerations of the leaf nodes and the edges within the star graph on the each scale $h$.

(1) **Weighs of $S_{c_z}^h$.**

To calculate the weight $\mathcal{W}_{c_z^h}$ of the star graph $S_{c_z}^h$, we begin by considering the weight $W_{c_{z_l}}^h$ of the edge within the graph:

$$\mathcal{W}_{c_z^h} = \frac{1}{N_{zh}} \sum_{l=1}^{N_{zh}} \exp(-w_{zl}) \quad (9)$$

where $w_{zl} \in W_{\boldsymbol{c}_{z_l}}^h$, $N_{zh}$ is the number of leafs in $\boldsymbol{S}_{\boldsymbol{c}_z}^h$; $z = \{1, \cdots, \mathcal{Z}\}$, $h = \{h_1, \cdots, h_{M_2}\}$. Exactly, this assigns weights to each correspondence at each scale by considering the edge weights within the star graph. Notably, the closer correspondence pairs contribute the higher weights. Subsequently, based on the evaluated weights, the correct correspondence at each scale can be filtered out by requiring its sum weight is higher than the average sum weight of the same level star graphs.

$$\mathcal{W}_{\boldsymbol{c}_z^h} > \frac{1}{\mathcal{Z}} \sum_{z=1}^{\mathcal{Z}} \mathcal{W}_{\boldsymbol{c}_z^h} \tag{10}$$

By referring to Eq. (8) and Eq. (9), we can observe that a higher weight assigned to $\boldsymbol{c}_z^h$ indicates a greater consistency with the surrounding corresponding relationships.

(2) **Leaf nodes of $\boldsymbol{S}_{\boldsymbol{c}_z}^h$.**

As the number of leaf nodes in $\boldsymbol{S}_{\boldsymbol{c}_z}^h$ reflects the number of corresponding relationships surrounding $\boldsymbol{c}_z^h$, it serves as a vivid manifestation of neighborhood consistency and can be used as a reference standard. Through extensive testing, we found that the optimal threshold is $2h$. Therefore, we establish a specific criteria based on the following factor:

$$N_{zh} > 2h \tag{11}$$

Consequently, correspondences in $\mathcal{C}$ that satisfy both Eq. (10) and Eq. (11) at the same scale $h$ are filtered out as the correct correspondences $\mathcal{C}^* = \{\boldsymbol{c}_{\check{z}}\}_{\check{z}=1}^{\check{\mathbf{Z}}}$, while others are dropped.

## 3.5 Decoding and Point Matching

Based on the established highly reliable correspondences for super-points, we can further refine these corresponding super-point pairs at a fine point level to extract correspondences for individual fine points. This refinement process aims to estimate the optimal transformation. For the specific refinement process, it is recommended to refer to the work of GeoTransformer [37].

## 3.6 Loss Function

Given the DMS is an explicit one-way module and its integration in the GeoTransformer [37] framework, we maintains the use of the original loss function without making improvements. The loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{oc} + \mathcal{L}_p \tag{12}$$

where $\mathcal{L}_{oc}$ represents an overlap-aware circle loss tailored for super-point matching, and $\mathcal{L}_p$ embodies a point matching loss for refining point correspondences. For an in-depth exposition, please consult [37].

## 4 EXPERIMENT

In this section, an extensive experimental evaluation of the proposed DMS is conducted. Firstly, the implementation details of the experiments are introduced. Subsequently, the method is evaluated and compared against previous state-of-the-art methods. Furthermore, an ablation study is performed to gain a comprehensive understanding of the design. Finally, the visual experiment of DMS and sota method is shown.

## 4.1 Experimental Setup

**Datasets.** Three datasets were considered in the study: 3DMatch [54] and 3DLoMatch [35] for indoor scene-scale data, KITTI [55] for outdoor scene-scale data, and ModelNet [56] for synthetic point cloud data, and 4DMatch [57] and 4DLoMatch [57] for non-rigid PCR. Within these datasets, 3DLoMatch is a subset of 3DMatch with point cloud pairs exhibiting an overlap rate between 10% and 30%, posing a significant challenge for algorithms.The ModelNet dataset consists of synthetic point clouds generated from 12,311 CAD models across 40 different classes, with ModelLoNet differing from ModelNet in terms of its average overlap rate (53.6% for ModelLoNet and 73.5% for ModelNet). 4DMatch [57] is a comprehensive benchmark created for the registration of non-rigid point clouds, employing animation sequences from DeformingThings4D [58]. Based on an overlap ratio threshold of 45%, the test sequence's point cloud pairs are divided into two categories: 4DMatch [57] and 4DLoMatch [57].

**Evaluation Criteria.** The evaluation metrics in LoR include: 1) Initial Ratio (IR) assesses the accuracy of the initial alignment by measuring the fraction of correspondences with residuals below a threshold $\mathcal{R}_I$. 2) Feature Matching Recall (FMR) evaluates the effectiveness of feature matching by measuring the proportion $\mathcal{R}_F$ of point cloud pairs with a high IR. 3) Registration Recall (RR) measures the fraction of point cloud pairs that achieve a transformation error below a specified threshold $\mathcal{R}_R$, providing an overall assessment of the registration quality. 4) Relative Rotation Error (RRE) measures the geodesic distance between the estimated rotation matrix and the ground truth rotation matrix. 5) Relative Translation Error (RTE) calculates the Euclidean distance between the estimated and ground truth translation vector. 6) The modified chamfer distance (CD) [59] is used to assess the alignment quality of our method compared to other methods on the ModelNet dataset. 7) The Non-rigid Inlier Ratio (NIR) [57], [60] is the percentage of tentative matches with residuals smaller than a specified limit (such as 0.04 meters) according to the ground-truth warping function. 8) Non-rigid Feature Matching Recall (NFMR) [57], [60] is the proportion of accurate matches from the ground truth that are correctly identified through the proposed matches.

**Implementation Details.** The multiscale sizes used in the experiments vary depending on the density and size of the point clouds from different benchmarks. The specific details in Table 2. As for the network parameter setting of GeoTransformer [37], it adheres to the standard setting of GeoTransformer [37]. The implementation and evaluation of our DMS approach were conducted using PyTorch [61] on a Intel(R) Xen(R) Silver 4116 CPU@2.10GHz and 4 NVIDIA RTX 3090 GPUs.

TABLE 2
Parameter Setting Details.

| Datasets | $h$ of MNFS | $h$ of MCS | K |
|---|---|---|---|
| 3DMatch | [2, 4, 8, 16] | [3, 5, 8] | [256, 128, 64, 32, 16] |
| KITI | [4, 8, 16, 32] | [4, 6, 8] | [256, 128, 64, 64, 32] |
| ModelNet | [2, 4, 8] | [3, 5, 7] | [128, 64, 32, 16] |
| 4DMatch | [1, 2, 4, 6] | [3, 5, 7] | [256, 128, 64, 32, 16] |

## 4.2 Comparative Experiment

**3DMatch and 3DLoMatch.** Following the methodology introduced by GeoTransformer [37], we conduct a comprehensive

TABLE 3
Scene-wise registration results on 3DMatch and 3DLoMatch. Boldfaced numbers indicate the DMS performances

| Dataset | 3DMatch | | | | | | | | 3DLoMatch | | | | | | | |
| Scene | Kitchen | Home_1 | Home_2 | Hotel_1 | Hotel_2 | Study | Lab | Mean | Kitchen | Home_1 | Home_2 | Hotel_1 | Hotel_2 | Study | Lab | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Registration Recall (%) RR** | | | | | | | | | | | | | | | | |
| FCGF | 98.0 | 94.3 | 68.6 | 96.7 | 91.0 | 76.1 | 71.1 | 85.1 | 60.8 | 42.2 | 53.6 | 53.1 | 38.0 | 16.1 | 30.4 | 40.1 |
| D3Feat | 96.0 | 86.8 | 67.3 | 90.7 | 88.5 | 78.2 | 64.4 | 81.6 | 49.7 | 37.2 | 47.3 | 47.8 | 36.5 | 15.7 | 31.9 | 37.2 |
| Predator | 97.6 | 97.2 | 74.8 | 98.9 | 96.2 | 85.9 | 73.3 | 89.0 | 71.5 | 58.2 | 60.8 | 77.5 | 64.2 | 45.8 | 39.1 | 59.8 |
| CoFiNet | 96.4 | 99.1 | 73.6 | 95.6 | 91.0 | 84.6 | 84.4 | 89.3 | 76.7 | 66.7 | 64.0 | 81.3 | 65.0 | 53.4 | 69.6 | 67.5 |
| GeoTransformer | 98.9 | 97.2 | 81.1 | 98.9 | 89.7 | 88.5 | 88.9 | 91.5 | 85.9 | 73.5 | 72.5 | 89.5 | 73.2 | 55.3 | 75.7 | 74.0 |
| **DMS** | **99.1** | **98.2** | **84.6** | **99.0** | **91.3** | **90.5** | **90.9** | **94.3** | **87.2** | **76.3** | **75.8** | **90.7** | **76.8** | **59.3** | **78.1** | **77.3** |
| **Relative Rotation Error (°) RRE** | | | | | | | | | | | | | | | | |
| FCGF | 1.767 | 1.849 | 2.210 | 1.867 | 1.667 | 2.417 | 1.792 | 1.949 | 2.904 | 3.229 | 3.277 | 2.768 | 2.801 | 3.372 | 4.006 | 3.147 |
| D3Feat | 2.016 | 2.029 | 2.425 | 1.990 | 1.967 | 2.400 | 2.114 | 2.161 | 3.226 | 3.492 | 3.373 | 3.330 | 3.165 | 3.708 | 3.619 | 3.361 |
| Predator | 1.861 | 1.806 | 2.473 | 2.045 | 1.600 | 2.458 | 1.926 | 2.029 | 3.079 | 2.637 | 3.220 | 2.694 | 2.907 | 3.046 | 3.412 | 3.048 |
| CoFiNet | 1.910 | 1.835 | 2.316 | 1.767 | 1.753 | 1.639 | 2.345 | 2.011 | 3.213 | 3.119 | 3.711 | 2.842 | 2.897 | 4.126 | 3.138 | 3.280 |
| GeoTransformer | 1.797 | 1.353 | 1.797 | 1.528 | 1.328 | 1.571 | 1.678 | 1.625 | 2.356 | 2.305 | 2.541 | 2.455 | 2.490 | 3.010 | 2.716 | 2.547 |
| **DMS** | **1.703** | **1.238** | **1.683** | **1.486** | **1.306** | **1.536** | **1.559** | **1.518** | **2.108** | **2.264** | **2.351** | **2.313** | **2.276** | **2.734** | **2.518** | **2.294** |
| **Relative Translation Error (m) RTE** | | | | | | | | | | | | | | | | |
| FCGF | 0.053 | 0.056 | 0.071 | 0.062 | 0.061 | 0.055 | 0.090 | 0.066 | 0.084 | 0.097 | 0.076 | 0.101 | 0.084 | 0.144 | 0.140 | 0.100 |
| D3Feat | 0.055 | 0.065 | 0.080 | 0.064 | 0.078 | 0.049 | 0.064 | 0.067 | 0.088 | 0.101 | 0.086 | 0.099 | 0.092 | 0.146 | 0.135 | 0.103 |
| Predator | 0.048 | 0.055 | 0.070 | 0.073 | 0.060 | 0.065 | 0.063 | 0.064 | 0.081 | 0.080 | 0.084 | 0.099 | 0.096 | 0.101 | 0.130 | 0.093 |
| CoFiNet | 0.047 | 0.059 | 0.063 | 0.063 | 0.058 | 0.044 | 0.075 | 0.062 | 0.080 | 0.078 | 0.078 | 0.099 | 0.086 | 0.131 | 0.123 | 0.094 |
| GeoTransformer | 0.042 | 0.046 | 0.055 | 0.055 | 0.046 | 0.050 | 0.053 | 0.053 | 0.062 | 0.070 | 0.071 | 0.080 | 0.075 | 0.107 | 0.083 | 0.074 |
| **DMS** | **0.033** | **0.034** | **0.042** | **0.043** | **0.041** | **0.047** | **0.046** | **0.045** | **0.057** | **0.067** | **0.063** | **0.068** | **0.063** | **0.082** | **0.071** | **0.058** |

comparison of our method with state-of-the-art approaches including PerfectMatch [62], FCGF [44], D3Feat [45], SpinNet [41], Predator [35], YOHO [63], CoFiNet [36], GeoTransformer [37], and PEAL [38] across three key metrics: FMR, IR, and RR. From the results presented in Table 4, it is evident that our method outperforms the previous approaches in all evaluation metrics. This demonstrates the superiority and effectiveness of our method compared to the existing methods.

TABLE 4
Evaluation results on 3DMatch and 3DLoMatch. Boldfaced numbers indicate the best performances, and the second best performances are underlined.

| Dataset | 3DMatch | | | | | 3DLoMatch | | | | |
| Samples | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Feature Matching Recall (%) FMR** | | | | | | | | | | |
| PerfetMatch | 95.0 | 94.3 | 92.9 | 90.1 | 82.9 | 63.6 | 61.7 | 53.6 | 75.2 | 34.2 |
| FCGF | 97.4 | 97.3 | 97.0 | 96.7 | 96.6 | 76.6 | 75.4 | 74.2 | 71.7 | 67.3 |
| D3Feat | 95.6 | 95.4 | 94.5 | 94.1 | 93.1 | 67.3 | 66.7 | 67.0 | 66.7 | 66.5 |
| SpinNet | 97.6 | 97.2 | 96.8 | 95.5 | 94.3 | 75.3 | 74.9 | 72.5 | 70.0 | 63.6 |
| Predator | 96.6 | 96.6 | 96.5 | 96.3 | 96.5 | 78.6 | 77.4 | 76.3 | 75.7 | 75.3 |
| YOHO | 98.2 | 97.6 | 97.5 | 97.7 | 96.0 | 79.4 | 78.1 | 76.3 | 73.8 | 69.1 |
| CoFiNet | 98.1 | 98.3 | 98.1 | 98.2 | 98.3 | 83.1 | 83.5 | 83.3 | 83.1 | 82.6 |
| GeoTransformer | 97.9 | 97.9 | 97.9 | 97.9 | 97.3 | 88.3 | 88.6 | 88.8 | 88.6 | 88.3 |
| PEAL | 99.0 | 99.0 | 99.1 | 99.1 | 98.8 | 91.7 | 92.4 | 92.5 | 92.9 | 92.7 |
| **DMS** | **99.1** | **99.2** | **99.2** | **99.1** | **98.8** | **92.3** | **92.8** | **93.1** | **93.5** | **93.3** |
| **Inlier Ratio (%) IR** | | | | | | | | | | |
| PerfetMatch | 36.0 | 32.5 | 26.4 | 21.5 | 16.4 | 11.4 | 10.1 | 8.0 | 6.4 | 4.8 |
| FCGF | 56.8 | 54.1 | 48.7 | 42.5 | 34.1 | 21.4 | 20.0 | 17.2 | 14.8 | 11.6 |
| D3Feat | 39.0 | 38.8 | 40.4 | 41.5 | 41.8 | 13.2 | 13.1 | 14.0 | 14.6 | 15.0 |
| SpinNet | 47.5 | 44.7 | 39.4 | 33.9 | 27.6 | 20.5 | 19.0 | 16.3 | 13.8 | 11.1 |
| Predator | 58.0 | 58.4 | 57.1 | 54.1 | 49.3 | 26.7 | 28.1 | 28.3 | 27.5 | 25.8 |
| YOHO | 64.4 | 60.7 | 55.7 | 46.4 | 41.2 | 25.9 | 23.3 | 22.6 | 18.2 | 15.0 |
| CoFiNet | 49.8 | 51.2 | 51.9 | 52.2 | 52.2 | 24.4 | 25.9 | 26.7 | 26.8 | 26.9 |
| GeoTransformer | 71.9 | 75.2 | 76.0 | 82.2 | 85.1 | 43.5 | 45.3 | 46.2 | 52.9 | 57.7 |
| PEAL | 72.4 | 79.1 | 84.1 | 86.1 | 87.3 | 45.0 | 50.9 | 57.4 | 60.3 | 62.2 |
| **DMS** | **73.6** | **80.8** | **85.5** | **87.3** | **88.1** | **47.8** | **51.6** | **58.2** | **61.5** | **63.8** |
| **Registration Recall (%) RR** | | | | | | | | | | |
| PerfetMatch | 78.4 | 76.2 | 71.4 | 67.6 | 50.8 | 33.0 | 29.0 | 23.3 | 17.0 | 11.0 |
| FCGF | 85.1 | 84.7 | 83.3 | 81.6 | 71.4 | 40.1 | 41.7 | 38.2 | 35.4 | 26.8 |
| D3Feat | 81.6 | 84.5 | 83.4 | 82.4 | 77.9 | 37.2 | 42.7 | 46.9 | 43.8 | 39.1 |
| SpinNet | 88.6 | 86.6 | 85.5 | 83.5 | 70.2 | 59.8 | 54.9 | 48.3 | 39.8 | 26.8 |
| Predator | 89.0 | 89.9 | 90.6 | 88.5 | 86.6 | 59.8 | 61.2 | 62.4 | 60.8 | 58.1 |
| YOHO | 90.8 | 90.3 | 89.1 | 88.6 | 84.5 | 65.2 | 65.5 | 63.2 | 56.5 | 48.0 |
| CoFiNet | 89.3 | 88.9 | 88.4 | 87.4 | 87.0 | 67.5 | 66.2 | 64.2 | 63.1 | 61.0 |
| GeoTransformer | 92.0 | 91.8 | 91.8 | 91.4 | 91.2 | 75.0 | 74.8 | 72.2 | 74.1 | 73.5 |
| PEAL | 94.6 | 93.7 | 93.7 | 93.9 | 93.4 | 81.7 | 81.2 | 80.8 | 80.4 | 80.1 |
| **DMS** | **95.2** | **94.3** | **94.1** | **95.0** | **94.6** | **82.3** | **81.8** | **81.5** | **81.2** | **80.7** |

In the comparison of registration results in Table 5 without using RANSAC under the experimental guidelines of GeoTransformer [37], the DMS (weighted SVD) approach achieved a registration recall of 90.3% on the 3DMatch dataset and a registration recall of 68.4% on the 3DLoMatch dataset. The result of 3DLoMatch surpassed GeoTransformer's performance by 8.5%, primarily due to the robustness of DMS. In the absence of outlier filtering provided by RANSAC, a high ratio of outliers can pose a significant challenge to PCR. The MCS technique show effctive performances in removing the outliers.

TABLE 5
Registration results w/o RANSAC on 3DMatch (3DM) and 3DLoMatch (3DLM). Boldfaced numbers highlight the best and the second best are underlined.

| Estimator | RANSAC-50k | | weighted SVD | | LGR | |
| Samples | 5000 | | 250 | | all | |
| Data | 3DM | 3DL | 3DM | 3DL | 3DM | 3DL |
|---|---|---|---|---|---|---|
| **Registration Recall (%) RR** | | | | | | |
| FCGF | 85.1 | 40.1 | 42.1 | 3.9 | null | null |
| D3Feat | 81.6 | 37.2 | 37.4 | 2.8 | null | null |
| SpinNet | 88.6 | 59.8 | 34.0 | 2.5 | null | null |
| Predator | 89.0 | 59.8 | 50.0 | 6.4 | null | null |
| CoFiNet | 89.3 | 67.5 | 64.6 | 21.6 | 87.6 | 64.8 |
| GeoTransformer | 92.0 | 75.0 | 86.5 | 59.9 | 91.5 | 74.0 |
| PEAL | 94.6 | 81.7 | null | null | 94.3 | 81.2 |
| **DMS** | **95.2** | **82.3** | **90.3** | **68.4** | **94.8** | **81.4** |

Similarly, Table 3 presents the registration results of 3DMatch and 3DLoMatch in various scenarios. It is evident from the table that DMS consistently achieves the highest registration recall and the lowest relative error across most scenarios. These results demonstrate the robustness and accuracy of DMS in handling different registration scenarios. The superior performance of DMS further validates its effectiveness in LoR tasks.

**KITTI.** The KITTI odometry dataset [55] comprises a sequence of 11 outdoor driving scenes captured by lidar sensors, we evaluate

our method based on the protocol described in GeoTransformer [37]. The Table 6 presents the comparison results with state-of-the-art methods, including 3DFeat-Net , FCGF [44], D3Feat [45] , SpinNet [41], Predator [35] , CoFiNet [36], and GeoTransformer [37]. It is evident from the table that our method achieves a registration recall rate comparable to the current state-of-the-art approaches. It is worth noting that our algorithm does not excel in relative error. This limitation arises because DMS is an explicit algorithm, and controlling its parameters requires consideration of an extremely large inlier system. As a future direction, researchers can explore learning-based methods to automatically determine suitable multi-scale settings, eliminating the need for manual configuration of multi-scale hyperparameters.

**TABLE 6**
Registration results on KITTI odometry. Boldfaced numbers highlight the best and the second best are underlined.

| Model | RTE(cm) | RRE(°) | RR(%) |
|---|---|---|---|
| RANSAC-*50k* | | | |
| 3DFeat-Net | 25.9 | **0.25** | 96.0 |
| FCGF | 9.5 | 0.30 | 96.6 |
| D3Feat | 7.2 | 0.30 | **99.8** |
| SpinNet | 9.9 | 0.47 | 99.1 |
| Preadator | **6.8** | <u>0.27</u> | **99.8** |
| CoFiNet | 8.2 | 0.41 | **99.8** |
| GeoTransformer | 7.4 | <u>0.27</u> | **99.8** |
| **DMS** | <u>6.9</u> | 0.29 | **99.8** |
| LGR | | | |
| FMR | ∼66 | 1.49 | 90.6 |
| DGR | ∼32 | 0.37 | 98.7 |
| HRegNet | ∼12 | 0.29 | 99.7 |
| GeoTransformer | <u>6.8</u> | **0.24** | **99.8** |
| **DMS** | **6.7** | <u>0.27</u> | **99.8** |

**ModelNet.** We conducted comparisons between our method and five other algorithms on the ModelNet and ModelLoNet datasets. The results of these comparisons are presented in Table 1. The experiments on these datasets were conducted following the experimental guidelines outlined by GeoTransformer [37]. From the results, it can be seen that our algorithm has achieved the lowest registration error in both ModelNet and ModelLoNet.

**TABLE 7**
Registration results on ModelNet40 and ModelLoNet. Boldfaced numbers hightlight the best and the second best are underlined.

| Model | ModelNet | | | ModelLoNet | | |
|---|---|---|---|---|---|---|
| | RRE | RTE | CD | RRE | RTE | CD |
| Small Rotation | | | | | | |
| RPM-Net | 2.357 | 0.028 | 0.00130 | 8.123 | 0.086 | 0.00611 |
| RGM | 4.548 | 0.049 | 0.00268 | 14.806 | 0.139 | 0.01482 |
| Predator | <u>2.064</u> | <u>0.023</u> | 0.00145 | 5.022 | 0.084 | 0.00734 |
| CoFiNet | 3.584 | 0.044 | 0.00205 | 6.992 | 0.091 | 0.00599 |
| GeoTransformer | 2.160 | 0.024 | <u>0.00143</u> | <u>3.638</u> | <u>0.064</u> | <u>0.00448</u> |
| **DMS** | **1.319** | **0.015** | **0.00086** | **2.266** | **0.030** | **0.00315** |
| Large Rotation | | | | | | |
| RPM-Net | 31.509 | 0.206 | 0.01074 | 51.478 | 0.346 | 0.01985 |
| RGM | 45.560 | 0.289 | 0.01697 | 68.724 | 0.442 | 0.03634 |
| Predator | 24.839 | 0.171 | 0.01940 | 46.990 | 0.378 | 0.05052 |
| CoFiNet | 10.496 | 0.084 | 0.00319 | 32.578 | 0.226 | 0.02273 |
| GeoTransformer | <u>6.436</u> | <u>0.047</u> | <u>0.00154</u> | <u>23.478</u> | <u>0.152</u> | <u>0.01296</u> |
| **DMS** | **5.538** | **0.043** | **0.00103** | **22.365** | **0.141** | **0.00963** |

**4DMatch and 4DLoMatch.** We conducted a comparison between DMS and four recent methods: D3Feat [45], Predator [35], Lepard [57], and GeoTransformer [60], with the results detailed in Table 8. DMS significantly outperforms D3Feat and Predator in both high-overlap and low-overlap scenarios and achieves performance very close to that of Lepard and GeoTransformer. It's important to note that Lepard benefits from a repositioning mechanism with coarse rigid registration, which effectively enhances its performance. Without repositioning, our DMS consistently exceeds Lepard across both benchmarks. Despite not being explicitly designed and optimized for handling deformation and non-rigid registration, DMS demonstrates commendable performance in non-rigid point cloud registration. In most instances, non-rigid deformation can be approximated by a series of local rigid transformations. Our MNFS is designed to capture the local rigidity consistency between two point clouds, aiding in the extraction of high-quality correspondences in non-rigid scenarios.

**TABLE 8**
Registration results on 4DMatch and 4DLoMatch. NIR and NFMR are measured in %. Boldfaced numbers hightlight the best and the second best are underlined.

| Model | 4DMatch | | | 4DLoMatch | | |
|---|---|---|---|---|---|---|
| | #Corr | NIR | NFMR | #Corr | NIR | NFMR |
| D3Feat | 697 | 55.3 | 56.1 | 204 | 21.3 | 28.1 |
| Predator | 698 | 59.3 | 56.8 | 480 | 25 | 32.1 |
| Lepard | 596 | **82.7** | **83.7** | 407 | 55.7 | **66.9** |
| Lepard (w/o repos) | 624 | 80.5 | 80.8 | 448 | 53.7 | 63.6 |
| GeoTransformer | 2331 | <u>82.2</u> | <u>83.2</u> | 1212 | **63.6** | <u>65.4</u> |
| DMS | 2653 | 81.9 | 83.0 | 1348 | <u>63.5</u> | <u>65.4</u> |

## 4.3  Ablation Experiment

In this section, our focus centers on conducting extensive ablation research and analysis experiments to investigate the crucial components of the DMS, namely MNFS and MCS. The evaluations are performed across multiple datasets, including 3DMatch, 3DLoMatch, ModelNet, and ModelLoNet, with a meticulous examination of their performance and impact. Furthermore, we delve deeper into the examination of the effects of MNFS, MCS, and DMS in other method frameworks, shedding light on the contributions and significance of each component in improving the overall performance of the algorithm. Finally, an essential aspect of our analysis revolves around the assessment of the time efficiency of the DMS algorithm.

**Performance Analysis of MNFS and MCS in DMS Framework.** We independently assessed MNFS and MCS contributions within the DMS framework. Table 9 displays results for FMR, IR, and RR across different sampling numbers on 3DMatch and 3DLoMatch datasets. We found MNFS slightly improved FMR by around 0.5% compared to MCS. MNFS consistently outperformed MCS in IR, often exceeding 1%. Remarkably, with 1000 sampling points, MNFS's improvement was about twice that of other settings. It is important to highlight that the impact of such a substantial IR increase has limited effect on RR in the 3DMatch dataset. However, in the case of 3DLoMatch, the increase in RR is more pronounced. This phenomenon can be explained by the fact that under high overlap, the inliers tend to cluster together, with these internal points playing a crucial role in low-overlap scenarios. Consequently, these results provide further evidence of

the robustness of DMS, particularly in challenging situations with low-overlap.

To assess the individual contributions of MNFS and MCS in the DMS framework, we conducted separate experiments using the ModelNet and ModelLoNet datasets. The results in Table 10 depict the relative error performance of each algorithm. The data in the table reveals that MNFS yields a more substantial improvement compared to MCS. Additionally, the improvement achieved on the ModelLoNet dataset surpasses that on the ModelNet dataset. It is important to note that DMS exhibits only a slight advantage in scenarios involving low rotation compared to those with large rotations. Hence, it can be inferred that rotation factors do not significantly impact DMS during registration. Consequently, DMS demonstrates adaptability not only to point clouds with low overlap but also to those with substantial rotations.

TABLE 9
Performance boosting on 3DMatch and 3DLoMatch with MNFS w/o peripheral points, MNFS, MCS, and DMS. Boldfaced numbers hightlight the best.

| Dataset | 3DMatch | | | | | 3DLoMatch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Sample | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 |
| Feature Matching Recall (%) FMR | | | | | | | | | | |
| GeoTransformer | 97.9 | 97.9 | 97.9 | 97.9 | 97.3 | 88.3 | 88.6 | 88.8 | 88.6 | 88.3 |
| MNFS w/o peripheral points | 98.3 | 98.3 | 98.3 | 98.3 | 97.9 | 89.3 | 90.3 | 90.8 | 90.7 | 90.6 |
| | 0.4↑ | 0.4↑ | 0.4↑ | 0.4↑ | 0.6↑ | 1.0↑ | 1.7↑ | 2.0↑ | 2.1↑ | 2.3↑ |
| MNFS | 98.8 | 98.8 | 98.7 | 98.7 | 98.5 | 90.8 | 91.3 | 91.5 | 91.9 | 91.5 |
| | 0.9↑ | 0.9↑ | 0.8↑ | 0.8↑ | 1.2↑ | 2.5↑ | 2.7↑ | 2.7↑ | 3.3↑ | 3.2↑ |
| MCS | 98.4 | 98.3 | 98.4 | 98.4 | 98.1 | 90.4 | 90.9 | 90.8 | 90.7 | 90.2 |
| | 0.5↑ | 0.4↑ | 0.5↑ | 0.5↑ | 0.8↑ | 2.1↑ | 2.3↑ | 2.0↑ | 2.1↑ | 1.9↑ |
| **DMS** | **99.1** | **99.2** | **99.2** | **99.1** | **98.8** | **92.3** | **92.8** | **93.1** | **93.5** | **93.3** |
| | **1.2↑** | **1.3↑** | **1.3↑** | **1.2↑** | **1.5↑** | **4.0↑** | **4.2↑** | **4.3↑** | **4.9↑** | **5.0↑** |
| Inlier Ratio (%) IR | | | | | | | | | | |
| GeoTransformer | 71.9 | 75.2 | 76.0 | 82.2 | 85.1 | 43.5 | 45.3 | 46.2 | 52.9 | 57.7 |
| MNFS w/o peripheral points | 72.6 | 78.5 | 81.9 | 83.8 | 86.5 | 45.0 | 47.3 | 53.5 | 56.1 | 60.1 |
| | 0.7↑ | 3.3↑ | 5.9↑ | 1.6↑ | 1.4↑ | 1.5↑ | 2.0↑ | 7.3↑ | 3.2↑ | 2.4↑ |
| MNFS | 73.2 | 79.1 | 83.3 | 85.6 | 87.2 | 46.2 | 48.9 | 56.1 | 58.5 | 61.3 |
| | 1.3↑ | 3.9↑ | 7.3↑ | 3.4↑ | 2.1↑ | 2.7↑ | 3.6↑ | 9.9↑ | 5.6↑ | 3.6↑ |
| MCS | 72.7 | 77.7 | 79.6 | 84.9 | 86.3 | 45.2 | 46.1 | 48.2 | 54.8 | 60.1 |
| | 0.8↑ | 2.5↑ | 3.6↑ | 2.7↑ | 1.2↑ | 1.7↑ | 0.8↑ | 2.0↑ | 1.9↑ | 2.4↑ |
| **DMS** | **73.6** | **80.8** | **85.5** | **87.3** | **88.1** | **47.8** | **51.6** | **58.2** | **61.5** | **63.8** |
| | **1.7↑** | **5.6↑** | **9.5↑** | **5.1↑** | **3.0↑** | **4.3↑** | **6.3↑** | **12.0↑** | **8.6↑** | **6.1↑** |
| Registration Recall (%) RR | | | | | | | | | | |
| GeoTransformer | 92.0 | 91.8 | 91.8 | 91.4 | 91.2 | 75.0 | 74.8 | 72.2 | 74.1 | 73.5 |
| MNFS w/o peripheral points | 93.7 | 93.1 | 92.6 | 92.8 | 92.7 | 78.2 | 77.6 | 77.1 | 77.4 | 76.0 |
| | 1.7↑ | 1.3↑ | 0.8↑ | 1.4↑ | 1.5↑ | 3.2↑ | 2.8↑ | 4.9↑ | 3.3↑ | 2.5↑ |
| MNFS | 94.4 | 93.7 | 93.3 | 93.7 | 93.5 | 80.7 | 79.8 | 79.6 | 79.4 | 78.1 |
| | 2.4↑ | 1.9↑ | 1.5↑ | 2.3↑ | 2.3↑ | 5.7↑ | 5.0↑ | 7.4↑ | 5.3↑ | 4.6↑ |
| MCS | 93.4 | 92.6 | 92.3 | 92.5 | 92.5 | 77.5 | 77.2 | 76.7 | 76.5 | 74.8 |
| | 1.4↑ | 0.8↑ | 0.5↑ | 1.1↑ | 1.3↑ | 2.5↑ | 2.4↑ | 4.5↑ | 2.4↑ | 1.3↑ |
| **DMS** | **95.2** | **94.3** | **94.1** | **95.0** | **94.6** | **82.3** | **81.8** | **81.5** | **81.2** | **80.7** |
| | **3.2↑** | **2.5↑** | **2.3↑** | **3.6↑** | **3.4↑** | **7.3↑** | **7.0↑** | **9.3↑** | **7.1↑** | **7.2↑** |

Additionally, we conducted experiments specifically to assess the impact of peripheral points on registration accuracy. This involved removing peripheral points from overlapping areas within the MNFS framework. The results, as detailed in the Table 9 and Table 10, reveal that the MNFS variant without peripheral points (labeled 'MNFS w/o peripheral points') consistently underperforms compared to the full MNFS method, particularly in scenarios with low overlap. For example, in the 3DLoMatch dataset at the 5000 sample size, MNFS w/o peripheral points registers a FMR of 89.3%, slightly lower than MNFS at 90.8%. Similarly, the IR and RR metrics also reflect lower scores when peripheral points are excluded, particularly in low-overlap scenarios. These findings underscore the critical role that peripheral points play in enhancing registration performance, their presence can significantly enhance

TABLE 10
Performance boosting on ModelNet and ModelLoNe with MNFS w/o peripheral points, MNFS, MCS, and DMS. Boldfaced numbers hightlight the best.

| Model | ModelNet | | | ModelLoNet | | |
|---|---|---|---|---|---|---|
| | RRE | RTE | CD | RRE | RTE | CD |
| Small Rotation | | | | | | |
| GeoTransformer | 2.160 | 0.024 | 0.00143 | 3.638 | 0.064 | 0.00448 |
| MNFS w/o peripheral points | 1.753 | 0.022 | 0.00121 | 3.186 | 0.049 | 0.00385 |
| | 0.407↑ | 0.002↑ | 0.00022↑ | 0.452↑ | 0.015↑ | 0.00063↑ |
| MNFS | 1.526 | 0.019 | 0.00103 | 2.863 | 0.043 | 0.00359 |
| | 0.634↑ | 0.005↑ | 0.00040↑ | 0.775↑ | 0.021↑ | 0.00089↑ |
| MCS | 1.723 | 0.021 | 0.00117 | 3.172 | 0.054 | 0.00396 |
| | 0.437↑ | 0.003↑ | 0.00026↑ | 0.466↑ | 0.010↑ | 0.00052↑ |
| **DMS** | **1.319** | **0.015** | **0.00086** | **2.266** | **0.030** | **0.00315** |
| | **0.841↑** | **0.009↑** | **0.00057↑** | **1.372↑** | **0.034↑** | **0.00133↑** |
| Large Rotation | | | | | | |
| GeoTransformer | 6.436 | 0.047 | 0.00154 | 23.478 | 0.152 | 0.01296 |
| MNFS w/o peripheral points | 6.184 | 0.046 | 0.00131 | 23.039 | 0.149 | 0.01192 |
| | 0.252↑ | 0.001↑ | 0.00023↑ | 0.439↑ | 0.003↑ | 0.00104↑ |
| MNFS | 5.882 | 0.045 | 0.00121 | 22.805 | 0.145 | 0.01085 |
| | 0.554↑ | 0.002↑ | 0.00033↑ | 0.673↑ | 0.007↑ | 0.00211↑ |
| MCS | 6.081 | 0.046 | 0.00135 | 23.061 | 0.148 | 0.01149 |
| | 0.355↑ | 0.001↑ | 0.00019↑ | 0.417↑ | 0.004↑ | 0.00147↑ |
| **DMS** | **5.538** | **0.043** | **0.00103** | **22.365** | **0.141** | **0.00963** |
| | **0.898↑** | **0.004↑** | **0.00051↑** | **1.113↑** | **0.011↑** | **0.00333↑** |

the accuracy and reliability of feature matching, especially in challenging low-overlap conditions.

**Evaluation of DMS in Other Method Frameworks.** Assessments were conducted using several state-of-the-art deep learning methods, namely SpinNet [41], Predator [35], and CoFiNet [36], all integrated with DMS. Each method underwent testing with varying numbers of samples, representing the number of sampling points or corresponding points. The results are shown in Table 11. It is important to note that DMS and its constituents, MNFS and MCS, demonstrated significant improvements in registration recall across all the tested methods on both the 3DMatch and

TABLE 11
Performance boosting for other method frameworks when combined with MNFS, MCS, and DMS. Boldfaced numbers hightlight the best.

| Dataset | 3DMatch RR(%) | | | | | 3DLoMatch RR(%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Samples | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 |
| SpinNet | 88.6 | 86.6 | 85.5 | 83.5 | 70.2 | 59.8 | 54.9 | 48.3 | 39.8 | 26.8 |
| SpinNet+MNFS | 91.2 | 89.7 | 89.3 | 88.2 | 77.8 | 62.4 | 59.4 | 56.3 | 49.1 | 38.2 |
| | 2.6↑ | 3.1↑ | 3.8↑ | 4.7↑ | 7.6↑ | 2.6↑ | 4.5↑ | 8.0↑ | 9.3↑ | 11.4↑ |
| SpinNet+MCS | 90.4 | 88.6 | 88.4 | 86.8 | 75.3 | 61.2 | 58.3 | 53.4 | 45.5 | 33.0 |
| | 1.8 ↑ | 2.0↑ | 2.9↑ | 3.3↑ | 5.1↑ | 1.4↑ | 3.4↑ | 5.1↑ | 5.7↑ | 6.2↑ |
| **SpinNet+DMS** | **94.1** | **92.5** | **92.2** | **91.3** | **82.8** | **65.2** | **63.2** | **60.6** | **54.8** | **43.7** |
| | **5.5↑** | **5.9↑** | **6.7↑** | **7.8↑** | **12.6↑** | **5.4↑** | **8.3↑** | **12.3↑** | **15.0↑** | **16.9↑** |
| Predator | 89.0 | 89.9 | 90.6 | 88.5 | 86.6 | 59.8 | 61.2 | 62.4 | 60.8 | 58.1 |
| Predator+MNFS | 92.6 | 93.0 | 93.6 | 91.8 | 89.3 | 66.3 | 67.9 | 68.6 | 66.4 | 64.2 |
| | 3.6↑ | 3.1↑ | 3.0↑ | 3.3↑ | 2.7↑ | 6.5↑ | 6.7↑ | 6.2↑ | 5.6↑ | 6.1↑ |
| Predator+MCS | 91.2 | 91.2 | 92.4 | 90.3 | 88.1 | 64.2 | 65.4 | 66.1 | 64.8 | 62.8 |
| | 2.2↑ | 1.3↑ | 1.8↑ | 1.8↑ | 1.5↑ | 4.4↑ | 4.2↑ | 3.7↑ | 4.0↑ | 4.7↑ |
| **Predator+DMS** | **93.4** | **94.1** | **94.6** | **93.2** | **91.0** | **68.3** | **70.3** | **72.8** | **70.6** | **68.1** |
| | **4.4↑** | **4.2↑** | **4.0↑** | **4.7↑** | **4.4↑** | **8.5↑** | **9.1↑** | **10.4↑** | **9.8↑** | **10.0↑** |
| CoFiNet | 89.3 | 88.9 | 88.4 | 87.4 | 87 | 67.5 | 66.2 | 64.2 | 63.1 | 61.0 |
| CoFiNet+MNFS | 92.6 | 92.0 | 91.4 | 91.3 | 90.2 | 73.3 | 72.3 | 70.5 | 69.7 | 68.1 |
| | 3.3↑ | 3.1↑ | 3.0↑ | 3.9↑ | 3.2↑ | 5.8↑ | 6.1↑ | 6.3↑ | 6.6↑ | 7.1↑ |
| CoFiNet+MCS | 91.4 | 91.1 | 90.0 | 90.5 | 89.4 | 70.7 | 69.9 | 68.8 | 68.2 | 66.8 |
| | 2.1↑ | 2.2↑ | 1.6↑ | 3.1↑ | 2.4↑ | 3.2↑ | 3.7↑ | 4.6↑ | 5.1↑ | 5.8↑ |
| **CoFiNet+DMS** | **94.8** | **94.0** | **93.8** | **93.1** | **92.7** | **76.2** | **75.9** | **74.7** | **73.2** | **72.5** |
| | **5.5↑** | **5.1↑** | **5.4↑** | **5.7↑** | **5.7↑** | **8.7↑** | **9.7↑** | **10.5↑** | **10.1↑** | **11.5↑** |

3DLoMatch datasets. Based on the data presented in Table 11, it is evident that both DMS and MNFS exhibit an improvement effect of over 5% on the 3DMatch dataset, while the enhancement on the 3DLoMatch dataset is nearly twice as much as that on 3DMatch. This finding once again validates that DMS is better suited for cloud registration with low overlapping points. Furthermore, it can be observed that when the number of sampling points reaches the extreme value of 250, all the proposed methods, including DMS, MNFS, and MCS, show the most substantial improvements. Notably, when SpinNet is tested with 250 sampling points on the 3DMLoMatch dataset, the application of DMS leads to a remarkable increase of 16.9%.

**Evaluate the effectiveness of multi-scale.** We evaluated the effectiveness of our multi-scale design in MNFS and MCS components on 3DMatch and 3DLoMatch, and the results of the experiments are shown in Table 12. It shows the registration recall percentage for various configurations of $h$ values in MCS and MNFS. Configurations range from no MNFS or MCS components (null), to a single scale, and to combinations of multiple scales. It can be seen from the data in the Table 12 that the multi-scale design greatly improves the registration recall rate. Specifically, when MCS is [3,5,8] and MNFS is [2,4,8,16] obtains the highest recall rate of 95.2% on 3DMatch, and is the same as MCS on 3DLoMatch when MNFS is [2, 4,8] scale tied for the highest recall rate of 82.3%, which highlights the advantages of comprehensive multi-scale configuration. Furthermore, the data in the Table 12 shows that recall increases as more scales are included, suggesting that wider scales enable the model to capture a more diverse set of features, thereby improving registration accuracy.

TABLE 12
Evaluate the effectiveness of our multi-scale design in both the MNFS and MCS components. Boldfaced numbers hightlight the best.

| $h$ of MCS | null | [3] | [5] | [8] | [3,5] | [3,5,8] |
|---|---|---|---|---|---|---|
| $h$ of MNFS | Registration Recall (%) of 3DMatch | | | | | |
| null | 92.0 | 92.4 | 92.5 | 92.4 | 92.9 | 93.4 |
| [2] | 89.9 | 89.8 | 90.7 | 91.4 | 92.1 | 92.8 |
| [4] | 88.8 | 89.5 | 90.5 | 90.8 | 91.8 | 92.3 |
| [8] | 89.2 | 89.7 | 90.6 | 90.4 | 91.4 | 92.1 |
| [2,4] | 92.0 | 92.3 | 92.6 | 92.8 | 93.3 | 93.7 |
| [2,4,8] | 93.5 | 93.3 | 93.7 | 93.5 | 93.9 | 94.2 |
| [2,4,8,16] | 94.4 | 94.6 | 94.7 | 94.6 | _94.9_ | **95.2** |
| $h$ of MNFS | Registration Recall (%) of 3DLoMatch | | | | | |
| null | 75.0 | 75.4 | 75.7 | 76.2 | 76.8 | 77.5 |
| [2] | 74.7 | 74.5 | 74.9 | 75.6 | 76.0 | 76.7 |
| [4] | 73.1 | 73.4 | 74.6 | 74.7 | 75.1 | 75.3 |
| [8] | 73.7 | 74 | 74.2 | 74.2 | 75.1 | 75.8 |
| [2,4] | 75.5 | 76.3 | 76.8 | 76.8 | 78.3 | 79.5 |
| [2,4,8] | 78.2 | 78.9 | 79.4 | 78.8 | 81.1 | **82.3** |
| [2,4,8,16] | 80.7 | 81.0 | 81.3 | 81.2 | _81.7_ | **82.3** |

In addition, Table 12 further proves the direct effectiveness of peripheral points in improving registration accuracy. Table 12 shows that as the value of $h$ increases, that is, as the multi-scale range expands, the registration recall rate increases significantly. Especially under the most complex configuration [2, 4, 8, 16], the recall rate reaches the highest, which shows that the peripheral points contained at different scales provide rich information for the registration process and help the model to better capture the correspondence between point clouds. By considering peripheral points at multi-scales, our method not only improves performance in 3DMatch but also shows superiority in more challenging low-overlap scenarios 3DLoMatch adaptability and accuracy. This is

mainly due to the fact that multi-scale design effectively utilizes peripheral points that may show higher similarity at smaller scales by aggregating information at different scales without affecting the similarity of internal points at other scales. This approach ensures that peripheral points can be accurately identified and utilized even when the overlapping areas are small, thereby improving the overall registration performance.

Overall, the data in this Table 12 highlight the effectiveness of multi-scale designs in including and exploiting peripheral points, and the direct contribution of these peripheral points to improving the accuracy and robustness of point cloud registration methods. Especially in low-overlap scenarios (3DLoMatch), the recall rate is significantly improved.

**Time consumption of DMS.** We conducted an analysis experiment on the time consumption of DMS. The experiment involved calculating the average time (ms) for processing all point clouds in the 3DMatch dataset. It is important to note that, in order to assess the impact of each parameter on DMS, the parameters in Table 13 were set appropriately, focusing on parameter influence rather than robustness considerations. From Table 13, the following observations can be made: 1) MNFS can be completed within tens of milliseconds; 2) MCS contributes significantly to the time consumption of DMS, but it remains controllable within $400ms$; 3) MCS is influenced by parameters $K$ and $h$, with $K$ having a much higher impact than $h$; 4) MNFS is greatly affected by the parameters $d$ and $h$.

TABLE 13
Average consumed time (ms) on the 3DMatch dataset.

| $h$ | $K$ | $h_{\text{init}}$ | $d$ | MNFS | MCS | DMS |
|---|---|---|---|---|---|---|
| [2, 4, 8, 16] | [256, 128, 64, 32, 16] | [3, 5, 8] | 256 | 35.453 | 363.042 | 398.495 |
| | | | | 26.064 | 370.174 | 396.238 |
| [2, 3, 4, 5] | [64, 32, 16, 8, 4] | | | 18.151 | 103.331 | 121.481 |
| | | [8, 16, 32] | | 18.385 | 101.452 | 119.837 |
| | | | 1024 | 41.961 | 101.676 | 143.637 |

## 4.4 Visual Experiment

We show the visualization of 3D rigid registration, as shown in Fig. 7. Note that to better visualize the quality of registration, simultaneous rotations were applied to both registration results and ground truth for enhanced visual clarity. These modifications do not impact the core objective of our methodology, which remains the precise alignment of the source point cloud to the target point cloud. This adjustment, made purely for visualization purposes, may lead to perceived discrepancies when directly compared with the input configurations, but it does not change the actual alignment of the source to the target point cloud.

It can be seen that the DMS method consistently produces results that closely match the Ground Truth. This is especially noticeable in complex datasets like 3DLoMatch and KITTI. For instance, in the 3DLoMatch dataset, DMS appears to maintain the structural relationships of the chairs and tables better than other methods. Similarly in the 3DMatch and ModelNet dataset. This robust performance is likely attributed to the unique design of the DMS, which incorporates a double-layer multi-scale star-graph to optimize the registration process, effectively handling the challenges of low-overlap scenarios where traditional methods might struggle. Additionally, the multi-scale approach of DMS
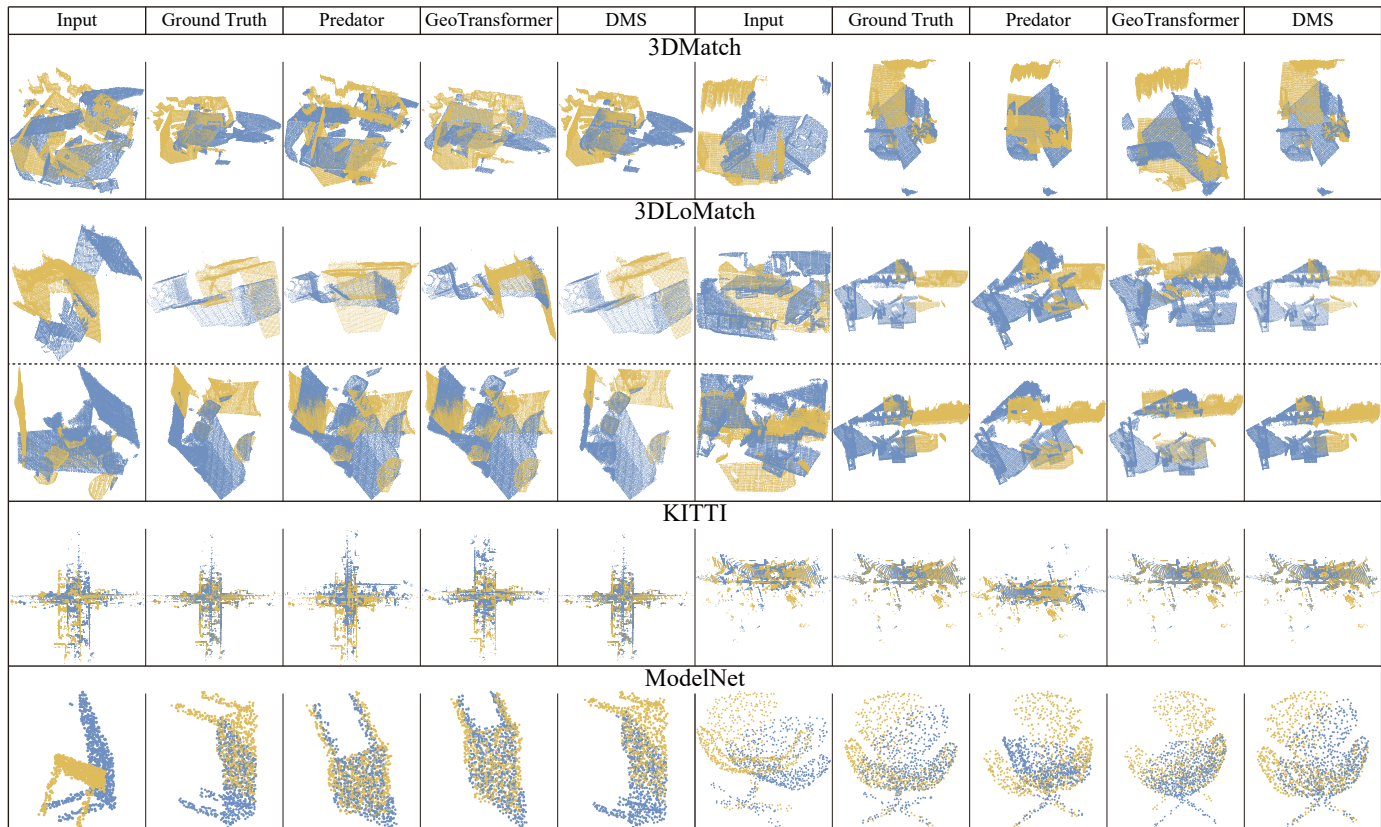
Fig. 7. Comprehensive visualization evaluation of different point cloud registration techniques across multiple datasets: 3DMatch, 3DLoMatch, KITTI, and ModelNet. Each row represents a dataset and each column corresponds to a step or method in the registration process: Input point clouds, Ground Truth alignment, and the results from Predator [35], GeoTransformer [37], and DMS. The source point clouds $\mathcal{P}$ are depicted in yellow, and the target point clouds $\mathcal{Q}$ in blue.

facilitates a more detailed understanding of the local geometries, which is critical for accurate registration of 3D point clouds.

Fig. 8 presents a visual comparison of point cloud registration outcomes using different methods on the 4DMatch and 4DLoMatch dataset. The first column displays the input point clouds with varying degrees of overlap, labeled with the specific model and the percentage of overlap. Subsequent columns illustrate the alignment results achieved by Lepard, GeoTransformer, and DMS respectively. In each model, the point clouds are color-coded to distinguish between the source and target datasets, with yellow representing the source and blue the target. Green lines indicate correct correspondences, while red lines indicate incorrect ones. Evidently, DMS, as shown in the final column, effectively aligns the point clouds across all tested scenarios, including those with lower overlap percentages. The results indicate that DMS maintains structure integrity and alignment accuracy, comparable to or surpassing the other methods, showcasing its proficiency in non-rigid point cloud registration tasks.

## 5 CONCLUSION

This study presented a robust LoR methodology founded upon the DMS framework. Following the coarse-to-fine strategy, we initially extract super-points and their respective features. Subsequently, we proposed the MNFS to establish multi-scale star graphs for these super-points while simultaneously fusing their features. By leveraging the neighborhoods and features of MNFS, we performed consistent super-point matching, yielding the initial



Fig. 8. Comparative Visualization of Non-Rigid PCR Using Lepard [57], GeoTransformer [60], and DMS on the 4DMatch and 4DLoMatch Benchmark.

correspondence set. Subsequently, we introduced the construction of MMS based on these initial correspondences. We then identified the correspondence set $M_{c_z}$ by leveraging the information within

MMS. Building upon $M_{c_z}$, we proceeded to construct MCS. Within the MCS framework, we introduced an effective mismatch elimination strategy, ultimately resulting in the establishment of a robust super-point correspondence. Finally, the points corresponding to each pair of super points are registered to restore the optimal transformation matrix. The results of the experiments achieved state-of-the-art results.

## ACKNOWLEDGMENT

## REFERENCES

[1] Y. Wu, J. Li, Y. Yuan, A. K. Qin, Q.-G. Miao, and M.-G. Gong, "Commonality autoencoder: Learning common features for change detection from heterogeneous images," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4257–4270, 2022.

[2] H. Cao, H. Wang, N. Zhang, Y. Yang, and Z. Zhou, "Robust probability model based on variational bayes for point set registration," *Knowledge-Based Systems*, vol. 241, p. 108182, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122000387

[3] Q. Mérigot, M. Ovsjanikov, and L. J. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 6, pp. 743–756, 2011.

[4] W. Zhang, H. Zhou, Z. Dong, Q. Yan, and C. Xiao, "Rank-pointretrieval: Reranking point cloud retrieval via a visually consistent registration evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 9, pp. 3840–3854, 2023.

[5] H. Wang, H. Cao, Y. Kai, H. Bai, X. Chen, Y. Yang, L. Xing, and C. Zhou, "Multi-source remote sensing intelligent characterization technique-based disaster regions detection in high-altitude mountain forest areas," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[6] A. Bletterer, F. Payan, and M. Antonini, "A local graph-based structure for processing gigantic aggregated 3d point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 8, pp. 2822–2833, 2022.

[7] Y. Wu, X. Hu, Y. Zhang, M. Gong, W. Ma, and Q. Miao, "Sacf-net: Skip-attention based correspondence filtering network for point cloud registration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3585–3595, 2023.

[8] Q. Liu, H. Su, Z. Duanmu, W. Liu, and Z. Wang, "Perceptual quality assessment of colored 3d point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 8, pp. 3642–3655, 2023.

[9] H. Cao, Q. He, H. Wang, Z. Xiong, N. Zhang, and Y. Yang, "An estimation of distribution algorithm based on variational bayesian for point-set registration," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 926–940, 2022.

[10] P. Ondrúška, P. Kohli, and S. Izadi, "Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1251–1258, 2015.

[11] Y. Wu, J. Liu, M. Gong, P. Gong, X. Fan, A. K. Qin, Q. Miao, and W. Ma, "Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding," *IEEE Transactions on Multimedia*, vol. 26, pp. 1626–1638, 2024.

[12] M. Arikan, R. Preiner, C. Scheiblauer, S. Jeschke, and M. Wimmer, "Large-scale point-cloud visualization through localized textured surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 9, pp. 1280–1292, 2014.

[13] N. Mellado, M. Dellepiane, and R. Scopigno, "Relative scale estimation and 3d registration of multi-modal geometry using growing least squares," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 9, pp. 2160–2173, 2016.

[14] Y. Liu, Q. Dai, and W. Xu, "A point-cloud-based multiview stereo algorithm for free-viewpoint video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 407–418, 2010.

[15] G. Yuan, Q. Fu, Z. Mi, Y. Luo, and W. Tao, "Ssrnet: Scalable 3d surface reconstruction network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 12, pp. 4906–4919, 2023.

[16] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, P. S. Schenker, Ed., vol. 1611, International Society for Optics and Photonics. SPIE, 1992, pp. 586 – 606. [Online]. Available: https://doi.org/10.1117/12.57955

[17] Y. Wu, P. Gong, M. Gong, H. Ding, Z. Tang, Y. Liu, W. Ma, and Q. Miao, "Evolutionary multitasking with solution space cutting for point cloud registration," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 110–125, 2024.

[18] L. Li, H. Fu, and M. Ovsjanikov, "Wsdesc: Weakly supervised 3d local descriptor learning for point cloud registration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 7, pp. 3368–3379, 2023.

[19] B. Tan, H. Qin, X. Zhang, Y. Wang, T. Xiang, and B. Chen, "Using multi-level consistency learning for partial-to-partial point cloud registration," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2023.

[20] X. Li, L. Wang, and Y. Fang, "Unsupervised category-specific partial point set registration via joint shape completion and registration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 7, pp. 3251–3265, 2023.

[21] Y. Wu, J. Liu, Y. Yuan, X. Hu, X. Fan, K. Tu, M. Gong, Q. Miao, and W. Ma, "Correspondence-free point cloud registration via feature interaction and dual branch [application notes]," *IEEE Computational Intelligence Magazine*, vol. 18, no. 4, pp. 66–79, 2023.

[22] A. Segal, D. Hähnel, and S. Thrun, "Generalized-icp," *Robotics: Science and Systems*, 06 2009.

[23] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2241–2254, 2016.

[24] Y. Liu, L. De Dominicis, B. Wei, L. Chen, and R. R. Martin, "Regularization based iterative point match weighting for accurate rigid transformation estimation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 9, pp. 1058–1071, 2015.

[25] M. Zhao, X. Huang, J. Jiang, L. Mou, D.-M. Yan, and L. Ma, "Accurate registration of cross-modality geometry via consistent clustering," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–12, 2023.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Association for Computing Machinery*, vol. 24, no. 6, 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[27] D. Barath and J. Matas, "Graph-cut ransac," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] S. Quan and J. Yang, "Compatibility-guided sampling consensus for 3-d point cloud registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7380–7392, 2020.

[29] J. Fan, J. Yang, Y. Zhao, D. Ai, Y. Liu, G. Wang, and Y. Wang, "Convex hull aided registration method (charm)," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2042–2055, 2017.

[30] Z.-J. Du, S.-S. Huang, T.-J. Mu, Q. Zhao, R. R. Martin, and K. Xu, "Accurate dynamic slam using crf-based long-term consistency," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 4, pp. 1745–1757, 2022.

[31] G. Mei, H. Tang, X. Huang, W. Wang, J. Liu, J. Zhang, L. Van Gool, and Q. Wu, "Unsupervised deep probabilistic approach for partial point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 611–13 620.

[32] S. Ao, Q. Hu, H. Wang, K. Xu, and Y. Guo, "Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 1255–1264.

[33] Z. Yan, Z. Yi, R. Hu, N. J. Mitra, D. Cohen-Or, and H. Huang, "Consistent two-flow network for tele-registration of point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4304–4318, 2022.

[34] Y. Wu, J. Liu, M. Gong, Z. Liu, Q. Miao, and W. Ma, "Mpct: Multiscale point cloud transformer with a residual network," *IEEE Transactions on Multimedia*, vol. 26, pp. 3505–3516, 2024.

[35] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4267–4276.

[36] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 23 872–23 884. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/c85b2ea9a678e74fdc8bafe5d0707c31-Paper.pdf

[37] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 143–11 152.

[38] J. Yu, L. Ren, Y. Zhang, W. Zhou, L. Lin, and G. Dai, "Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 702–17 711.

[39] Y. Wu, Y. Zhang, W. Ma, M. Gong, X. Fan, M. Zhang, A. K. Qin, and Q. Miao, "Rornet: Partial-to-partial registration network with reliable overlapping representations," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.

[40] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3d registration with maximal cliques," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 745–17 754.

[41] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 753–11 762.

[42] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[43] 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 Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[44] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[45] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[46] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[47] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[48] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, and C.-L. Tai, "Pointdsc: Robust point cloud registration using deep spatial consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 859–15 869.

[49] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8893–8902.

[50] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[51] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[52] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8922–8931.

[53] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[54] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[55] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[56] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[57] Y. Li and T. Harada, "Lepard: Learning partial point cloud matching in rigid and deformable scenes." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[58] T. T. B. Z. Yang Li, Hikari Takehara and M. Nießner, "4dcomplete: Non-rigid motion estimation beyond the observable surface." *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[59] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[60] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, S. Ilic, D. Hu, and K. Xu, "Geotransformer: Fast and robust point cloud registration with geometric transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 9806–9821, 2023.

[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[62] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[63] H. Wang, Y. Liu, Z. Dong, and W. Wang, "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors," in *Proceedings of the 30th ACM International Conference on Multimedia*, ser. MM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1630–1641. [Online]. Available: https://doi.org/10.1145/3503161.3548023

**Hualong Cao** received his B.S. degree from Suzhou University in 2019 and his master's degree from Yunnan Normal University in 2022. Currently, he is pursuing a Ph.D. at the School of Information, Renmin University of China. His current research interests are mainly computer vision, SLAM and robotics.

**Yongcai Wang** (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Automation Sciences and Engineering, Tsinghua University, in 2001 and 2006, respectively. From 2007 to 2009, he was an Associate Researcher with NEC Labs., China. From 2009 to 2015, he was a Research Scientist with the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University. In 2015, he was a Visiting Scholar with Cornell University. He is currently an Associate Professor with the Department of Computer Sciences, Renmin University of China. His research interests include network localization, and combinatorial optimization and applications.

**Deying Li** received the M.S. degree in mathematics from Huazhong Normal University in 1988 and the Ph.D. degree in computer science from the City University of Hong Kong in 2004. She is currently a Professor with the Department of Computer Science, Renmin University of China. Her research interests include wireless networks, mobile computing, and algorithm design and analysis.