PointMC: Multi-instance Point Cloud Registration based on Maximal Cliques

Yue Wu [1 2] **Xidao Hu** [1 2] **Yongzhe Yuan** [1 2] **Xiaolong Fan** [1 3] **Maoguo Gong** [1 3] **Hao Li** [1 3] **Mingyang Zhang** [1 3] **Qiguang Miao** [1 2] **Wenping Ma** [4]

Abstract

Multi-instance point cloud registration is the problem of estimating multiple rigid transformations between two point clouds. Existing solutions rely on global spatial consistency of ambiguity and the time-consuming clustering of high-dimensional correspondence features, making it difficult to handle registration scenarios where multiple instances overlap. To address these problems, we propose a maximal clique based multi-instance point cloud registration framework called PointMC. The key idea is to search for maximal cliques on the correspondence compatibility graph to estimate multiple transformations, and cluster these transformations into clusters corresponding to different instances to efficiently and accurately estimate all poses. PointMC leverages a correspondence embedding module that relies on local spatial consistency to effectively eliminate outliers, and the extracted discriminative features empower the network to circumvent missed pose detection in scenarios involving multiple overlapping instances. We conduct comprehensive experiments on both synthetic and real-world datasets, and the results show that the proposed PointMC yields remarkable performance improvements.

1. Introduction

With the development of high-precision sensors, pairwise point cloud registration techniques have been widely applied in fields such as autonomous driving (Zhang et al., 2023b; Zhao et al., 2024; Li et al., 2015) and 3D reconstruction (Cheng et al., 2023; Yu et al., 2023b; Merras et al., 2017). The task involves estimating a single rigid transformation between two frames of point clouds. However, due to the possibility that the target scene point cloud may contain multiple instances of the same source point cloud, we need to estimate multiple transformations between them. As shown in Figure 1 (right), in the case where there is a chair modeled as the source point cloud, we aim to find the poses of the same object in the target indoor scene point cloud. The existing literature has relatively little research on this challenging problem, which is referred to as multi-instance point cloud registration.

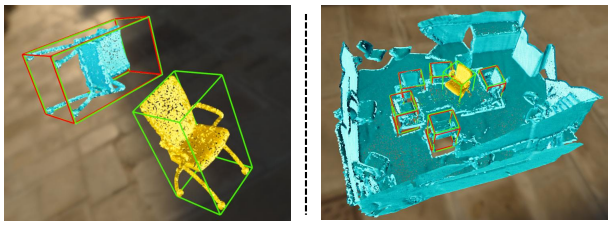

Figure 1. Given two frame point clouds, pairwise point cloud registration (left) focuses on estimating a single rigid transformation between the source and target point clouds, while multi-instance point cloud registration (right) aims to estimate multiple poses of objects identical to the source point cloud within the target point cloud.

For the problem of multi-instance point cloud registration, due to its similarity to 3D object detection tasks (Li et al., 2023; Zhou et al., 2023), one solution is to utilize a 3D object detector to detect and segment instances within the target point cloud, and then transform it into a traditional pairwise point cloud registration problem. However, since the object detector is trained for a specific object, it is less robust when encountering point clouds of unknown objects. Another solution relies on multi-model fitting algorithms (Kluger et al., 2020; Magri & Fusiello, 2016; 2014). Nevertheless, traditional multi-model fitting algorithms generate significant computational costs in scenarios with a large number of outliers in multi-instance point cloud registration, as they require a large number of samples to generate

[1]MoE Key Lab of Collaborative Intelligence Systems, Xidian University, Xi'an, China [2]School of Computer Science and Technology, Xidian University, Xi'an, China [3]School of Electronic Engineering, Xidian University, Xi'an, China [4]School of Artificial Intelligence, Xidian University, Xi'an, China. Correspondence to: Maoguo Gong <gong@ieee.org>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

hypotheses. Recently, there have been several deep learning-based approaches (Tang & Zou, 2022; Yuan et al., 2022) that leverage global spatial consistency and spectral clustering algorithms to extract high-dimensional corresponding features and perform clustering. Despite this, these methods encounter two critical issues. First, due to the ambiguity of global spatial consistency, they struggle to effectively differentiate multiple overlapping instances in multi-instance registration scenarios. Second, the utilization of spectral clustering algorithms incurs high computational costs when clustering high-dimensional features, leading to longer registration times.

There is an inspiring recent work (Zhang et al., 2023a) on pairwise point cloud registration using maximal cliques. The key idea is to relax the previous maximum clique constraint and mine more local consensus information within the graph to accurately generate pose hypotheses. Since the process generates numerous pose hypotheses that can be represented by low-dimensional vectors, we contemplate whether it is possible to perform clustering on these low-dimensional pose hypotheses and then directly select the corresponding transformation for each instance. Compared to clustering and iteratively solving transformations on high-dimensional correspondence features, this method will undoubtedly be more efficient. Meanwhile, we consider utilizing local spatial consistency instead of global spatial consistency to extract correspondence features with rich local information in order to better distinguish overlapping instances.

In this paper, we propose a multi-instance point cloud registration framework PointMC based on maximal cliques. The key idea is to estimate multiple transformations by searching maximal cliques on the correspondence compatible graph, and then cluster the transformations into in different clusters. Specifically, we first utilize a graph-based correspondence embedding module to extract local spatial consistency aware features of putative correspondences, and use them to distinguish inliers and outliers. Since this method has strong local spatial consistency, it can effectively remove outliers in putative correspondences while allowing their features to retain as much local information as possible. We then model the filtered correspondence set as a compatibility graph, search for maximal cliques representing a single instance consensus set in the graph, and then use the SVD algorithm (Sorkine-Hornung & Rabinovich, 2017) to perform transformation assumptions on all cliques to obtain the transformation set. Each node in the graph represents a single correspondence, and each edge between two nodes represents a pair of compatible correspondences. Finally, the transformations in the transformation set are clustered into different groups. Each group corresponds to the transformation pose of an instance, and the transformation that causes the associated correspondences to have the smallest transformation error is selected from the group as the final transformation of the instance. Our main contributions can be summarized as follows:

- We introduce a graph-based method for local spatial consistency to measure the geometric compatibility between correspondences within local regions, aiming to enhance the filtering of outlier correspondences.
- We propose to search for maximal cliques on the correspondence compatibility graph to estimate multiple transformations, and cluster these transformations into clusters corresponding to different instances to precisely estimate all poses.
- We provide qualitative and quantitative comparisons under synthetic and real-world datasets, showing the state-of-the-art performance.

2. Related Work

2.1. Pairwise Point Cloud Registration

Pairwise point cloud registration can be decomposed into three subtasks: point matching, outlier rejection, and transformation estimation. Traditional point matching methods typically rely on hand-crafted descriptors (Rusu et al., 2009; Ma et al., 2019; Drost et al., 2010) that capture local information, but they tend to lack robustness against noise and outliers. Recent works(Cao et al., 2021; Yew & Lee, 2022; Liu et al., 2023; Yu et al., 2023a; Yuan et al., 2022; Yang et al., 2022; Yuan et al., 2024) have embraced the utilization of deep networks for feature learning, leveraging these learned features to establish correspondences using various methodologies. PCAM (Cao et al., 2021) multiplies cross-attention matrices in multiple levels in the encoder to establish initial point matching. REGTR (Yew & Lee, 2022) establishes correspondences using a network architecture consisting of transformer layers with self and cross attention. RegFormer (Liu et al., 2023) proposed a Bijective Association Transformer (BAT) to address significant mismatches caused by potential descriptor errors. Attaining perfect matches is challenging, and a robust outlier rejection mechanism is essential. RANSAC (Fischler & Bolles, 1981) and its variants (Le et al., 2019; Barath & Matas, 2018) are widely regarded as the most popular traditional outlier rejection methods. SACF-Net (Wu et al., 2023), Predator (Huang et al., 2021), and DGR (Choy et al., 2020) treat outlier rejection as a binary classification task and output a confidence score for each correspondence. MAC(Zhang et al., 2023a) conducted correspondence screening by constructing a maximal clique based on the spatial consistency among correspondences. Correspondence-based methods (Wu et al., 2023; Huang et al., 2021; Choy et al., 2020) commonly employ a differentiable weighted procrustes method

(Arun et al., 1987) based on SVD (Sorkine-Hornung & Rabinovich, 2017) to obtain the final transformation. Several end-to-end models (Aoki et al., 2019; Wang & Solomon, 2019; Yew & Lee, 2020) seamlessly incorporate the complete transformation estimate into the training pipeline.

2.2. Multi-instance Point Cloud Registration

In contrast to pairwise registration, which estimates a single transformation between two frame point clouds, multi-instance registration involves estimating multiple transformations for the source point cloud and multiple instances within the target point cloud. Multi-instance registration requires not only filtering outliers in noisy correspondences, but also clustering the remaining correspondences into individual instances. The current methods (Yuan et al., 2022; Tang & Zou, 2022) primarily focus on conducting correspondence clustering using deep representations of correspondence, followed by iterative estimation of the transformation for each individual instance. ECC (Tang & Zou, 2022) utilizes the global spatial consistency (Leordeanu & Hebert, 2005) of the point cloud rigid transformation to directly group the noise correspondence sets into different clusters based on the distance invariance matrix. However, the reliability of the distance invariant matrix is compromised in scenarios involving dense noisy correspondences caused by the presence of multiple instances, particularly when outliers closely resemble inliers. In addition to utilizing global spatial consistency, PointCLM (Yuan et al., 2022) also obtains discriminative high-dimensional corresponding representations based on contrastive learning. After a specific pruning strategy, the spectral clustering algorithm is used to cluster the high-dimensional corresponding features. Based on experimental results, the learned high-dimensional features provide limited improvement to the overall results, while the process of clustering such features proves to be time-consuming. In this paper, we introduce a multi-instance point cloud registration method based on maximal cliques (Zhang et al., 2023a) to obtain low-dimensional representations of multiple rigid transformations, and obtains the final result through low-dimensional representation clustering with low time consumption.

3. Problem Setting

Consider two point clouds to be registered: X and $Y = Y_0 \cup Y_1 \cup ... \cup Y_K$. Point cloud X consists of a 3D model, and point cloud Y represents a scene containing K instances of the same model ($Y_1 ... Y_K$) as well as some other points (Y_0), where these instances may partially overlap with the 3D model. After obtaining the putative correspondence set $C = \{c_i = (x_i, y_i) \in \mathbb{R}^6\}_{i=1}^{M}$ through point feature matching (Huang et al., 2021; Choy et al., 2019; Thomas et al., 2019), an important step in the previous method was to segment it into different subsets $C_0, C_1, ..., C_k$ satisfying $C = C_0 \cup C_1 \cup ... \cup C_k$, where C_0 denotes the set of outliers and the rest denotes the set corresponding to each instance. The task of multi-instance point cloud registration is to derive the rigid transformations $T = \{T_k = (R_k, t_k)\}_{k=1}^{K}$ that align point cloud X to each instance point cloud Y_k, where $R_k \in SO(3)$ denotes the rotation matrix and $t_k \in \mathbb{R}^3$ denotes the translation vector. The R_k and t_k can be described as

$$R_k, t_k = \underset{(R,t)}{\operatorname{argmin}} \sum_{i=1} \|y_{ki} - (Rx_i + t)\|^2 \tag{1}$$

where x_i and y_{ki} denotes the truth corresponding points between X and Y_k. Due to the presence of numerous abnormal correspondences and the typically unknown number of real instances in the scene point cloud, this is a rather challenging task.

4. Method

The overview of the proposed PointMC is shown in Figure 2. We first take the putative correspondences as the input, use the graph-based correspondence embedding module to extract the features of the putative correspondences (section 4.1), and combine the classification head to distinguish the inliers and outliers (section 4.2). Subsequently, we model the filtered correspondence set as a compatibility graph, search for maximal cliques in the graph, and solve transformations for all cliques to obtain a set of transformations (section 4.3). Finally, we cluster the transformations in the transformation set into different groups and select the poses corresponding to each instance (section 4.4).

4.1. Local Spatial Consistency

The widely utilized property of global spatial consistency (Bai et al., 2021) in pairwise point cloud registration ensures that the distance between each pair of points is preserved under any rigid transformation. Consider two correspondences $c_i = (x_i, y_i)$ and $c_j = (x_j, y_j)$, the global spatial consistency can be computed as

$$\theta_{ij} = \left[1 - \frac{d_{ij}^2}{\sigma_d^2}\right]_+, d_{ij} = |\|x_i - x_j\| - \|y_i - y_j\|| \tag{2}$$

where $[\cdot]_+ = \max(0, \cdot)$ and σ_d is a distance parameter to control the sensitivity to the difference in distance. If c_i and c_j are inliers compatible with the same instance, then θ_{ij} is close to 1. However, if outliers exist, θ_{ij} is close to 0.

However, in the context of multi-instance point cloud registration, the applicability of such global spatial consistency is not pronounced. This is due to the possibility of needing to estimate the poses of multiple overlapping instances in multi-instance registration scenarios, which results in

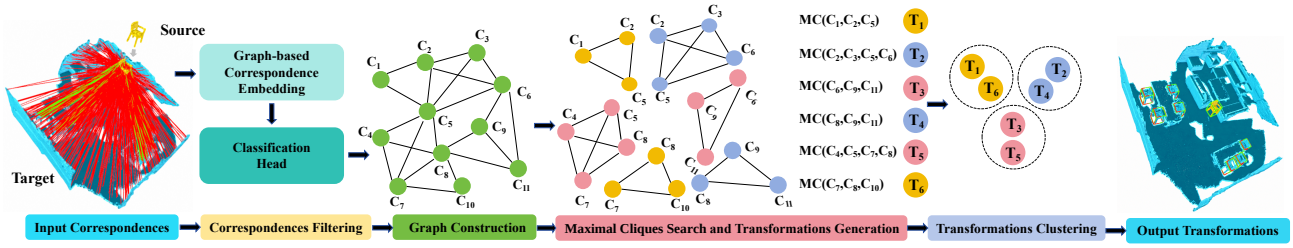

Figure 2. The pipeline of the proposed **PointMC** for multi-instance point cloud registration. It takes putative correspondences as input, and output K rigid transformations. The green lines and red lines represent inliers and outliers, respectively. The green bounding boxes in output transformations represent the ground truth poses of instances in the target point cloud and the red bounding boxes represent our predictions.

a decrease in the reliability of global spatial consistency. Therefore, we adopt a graph-based local spatial consistency method (Qin et al., 2023), which aims to confine correspondences within a single instance as much as possible. It is defined as

$$\omega_{ij} = \begin{cases} \left[1 - \frac{d_{ij}^2}{\sigma_d^2}\right]_+, & c_i \in \mathcal{C} \wedge c_j \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathcal{C}$ denotes the set of correspondences within a local region, which can be acquired through uniform farthest point sampling.

4.2. Correspondences Filtering

Due to the presence of a large number of outliers in the input putative correspondences, it can significantly impact the subsequent pipeline process. Therefore, we apply a filtering and selection process based on local spatial consistency to refine the initial correspondences.

4.2.1. Feature Extractor

Taking M putative correspondences $C = \{c_i = (x_i, y_i) \in \mathbb{R}^6\}_{i=1}^M$ as the input to our pipeline, we first concatenate each correspondence with its low-frequency encoding result to obtain

$$\theta_i = [c_i, sin(2^{-1}c_i), cos(2^{-1}c_i)] \in \mathbb{R}^{18} \tag{4}$$

which allowing for the incorporation of additional local information. Subsequently, a three-layer MLP is employed to project the constructed correspondence matrix $\theta \in \mathbb{R}^{M\times 18}$ into a initial high-dimensional feature matrix $F_{init} \in \mathbb{R}^{M\times d}$, with batch normalization and LeakyReLU applied after each layer of the MLP. Following the concept of local spatial consistency, we adopt a graph-based correspondence embedding module (Qin et al., 2023) to further enhance the discriminability of features. The module consists of a stack of spatial-consistency-aware self-attention (SCASA) module used to refine features based on attention mechanisms.

4.2.2. Correspondences Classification

After obtaining the enhanced feature $F_{out} \in \mathbb{R}^{M\times D}$, a three-layer MLP is used to estimate the confidence score o_i for each correspondence. Apply batch normalization and ReLU after the first two layers, and use sigmoid activation after the last layer. The correspondences with confidence scores higher than the threshold τ are considered inliers, while the remaining correspondences are treated as outliers and removed. We adopt the binary focal loss to supervise the confidence scores, and the loss calculation for each correspondence is as

$$\mathcal{L}_{cls} = \frac{1}{M}\sum_{i=1}^{M} -\log(o_i)\cdot o_i^* - \log(1-o_i)\cdot(1-o_i^*) \tag{5}$$

where

$$o_i^* = \begin{cases} 1, & \|\varphi^*(\mathbf{x}_i) - \mathbf{y}_i\|^2 < \varepsilon^2 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

ε is a threshold that controls the minimum radius at which two points are considered to be corresponding points, and φ^* is the ground-truth transformation between x_i and y_i.

4.3. Search Transformations

We construct a compatibility graph on the obtained clean set of correspondences and employ a method based on searching for maximal cliques (Zhang et al., 2023a) to obtain potential transformations.

4.3.1. Correspondence-Compatible Graph Construction

Graph space accurately describes the compatibility between correspondences, thus we model the filtered set of corre-

spondences as an undirected graph. Nodes on the graph represent correspondences, and the compatibility between nodes can be calculated as

$$G_{ij}\left(c_i, c_j\right) = \left[\exp\left(-\frac{d_{ij}^2}{2d_{thr}^2}\right)\right]_{t_c} \tag{7}$$

where d_{thr} is a distance parameter and $[\phi]_{t_c} = \max(0, \phi - t_c)$. t_c denotes the compatibility threshold, and only if the compatibility between two nodes (c_i, c_j) exceeds t_c, a weighted edge with a value of G_{ij} will be formed between them. After obtaining the symmetric weight matrix M_f of an undirected graph, we further adopt a second-order compatibility measure (Chen et al., 2022), which can be computed as

$$M_s = M_f \cdot (M_f \times M_f) \tag{8}$$

This second-order measure encodes richer information beyond the first-order measurements, thereby enhancing robustness to outliers and promoting sparsity, which helps improve the speed of cliques search.

4.3.2. Maximal Cliques Search and Filtering

Given an undirected graph, a maximal clique is a clique that cannot be extended by adding any node, and the maximal clique with the most nodes is the maximum clique of the graph. A large number of maximal cliques in an undirected graph are associated with multiple instances, while a small number of maximum cliques are likely to be associated with only one instance. Therefore, adopting a maximal cliques search strategy is more suitable for multi-instance point cloud registration tasks.

We use the improved Bron-Kerbosch algorithm (Wei et al., 2021) encapsulated in the igraph C++ library to perform the maximal clique search task. It guarantees completeness in finding all maximal cliques, and its backtracking technique helps optimize the search process. Additionally, the algorithm's recursive nature presents opportunities for parallelization, potentially improving performance in parallel computing environments.

After conducting the maximal clique search, we obtain MAC_{ini}, a set of maximal cliques that typically contains tens of thousands of elements. In order to reduce time consumption, we employ a filtering process to reduce the size of MAC_{ini}. Given a clique $C_j = (V_j, E_j)$, we compute its weight as

$$w_{C_j} = \sum_{e_i \in E_j} w_{e_i} \tag{9}$$

where w_{e_i} denotes the weight of edge e_i in M_s. Since a node may exist in multiple maximal cliques, we enforce it to belong only to the one with the maximum weight, while deleting the remaining maximal cliques that contain the same node. This filtering process ensures that the resulting set of maximal cliques, denoted as MAC_{flt}, contains fewer maximal cliques than the total number of nodes in the graph. In the case of a large number of graph nodes, we can further rank the weights of the maximal cliques in MAC_{flt} and select the top K maximal cliques for subsequent processing.

4.3.3. Transformations Generation

After filtering, each maximal clique contains a set of compatible correspondences. By applying the SVD algorithm (Sorkine-Hornung & Rabinovich, 2017) or the weighted SVD algorithm (Choy et al., 2020) to each set of compatible correspondences, we can obtain a collection of transformations T_{all} composed of 7D or 6D vectors. The first four dimensions of the 7D vectors represent rotation expressed as quaternions, while the first three dimensions of the 6D vectors represent rotation expressed as Euler angles. The last three dimensions of both vectors represent translation. Compared to clustering the high-dimensional correspondence features, clustering the low-dimensional pose vectors is computationally more efficient, and the final transformations can be obtained without the need for iterative optimization.

4.4. Transformations Clustering

Upon obtaining the transformation set, the next step is to partition these transformations into multiple subsets belonging to different instances and select the transformation corresponding to each instance from the subset. The transformation partition can be regarded as a clustering problem, and the number of instances should be equal to the number of clusters.

To conduct our experiments, we individually selected the density-based clustering algorithm DBSCAN (Ester, 1996) and the hierarchical algorithm Chameleon (Karypis et al., 1999). Chameleon exhibits stronger clustering effectiveness compared to DBSCAN, as it utilizes a two-step clustering strategy with a merging strategy to explore more hidden clusters in the data. DBSCAN has simpler parameter settings and lower computational complexity, making it more efficient than Chameleon, especially for large-scale datasets. We will quantitatively compare the performance of these two clustering algorithms.

Given a subset of transformations T_{sub} obtained for a single instance, we select the final transformation T^* corresponding to the instance using the following formula

$$T^* = \underset{T_i \in T_{sub}}{\operatorname{argmin}} \sum_{c_k \in C_{sub}} \|T_i(x_k) - y_k\|^2 \tag{10}$$

where C_{sub} denotes the set of correspondences contained in the maximal clique associated with each transformation in the transformation subset T_{sub}, and $c_k = (x_k, y_k)$.

5. Experiments

5.1. Experimental Setup

Datasets. We employ Scan2CAD (Avetisyan et al., 2019) as the real-world dataset, which aligns object instances in ScanNet (Dai et al., 2017) with CAD models in ShapeNet (Chang et al., 2015). In Scan2CAD, multiple real-world scan scenes contain identical 2-5 CAD instances, and provide accurate rigid transformation annotations. We fully utilize the annotation information and conduct experiments by respectively sampling the target point cloud from the scene point cloud and the source point cloud from the CAD model. After obtaining 2,175 sets of point clouds, we used 1,523 scenes for training, 326 scenes for validation, and 326 scenes for testing. We utilize the fine-tuned Predator (Huang et al., 2021) for point matching to establish the initial putative correspondence set.

To evaluate synthetic objects, we use the ModelNet40 (Wu et al., 2015), which contains 12,311 CAD models belonging to 40 categories. We sample 512 points from the CAD model as the source point cloud, and then repeat its rigid transformation 3-10 times to generate multiple instances. We merge these instances with randomly generated outliers to create the target point cloud. We use 9,843 models for training and 2,468 models for testing.

Metrics. We follow the evaluation procedure of PointCLM (Yuan et al., 2022), where the rotation error is defined as

$$RE = \arccos\left[\left(\mathrm{Tr}\left(R_{gt}^{T}R_{est}\right)-1\right)/2\right] \tag{11}$$

and translation error is defined as

$$TE = \|t_{est} - t_{gt}\|_2 \tag{12}$$

Success in registering an instance is indicated by $RE < 15^\circ$ and $TE < 0.1$. We use mean recall (MR), mean precision (MP), and their harmonic mean (MF) as evaluation metrics, which are defined as

$$MR = \frac{1}{N}\sum_{i=1}^{N}\frac{M_i^{suc}}{M_i^{gt}} \tag{13}$$

$$MP = \frac{1}{N}\sum_{i=1}^{N}\frac{M_i^{suc}}{M_i^{pred}} \tag{14}$$

$$MF = \frac{2\times MR\times MP}{MR+MP} \tag{15}$$

where N represents the number of paired point clouds, M^{suc} represents the number of successful registration instances, M^{gt} represents the actual number of instances, and M^{pred} represents the number of predicted transformations.

Implementation Details. We optimize the network using the Adam optimizer with a weight decay of 0.001, a learning rate of 0.01. Our network is trained using PyTorch, and we train the network for 1000 epochs. All the point clouds were downsampled in 0.05m voxel size. The distance parameter σ_d is set to 0.05 for the synthetic dataset and 0.1 for the real-world dataset. The distance parameter d_{thr} is set to 10 pr, where 'pr' is a distance unit called point cloud resolution (Yang et al., 2019). Default value for compatibility threshold t_c is 0.99. We select the correspondences whose confidence scores are above $\tau = 0.6$ as inliers and the others are removed as outliers. When compared with other methods, second-order compatibility measure and the DBSCAN clustering algorithm are utilized.

Baseline Methods. We compared PointMC with three multi-model fitting methods (RansaCov (Magri & Fusiello, 2016), CONSAC (Kluger et al., 2020), T-linkage (Magri & Fusiello, 2014)) and two state-of-the-art multi-instance point cloud registration methods (ECC (Tang & Zou, 2022), PointCLM (Yuan et al., 2022)). We adjusted all methods to achieve the best performance on the evaluation dataset within reasonable time and memory consumption ranges. To ensure a fair comparison, all methods used the same assumed correspondences as input.

5.2. Evaluation on Synthetic Dataset

We first compare our PointMC with other competitors' methods on synthetic dataset ModelNet40, and the quantitative results are shown in Table 1. Our PointMC outperforms all other methods, with a 1.75% increase in registration recall rate and a 0.01s reduction in computation time compared to the closest competitor. With the aid of global spatial consistency and an effective correspondence clustering strategy, PointCLM and ECC have also achieved impressive results. However, the remaining three multi-model fitting methods exhibited poor performance on the multi-instance registration task.

Table 1. Multi-instance registration results on ModelNet40 dataset.

	MR(%)	MP(%)	MF(%)	Runtime(s)
T-linkage	0.83	2.17	1.20	3.66
RansaCov	1.22	7.35	2.09	0.13
CONSAC	2.12	10.25	3.51	0.54
ECC	84.81	93.22	88.81	3.21
PointCLM	93.10	99.71	96.29	0.05
Ours	**94.85**	**99.76**	**97.24**	**0.04**

We show the visualization of the multi-instance point cloud registration results in Figure 3. In scenarios with multiple overlapping instances, our PointMC accurately estimates the number of rigid transformations and accurately estimates all transformation poses with small errors. Due to the use of powerful local spatial consistency to encode the correspondences of instances within local regions, our PointMC excellently distinguishes all instances. However, due to the

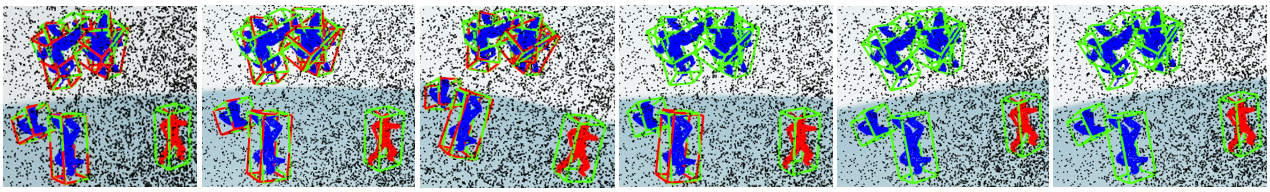

Figure 3. Examples of multi-instance registrations on ModelNet40 dataset. From (a) to (f): The source point cloud is shown in red, the transformed point clouds in the target point cloud are shown in blue, and the outlier points are shown in black. The green bounding box represents the actual poses of the instance in the target point cloud, and the red bounding box represents the predicted poses. The instance being surrounded by both red and green bounding boxes indicates successful detection of the current instance, while being surrounded only by green bounding boxes indicates missed detection.

limitations of global spatial consistency in the presence of a large number of outliers, PointCLM and ECC failed to distinguish between multiple overlapping instances, resulting in missed detections. The remaining three multi-model fitting methods, with the exception of CONSAC successfully registering one obvious instance with a large error, failed to register any instances.

5.3. Evaluation on Real-World Dataset

We then compared our PointMC with other competitors' methods on the real-world dataset Scan2CAD, and the quantitative results are shown in Table 2. Our PointMC outperforms all other methods in three metrics (MR, MP, MF), with average improvements of 25.94%, 21.92%, and 25.81%, respectively. It is comparable to CONSAC in terms of time consumption, but it improves registration recall by 27.4% and registration accuracy by 21.12%. Compared to three other multi-modal fitting methods, PointCLM and ECC also achieved decent results.

Table 2. Multi-instance registration results on Scan2CAD dataset.

	MR(%)	MP(%)	MF(%)	Runtime(s)
T-linkage	32.56	40.42	30.06	5.59
RansaCov	56.23	30.85	39.84	0.13
CONSAC	57.25	55.24	56.22	**0.06**
ECC	66.22	72.25	69.10	1.56
PointCLM	81.26	73.44	77.15	0.09
Ours	**84.65**	**76.36**	**80.29**	0.07

We present a set of visual registration results for an indoor point cloud scene with 16 identical chair instances to qualitatively compare with other competitors. Our PointMC still accurately predicts the number of instances in the scene and estimates the transformation poses of all source point clouds with small errors. PointCLM and ECC miss 1 and 2 instances, respectively, among the closely located ones. CONSAC only registers four easily distinguishable instances with larger errors. T-linkage and RansaCov also register three easily distinguishable instances with larger errors.

5.4. Ablation Study

In this section, we conduct several ablation experiments to investigate the effect of each essential component of PointMC. We first examine the effect of the local spatial consistency, and secondly demonstrate the benefits of the second-order compatibility measure and different clustering algorithms. The performance of the trained models are evaluated on the validation set of the considered datasets.

Table 3. The effect of local spatial consistency.

	GSC	LSC	MR(%)	MP(%)	MF(%)
ModelNet40	✓		93.31	99.39	96.25
ModelNet40		✓	**94.85**	**99.76**	**97.24**
Scan2CAD	✓		82.75	74.96	78.66
Scan2CAD		✓	**84.65**	**76.36**	**80.29**

Effect of local spatial consistency. In order to quantitatively study the effectiveness of local spatial consistency compared to global spatial consistency in the task of multi-instance point cloud registration, we replaced the correspondence feature extraction module in PointMC with the SC-Nonlocal module (Bai et al., 2021) based on global spatial consistency in PointCLM, and then compared the performance before and after the replacement. The comparative results are shown in Table 3, where "GSC" represents global spatial consistency, and "LSC" represents local spatial consistency. PointMC combined with local spatial consistency improved the average recall rate by 1.54% and the average accuracy by 0.37% on the ModelNet40 dataset, and increased the average recall rate by 1.9% and the average accuracy by 1.4% on the Scan2CAD dataset. We found mainly stems from registration scenes with multiple instances and instance overlaps, indicating the effectiveness of local spatial consistency in registering such scenes.

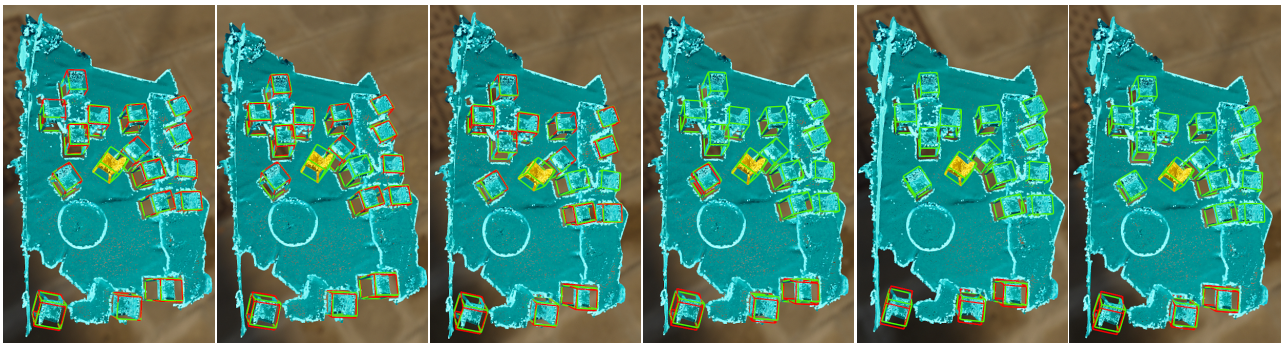

Figure 4. Examples of multi-instance registrations on Scan2CAD dataset. From (a) to (f): The source point cloud is shown in gold and the transformed point clouds in the target point cloud are shown in cyan. The green bounding box represents the actual poses of the instance in the target point cloud, and the red bounding box represents the predicted poses. The instance being surrounded by both red and green bounding boxes indicates successful detection of the current instance, while being surrounded only by green bounding boxes indicates missed detection.

Table 4. The effect of second-order compatibility measure.

	FOC	SOC	MR(%)	MP(%)	Runtime(s)
ModelNet40	✓		93.24	99.25	0.06
ModelNet40		✓	**94.85**	**99.76**	**0.04**
Scan2CAD	✓		82.26	73.89	0.10
Scan2CAD		✓	**84.65**	**76.36**	**0.07**

Table 5. The effect of DBSCAN and Chameleon.

	DA	CA	MR(%)	MP(%)	Runtime(s)
ModelNet40	✓		94.85	99.76	**0.04**
ModelNet40		✓	**95.11**	**99.79**	0.07
Scan2CAD	✓		84.65	76.36	**0.07**
Scan2CAD		✓	**86.53**	**79.12**	0.11

Effect of second-order compatibility measure. To justify the advantages of second-order compatibility measure, we change the compatibility graph construction process of PointMC to first-order compatibility measure and test the registration performance. The results are reported in Table 4, where "FOC" denotes the first-order compatibility measure and "SOC" denotes the second-order compatibility measure. Combining second-order compatibility measure, PointMC achieved an average recall improvement of 1.61% and an average precision improvement of 0.51% on the ModelNet40 dataset. It also reduced the runtime by 0.02s. On the Scan2CAD dataset, it achieved an average recall improvement of 2.39% and an average precision improvement of 2.47%. The runtime was reduced by 0.03s. By utilizing SOC to construct the compatibility graph, not only does it consider the geometric consistency of correspondences, but it also focuses on the commonly compatible matches in the correspondence set, making it more robust compared to FOC, especially in scenarios with high outlier rates. Meanwhile, SOC is sparser than FOC, which proves beneficial for faster searching of maximal cliques on the compatibility graph.

Effect of DBSCAN and Chameleon. We have quantitatively compared the effectiveness of the clustering algorithms DBSCAN and Chameleon in clustering low-dimensional rigid transformation datasets. The results are shown in Table 5, where "DA" represents the DBSCAN algorithm and "CA" represents the Chameleon algorithm. PointMC combined with Chameleon algorithm improved the average recall rate by 0.26% and the average accuracy by 0.03% on ModelNet40 dataset, and the average recall rate by 1.88% and the average accuracy by 2.76% on Scan2CAD dataset. However, PointMC combined with DBSCAN algorithm had a 42.9% reduction in runtime on ModelNet40 datasets and a 36.4% reduction on Scan2CAD datasets. Due to the higher density of data points in low-dimensional space, DBSCAN can accurately identify clustering structures and handle noise points and boundary points effectively. Chameleon algorithm may be slightly more complex in dealing with low-dimensional data. The graph construction process may introduce some overhead. However, Chameleon can provides better performance due to the better visualization and definition of clustering structures in low-dimensional space.

6. Discussions

Limitations. It is worth noting that the model has some limitations, and we leave it for future works. Firstly, the search and filtering process of maximal cliques is relatively time-consuming. Although the registration time required

by our framework is minimal, the performance could be further improved if a faster algorithm for maximal clique search could be introduced. In addition, the employed pose clustering algorithm is sensitive to parameters, affecting the generalization of the model.

7. Conclusion

In this work, we propose a framework called PointMC that leverages maximal cliques to address the problem of multi-instance point cloud registration. We utilize a module based on local spatial consistency to extract discriminative features from putative correspondences, enabling us to filter out true correspondences from a substantial amount of outliers. Based on this, we construct a compatibility graph of correspondences and search for maximal cliques to obtain a set of transformations. We then apply a clustering algorithm to efficiently cluster the transformations and obtain the transformation poses of the instances. The results on both synthetic and real-world datasets demonstrate that our method achieves state-of-the-art performance in terms of both accuracy and efficiency.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62036006,62276200), the CAAI-Huawei MINDSPORE Academic Open Fund, and the Fundamental Research Funds for the Central Universities (QTZX24074).

Impact Statement

This paper proposes a novel multi-instance point cloud registration framework. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aoki, Y., Goforth, H., Srivatsan, R. A., and Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, pp. 7163–7172, 2019.

Arun, K. S., Huang, T. S., and Blostein, S. D. Least-squares fitting of two 3-d point sets. *IEEE TPAMI*, pp. 698–700, 1987.

Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A. X., and Niessner, M. Scan2cad: Learning cad model alignment in rgb-d scans. In *CVPR*, 2019.

Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., and Tai, C.-L. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, pp. 15859–15869, 2021.

Barath, D. and Matas, J. Graph-cut ransac. In *CVPR*, pp. 6733–6741, 2018.

Cao, A.-Q., Puy, G., Boulch, A., and Marlet, R. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *CVPR*, pp. 13229–13238, 2021.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, Z., Sun, K., Yang, F., and Tao, W. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *CVPR*, pp. 13221–13231, 2022.

Cheng, Y.-C., Lee, H.-Y., Tulyakov, S., Schwing, A. G., and Gui, L.-Y. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, pp. 4456–4465, 2023.

Choy, C., Park, J., and Koltun, V. Fully convolutional geometric features. In *ICCV*, pp. 8958–8966, 2019.

Choy, C., Dong, W., and Koltun, V. Deep global registration. In *CVPR*, pp. 2514–2523, 2020.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pp. 5828–5839, 2017.

Drost, B., Ulrich, M., Navab, N., and Ilic, S. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, pp. 998–1005, 2010.

Ester, M. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc.int.conf.knowledg Discovery and Data Mining*, 1996.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pp. 381–395, 1981.

Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., and Schindler, K. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pp. 4267–4276, 2021.

Karypis, G., Han, E.-H., and Kumar, V. Chameleon: hierarchical clustering using dynamic modeling. *Computer*, pp. 68–75, 1999.

Kluger, F., Brachmann, E., Ackermann, H., Rother, C., Yang, M. Y., and Rosenhahn, B. Consac: Robust multi-model fitting by conditional sample consensus. In *CVPR*, pp. 4634–4643, 2020.

Le, H. M., Do, T.-T., Hoang, T., and Cheung, N.-M. Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In *CVPR*, pp. 124–133, 2019.

Leordeanu, M. and Hebert, M. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pp. 1482–1489, 2005.

Li, H., Liu, Y., Xiong, S., and Wang, L. Pedestrian detection algorithm based on video sequences and laser point cloud. *Frontiers of Computer Science*, 9:402–414, 2015.

Li, J., Luo, C., and Yang, X. Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. In *CVPR*, pp. 17567–17576, 2023.

Liu, J., Wang, G., Liu, Z., Jiang, C., Pollefeys, M., and Wang, H. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. In *ICCV*, pp. 8451–8460, 2023.

Ma, J., Zhao, J., Jiang, J., Zhou, H., and Guo, X. Locality preserving matching. *IJCV*, pp. 512–531, 2019.

Magri, L. and Fusiello, A. T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In *CVPR*, pp. 3954–3961, 2014.

Magri, L. and Fusiello, A. Multiple model fitting as a set coverage problem. In *CVPR*, pp. 3318–3326, 2016.

Merras, M., El Hazzat, S., Saaidi, A., Satori, K., and Nazih, A. G. 3d face reconstruction using images from cameras with varying parameters. *International Journal of Automation and Computing*, 14:661–671, 2017.

Qin, Z., Yu, H., Wang, C., Peng, Y., and Xu, K. Deep graph-based spatial consistency for robust non-rigid point cloud registration. In *CVPR*, pp. 5394–5403, 2023.

Rusu, R. B., Blodow, N., and Beetz, M. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, pp. 3212–3217, 2009.

Sorkine-Hornung, O. and Rabinovich, M. Least-squares rigid motion using svd. *Computing*, pp. 1–5, 2017.

Tang, W. and Zou, D. Multi-instance point cloud registration by efficient correspondence clustering. In *CVPR*, pp. 6667–6676, 2022.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pp. 6411–6420, 2019.

Wang, Y. and Solomon, J. Deep closest point: Learning representations for point cloud registration. In *ICCV*, pp. 3522–3531, 2019.

Wei, Y.-W., Chen, W.-M., and Tsai, H.-H. Accelerating the bron-kerbosch algorithm for maximal clique enumeration using gpus. *IEEE Transactions on Parallel and Distributed Systems*, pp. 2352–2366, 2021.

Wu, Y., Hu, X., Zhang, Y., Gong, M., Ma, W., and Miao, Q. Sacf-net: Skip-attention based correspondence filtering network for point cloud registration. *IEEE TCSVT*, 33(8): 3585–3595, 2023.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pp. 1912–1920, 2015.

Yang, F., Guo, L., Chen, Z., and Tao, W. One-inlier is first: Towards efficient position encoding for point cloud registration. *Advances in Neural Information Processing Systems*, 35:6982–6995, 2022.

Yang, J., Xiao, Y., Cao, Z., and Yang, W. Ranking 3d feature correspondences via consistency voting. *Pattern Recognition Letters*, pp. 1–8, 2019.

Yew, Z. J. and Lee, G. H. Rpm-net: Robust point matching using learned features. In *CVPR*, pp. 11824–11833, 2020.

Yew, Z. J. and Lee, G. H. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, pp. 6677–6686, 2022.

Yu, J., Ren, L., Zhang, Y., Zhou, W., Lin, L., and Dai, G. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *CVPR*, pp. 17702–17711, 2023a.

Yu, N., Li, H., Xu, Q., Sie, O., and Firdaous, E. 3d reconstruction and defect pattern recognition of bonding wire based on stereo vision. *CAAI Transactions on Intelligence Technology*, 2023b.

Yuan, M., Li, Z., Jin, Q., Chen, X., and Wang, M. Pointclm: A contrastive learning-based framework for multi-instance point cloud registration. In *ECCV*, pp. 595–611, 2022.

Yuan, M., Fu, K., Li, Z., and Wang, M. Decoupled deep hough voting for point cloud registration. *Frontiers of Computer Science*, 18(2):182703, 2024.

Zhang, X., Yang, J., Zhang, S., and Zhang, Y. 3d registration with maximal cliques. In *CVPR*, pp. 17745–17754, 2023a.

Zhang, Z., Chen, J., Xu, X., Liu, C., and Han, Y. Hawk-eye-inspired perception algorithm of stereo vision for obtaining orchard 3d point cloud navigation map. *CAAI Transactions on Intelligence Technology*, 8(3):987–1001, 2023b.

Zhao, H., Zhang, J., Chen, Z., Yuan, B., and Tao, D. On robust cross-view consistency in self-supervised monocular depth estimation. *Machine Intelligence Research*, 21(3): 495–513, 2024.

Zhou, C., Zhang, Y., Chen, J., and Huang, D. Octr: Octree-based transformer for 3d object detection. In *CVPR*, pp. 5166–5175, 2023.