Mutual Voting for Ranking 3D Correspondences

Jiaqi Yang, Xiyu Zhang, Shichao Fan, Chunlin Ren and Yanning Zhang, *Senior Member, IEEE*

Abstract—Consistent correspondences between point clouds are vital to 3D vision tasks such as registration and recognition. In this paper, we present a mutual voting method for ranking 3D correspondences. The key insight is to achieve reliable scoring results for correspondences by refining both voters and candidates in a mutual voting scheme. First, a graph is constructed for the initial correspondence set with the pairwise compatibility constraint. Second, nodal clustering coefficients are introduced to preliminarily remove a portion of outliers and speed up the following voting process. Third, we model nodes and edges in the graph as candidates and voters, respectively. Mutual voting is then performed in the graph to score correspondences. Finally, the correspondences are ranked based on the voting scores and top-ranked ones are identified as inliers. Feature matching, 3D point cloud registration, and 3D object recognition experiments on various datasets with different nuisances and modalities verify that MV is robust to heavy outliers under different challenging settings, and can significantly boost 3D point cloud registration and 3D object recognition performance. Code will be available at: https://github.com/NWPU-YJQ-3DV/2022_Mutual_Voting.

Index Terms—3D point clouds, mutual voting, feature matching, point cloud registration, object recognition

1 Introduction

POINT cloud feature matching is one of the key problems in the field of 3D computer vision. As the basis of 3D computer vision tasks such as 3D reconstruction, 3D object recognition, 3D object tracking, and point cloud registration, point cloud feature matching requires the establishment of correct point-to-point correspondences (a.k.a matches) in the point cloud sequence. Such correspondences are usually produced through a two-step procedure: initial correspondence generation and inlier selection.

However, most existing 3D keypoint detectors and descriptors still suffer from limited performance, resulting in outliers in the initial correspondence set. Additionally, in real-world applications, nuisances such as noise, varying data resolutions, clutter, and occlusion will further generate outliers. In that case, inlier selection is a vital step for obtaining geometrically consistent correspondences, however, remains a very challenging issue at present [1].

For 3D inlier selection, existing geometry-based works can be categorized as *label*-based and *score*-based. For label-based methods [2], [3], [4], a two-class classifier is designed to assign binary labels to correspondences. Typically, these methods generally assume that the correct correspondences can form a cluster and tend to find these matches with a one-shot manner. These methods are hand-crafted two-class classifiers, and usually suffer from limited discriminative power to distinguish inliers and outliers. For score-based methods [5], [6], [7], [8], they first calculate the confidence scores for input correspondences and then rank correspondences based on the scoring results. Feature similarities and geometric constraints are two common cues for defining correspondence scoring functions. Methods relying on feature similarity constraints are not robust, as the descriptiveness of feature descriptors is not guaranteed. Methods relying on geometric constraints are more common to 3D feature matching, whereas voting-based ones have attracted increasing attention due to their effectiveness and simplicity. Prior voting-based methods [5], [7] mainly vote in the Euclidean space, which makes it difficult to accurately model the compatibility relationship among correspondences and fully exploit the consistency between the inliers. Thus, existing voting-based methods are still sensitive to heavy outliers. For deep-learned works, they generally need massive data for training and hold poor generalization ability to different down-stream tasks [1].

Under these considerations, we propose a mutual voting (MV) method for 3D inlier selection. Existing voting-based methods follow a "voter→candidiate" one-way voting scheme, which ignores the potential existence of unreliable voters in the voting set and leads to less convincing voting scores. By contrast, the key insight of our MV is to achieve reliable scoring results for correspondences by refining both voters and candidates in a mutual voting scheme. To the best of our knowledge, MV proposes the first mutual voting scheme for scoring 3D correspondences. It first models the initial correspondence set as a compatibility graph, where each nodal represents a single correspondence and each edge between two nodes indicates a pair of geometrically compatible correspondences. Second, node clustering coefficients are introduced to preliminarily remove a portion of outliers and speed up the following voting process. Third, nodes and edges in the graph are rendered as candidates and voters, respectively. Subsequently, "node↔edge" mutual voting is performed in the graph, and voting scores are calculated for all correspondences. Finally, input correspondences are ranked in a descending order based on voting scores, and top-ranked ones are recognized as inliers. To verify the robustness and effectiveness of MV, we conduct feature matching experiments on datasets

- *This work is supported in part by the National Natural Science Foundation of China (NFSC) under Grants U19B2037 and 62002295, Natural Science Basic Research Program of Shaanxi (No. 2021JCW-03), and Shaanxi Provincial Key R&D Program (No. 2021KWZ-03).*

- *Jiaqi Yang, Xiyu Zhang, Shichao Fan, Chunlin Ren and Yanning Zhang are with the National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, China. E-mail: jqyang@nwpu.edu.cn; {2426988253, fsc_smile, renchunlin}@mail.nwpu.edu.cn; ynzhang@nwpu.edu.cn. (Corresponding author: Yanning Zhang)*

incorporating various nuisances such as limited overlap, clutter, occlusion, noise, data resolution variation, and the results indicate that MV is robust to common nuisances. In addition, MV's performance in down-stream tasks are experimentally verified. Specifically, 3D point cloud registration and 3D object recognition experiments are deployed paired with comparisons with the state-of-the-arts. MV also achieves outstanding point cloud registration and object recognition performance when served as a drop-in replacement in existing pipelines. The distinctions of MV against existing methods are:

- **Mutual voting: voters and candidates are voted mutually to refine each other and finally improve the confidence of voting scores.** Unlike existing voting-based methods [5], [7] that perform voting in a single ”voter→candidate” line, our MV method performs voting in a dual ”voter↔candidate” way. The motivation behind is that convincing voting scores are based on convincing voters, and the reliability of voters can also be evaluated from voters cast by candidates. This mechanism clearly makes MV distinctive against existing methods. MV is also different from a recent graph-based method called SC^2-PCR [9] from two aspects. 1) SC^2-PCR is a transformation estimator that outputs a rigid transformation from correspondences, while MV performs correspondence selection. 2) Although both methods execute in a graph space, the definition of the label/score for a correspondence plays a more vital role. MV scores correspondences with a novel mutual voting scheme, while SC^2-PCR tends to find a set of seed correspondences and generate hypotheses from a few reliable seeds.
- **Nodal clustering coefficient: it is introduced to preliminarily remove outliers.** We introduce the nodal clustering coefficient to remove a portion of outlier nodes in the graph. This can be served as a coarse module for 3D outlier rejection. It can alleviate the negative impact of outliers on the following refine module, i.e., mutual voting.

In a nutshell, this paper presents the following contributions:

- **We propose an MV method for ranking 3D correspondences.** A mutual voting mechanism is presented by refining both voters and candidates to achieve reliable scoring results. MV is highly selective even in the presence of limited overlap, clutter, occlusion, noise, and data resolution variation.
- **MV can effectively boost the performance of correspondence-based down-stream tasks.** We feed the selected correspondences to correspondence-based pipelines of 3D point cloud registration and object recognition, and demonstrate that MV can significantly boost the performance of downstream tasks including 3D point cloud registration and object recognition.

The remainder of this paper is organized as follows. Sect. 2 gives a brief review on correspondence selection methods. Sect. 3 provides a detailed description of our MV method. Sect. 4 presents the experimental setup, results, and discussions. Sect. 5 draws conclusions and presents future research directions.

2 Related Work

This paper focuses on the selection process for correct correspondences, and we will give a review on correspondence selection methods in 2D and 3D domains.

2.1 2D Feature Matching

Traditional methods. Fischler et al. [2] proposed the random sampling consistency (RANSAC) method, which can find the optimal parameter model in a set of datasets containing outliers using an iterative approach, and has been widely used in image alignment and stitching. For estimating transformations from correspondences, RANSAC iteratively performs a hypothesis generation-and-verification process by randomly draw samples from the given correspondence set. However, RANSAC suffers from limited accuracy and efficiency in the presence of heavy outliers. Later, many variants of RANSAC have been proposed to alleviate these issues [10], [11].

Leordeanu and Hebert [3] proposed a spectral analysis approach for finding consistent correspondences in the initial feature matching set by assuming that inliers usually form a cluster. Torresani et al. [12] scored the initial correspondences by minimizing an energy function on correspondence similarity and spatial compatibility. These two methods usually require the initial correspondence set to possess a high inlier ratio. To deal with this issue, some graph-matching-based methods have emerged [13], [14], [15], [16], [17]. However, most graph-matching-based methods suffer from limitations such as poor scalability and limited efficiency [18], [19]. Although some methods can achieve robust feature matching performance or speed up the process of outlier rejection [20], [21], [22], it is still a challenging problem when dealing with heavy outliers [23].

Learning-based methods. Learning-based methods for 2D feature matching serve the inlier selection problem as a binary classification problem. Yi et al. [24] presented the first deep-learned feature matcher in 2D domain based on multi-layer perception networks, however, it fails to mine local features among correspondences. To address this issue, several works such as NM-Net [25], and OA-Net [26] introduce various local feature mining modules to capture local information. Dai et al. [27] proposed MS^{2}DG-Net, which considers the semantic information between two given images' sparse correspondences. Sun et al. [28] normalized the feature map using weights estimated in a permutation equivalent network, and excluded outliers from this normalization.

In 2D domain, when sufficient training samples are available, learning-based methods generally achieve better performance compared to traditional methods. 2D feature matching concentrates on matching regular images. By contrast, MV focuses on the matching of unordered and irregular 3D point clouds. It is more challenging due to higher data dimensionality and data irregularity.

2.2 3D Feature Matching

Score-based methods. Most of the early score-based methods [29], [30] rank 3D correspondences on feature similarity scores, while local features could be sensitive with respect to noise and data resolution variation. Rodolà et al. [6] proposed a game theory matching (GTM) method, which seeks global consistency between surface points while operating locally. It extends the scope of local descriptors by actively selecting correspondences that satisfy the global consistency constraint, enabling generalization to more challenging scenarios where two point clouds have different unknown scales. Buch et al. [5] proposed a voting-based method that combines both local and global constraints to score correspondences. Yang et al. [7] proposed a consistency voting method, which uses geometric and spatial constraints to calculate the consistency scores of the predefined voting set and the initial correspondence set for correspondence ranking. Sahloul et al. [31] followed a two-stage scoring scheme to rank correspondences, and proposed a single-point superimposition transforms to improve the stability of local constraint. Chen et al. [9] proposed SC^2-PCR that introduces a second-order spatial compatibility metric to measure the affinity between correspondences and employs global spectral technique to rank correspondences; then, top-scored correspondences are served as seeds to compute a final rigid transformation for 3D point cloud registration.

Label-based methods. There are two types of label-based methods: geometric and deep-learned. *1) Geometric methods:* some label-based methods assume that the correct correspondences can form a cluster and tend to find these correspondence in a one-shot manner. For instance, Chen et al. [32] formed a cluster for each correspondence by ensuring matches in the cluster are compatible with the query match, and the largest cluster is supposed to be the inlier set; Tombari et al. [4] proposed 3D Hough voting, which first transforms correspondence to a 3D Hough space and then finds a tight cluster formed by inliers in the Hough space. Recently, Bustos and Chin [33] presented a global optimization method called guaranteed outlier removal (GORE) based on branch-and-bound for six-degree-of-freedom (6-DoF) Euclidean registration; Yang et al. [34] proposed TEASER++, which reformulates the registration problems using a truncated least squares (TLS) cost and introduces a general graph-theoretic framework for outlier removal. *2) Deep-learned methods:* similar to 2D deep-learned methods, they try to design a distinctive deep-learned classifier. Deep global registration (DGR) [35] and 3DRegNet [36] classify a given correspondence by training end-to-end neural networks and using operators such as sparse convolution and point-by-point MLP; Yu et al. [37] proposed CoFiNet, a coarse-to-refine learning framework, which extracts correspondences from coarse to fine without keypoint detection; Bai et al. [38] proposed PointDSC, a deep neural network that explicitly explores spatial consistency for removing outlier correspondences and 3D point cloud registration.

Although deep-learned methods have achieved remarkable performance, they usually require plenty of training data, which are not always available in real-world applications. In addition, the generalization ability still remains an issue for deep-learned methods. As such, this paper focuses on geometric methods. Compared with existing methods, MV has two distinctions. First, instead of voting in the Euclidean space, MV performs voting in a graph space to better model the compatibility relationship among correspondences. Second, existing voting-based methods are in a one-way voting fashion, which ignore the fact that unreliable voters commonly exist and result in performance deterioration. On the contrary, MV is a mutual voting method that additionally refines voters based on "candidate→voter" voting to improve the quality of the voting set. We will show that our method, without massive training data and GPU computing, yields even better performance than deep-learned ones and achieves pleasurable performance in both 3D point cloud registration and 3D object recognition applications.

3 Methodology

The pipeline of our method is presented in Fig. 1. There are mainly four steps involved: graph construction, nodal clustering coefficients calculation, mutual voting, and correspondence ranking. The role of each step in the pipeline is as follows:

- **Graph construction:** the initial correspondence set is modeled as a compatibility graph, where each node represents a single correspondence, and each edge between two nodes indicates a pair of geometrically compatible correspondences. The motivation is to accurately render the affinity relationship among unordered correspondences.
- **Nodal clustering coefficient calculation:** in complex networks, clustering coefficients are used to measure the degree to which graph nodes hug the surroundings, portraying how dense or sparse the network is. This concept is introduced to our MV method. It aims at preliminarily removing a portion of outliers to present a better base for the following mutual voting.
- **Mutual voting:** the score of each correspondence is assigned with a mutual voting scheme, where voters and candidates are mutually refined to achieve a convincing scoring result. As high-quality voting sets are fundamental to convincing voting results, we enforce "candidate→voter" voting to reduce unreliable voters, in addition to "voter→candidiate" voting. This forms a mutual voting scheme.
- **Correspondence ranking:** all input correspondences are sorted in a descending order based on the voting scores. Top-scored ones are served as selected inliers by MV.

To solve the 3D feature matching problem with heavy outliers, we present MV that first constructs a graph to accurately model the compatibility relationship among correspondences and then performs mutual voting to refine both voters and candidates to achieve convincing scoring results for correspondences. MV is detailed in the following.

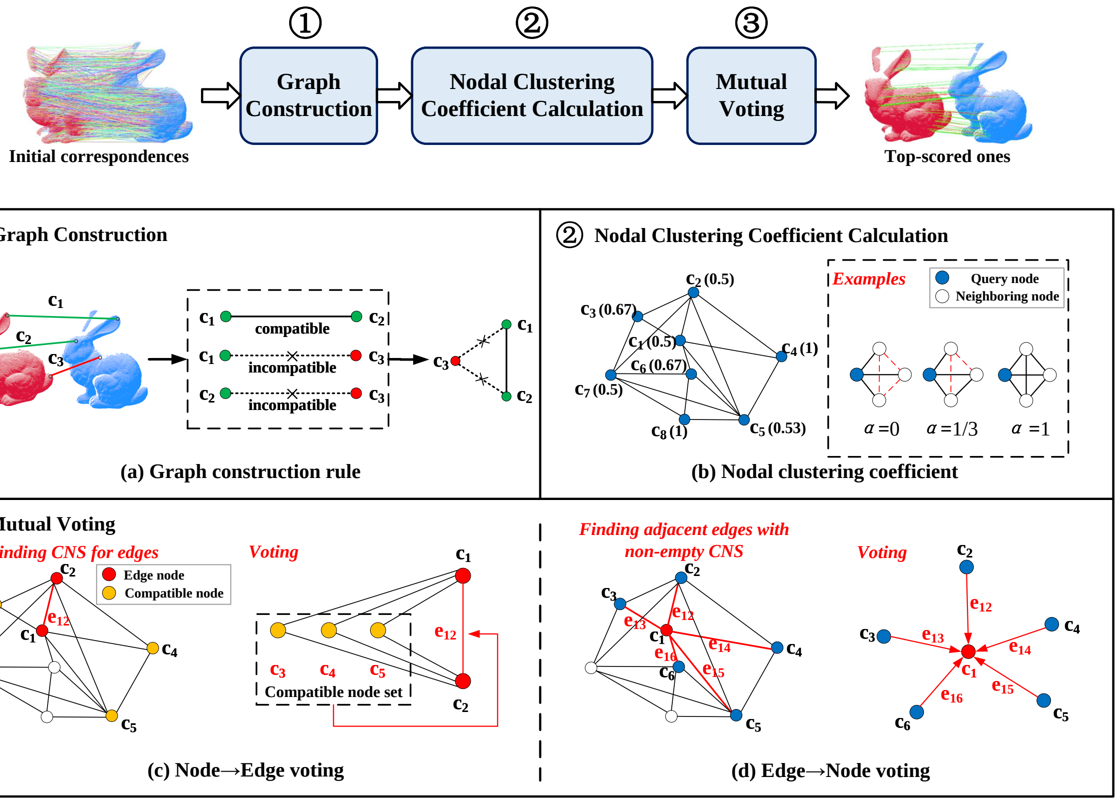


Fig. 1. Pipelines of the proposed MV method. The input to MV is the initial correspondence set with outliers. First, a compatibility graph is constructed for the initial correspondence set based on the compatibility score between correspondences. Second, the nodal clustering coefficients are calculated to preliminarily remove a portion of outliers. Third, mutual voting is performed between nodes and edges. At the node→edge voting stage, each edge will find its compatible node set (CNS), nodes in this set vote for the edge. At the edge→node voting stage, a node's adjacent edges with non-empty CNS vote for the node. Finally, all correspondences are ranked according to the voting scores, and top-scored ones are served as inliers.

3.1 Graph Construction

The voting process will be performed in a graph. Compared with the Euclidean space, the graph space can better render the affinity relationship among correspondences. The initial correspondence set is modeled as a compatibility graph, where nodes represent correspondences and edges connect geometrically compatible nodes.

In particular, let $\mathbf{p}_i^s$ and $\mathbf{p}_i^t$ denote the points in the source point cloud $\mathbf{P}^s$ and target point cloud $\mathbf{P}^t$, respectively. Then, the rigidity between the two correspondences $\mathbf{c}_i$ and $\mathbf{c}_j$ can be quantitatively measured as:

$$S_{dist}(\mathbf{c}_i, \mathbf{c}_j) = \left| \left\| \mathbf{p}_i^s - \mathbf{p}_j^s \right\| - \left\| \mathbf{p}_i^t - \mathbf{p}_j^t \right\| \right|. \tag{1}$$

The compatibility score between $\mathbf{c}_i$ and $\mathbf{c}_j$ is given as:

$$S_{cmp}(\mathbf{c}_i, \mathbf{c}_j) = \exp(-\frac{S_{dist}(\mathbf{c}_i, \mathbf{c}_j)^2}{2d_{cmp}^2}), \tag{2}$$

where d_{cmp} is a distance parameter and $S_{cmp} \in [0, 1]$. Ideally, $S_{cmp}(\mathbf{c}_i, \mathbf{c}_j) = 1$ if $\mathbf{c}_i$ and $\mathbf{c}_j$ are inliers.

Subsequently, as shown in Fig.1(a), given a set of initial correspondences $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$, we model them as a graph $G = (\mathbf{V}, \mathbf{E})$. Here, $\mathbf{V} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$ and $\mathbf{E} = \{\mathbf{e}_{12}, \mathbf{e}_{13}, ..., \mathbf{e}_{ij}\}$ with $\mathbf{e}_{ij} = (\mathbf{c}_i, \mathbf{c}_j)$. Notably, if $S_{cmp}(\mathbf{c}_i, \mathbf{c}_j)$ is greater than a threshold t_{cmp}, $\mathbf{c}_i$ and $\mathbf{c}_j$ form an edge and $S_{cmp}(\mathbf{c}_i, \mathbf{c}_j)$ is the weight of the edge. To this end, a graph is generated for $\mathbf{C}$.

3.2 Nodal Clustering Coefficient Calculation

In complex networks, the clustering coefficients not only portray how dense and sparse the network is, but also describe the degree to which the nodes' neighbors hug each other [39]. We introduce this to the problem of 3D feature matching.

3.2.1 *Basic Concepts of the Clustering Coefficient*

Let the degree of node $\mathbf{c}_i$ be d_i, then there are at most $d_i(d_i - 1)/2$ edges among d_i neighbor nodes. Let w_i denote the number of edges that actually exist among the neighboring nodes of node c_i. In a weighted network, w_i denotes the sum of the weights of these edges. The clustering coefficient α_i for $\mathbf{c}_i$, as illustrated in Fig. 1(b), can be defined as:

$$\alpha_i = \frac{w_i}{(d_i * (d_i - 1))/2}. \tag{3}$$

The nodal clustering coefficients reflect the significance of the nodes in the network and the degree of local aggregation of the network. In addition, average clustering coefficient $\overline{\alpha}$ and overall clustering coefficient $\alpha_{overall}$ can be used to express the degree of aggregation of the whole network, as defined in the following:

$$\overline{\alpha}=\frac{1}{n}\sum_{i=1}^{n}\alpha_i, \tag{4}$$

$$\alpha_{overall}=\frac{\sum\limits_{i=1}^{n} w_i}{\sum\limits_{i=1}^{n}(d_i*(d_i-1))/2}. \tag{5}$$

3.2.2 *Application of Nodal Clustering Coefficient in MV*

i) Remove outliers preliminarily. Inliers are consistent, and therefore are supposed to be compatible with each other. As such, inliers are more likely to form cliques in a graph. It is interesting to note that nodes with greater nodal clustering coefficients are more likely to be in cliques. Therefore, we set an adaptive threshold t_α to eliminate nodes with low nodal clustering coefficients, which is defined as:

$$t_\alpha=\min(\alpha_{overall},\overline{\alpha},otsu_\alpha), \tag{6}$$

where $otsu_\alpha$ represents the OTSU [40] threshold based on the nodal clustering coefficients of all nodes. The leverage of nodal clustering coefficient has two merits. First, a portion of outliers can be removed, which would alleviate the negative effects of outliers in the following mutual voting process. Second, less nodes will be involved in the voting process, therefore speeding up the selection process.

ii) Be the weight in the voting process. The nodal clustering coefficient is an informative cue for nodes. It will participate the "node→edge" voting process (Sect. 3.3)

3.3 Mutual Voting

All nodes in G will be scored in a mutual voting process. It is intuitive that candidates are nodes. For the definition of voters, we choose edges. Admittedly, there are other choices of voters, such as loops and cliques in the graph. However, searching these types of voters is time-consuming especially in large-scale graphs. In addition, edges are essential components for these voters. Therefore, "node↔edge" mutual voting, as illustrated in Fig. 1(c) and (d), is designed for scoring correspondences.

i) Node→edge voting. We perform node→edge voting to assign weights to edges in the subsequent edge→node voting process. For each edge, we try to find its "compatible node set" (CNS). In particular, a node $\mathbf{c}_k$ is judged as a compatible node with respect to edge $\mathbf{e}_{ij}$ if $\mathbf{c}_k$ is connected with both $\mathbf{c}_i$ and $\mathbf{c}_j$. The set of such nodes for edge $\mathbf{e}_{ij}$ is defined as its CNS, denoted by $\mathbf{C}_{cmp}(\mathbf{e}_{ij})$. As shown in Fig. 1(c), nodes $\mathbf{c}_3$, $\mathbf{c}_4$, and $\mathbf{c}_5$ constitute the CNS for edge $\mathbf{e}_{12}$.

Specifically, the CNS for an edge can be efficiently retrieved as follows. For a node $\mathbf{c}_i$ in the compatibility graph, its neighboring node set $\mathbf{N}(\mathbf{c}_i)$ is given as:

$$\mathbf{N}(\mathbf{c}_i)=\{\mathbf{c}_j|\mathbf{c}_j\in\mathbf{V},\mathbf{e}_{ij}\in\mathbf{E}\}. \tag{7}$$

Then, the CNS of $\mathbf{e}_{ij}$ can be derived by:

$$\mathbf{C}_{cmp}(\mathbf{e}_{ij})=\mathbf{N}(\mathbf{c}_i)\cap\mathbf{N}(\mathbf{c}_j). \tag{8}$$

At this point, edges are associated with CNSs (could be empty sets). The voting score of $\mathbf{e}_{ij}$ is defined as:

$$\begin{aligned}S(\mathbf{e}_{ij})=\sum_{\mathbf{c}_k\in\mathbf{C}_{cmp}(\mathbf{e}_{ij})}\frac{\alpha_i+\alpha_j+\alpha_k}{3}[S_{cmp}(\mathbf{c}_i,\mathbf{c}_j)+\\ S_{cmp}(\mathbf{c}_i,\mathbf{c}_k)+S_{cmp}(\mathbf{c}_j,\mathbf{c}_k)].\end{aligned} \tag{9}$$

Therefore, edges are assigned with voting scores.

ii) Edge→node voting. In this stage, edges become voters. As illustrated in Fig. 1(d), nodes are voted by its adjacent edges whose CNS is not empty. More specifically, the voting score for $\mathbf{c}_i$ is defined as:

$$S(\mathbf{c}_i)=\sum_{\mathbf{C}_{cmp}(\mathbf{e}_{ij})\neq\emptyset}S(\mathbf{e}_{ij}) \tag{10}$$

The initial set of correspondences is then sorted in a descending order according to the voting score $S(\mathbf{c}_i)$. The determination of the output of MV is flexible. One is using OTSU [40] thresholding strategy. The other is selecting the top-K ones as inliers, where K can be tuned according to a particular application scenario. By default, we choose the former one for automatic inlier selection.

3.4 Computational Complexity Analysis

Finally, we present the computational complexity analysis of MV. The main steps of MV include: graph construction, nodal clustering coefficients calculation, mutual voting, and correspondence ranking. Assuming a compatibility graph with n nodes and m edges: first, the time complexity of modeling the initial correspondence set into a compatibility graph is $O(n^2)$; second, calculating the clustering coefficients for all nodes indicates a computational compatibility $O(n^3)$; third, finding the CNSs for edges and the mutual voting process have computational complexities of $O(n^3)$ and $O(n^2)$, respectively; finally, the computational complexity of ranking correspondences with a fast sorting algorithm is $O(n\log(n))$. Therefore, the overall computational complexity of the method is $O(n^2)+O(n^3)+O(n^3)+O(n^2)+O(n\log(n))=O(n^3)$. In practical applications, the number of correspondences are generally at the magnitude of a few thousand [1], and our method is still efficient as compared with many other 3D feature matchers (verified in Sect. 4.1.5).

4 Experiments

This section presents three types of experiments for evaluating our proposed method: feature matching, point cloud registration and 3D object recognition experiments.

4.1 Feature Matching Experiments

4.1.1 *Experimental Setup*

Dataset. The datasets used for the experiments in this section are UWA 3D modeling (U3M) [41], Bologna Mesh Registration (BMR) [42], UWA 3D object recognition (U3OR) [43], [44], and Bologna Dataset5 (BoD5) [42]. These datasets have different data modalities, a variety of nuisances, and different application scenarios that allow for an in-depth feature matching performance evaluation. The details for these datasets are shown in Table 1. Several

TABLE 1
Properties of feature matching experimental datasets.

Dataset	Data modality	Nuisances	Application scenario	# Matching pairs	Avg. inlier ratio
U3M [41]	LiDAR	Limited overlap, self-occlusion	Object Registration	496	14.80%
BMR [42]	Kinect	Limited overlap, self-occlusion, real noise	Object Registration	485	5.63%
U3OR [43], [44]	LiDAR	Clutter, occlusion	Object Recognition	188	8.09%
BoD5 [42]	Kinect	Clutter, occlusion, real noise, holes	Object Recogniiton	43	15.75%

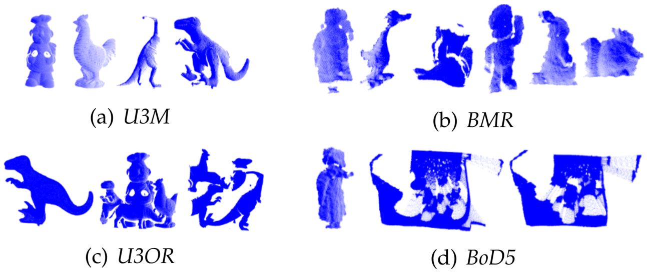

(a) *U3M*
(b) *BMR*
(c) *U3OR*
(d) *BoD5*

Fig. 2. Sample point clouds from feature matching experimental datasets.

sample views of these datasets are visualized in Fig. 2.

Evaluation metric. For a correspondence $\mathbf{c} = (\mathbf{p}^s, \mathbf{p}^t)$, it is judged as correct if it satisfies:

$$\left\| \mathbf{R}_{gt}\mathbf{p}^s + \mathbf{t}_{gt} - \mathbf{p}^t \right\| < d_{inlier}, \tag{11}$$

where $\mathbf{R}_{gt}$ and $\mathbf{t}_{gt}$ denote the ground-truth rotation matrix and translation vector respectively; d_{inlier} is a distance threshold, which is set to 5 pr in the experiment. Here, 'pr' is a distance unit called point cloud resolution, which is the mean of the closest distance of a point to the nearest neighbor in a point cloud. Following [7], [8], we use the recall of inliers with respect to top-K correspondence subset as the evaluation metric. By varying the value of K and recording the number of inliers in the corresponding subset, a curve can be plotted with respect to different settings of K. Let $\mathbf{C}_K$ be the top-K correspondence subset and $\mathbf{C}_{initial}$ be the initial correspondence set, the recall of inliers with respect to K is defined as:

$$recall_K = \frac{\#inliers\ in\ \mathbf{C}_K}{\#inliers\ in\ \mathbf{C}_{initial}}. \tag{12}$$

Compared methods. We compare MV with nine existing state-of-the-art feature matching methods. These methods are similarity score (SS) [43], [45], nearest neighbor similarity ratio (NNSR) [46], spectral technique (ST) [3], search of inliers (SI) [47], game theory matching (GTM) [6] and consistency voting (CV) [7], PointDSC [38], TEASER++ [34], SC^2-PCR [9], where SI and CV are voting-based methods as well.

Implementation details. The experiments were implemented using the point cloud library (PCL) [48]. A previous research [1] has verified that the correspondence sets computed using the combination of Harris3D [49]+SHOT [42] have different spatial distributions, different scales, and different inlier ratios, allowing for a comprehensive evaluation of feature matching methods. Therefore, we employ the Harris3D detector for keypoint detection and the SHOT descriptor for local geometric feature extraction to generate initial correspondences. We also have tried the other combinations in Sec. 4.1.4. By default, we empirically set the compatibility threshold t_{cmp} and the distance parameter d_{cmp} mentioned in Sec. 3.1 to 0.9 and 10 pr, respectively.

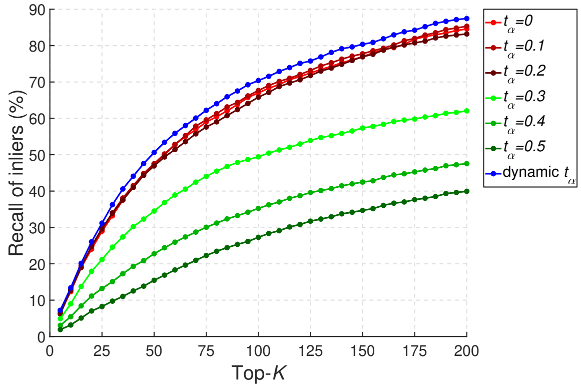


Fig. 3. Performance of MV with different settings of t_α.

TABLE 2
Recall performance (%) of applying different constraints to MV on the U3M dataset.

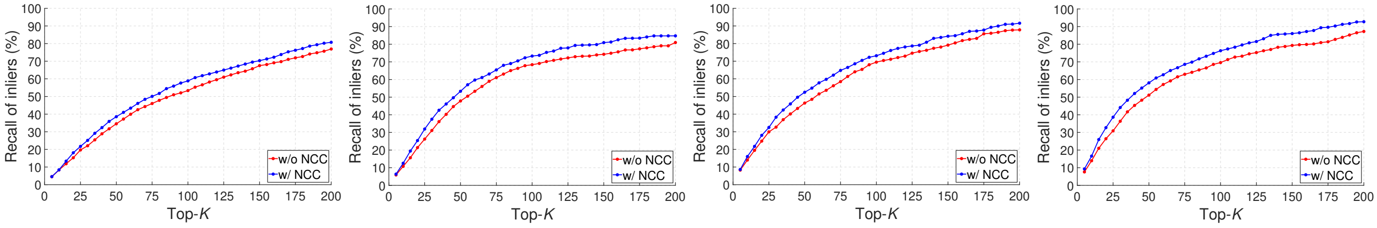
Std. of Gaussian noise (pr)	1.0	1.5	2.0	2.5	3.0
Distance+angular	71.39	68.32	60.17	51.51	48.00
Distance-only	**74.43**	**72.92**	**65.45**	**64.21**	**62.95**

4.1.2 Method Analysis

Compatibility constraints. In addition to the distance constraint, the angular constraint, i.e., deviation angles between normals, is also available for measuring the rigidity between correspondences during graph construction. To verify the rationality of applying the distance-only constraint, we compared the recall performance of MV and its variant with the distance+angular constraint on top-100 sets under different levels of Gaussian noise. This experiment was conducted on the U3M dataset.

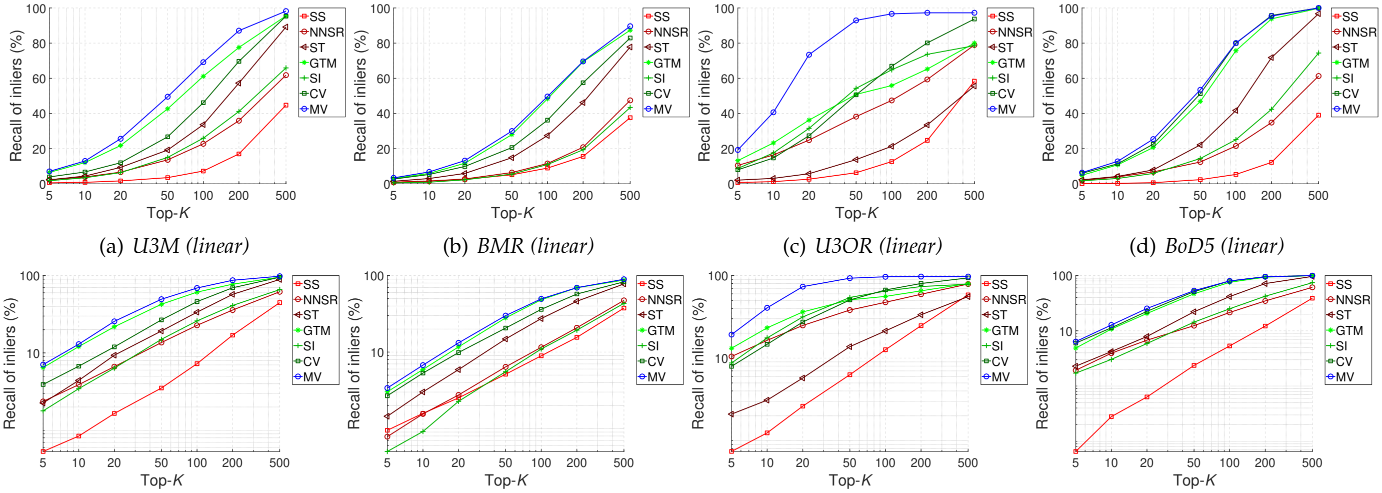
Results in Table 2 show that adding angular constraint does not improve the recall performance, because the angular constraint is very sensitive to noise. By contrast, the distance-only constraint achieves more stable performance.

The rationality of setting clustering coefficient threshold adaptive. The clustering coefficient threshold t_α in Eq. 6 is used to preliminarily reject outliers. To very the rationality of making it adaptive, we vary t_α from 0 to 0.5 with a step of 0.1, and compare with the adaptive threshold. The



(a) *Chef* (b) *Chicken* (c) *Parasaurolophus* (d) *T-rex*

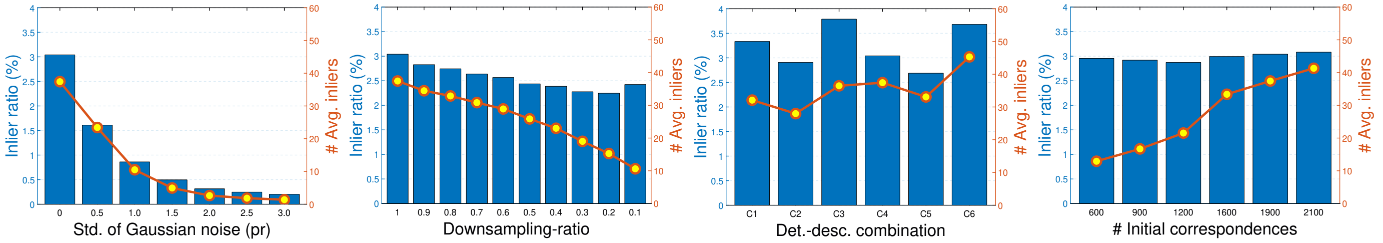
Fig. 4. Comparison of MV methods with and w/o nodal clustering coefficient calculation on the four subsets of U3M (NCC denotes nodal clustering coefficients).



(a) *U3M (linear)* (b) *BMR (linear)* (c) *U3OR (linear)* (d) *BoD5 (linear)*

(e) *U3M (log)* (f) *BMR (log)* (g) *U3OR (log)* (h) *BoD5 (log)*

Fig. 5. Feature matching performance of tested methods on four feature matching datasets.



(a) *Gaussian noise* (b) *Down-sampling* (c) *Different det.-desc.* (d) *Diff. input scales*

Fig. 6. Information in terms of the inlier ratio and the number of inliers of the input correspondence sets under different experimental configurations on the U3M dataset.

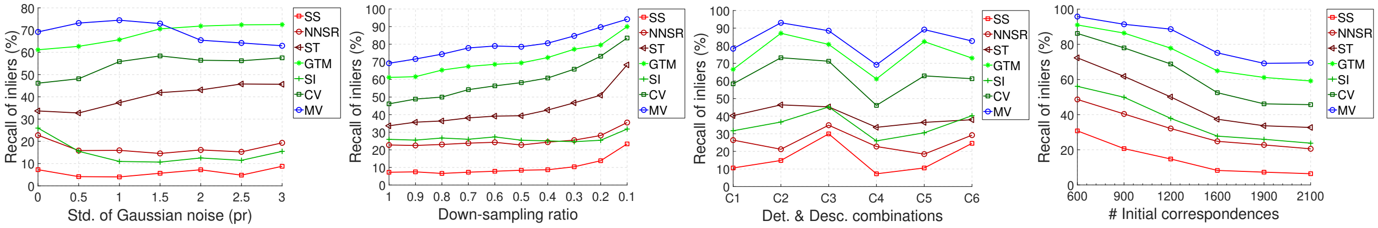
TABLE 3
Feature matching performance (%) on U3M.

	Precision	Recall	F-score
SS [43], [45]	4.81	63.39	8.94
NNSR [46]	9.67	41.63	15.69
ST [3]	4.31	**100.00**	8.26
GTM [6]	35.63	46.22	40.24
SI [47]	14.28	43.81	21.54
CV [7]	13.31	94.92	23.35
PointDSC [38]	9.09	0.23	0.45
SC^2-PCR [9]	**49.81**	20.74	29.29
TEASER++ [34]	1.03	0.56	0.73
MV	31.00	78.26	**44.41**

TABLE 4
Feature matching performance (%) on BMR.

	Precision	Recall	F-score
SS [43], [45]	2.68	54.41	5.11
NNSR [46]	4.41	29.11	7.66
ST [3]	2.89	**100.00**	5.62
GTM [6]	**23.07**	26.07	24.48
SI [47]	4.57	14.00	6.89
CV [7]	8.04	88.29	14.74
PointDSC [38]	5.43	86.03	10.21
SC^2-PCR [9]	3.11	**100.00**	6.03
TEASER++ [34]	19.13	42.53	26.39
MV	18.19	62.8	**28.21**

experiment is conducted on the U3M dataset, and the results of MV under different parameter settings are shown in



(a) *Varying noise* (b) *Density variation* (c) *Varying det.-desc.* (d) *Varying input scales*

Fig. 7. Robust performance of tested methods with respect to different nuisances on U3M.

TABLE 5
Feature matching performance (%) on U3OR.

	Precision	Recall	F-score
SS [43], [45]	2.40	80.53	4.66
NNSR [46]	6.42	63.45	11.66
ST [3]	1.69	**99.49**	3.32
GTM [6]	43.55	43.28	43.41
SI [47]	24.27	60.11	34.58
CV [7]	7.34	90.63	13.58
PointDSC [38]	3.84	0.27	0.50
SC^2-PCR [9]	62.34	45.14	52.36
TEASER++ [34]	0.19	1.30	0.34
MV	**64.62**	90.39	**75.36**

TABLE 6
Feature matching performance (%) on BoD5.

	Precision	Recall	F-score
SS [43], [45]	6.29	52.03	11.22
NNSR [46]	12.84	39.61	19.39
ST [3]	11.12	**100.00**	20.01
GTM [6]	**65.23**	50.18	56.72
SI [47]	27.27	41.80	33.01
CV [7]	37.96	99.98	55.03
PointDSC [38]	28.59	96.12	44.07
SC^2-PCR [9]	6.59	**100.00**	12.37
TEASER++ [34]	52.99	82.33	64.48
MV	55.12	94.39	**69.60**

Fig. 3.

It can be seen from the figure that changing t_α has a clear impact on the feature matching performance. Moreover, our adaptive threshold achieves the best performance.

Ablation study. To verify the necessity of using the nodal clustering coefficient to remove a portion of outliers, ablation experiments were conducted to compare the recall performance of MV with and without the nodal clustering coefficient calculation step. The experimental results are presented in Fig. 4.

On the four subsets of U3M, MV with nodal clustering coefficient calculation consistently achieves the best performance. This is because 1) the nodal clustering coefficient utilizes the geometric information of the cluster structures in the compatibility graph and has a strong discriminative power; 2) it reduces the impact of outliers on the subsequent mutual voting process, which results in a more convincing judgement on the correctness of correspondences.

4.1.3 Feature Matching Results

Feature matching results of tested score-based methods on experimental datasets are shown in Fig. 5. For each dataset, we give two figures with linear and logarithmic recall axes plots, which can more clearly reflect the performance gap with small and large values of K, respectively.

On all the experimental datasets, MV outperforms the others, indicating that our method can be generalized to different application scenarios and data modalities. On the U3OR dataset, MV has a more obvious gap over other methods. Table 1 shows that the average inlier ratio of U3OR is relatively low (8.09%), thus validating the robustness of MV to low inlier ratio. Fig. 8 visualizes the feature matching results of several tested methods.

To compare with label-based methods as well, we further present precision, recall, and F-score performance [1] of both label-based and score-based methods in Tables 3-6. For score-based methods, correspondence grouping is achieved using the OTSU technique. For PointDSC, we directly use the model trained on the 3DMatch dataset on object datasets to test its generalization ability. For SC^2-PCR, because it is a rigid transformation estimator, we use the correspondences that are consistent with its estimated transformation matrix as its judged inliers.

As shown in Tabels 3-6, MV is the best competitor in terms of the F-score performance on all datasets. The performance of PointDSC is very poor when applied to object datasets. The performance of all compared methods fluctuates significantly in cross-dataset experiments. By contrast, our MV achieves stable and outstanding performance in this case.

4.1.4 Robustness Results

The experiments in this section analyze the robustness of all compared methods on the U3M dataset in the presence of Gaussian noise, point density variation, different detector-descriptor combinations, and varying the number of initial feature matches. The impacts of these nuisances on the inputs are statistically shown in Fig. 6. Here, we set K to 100, and the results are shown in Fig. 7.

Gaussian noise. To simulate the real case with noisy point cloud data, different levels of Gaussian noise are added to the target point cloud during the experiments, and the standard deviation of Gaussian noise varies from 1.5 pr to 3.0 pr with a gap of 0.5 pr.

The results in Fig. 7(a) show that MV ranks first when the standard deviation is less than 1.5 pr, and is only inferior to

TABLE 7
Varying the number of input correspondences (ms).

# Initial correspondences	600	900	1200	1600	1900
SS [43], [45]	54.40	86.39	135.00	**278.64**	**319.18**
NNSR [46]	61.73	97.90	154.56	318.31	366.26
ST [3]	1946.24	3560.74	5983.80	15007.80	17611.20
GTM [6]	**36.63**	**65.65**	**112.99**	288.72	336.87
SI [47]	95.52	150.32	241.58	519.92	600.39
CV [7]	102.76	177.85	297.38	709.77	844.76
MV	39.75	73.52	133.09	399.95	510.14

TABLE 8
Varying the inlier ratio of input correspondences (ms).

Inlier ratio	0%∼5%	5%∼10%	10%∼15%	15%∼20%	20%∼25%	25%∼30%
SS [43], [45]	53.79	54.76	**51.23**	34.89	**41.00**	31.00
NNSR [46]	60.94	63.54	59.07	43.89	50.50	38.00
ST [3]	2981.73	3099.22	3060.30	2017.33	2548.50	1580.00
GTM [6]	50.93	**52.89**	52.77	**34.44**	43.33	**26.67**
SI [47]	107.18	112.05	106.07	80.78	93.33	72.33
CV [7]	145.16	151.95	147.43	97.44	130.33	80.00
MV	**35.86**	57.84	103.30	107.11	207.33	115.33

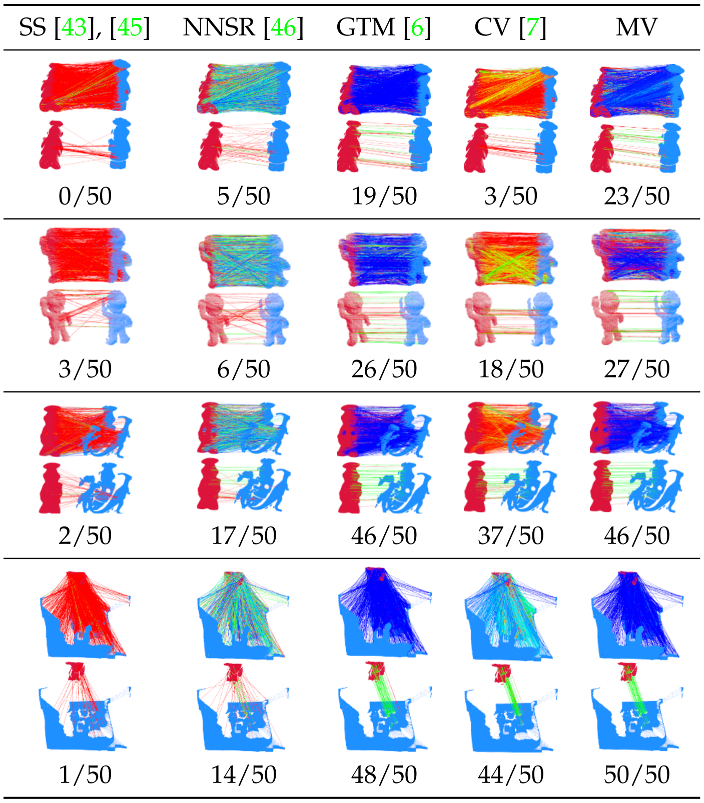


Fig. 8. Visual feature matching results of several tested methods. Each method has two visual results, where the top one is the scoring result rendered by pseudo-color (red→blue: high → low confidence scores) and the bottom one is the selection result (x/K indicates x inliers in the selected K correspondences). Green and red lines refer to correct and incorrect correspondences, respectively.)

GTM as the standard deviation further increases. This shows that MV is robust to different levels of noise.

Density variation. To investigate the effect of point cloud density variation on the performance of all methods, the target point cloud is down-sampled. The down-samping ratio varies from 1 to 0.1 with an interval of 0.1. The experimental results are shown in Fig. 7(b).

As witnessed by the figure, the performance of MV is more advantageous than other competitors. Moreover, MV is still able to achieve a 93% recall with 90% points removed.

Different detector-descriptor combinations. Different combinations of keypoint detectors and local feature descriptors generate different numbers and spatial distributions of initial correspondences. Two detectors, i.e., Harris3D [49] and ISS [50], and three local feature descriptors, i.e., SHOT [42], LFSH [45] and RCS [51], are used in the experiments, resulting a total of six different combinations. We use C1∼C6 to represent the combinations of Harris3D+SHOT, Harris3D+LFSH, Harris3D+RCS, ISS+SHOT, ISS+LFSH, and ISS+RCS, respectively. The experimental results are shown in Fig. 7(c).

The results suggest that MV is the best performer under all combinations. This indicates that MV is suitable for inputs generated from different detectors and descriptors.

Varying numbers of initial correspondences. Changing the scale of the input correspondences can be achieved by varying the non-maximum-suppression radius of the employed keypoint detector. The results are reported in Fig. 7(d).

When faced with inputs at different scales, our MV consistently achieves the best performance. The gap becomes more clear for inputs wither greater scales.

4.1.5 Time efficiency

To compare the time efficiency of tested feature matching methods, two cases are analyzed here: inputs with different magnitudes and different inlier ratios. For the former, the number of initial correspondences is set to 600, 900, 1200, 1600, and 1900; for the latter, the input magnitude is fixed to 1200 and point cloud pairs are divided into six groups, whose inlier ratios range from 0% to 30% with a step of 5%. The experiments were conducted on the U3M dataset and the average time costs for matching a single point cloud pair of tested methods are presented in Tables 7 and 8.

From Table 7, it can be observed that the time cost taken by each method tends to increase as the input correspondence magnitude increase. In particular, MV is a top-ranked performer when the number of input correspondences is smaller than 1600. From Table 8, it can be found that MV is the most efficient one when the inlier ratio is less than 5%. As the inlier ratio further increases, the time cost of MV improves but still remains efficient, because it is able to finish inlier selection with around 0.1 seconds when the inlier ratio is between 10% and 30%. With more inliers in the initial correspondence set, more edges could possess CNSs and the cardinality of CNS will increase, resulting in more time costs.

Overall, MV is an efficient method for 3D feature matching, and the recall performance of MV is clearly better than other compared methods.

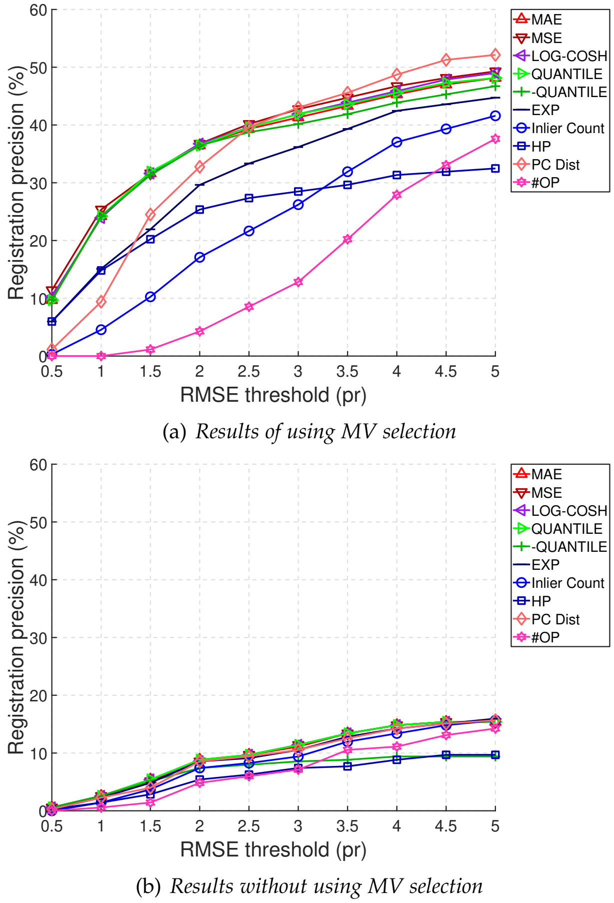
4.2 Point Cloud Registration Experiments

4.2.1 Experimental Setup

Datasets. We consider four datasets, i.e., the object-scale dataset U3M, the scene-scale indoor datasets 3DMatch [52] & 3DLoMatch [53], and scene-scale outdoor dataset KITTI [54]. 3DLoMatch is the subset of 3DMatch, where the

TABLE 9
3DMatch and 3DLoMatch test set statistics.

Scene	# Point Clouds	# Point Cloud Pairs of 3DMatch	# Point Cloud Pairs of 3DLoMatch
7-scenes-redkitchen	60	506	525
sun3d-home_at-home_at_scan1_2013_jan_1	60	156	289
sun3d-home_md-home_md_scan9_2012_sep_30	60	208	230
sun3d-hotel_uc-scan3	55	226	218
sun3d-hotel_umd-maryland_hotel1	57	104	158
sun3d-hotel_umd-maryland_hotel3	37	54	49
sun3d-mit_76_studyroom-76-1studyroom2	66	292	240
sun3d-mit_lab_hj-lab_hj_tea_nov_2_2012_scan1_erika	38	77	72
Total	433	1623	1781



(a) *Results of using MV selection*

(b) *Results without using MV selection*

Fig. 9. Registration performance of RANSAC-based methods with and without MV-selected correspondences when using different hypothesis evaluation metrics on U3M.

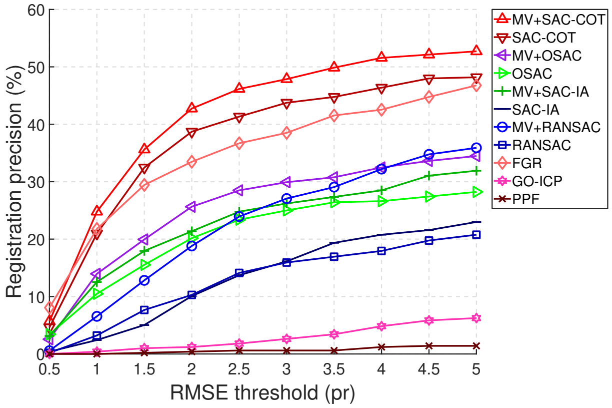


Fig. 10. Registration performance of tested point cloud registration methods on U3M.

overlap rate of the point cloud pairs ranges from 10% to 30%, which is very challenging. The statistics of 3DMatch and 3DLoMatch test set are shown in Table 9. For KITTI, we follow [9], [38] and obtain 555 pairs of point clouds for testing. We use both FPFH [55] (handcrafted descriptor) and FCGF [56] (learned descriptor) as feature descriptors for correspondence generation on scene-scale datasets.

Evaluation metric. We follow [57] that employs the root mean square error (RMSE) metric to evaluate the 3D point cloud registration performance on the object-scale dataset, e.g., U3M. Given the estimated rotation matrix $\mathbf{R}_{est}$ and translation vector $\mathbf{t}_{est}$, the point-wise error ε_{p} between two truly corresponding points $\mathbf{p}^s$ and $\mathbf{p}^t$ is defined as:

$$\varepsilon_{\mathrm{p}}(\mathbf{p}^s, \mathbf{p}^t) = ||\mathbf{R}_{est}\mathbf{p}^s + \mathbf{t}_{est} - \mathbf{p}^t||. \tag{13}$$

Then, the definition of RMSE is:

$$\mathrm{RMSE} = \sqrt{\sum_{\mathbf{p}^s,\mathbf{p}^t \in \mathbf{C}_{gt}} \frac{\varepsilon_{\mathrm{p}}^2\left(\mathbf{p}^s, \mathbf{p}^t\right)}{|\mathbf{C}_{gt}|}}, \tag{14}$$

where $\mathbf{C}_{gt}$ represents the ground-truth set of corresponding points between two point clouds. When the RMSE of a registration is smaller than a threshold t_{rmse}, we judge it as success.

Also, we employ the rotation error (RE) and translation error (TE) to evaluate the registration results on scene-scale dataset. Given the estimated rotation matrix $\mathbf{R}_{est}$ and ground-truth rotation matrix $\mathbf{R}_{gt}$, estimated translation vector $\mathbf{t}_{est}$ and ground-truth translation vector $\mathbf{t}_{gt}$, RE and TE can be defined as:

$$\mathrm{RE}(\mathbf{R}_{est}) = \arccos \frac{\mathrm{Tr}(\mathbf{R}_{est}{}^{\top}\mathbf{R}_{gt}) - 1}{2}, \tag{15}$$

$$\mathrm{TE}(\mathbf{t}_{est}) = ||\mathbf{t}_{est} - \mathbf{t}_{gt}||_2. \tag{16}$$

By referring to the settings in [35], the registration is considered successful when the RE $\leq$ 15°, TE $\leq$ 30 cm on 3DMatch & 3DLoMatch datasets, and RE $\leq$ 5°, TE $\leq$ 60 cm on KITTI dataset. For a dataset, we define its registration accuracy as the ratio of success cases to the total number of point cloud pairs to be registered.

TABLE 10
Registration results (%) on 3DMatch dataset under 1k correspondences setting. The symbol '-' denotes unavailable benchmark record, **bold** and underlining indicate the best and the second best results, respectively.

Descriptor	Method	Kitchen	Home1	Home2	Hotel1	Hotel2	Hotel3	Study Room	MIT Lab	Avg.
FPFH	*i) Traditional*									
	SM [3]	-	-	-	-	-	-	-	-	55.88
	FGR [58]	44.07	51.28	37.98	46.9	38.46	48.15	26.71	46.75	41.16
	RANSAC-1K [2]	60.87	71.15	53.37	67.26	58.65	70.37	44.52	51.95	58.60
	RANSAC-10K [2]	77.08	**82.69**	67.31	82.30	73.08	81.48	61.99	66.23	73.75
	GORE [33]	78.06	80.12	66.34	86.54	75.96	83.30	66.09	59.74	74.79
	TEASER++ [34]	-	-	-	-	-	-	-	-	76.27
	LTGV [8]	80.63	**82.69**	67.79	**86.73**	75.00	**85.19**	67.47	**67.53**	76.83
	ii) Deep learned									
	3DRegNet [36]	-	-	-	-	-	-	-	-	26.31
	DGR [35]	69.17	74.36	57.69	68.58	65.38	74.07	56.16	55.84	65.19
	PointDSC [38]	76.88	81.41	65.87	83.63	70.19	79.63	61.99	62.34	73.14
	MV	**82.21**	**82.69**	**71.15**	**86.73**	**78.85**	83.33	**69.18**	**67.53**	**78.25**
FCGF	*i) Traditional*									
	SM [3]	-	-	-	-	-	-	-	-	86.57
	RANSAC-1K [2]	71.15	72.44	58.17	80.97	71.15	72.22	63.36	63.64	69.25
	RANSAC-10K [2]	74.90	73.72	59.62	81.86	70.19	70.37	62.33	66.23	70.67
	TEASER++ [34]	-	-	-	-	-	-	-	-	86.07
	LTGV [8]	95.65	91.67	**76.44**	94.69	89.42	81.48	82.53	**75.32**	88.48
	ii) Deep learned									
	3DRegNet [36]	-	-	-	-	-	-	-	-	77.76
	DGR [35]	93.68	91.03	75.00	95.13	89.42	**85.19**	81.85	67.53	87.31
	PointDSC [38]	94.86	91.03	75.48	92.48	87.5	81.48	83.22	71.43	87.55
	MV	**96.64**	**93.59**	75.96	**95.58**	**93.27**	83.33	**84.93**	74.03	**89.71**

TABLE 11
Registration results (%) on 3DLoMatch dataset under 1k correspondences setting.

Descriptor	Method	Kitchen	Home1	Home2	Hotel1	Hotel2	Hotel3	Study Room	MIT Lab	Avg.
FPFH	*i) Traditional*									
	FGR [58]	0.38	2.77	4.78	1.83	1.27	4.08	0.42	1.39	1.74
	RANSAC-1K [2]	10.29	9.34	16.96	17.43	12.03	32.65	2.92	4.17	11.40
	RANSAC-10K [2]	23.24	15.92	27.83	25.69	18.35	38.78	6.25	11.11	20.16
	GORE [33]	29.90	21.45	29.56	44.59	26.58	**40.82**	11.25	11.11	26.50
	TEASER++ [34]	-	-	-	-	-	-	-	-	28.9
	LTGV [8]	33.10	22.30	36.90	45.00	30.70	34.10	12.30	**21.70**	30.38
	ii) Deep learned									
	DGR [35]	19.05	14.53	20.00	37.61	24.05	34.69	8.33	13.89	19.93
	PointDSC [38]	24.90	16.00	27.00	27.30	14.60	29.30	5.10	15.90	19.99
	MV	**34.20**	**24.50**	**37.80**	**47.80**	**32.10**	36.60	**13.60**	20.30	**32.17**
FCGF	*i) Traditional*									
	RANSAC-1K [2]	18.67	9.69	20.00	19.27	19.62	22.45	12.08	16.67	16.68
	RANSAC-10K [2]	18.48	10.03	23.91	17.89	19.62	20.41	7.92	13.89	16.28
	TEASER++ [34]	-	-	-	-	-	-	-	-	42.11
	LTGV [8]	55.10	39.70	50.50	57.90	**39.40**	**36.60**	40.70	36.20	48.29
	ii) Deep learned									
	DGR [35]	37.90	20.07	35.22	31.19	28.48	28.57	17.50	23.61	29.42
	PointDSC [38]	51.40	34.00	**52.30**	57.40	38.00	**36.60**	33.50	**40.60**	45.14
	MV	**56.40**	**40.80**	50.00	**59.30**	38.00	**36.60**	**41.10**	39.10	**48.91**

Implementation details. We follow RANSAC-based methods that estimate a registration pose from a set of correspondences to perform registration. More specifically, we perform correspondence selection using MV and employ the output of MV as the input of RANSAC estimator. By default, we use 5k RANSAC iterations to perform registration. Note that better estimators could also be considered, which are complementary to MV because MV's output is the input of rigid transformation estimators.

4.2.2 Results on U3M Dataset

We first compare the performance of RANSAC methods with and without MV, respectively. In particular, ten RANSAC hypothesis evaluation metrics investigated in [59] (MAE, MSE, LOG-COSH, EXP, QUANTILE, -QUANTILE, Inlier Count, HP, PC Dist, #OP) are considered. The RMSE threshold is varied from 0.5 pr to 5 pr with a step of 0.5 pr. The results are presented in Fig. 9.

Two observations can be made from the figure. First, significant performance boosting is achieved under all RANSAC hypothesis evaluation metrics. For instance, when t_{rmse} equals 5 pr, RANSAC with inlier count metric achieves more than 30 percentages improvement when equipped with our MV-selected correspondences than using the raw correspondence set as the input. Second, MV+RANSAC achieves more advanced performance with MAE, MSE, LOG-COSH, QUANTILE, -QUANTILE and PC Dist hypothesis metrics.

In addition, we perform a more extensive comparison in Fig. 10. Here, the following methods are tested, includ-

TABLE 12
Registration results on 3DMatch dataset under 5k correspondences setting.

	FPFH			FCGF		
	RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
i) Traditional						
SM [3]	55.88	2.94	8.15	86.57	2.29	7.07
FGR [58]	40.91	4.96	10.25	78.93	2.90	8.41
RANSAC-1M [2]	64.20	4.05	11.35	88.42	3.05	9.42
RANSAC-4M [2]	66.10	3.95	11.03	91.44	2.69	8.38
GC-RANSAC [10]	67.65	2.33	6.87	92.05	2.33	7.11
TEASER++ [34]	75.48	2.48	7.31	85.77	2.73	8.66
CG-SAC [62]	78.00	2.40	6.89	87.52	2.42	7.66
SC^2-PCR [9]	**83.73**	**2.18**	6.70	93.16	**2.09**	**6.51**
ii) Deep learned						
3DRegNet [36]	26.31	3.75	9.60	77.76	2.74	8.13
DGR [35]	32.84	2.45	7.53	88.85	2.28	7.02
DHVR	67.10	2.78	7.84	91.93	2.25	7.08
PointDSC [38]	72.95	**2.18**	**6.45**	91.87	2.10	6.54
MV	82.62	3.13	9.04	**93.47**	3.30	9.46

TABLE 13
Registration results on 3DLoMatch dataset under 5k correspondences setting.

	FPFH			FCGF		
	RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
i) Traditional						
RANSAC-1M [2]	0.67	10.27	15.06	9.77	7.01	14.87
RANSAC-4M [2]	0.45	10.39	20.03	10.44	6.91	15.14
TEASER++ [34]	35.15	4.38	10.96	46.76	4.12	12.89
SC^2-PCR [9]	**38.57**	4.03	10.31	58.73	**3.80**	**10.44**
ii) Deep learned						
DGR [35]	19.88	5.07	13.53	43.80	4.17	10.82
PointDSC [38]	20.38	4.04	10.25	56.20	3.87	10.48
MV	36.16	5.07	12.73	**59.18**	4.99	12.92

ing SAC-COT [57], MV+SAC-COT, OSAC [45], MV+OSAC, SAC-IA [55], MV+SAC-IA, RANSAC [2], MV+RANSAC, FGR [58], GO-ICP [60], and PPF [61], where the latter three are RANSAC-independent methods.

The results indicate that MV+SAC-COT achieves the best performance. Notably, MV significantly improves all tested RANSAC-fashion estimators, such as SAC-COT, OSAC, SAC-IA, and RANSAC.

4.2.3 Results on 3DMatch & 3DLoMatch Datasets

Feature matching are usually performed using the nearest neighbor search [1] or the mutual nearest neighbor search [58] in the descriptor space. The two options generate two settings on 3DMatch and 3DLoMatch datasets, i.e., “5k correspondences setting” and “1k correspondences setting”. By default, we use the 5k correspondence setting.

1k correspondences setting. The results under 1k correspondences setting are reported in Tables 10 and 11. The benchmark records under this setting are taken from [8].

The following conclusions can be drawn: 1) regardless of which descriptor is used, MV outperforms all compared methods on both 3DMatch and 3DLoMatch datasets, indicating its strong ability of registering indoor scene point clouds; 2) even compared with deep-learned methods, our MV still achieves better performance without any data training. Fig. 11 gives some visualization examples of the feature matching and registration results by MV on the 3DMatch dataset.

5k correspondences setting. Following [9], [38], results with 5k correspondences setting are shown in Tables 12 and 13. When using FCGF descriptor, our MV achieves the best performance on both 3DMatch and 3DLoMatch dataset, indicating that MV is very flexible. When using FPFH descriptor, MV is the second best one, being slightly inferior to SC^2-PCR.

TABLE 14
Registration results on KITTI dataset.

	FPFH			FCGF		
	RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
i) Traditional						
FGR [58]	5.23	0.86	43.84	89.54	0.46	25.72
TEASER++ [34]	91.17	1.03	17.98	94.96	0.38	**13.69**
RANSAC [2]	74.41	1.55	30.20	80.36	0.73	26.79
CG-SAC [62]	74.23	0.73	14.02	83.24	0.56	22.96
SC^2-PCR [9]	**99.28**	0.39	8.68	97.84	**0.33**	20.58
ii) Deep learned						
DGR [35]	77.12	1.64	33.10	96.90	0.34	21.70
PointDSC [38]	98.92	**0.38**	**8.35**	97.84	**0.33**	20.32
MV	98.92	0.56	10.82	**98.20**	0.35	20.41

TABLE 15
Registration results under different descriptor settings.

	3DMatch			3DLoMatch		
	RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
FCGF [56]	85.10	3.05	9.42	40.10	6.01	10.32
SpinNet [63]	88.60	2.12	6.86	59.80	3.78	10.75
Predator [53]	89.00	2.11	6.03	59.80	3.12	8.91
FCGF+MV	93.47 (8.37↑)	3.30	9.46	59.18 (19.08↑)	4.99	12.92
SpinNet+MV	95.13 (6.53↑)	2.18	7.01	70.63 (10.83↑)	3.82	11.88
Predator+MV	92.73 (3.73↑)	2.15	6.27	68.60 (8.80↑)	3.38	9.53

Different descriptor settings. In addition to FPFH and FCGF, more recent deep-learned descriptors such as SpinNet [63] and Predator [53] are combined with MV for evaluation.

As show in Table 15, applying MV for outlier rejection can greatly improve all tested methods. It indicates that MV has good generalization ability and can be integrated into deep-learned methods to boost their registration performance.

Potential improvements for MV. Our MV is flexible and still has a large room for improvement. For instance, we tried two different methods for graph construction in the MV pipeline, i.e., the second-order graph (SOG) in SC^2-PCR and the “ground-truth” graph (GTG). In particular, GTG indicates that edges only connect true inliers in the graph using the ground-truth information. It can be served as an ideal case for graph construction. The results are presented in Table 16.

It can be seen that MV+SOG effectively reduces RE and TE values, generating more accurate registrations. MV+GTG significantly improves the RR performance on 3DLoMatch dataset. It verifies that MV provides a very flexible pipeline and has a large room for further improvements.

4.2.4 Results on KITTI dataset

In Table 14, the results of DGR [35], PointDSC [38], TEASER++ [34], RANSAC [2], CG-SAC [62], SC^2-PCR [9] and MV are reported for comparison.

As shown by the table, in terms of the registration recall performance, MV presents the best and the second best (0.36% behind the best method) results with FPFH and FCGF descriptor settings, respectively. Note that outdoor

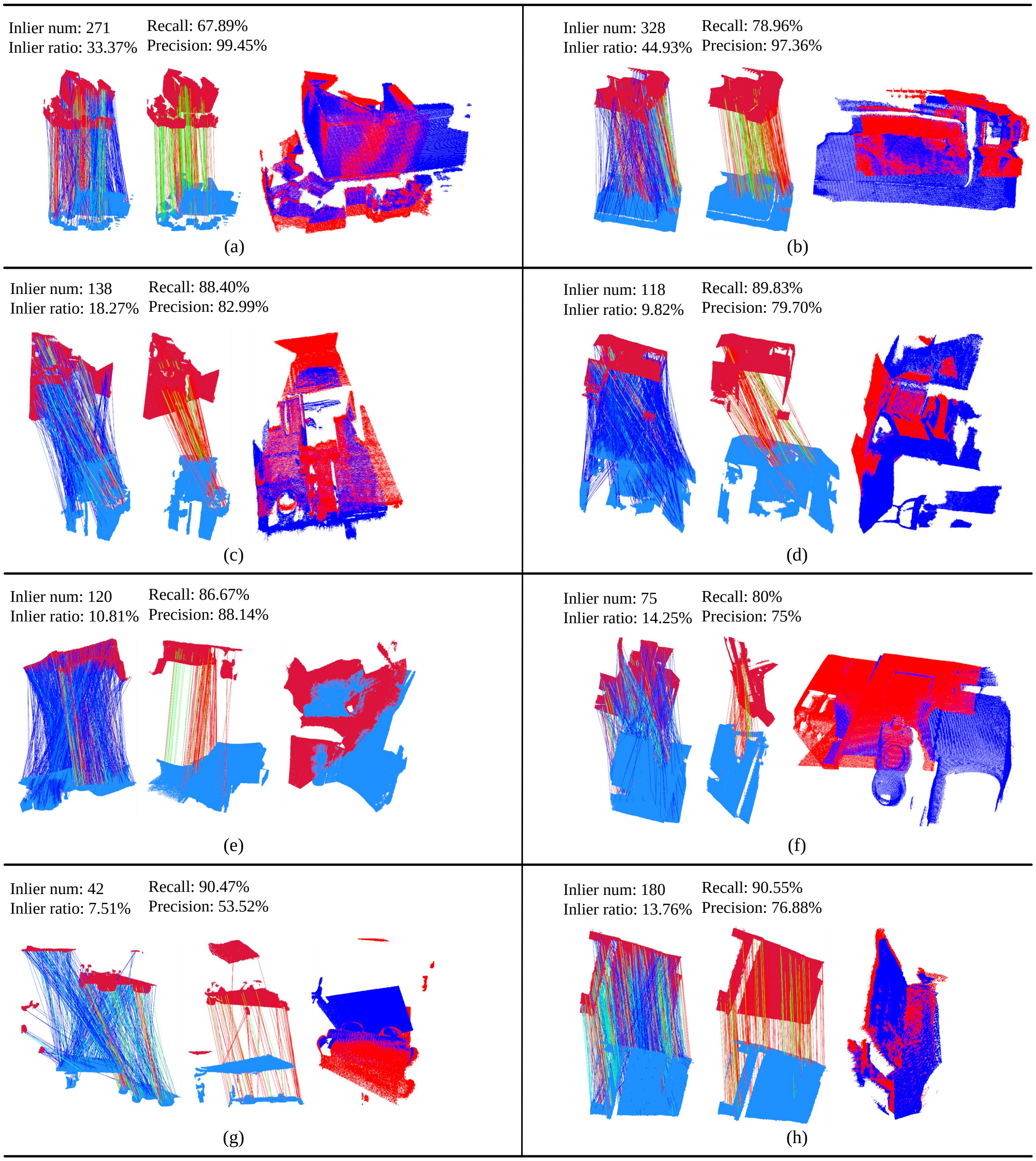


Fig. 11. Visual feature matching and registration results by MV on 3DMatch dataset. For each result, from left to right: initial correspondences rendered based on voting scores (red→ blue: high→low voting scores), selected correspondences by MV (green and red lines respectively denote correct and incorrect correspondences), and the registration result.

point clouds are significantly sparse and non-uniformly distributed. The registration experiments on object, indoor scene, and outdoor scene consistently verify that MV-selected correspondences can effectively boost point cloud registration performance in different application contexts.

4.3 3D Object Recognition Experiments

4.3.1 Experimental Setup

Datasets. The two datasets used for the experiments in this section are Queen [70] and U3OR [43], [44]. The Queen dataset contains 5 models and 84 scenes, and the U3OR dataset contains 5 models and 50 scenes. The two datasets are acquired through different sensing techniques and possess complex backgrounds, real noise, clutter and occlusion.

Evaluation metrics. We follow [30] and compute the ε_{res} and r_{ov} metrics. Assume that the transformation estimation matrix $\mathbf{M}_i = (\mathbf{R}_i, \mathbf{t}_i)$ is obtained in the i-th iteration of RANSAC, and the model point cloud $\mathbf{P}^s$ is transformed to obtain the point cloud $\mathbf{P}^{trans}$. Let $\mathbf{p}_i^t$ be the point of

TABLE 16
Potential improvements for MV. **SOG**: Construct a second-order compatibility graph instead. **GTG**: Construct a "ground truth" graph instead.

		3DMatch			3DLoMatch		
		RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
FPFH	MV	82.62	3.13	9.04	36.16	5.07	12.73
	MV+SOG	82.93	2.22	6.78	38.41	4.08	10.34
	MV+GTG	98.58	1.61	5.35	86.64	3.26	8.74
FCGF	MV	93.47	3.30	9.46	59.18	4.99	12.92
	MV+SOG	93.22	2.13	6.61	58.23	3.81	10.35
	MV+GTG	98.40	1.66	5.66	91.63	2.83	7.84

TABLE 17
Object recognition results on Queen dataset (%).

Method	Angle	Bigbird	Gnome	Kid	Zoe	Average
EM [64]	-	-	-	-	-	81.90
VD-LSD(SQ) [65]	89.70	**100.00**	70.50	84.60	71.80	83.80
(500 iterations)						
Spin image [66]	2.63	51.16	18.42	36.59	9.52	24.14
RoPS [30]	34.21	34.88	44.74	17.07	11.90	28.57
RCS [67]	63.16	76.74	86.84	82.93	57.14	74.38
VOID [68]	84.21	90.70	92.11	95.12	61.94	86.21
Spin image+MV	21.05	67.44	47.37	78.05	47.62	53.69 (29.55↑)
RoPS+MV	60.53	72.09	68.42	58.54	73.81	67.49 (38.92↑)
RCS+MV	92.11	88.37	**94.74**	95.12	92.86	94.58 (20.20↑)
VOID+MV	**94.74**	83.72	**94.74**	**100.00**	**95.24**	**97.04** (10.83↑)
(1000 iterations)						
Spin image [66]	2.63	48.84	18.42	43.90	19.05	27.09
RoPS [30]	39.47	39.53	50.00	24.39	11.90	33.00
RCS [67]	68.42	79.07	86.84	87.80	69.05	79.31
VOID [68]	84.21	83.72	**94.74**	90.24	71.43	88.18
Spin image+MV	23.68	67.44	50.00	85.37	42.86	56.65 (29.56↑)
RoPS+MV	63.16	74.42	68.42	63.41	69.05	68.47 (35.47↑)
RCS+MV	**92.11**	86.05	**94.74**	95.12	**95.24**	95.07 (15.76↑)
VOID+MV	**92.11**	74.42	**94.74**	**100.00**	**95.24**	**97.04** (8.86↑)
(2000 iterations)						
Spin image [66]	5.26	51.16	23.68	46.34	26.19	31.03
RoPS [30]	50.00	51.16	68.42	41.46	26.19	47.52
RCS [67]	71.05	83.72	86.84	95.12	69.05	82.76
VOID [68]	**94.74**	95.24	**100.00**	97.56	78.57	93.07
Spin image+MV	23.68	65.12	50.00	80.49	47.62	56.16 (25.13↑)
RoPS+MV	63.16	72.09	71.05	70.73	73.81	70.94 (23.42↑)
RCS+MV	92.11	79.07	94.74	**100.00**	92.86	95.57 (12.81↑)
VOID+MV	**94.74**	95.24	**100.00**	**100.00**	**97.62**	**97.52** (4.45↑)

$\mathbf{P}^s$, $\mathbf{p}_i^{trans}$ be the point of $\mathbf{P}^{trans}$ and the nearest neighbor to the point in the point cloud $\mathbf{P}^t$. If the distance $d(\mathbf{p}_i^t, \mathbf{p}_i^{trans}) = ||\mathbf{p}_i^t - \mathbf{p}_i^{trans}||$ is less than the threshold d_{rec}, which is set to 2 pr, $\mathbf{p}_i^{trans}$ will be classified into the point set $\mathbf{P}_{overlap}^{trans}$. The point cloud residual error ε_{res} and the point cloud overlap rate r_{ov} are defined as:

$$\varepsilon_{res} = \frac{\sum\limits_{\mathbf{p}_i^{trans} \in \mathbf{P}_{overlap}^{trans}} d(\mathbf{p}_i^{trans}, \mathbf{p}_i^t)}{|\mathbf{P}_{overlap}^{trans}|}, \tag{17}$$

$$r_{ov} = \frac{|\mathbf{P}_{overlap}^{trans}|}{|\mathbf{P}^t|}. \tag{18}$$

When the residual is less than the threshold d_{res} and the overlap rate is greater than the threshold $t_{overlap}$, the transformation estimation matrix $\mathbf{M}_i$ is considered to satisfy the condition and the iteration is stopped. Using this transformation matrix the model $\mathbf{P}^s$ can be identified from the scene $\mathbf{P}^t$. Following [68], we let d_{res} be 0.75 pr and $t_{overlap}$ be 0.04, or let d_{res} be 1.5 pr and $t_{overlap}$ be 0.2.

4.3.2 Comparative Results

For 3D object recognition, descriptor-based methods are main solutions [30]. We tested the recognition performance of several representative descriptors with and without MV-selected correspondences under different RANSAC iterations. Results on Queen and U3OR datasets are shown in Tables 17 and 18, respectively.

TABLE 18
Object recognition results on U3OR dataset (%).

Method	T-rex	Chef	Chicken	parasaurolophus	Average
RoPS [30]	-	-	-	-	98.90
TriLCI [69]	97.78	**100.00**	**100.00**	62.22	98.90
(500 iterations)					
Spin image [66]	35.56	92.00	62.50	40.00	58.51
RCS [67]	88.89	96.00	97.92	84.44	92.02
SHOT [42]	35.56	82.00	52.08	42.22	53.72
FPFH [55]	46.67	**100.00**	68.75	46.67	66.49
VOID [68]	97.78	**100.00**	**100.00**	95.56	98.40
Spin image+MV	68.89	**100.00**	75.00	57.78	76.06 (17.55↑)
RCS+MV	93.33	**100.00**	95.83	93.33	95.74 (3.72↑)
SHOT+MV	40.00	98.00	66.67	44.44	63.30 (9.58↑)
FPFH+MV	64.44	**100.00**	72.92	60.00	75.00 (8.51↑)
VOID+MV	**100.00**	**100.00**	**100.00**	**97.78**	**99.47** (1.07↑)
(1000 iterations)					
Spin image [66]	46.67	98.00	64.58	46.67	64.89
RCS [67]	93.33	**100.00**	95.83	91.11	95.21
SHOT [42]	44.44	84.00	62.50	46.67	60.11
FPFH [55]	44.44	**100.00**	66.67	51.11	66.49
VOID [68]	**97.78**	**100.00**	**100.00**	**97.78**	**98.94**
Spin image+MV	68.89	**100.00**	72.92	57.78	75.53 (10.41↑)
RCS+MV	95.56	**100.00**	97.92	93.33	96.81 (1.60↑)
SHOT+MV	44.44	**100.00**	70.83	44.44	65.96 (5.85↑)
FPFH+MV	60.00	**100.00**	72.92	60.00	73.94 (7.45↑)
VOID+MV	**97.78**	**100.00**	**100.00**	**97.78**	**98.94**
(2000 iterations)					
Spin image [66]	51.11	**100.00**	58.33	97.78	68.62
RCS [67]	**100.00**	**100.00**	93.75	97.78	97.87
SHOT [42]	90.00	48.89	37.78	58.33	59.57
FPFH [55]	**100.00**	53.33	48.89	68.75	68.62
VOID [68]	**100.00**	**100.00**	**100.00**	97.78	**99.47**
Spin image+MV	71.11	**100.00**	77.08	57.78	77.13 (8.51↑)
RCS+MV	**100.00**	95.56	97.78	**100.00**	98.40 (0.53↑)
SHOT+MV	**100.00**	51.11	48.89	77.08	70.21 (10.64↑)
FPFH+MV	**100.00**	55.56	62.22	70.83	72.87 (4.25↑)
VOID+MV	**100.00**	97.78	**100.00**	**100.00**	**99.47**

TABLE 19
Comparative object recognition results on Queen dataset (%), where IR denotes inlier ratio.

Method	Angle (IR=1.57%)	Bigbird (IR=1.54%)	Gnome (IR=2.57%)	Kid (IR=1.72%)	Zoe (IR=0.74%)	Average (IR=1.61%)
RCS+PointDSC	39.47	60.47	34.22	19.51	9.52	33.50
RCS+TEASER++	86.84	60.47	31.58	24.39	9.52	43.35
RCS+SC^2-PCR	42.11	58.14	28.95	29.27	11.98	34.98
RCS+MV	**92.11**	**86.05**	**94.74**	**95.12**	**95.24**	**95.07**

It can be seen that MV dramatically improves the 3D object recognition performance on all datasets under all tested descriptors. This phenomenon is more salient with a lower number of RANSAC iterations. On the Queen dataset, MV achieves the most significant improvement for RoPS, with a 38.92% improvement under 500 RANSAC iterations; it also has a 10.83% improvement for the best performing VOID descriptor with 500 iterations. On the U3OR dataset, MV achieves a 17.55% improvement for spin image with 500 iterations and a 10.64% improvement for SHOT with 2000 iterations. The results suggest that MV holds good generalization ability and can adapt to different descriptors. Fig. 12 visualizes several 3D object recognition results by MV and other competitors.

We also tested the object recognition performance of three compared feature matching methods, i.e., PointDSC, TEASER++, and SC^2-PCR. In this experiment, 1000 RANSAC iterations are performed on the correspondence subsets given by these methods. Results are shown in Tables 19 and 20.

One can see that the results of the three compared methods are not satisfactory, mainly due to the extremely low inlier ratio of initial correspondences in the object recog-

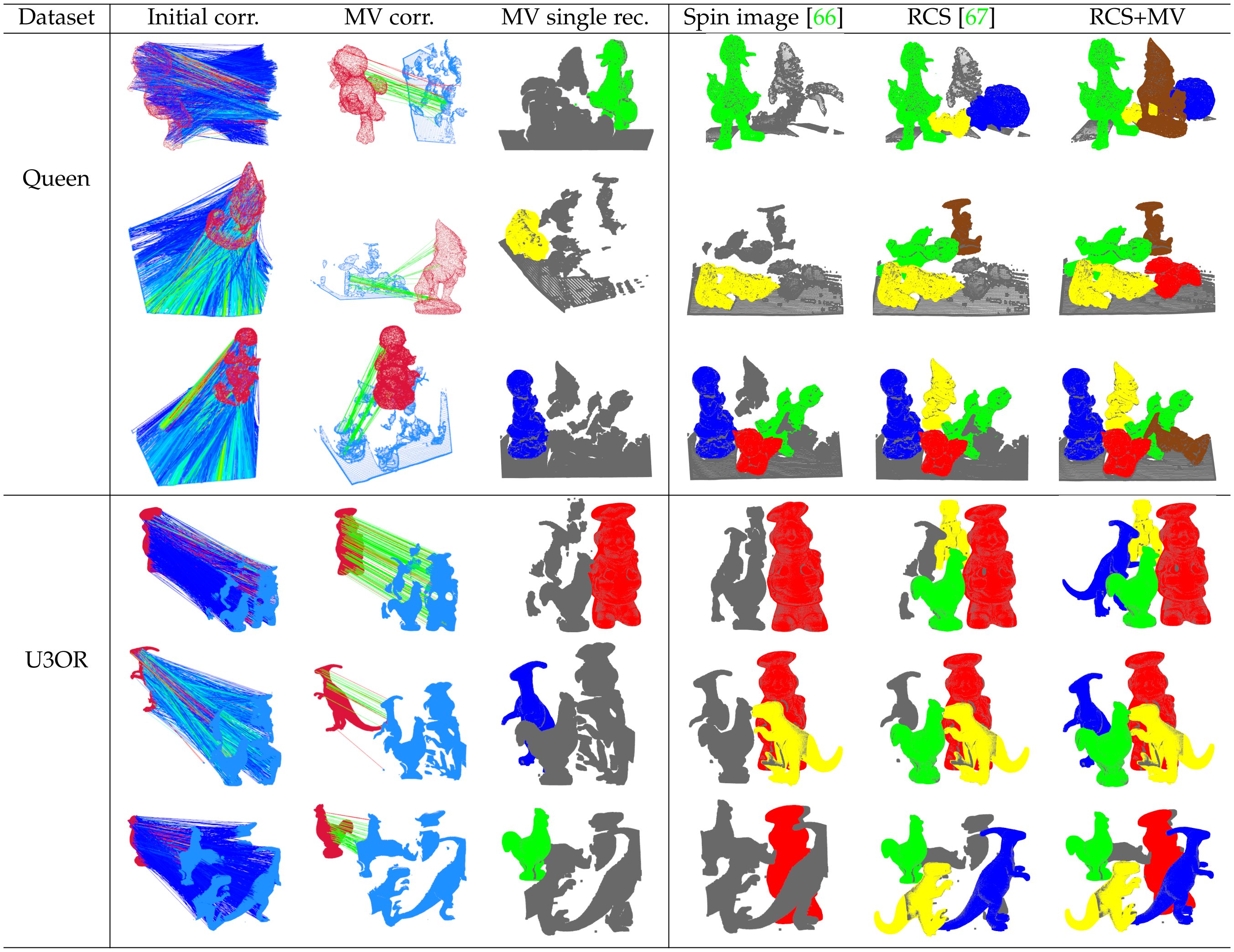


Fig. 12. Visualization of 3D object recognition results on Queen and U3OR datasets. From left to right: initial feature correspondences (red→blue: high→low voting scores), MV-selected correspondences (green and red lines respectively denote correct and incorrect correspondences), single-object recognition result by RCS+MV, multi-object recognition results by spin image, RCS, and RCS+MV, respectively.

TABLE 20
Comparative object recognition results on U3OR dataset (%).

Method	T-rex (IR=0.94%)	Chef (IR=3.06%)	Chicken (IR=1.46%)	parasaurolophus (IR=1.18%)	Average (IR=1.69%)
Spin image+PointDSC	0	0	0	2.22	0.53
Spin image+TEASER++	4.44	30.00	8.33	8.89	13.30
Spin image+SC^2-PCR	4.44	**100.00**	12.5	8.89	32.98
Spin image+MV	**68.89**	**100.00**	**72.92**	**57.78**	**75.53**

nition scenario. For instance, only 1.61% inliers are found in the initial correspondence sets on the Queen dataset. In this case, SC^2-PCR, which achieves outstanding performance on the 3DMatch and 3DLoMatch datasets, is not able to boost the 3D object recognition performance. This indicates that these compared methods can hardly handle cases with extremely low inlier ratios. On the contrary, MV manages to improve the 3D object recognition performance with scarce inliers.

5 Conclusions

In this paper, we presented a novel mutual voting method for ranking 3D correspondences. It reliably assigns a voting sore to each correspondence by refining both voters and candidates in a mutual voting scheme. Feature matching, 3D point cloud registration, and 3D object recognition experiments on various datasets with different challenges and modalities verify two conclusions: 1) MV is robust to heavy outliers under different challenging settings; 2) MV can significantly boost 3D point cloud registration and 3D object recognition performance with existing pipelines.

In the future, we plan to further investigate the following two problems. *1) Compatibility metric:* MV is based on the compatibility graph, which is built upon the compatibility scores between every two correspondences; thus, developing more advanced compatibility metrics may further improve MV's performance. *2) End-to-end voting.* At present, MV-selected correspondences still need an estimator (e.g., RANSAC) for 3D point cloud registration and object recognition; we wish to investigate an end-to-end voting process that directly computes a rigid transformation from raw correspondences with outliers.

Acknowledgments

The authors would like to thank the publishers of datasets used in our experiments for making them publicly

available.

References

[1] J. Yang, K. Xian, P. Wang, and Y. Zhang, "A performance evaluation of correspondence grouping methods for 3d rigid data matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1859–1874, 2019.

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[3] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2005, pp. 1482–1489.

[4] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *Pacific-Rim Symposium on Image and Video Technology*. IEEE, 2010, pp. 349–355.

[5] A. Glent Buch, Y. Yang, N. Kruger, and H. Gordon Petersen, "In search of inliers: 3d correspondence by local and global voting," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 2067–2074.

[6] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello, "A scale independent selection process for 3d object recognition in cluttered scenes," *International Journal of Computer Vision*, vol. 102, no. 1, pp. 129–145, 2013.

[7] J. Yang, Y. Xiao, Z. Cao, and W. Yang, "Ranking 3d feature correspondences via consistency voting," *Pattern Recognition Letters*, vol. 117, pp. 1–8, 2019.

[8] J. Yang, J. Chen, S. Quan, W. Wang, and Y. Zhang, "Correspondence selection with loose-tight geometric voting for 3d point cloud registration," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[9] Z. Chen, K. Sun, F. Yang, and W. Tao, "Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 13 221–13 231.

[10] D. Barath and J. Matas, "Graph-cut ransac," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 6733–6741.

[11] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 124–133.

[12] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. European Conference on Computer Vision*, 2008, pp. 596–609.

[13] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 398–405.

[14] F. Zhou and F. De la Torre, "Factorized graph matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 127–134.

[15] C. Olsson, O. Enqvist, and F. Kahl, "A polynomial-time bound for matching and registration with outliers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[16] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 1295–1302.

[17] H. Wang, G. Xiao, Y. Yan, and D. Suter, "Mode-seeking on hypergraphs for robust geometric model fitting," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 2902–2910.

[18] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Proc. ACM on International Conference on Multimedia Retrieval*. ACM, 2016, pp. 167–174.

[19] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 4181–4190.

[20] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1706–1721, 2014.

[21] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 512–531, 2019.

[22] X. Jiang, J. Ma, J. Jiang, and X. Guo, "Robust feature matching using spatial clustering with heavy outliers," *IEEE Transactions on Image Processing*, vol. 29, pp. 736–746, 2019.

[23] C. Zhao, Z. Cao, J. Yang, K. Xian, and X. Li, "Image feature correspondence selection: A comparative study and a new contribution," *IEEE Transactions on Image Processing*, vol. 29, pp. 3506–3519, 2020.

[24] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 2666–2674.

[25] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, "Nm-net: Mining reliable neighbors for robust feature correspondences," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 215–224.

[26] W. Zheng, Z. Shi, G. Xiao, and J. Ma, "Learning two-view correspondences and geometry using neighbor-aware network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[27] L. Dai, Y. Liu, J. Ma, L. Wei, T. Lai, C. Yang, and R. Chen, "Ms2dgnet: Progressive correspondence learning via multiple sparse semantics dynamic graph," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 8973–8982.

[28] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "Acne: Attentive context normalization for robust permutation-equivariant learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 11 286–11 295.

[29] A. S. Mian, M. Bennamoun, and R. A. Owens, "Automatic correspondence for 3d modeling: an extensive review," *International Journal of Shape Modeling*, vol. 11, no. 02, pp. 253–291, 2005.

[30] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3d local surface description and object recognition," *International Journal of Computer Vision*, vol. 105, no. 1, pp. 63–86, 2013.

[31] H. Sahloul, S. Shirafuji, and J. Ota, "An accurate and efficient voting scheme for a maximally all-inlier 3d correspondence set," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2287–2298, 2020.

[32] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252–1262, 2007.

[33] A. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2868–2882, 2017.

[34] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[35] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 2514–2523.

[36] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 7193–7203.

[37] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[38] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2021, pp. 15 859–15 869.

[39] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[40] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[41] A. S. Mian, M. Bennamoun, and R. A. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 19–40, 2006.

[42] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[43] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1584–1601, 2006.

[44] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, vol. 89, no. 2, pp. 348–361, 2010.

[45] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3d point cloud registration," *Information Sciences*, vol. 346, pp. 163–179, 2016.

[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[47] A. G. Buch, Y. Yang, N. Krüger, and H. G. Petersen, "In search of inliers: 3d correspondence by local and global voting," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 2075–2082.

[48] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1–4.

[49] I. Sipiran and B. Bustos, "Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes," *The Visual Computer*, vol. 27, no. 11, pp. 963–976, 2011.

[50] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Proc. International Conference on Computer Vision Workshops*. IEEE, 2009, pp. 689–696.

[51] J. Yang, Q. Zhang, K. Xian, Y. Xiao, and Z. Cao, "Rotational contour signatures for robust local surface description," in *Proc. IEEE International Conference on Image Processing*. IEEE, 2016, pp. 3598–3602.

[52] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1802–1811.

[53] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4267–4276.

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

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

[56] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE International Conference on Computer Vision*, 2019, pp. 8958–8966.

[57] 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, 2021.

[58] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[59] J. Yang, Z. Huang, S. Quan, Q. Zhang, Y. Zhang, and Z. Cao, "Toward efficient and robust metrics for ransac hypotheses and 3d rigid registration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 893–906, 2021.

[60] 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, 2015.

[61] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Ieee, 2010, pp. 998–1005.

[62] 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.

[63] 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 Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 753–11 762.

[64] J. Novatnack and K. Nishino, "Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images," in *Proc. European Conference on Computer Vision*. Springer, 2008, pp. 440–453.

[65] B. Taati and M. Greenspan, "Local shape descriptor selection for object recognition in range data," *Computer Vision and Image Understanding*, vol. 115, no. 5, pp. 681–694, 2011.

[66] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[67] J. Yang, Q. Zhang, K. Xian, Y. Xiao, and Z. Cao, "Rotational contour signatures for both real-valued and binary feature representations of 3d local shape," *Computer Vision and Image Understanding*, vol. 160, pp. 133–147, 2017.

[68] J. Yang, S. Fan, Z. Huang, S. Quan, W. Wang, and Y. Zhang, "Void: 3d object recognition based on voxelization in invariant distance space," *The Visual Computer*, pp. 1–17, 2022.

[69] W. Tao, X. Hua, K. Yu, X. Chen, and B. Zhao, "A pipeline for 3-d object recognition based on local shape description in cluttered scenes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 801–816, 2020.

[70] A. Petrelli and L. Di Stefano, "On the repeatability of the local reference frame for partial shape matching," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2244–2251.

Jiaqi Yang received his B.S. degree and PhD degree from Huazhong University of Science and Technology, China in 2014 and 2019, respectively. Sponsored by China Scholarship Council, he visited the GRASP laboratory, University of Pennsylvania from 2017 to 2018. Currently he is an associate professor at the School of Computer Science, Northwestern Polytechnical University. His research interests include local geometric description, 3D registration, 3D feature matching, and 3D object recognition.

He served as a reviewer for several international journals and conferences, including the Pattern Recognition, Computer Vision and Image Understanding, International Journal of Remote Sensing, International Conference of Computer Vision, and International Conference of 3D Vision.

Xiyu Zhang received his B.S. degree from North-western Polytechnical University in 2022. He is currently pursuing the Master's degree at School of Computer Science, Northwestern Polytechnical University.

His research interests include 3D feature matching and 3D point cloud registration.

Shichao Fan received his B.S. degree from Nanchang Hangkong University, Nanchang, China in 2019. He is currently pursuing the M.S. degree at the School of Computer Science, Northwestern Polytechnical University.

His research interests include local geometric description, and 3D object recognition.

Chunlin Ren received his B.S. degree from North-western Polytechnical University in 2022. He is currently pursuing the Master's degree at School of Computer Science, Northwestern Polytechnical University.

His research interests include multi-view stereo.

Yanning Zhang received the B.S. degree from the Department of Electronic Engineering, Dalian University of Technology, Dalian, China, in 1988, the M.S. degree from the School of Electronic Engineering, and the Ph.D. degree from the School of Marine Engineering, North-western Polytechnical University, Xian, China, in 1993 and 1996, respectively. She is currently a Professor at the School of Computer Science, Northwestern Polytechnical University. Her current research interests include computer vision and pattern recognition, image and video processing, and intelligent information processing. Dr. Zhang was the Organization Chair of the Asian Conference on Computer Vision 2009, and served as the program committee chairs of several international conferences.