澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

An Mutual Voting method for ranking 3D correspondences

翁霄羽
wengxiaoyu2009@163.com

Introduction

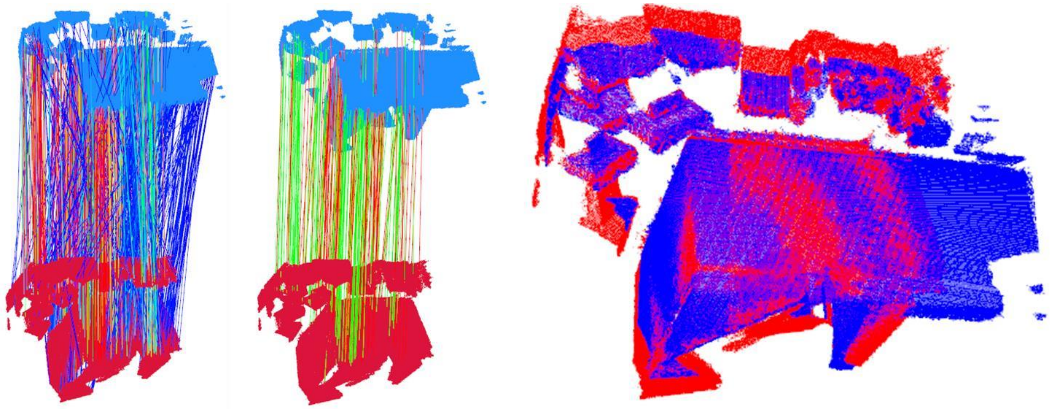
Yang J, Zhang X, Fan S, et al.

Mutual Voting for Ranking 3D Correspondences[J].

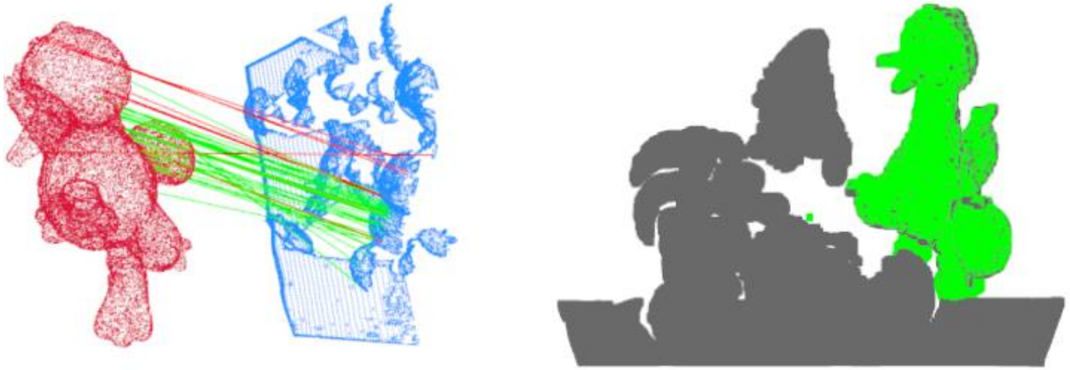
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

Introduction

why



Visual feature matching and registration results by MV



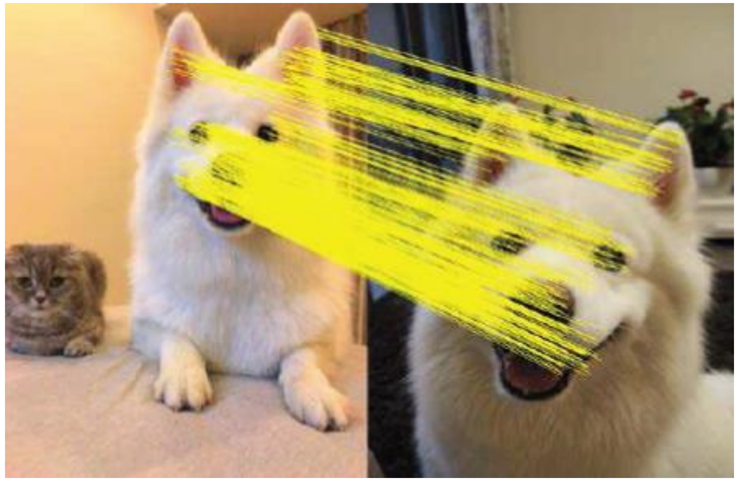
Visualization of 3D object recognition results
MV-selected correspondences
Single object recognition result by RCS+MV

MV can effectively boost the performance of correspondence-based down-stream tasks.
The selected correspondences are feed 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.

Introduction

Feature Matching

Purpose： Through the difference of descriptors to match the feature points in the two graphs.



2D Feature Matching： The random sampling consistency (RANSAC) method

3D Feature Matching： SC2-PCR ,MV

Introduction

RANSAC

The RANSAC 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.

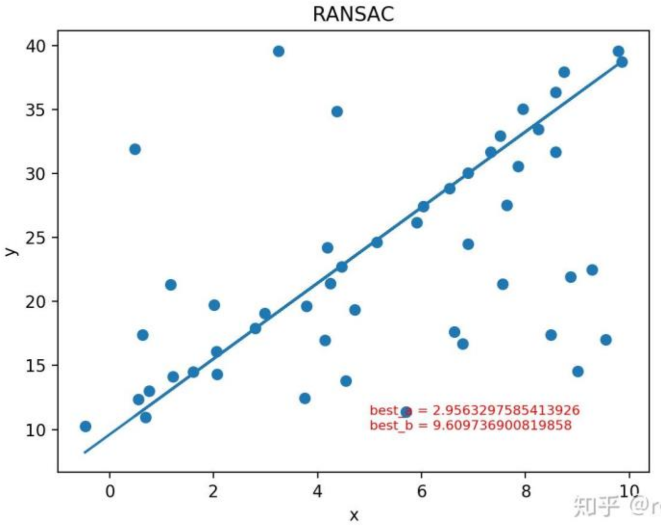
Step:

1.假定模型，并随机抽取N个样本点，对模型进行拟合

2.由于不是严格线性，数据点都有一定波动，假设容差范围为：sigma，找出距离拟合曲线容差范围内的点，并统计点的个数

3.重新随机选取Nums个点，重复第一步~第二步的操作，直到结束迭代

4.每一次拟合后，容差范围内都有对应的数据点数，找出数据点个数最多的情况，就是最终的拟合结果



RANSAC拟合曲线效果

Disadvantage : RANSAC suffers from limited accuracy and efficiency in the presence of heavy outliers. 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

Introduction

SC2-PCR

一种以RANSAC思想为基础的点云配准方法。流程如下图，先输入两组点云的Correspondence匹配，使用二阶空间兼容矩阵SC2构建每对Correspondence的相似度矩阵，再从中选择可靠的种子点，对于每个种子点还选择一些附近的Correspondence构成一个一致性集合，对每个一致性集合使用加权SVD计算出变换矩阵，再从中选择效果最好的变换。

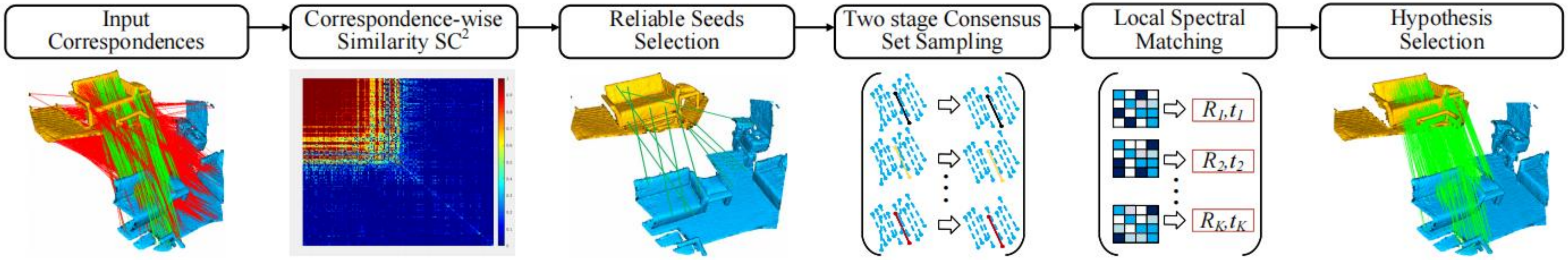


Figure 2. Pipeline of our method. **1.** Computing correspondence-wise second order spatial compatibility measure. **2.** Selecting some reliable correspondences as seeds. **3.** Performing the two-stage sampling around each seed. **4.** Performing local spectral matching to generate an estimation of R and t for each seed. **5.** Selecting the best estimation as final result.

Mutual Voting

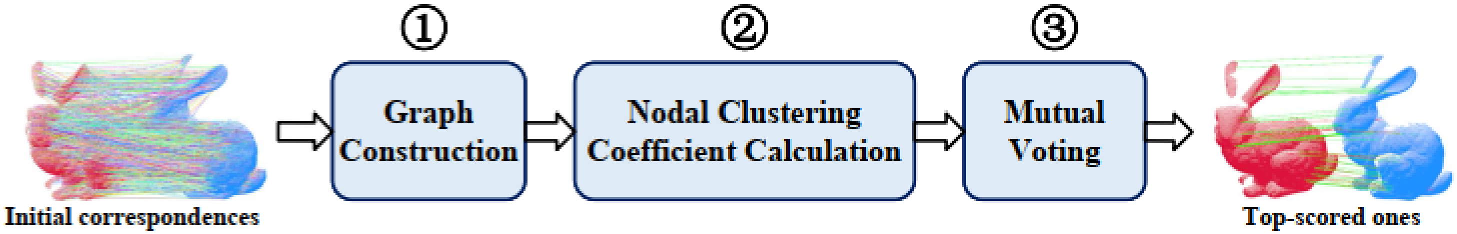
MV differs from other methods: Compared with existing methods, MV has two distinctions.

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.

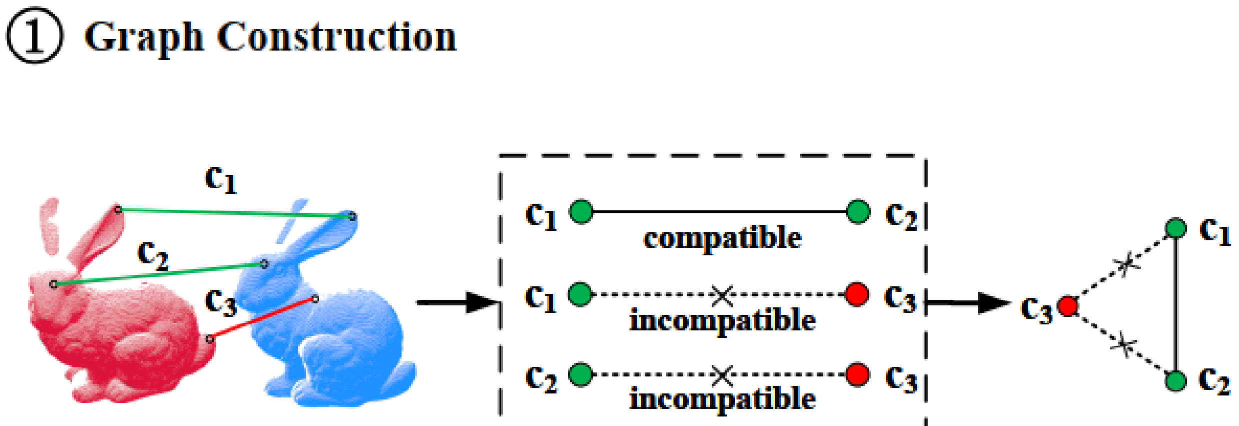
Without massive training data and GPU computing, MV yields even better performance than deep-learned ones and achieves pleasurable performance in both 3D point cloud registration and 3D object recognition applications.

There are mainly four steps involved: graph construction, nodal clustering coefficients calculation, mutual voting, and correspondence ranking.

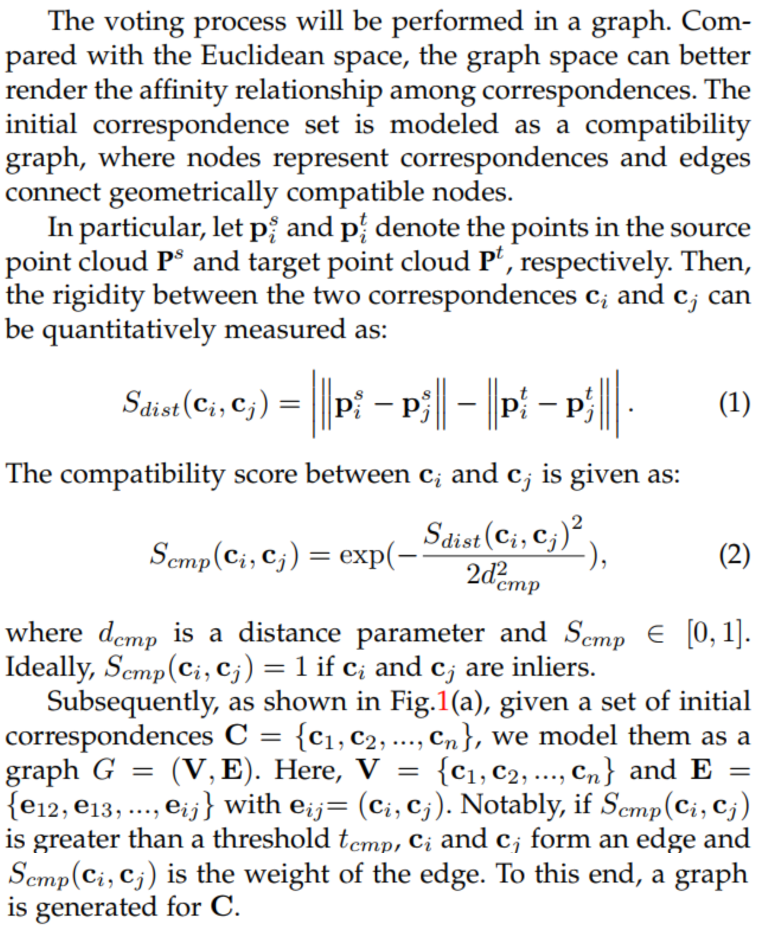


Mutual Voting

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.



(a) Graph construction rule



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:

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

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

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

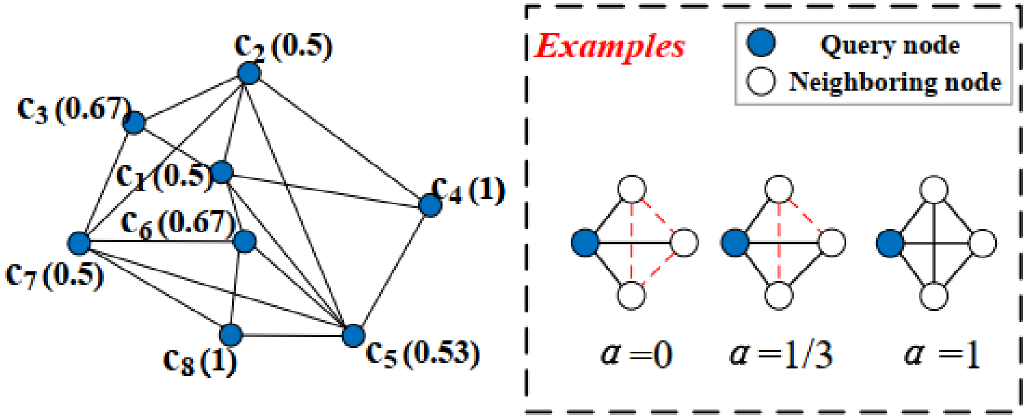
where d_{cmp} is a distance parameter and $S_{cmp} \in [0, 1]$. Ideally, $S_{cmp}(\mathbf{c}_i, \mathbf{c}_j) = 1$ if $\mathbf{c}_i$ and $\mathbf{c}_j$ are inliers.

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 $\mathbf{C}$.

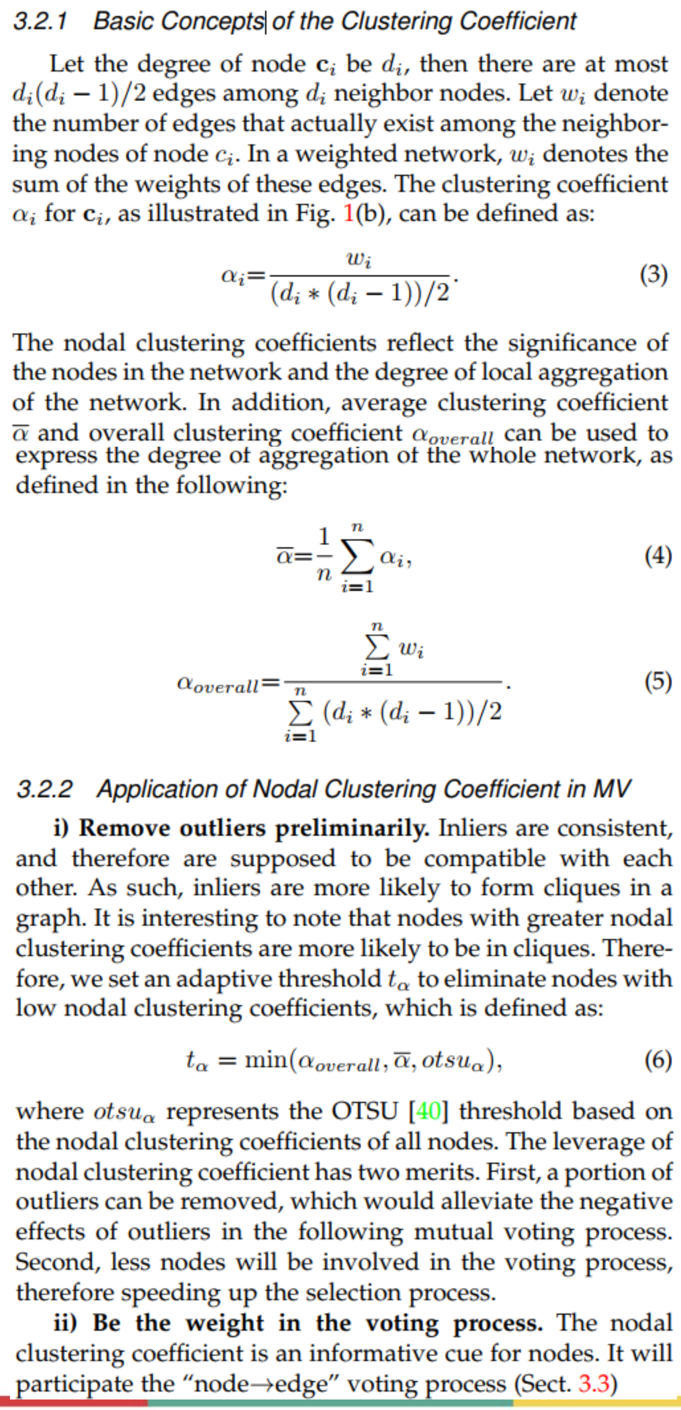
Mutual Voting

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.

② Nodal Clustering Coefficient Calculation



(b) Nodal clustering coefficient



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:

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

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

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:

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

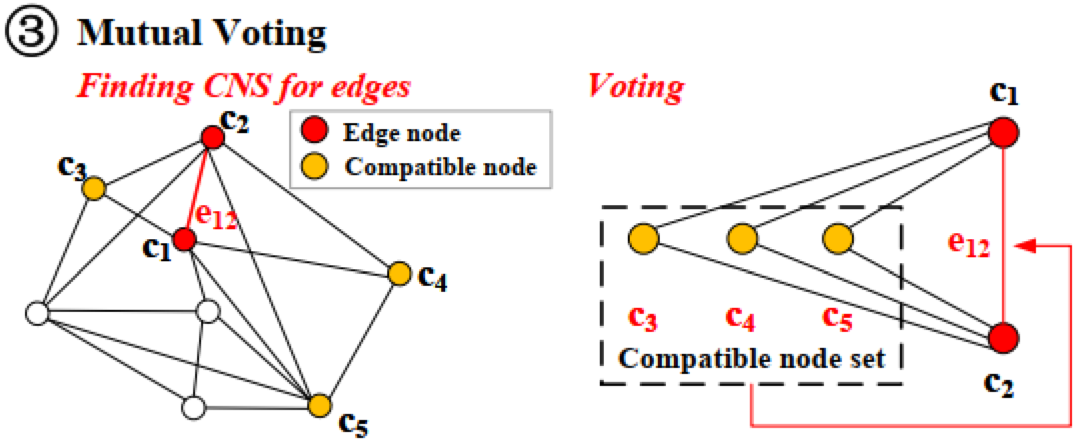
where $otsu_\alpha$ represents the OTSU [40] threshold based on the nodal clustering coefficients of all nodes. The leverage of nodal clustering coefficient has two merits. First, a portion of outliers can be removed, which would alleviate the negative effects of outliers in the following mutual voting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

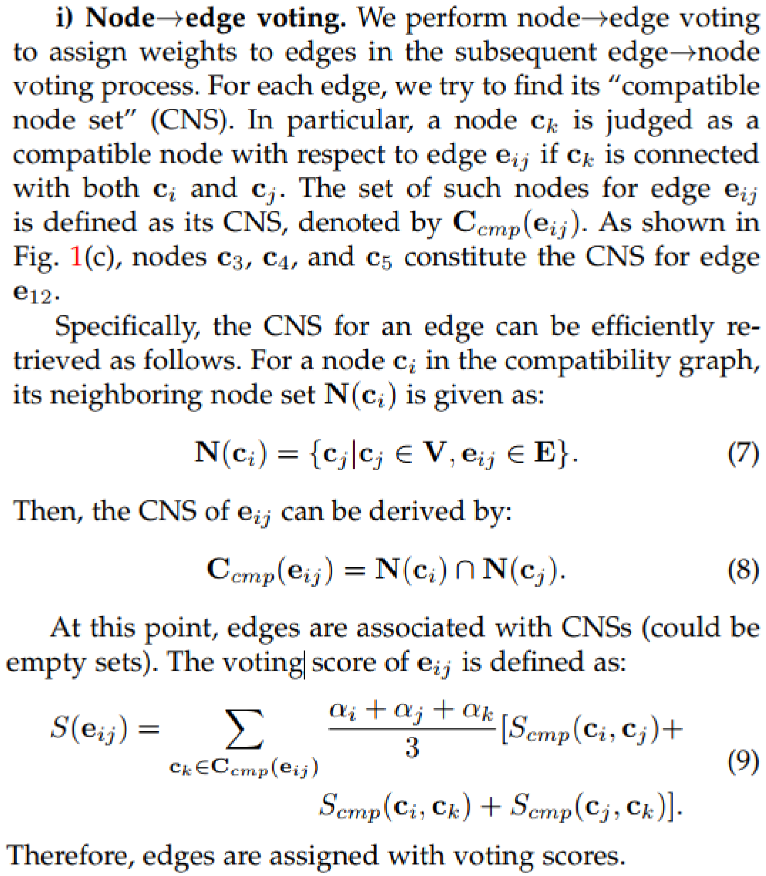
Mutual Voting

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.

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. “node↔edge” mutual voting, is designed for scoring correspondences.



(c) Node→Edge voting



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:

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

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

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

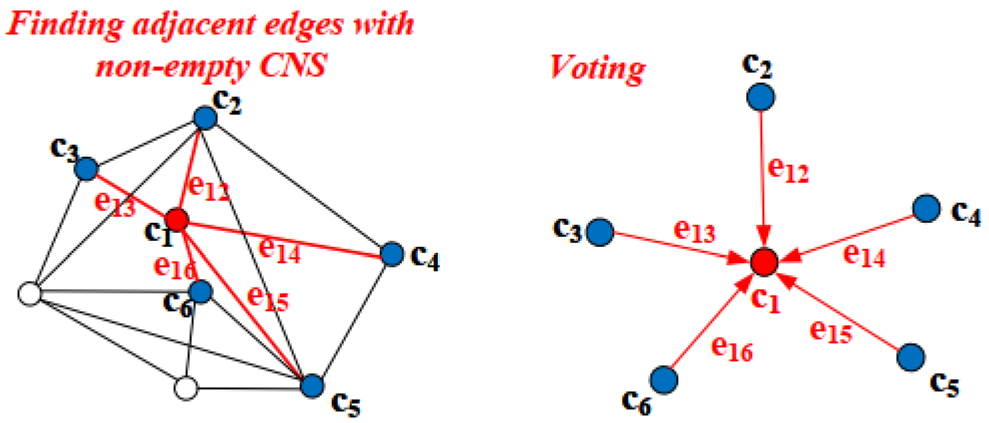
At this point, edges are associated with CNSs (could be empty sets). The voting score of $\mathbf{e}_{ij}$ is defined as:

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

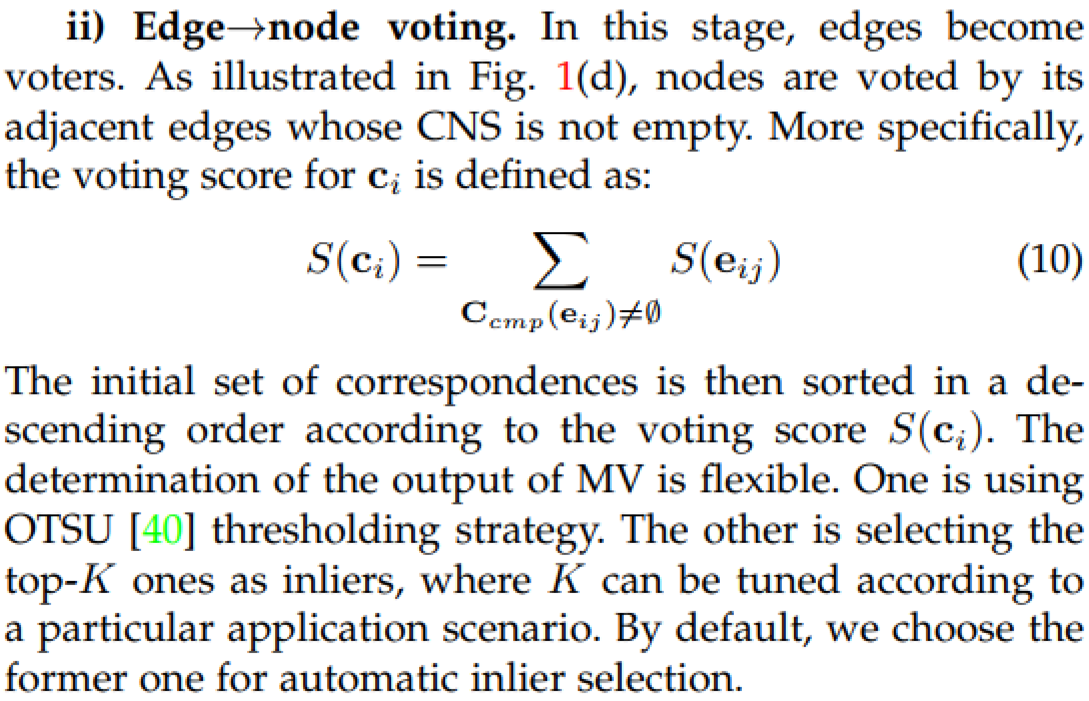
Therefore, edges are assigned with voting scores.

Mutual Voting

Mutual voting



(d) Edge→Node voting



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.

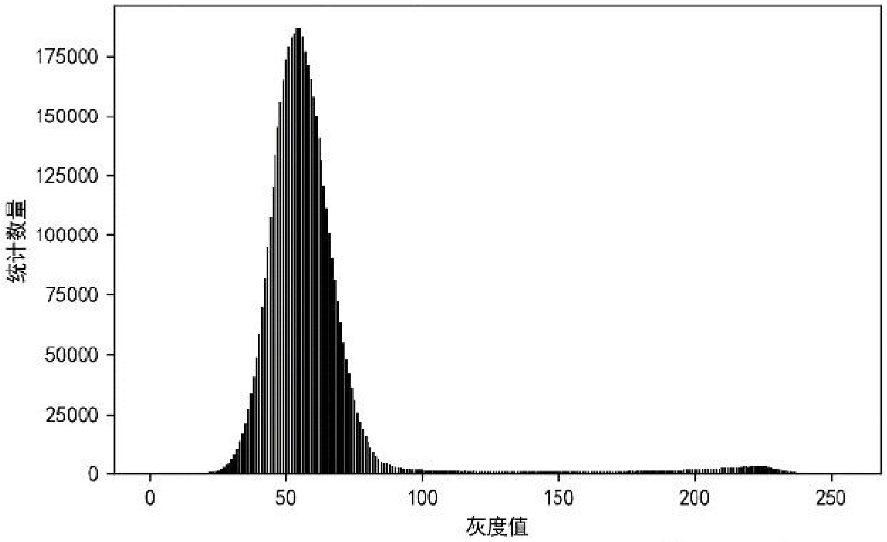
Mutual Voting

OTSU简介

OTSU算法，又被称为最大类间方差法（大津算法）是一种确定阈值的算法，是由日本学者大津展之于1979 年提出的。该方法常用于图像进行二值分割时的自适应阈值计算。

它是按图像的灰度分布特性,将图像分成背景(background)和目标(object)两部分。分割的依据是两类之间的间类方差最大，即类别内的差异最小化。

如下图所示，假定图像包含前景和背景两类像素，则其灰度直方图为双峰直方图，然后计算使得两类像素能分开的类内方差或等价的类间方差作为最佳阈值，并以此作为分割依据。



图· 指数图像灰度直方图

Mutual Voting

Computational Complexity Analysis

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 $O(n2)$;

second, calculating the clustering coefficients for all nodes indicates a computational compatibility $O(n3)$;

third, finding the CNSs for edges and the mutual voting process have computational complexities of $O(n3)$ and $O(n2)$, 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 $O(n2)+O(n3)+O(n3)+ O(n2) + O(n\log(n)) = O(n3)$.

Experiment- Feature Matching Experiments

Dataset

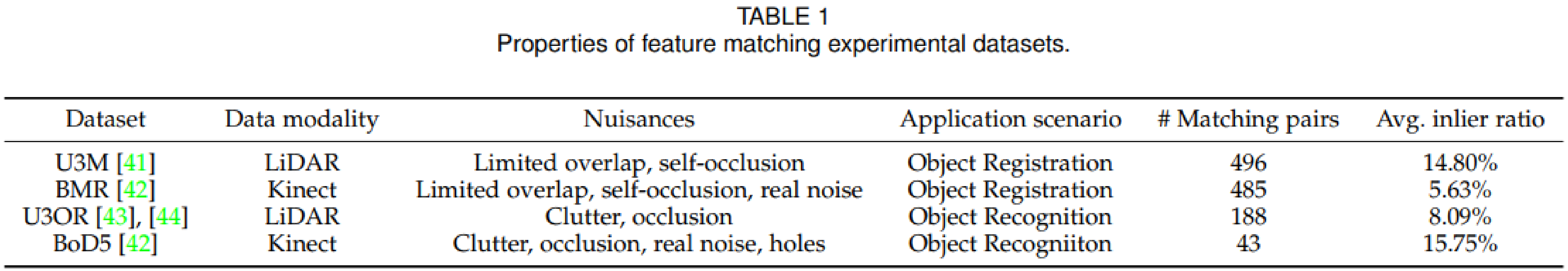
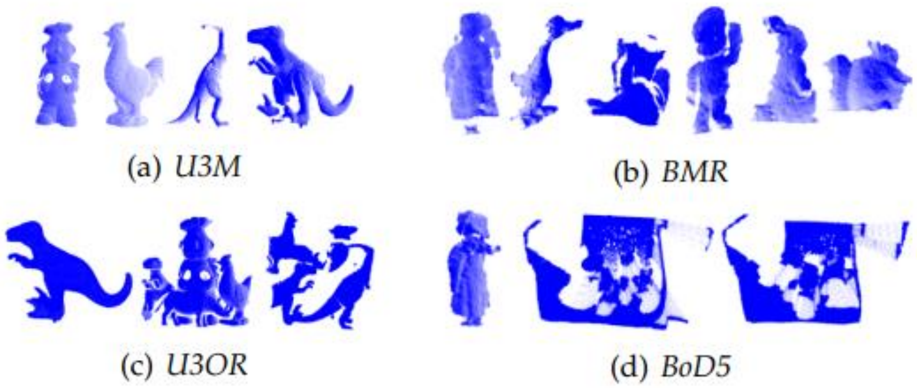


TABLE 1
Properties of feature matching experimental datasets.

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%

Sample point clouds from feature matching experimental datasets



(a) *U3M* (b) *BMR*

(c) *U3OR* (d) *BoD5*

Experiment- Feature Matching Experiments

Evaluation metric

For a correspondence **c** = (**p***s*, **p***t*), it is judged as correct if it satisfies

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

where R*gt* and t*gt* denote the ground-truth rotation matrix and translation vector respectively; *dinlier* 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.

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 C_k be the top-*K* correspondence subset and $C_{initial}$ be the initial correspondence set, the recall of inliers with respect to *K* is defined as

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

Experiment- Feature Matching Experiments

The rationality of setting clustering coefficient threshold adaptive.

The clustering coefficient threshold *tα* in Eq. 6 is used to preliminarily reject outliers. To very the rationality of making it adaptive, we vary *tα* from 0 to 0.5 with a step of 0.1, and compare with the adaptive threshold. The Fig. 3

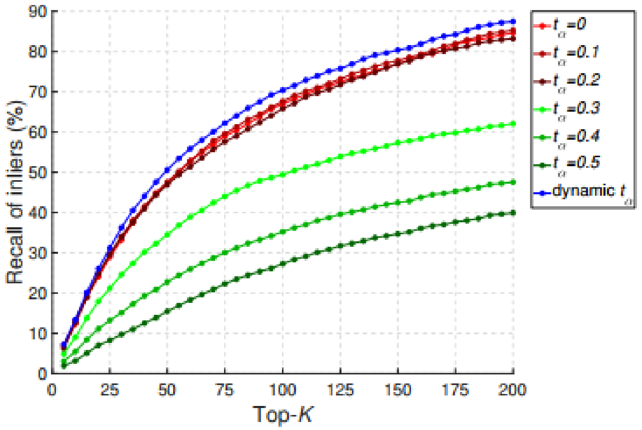


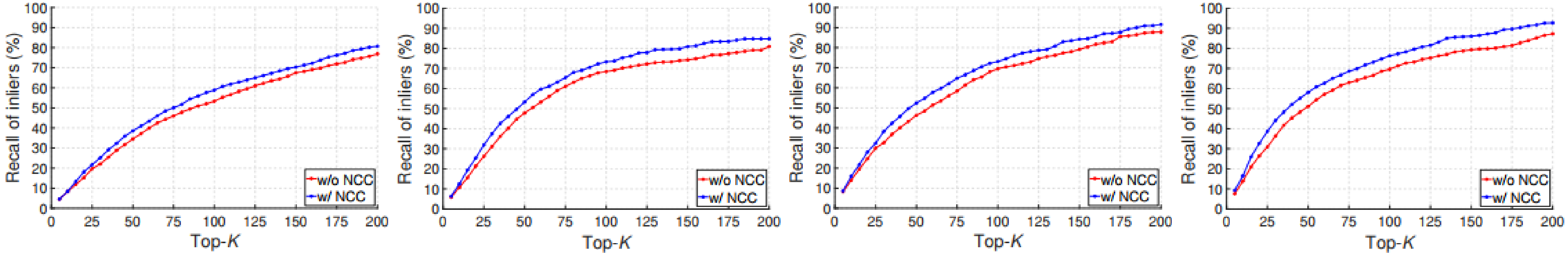
Fig. 3. Performance of MV with different settings of t_α.

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

Experiment- Feature Matching Experiments

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

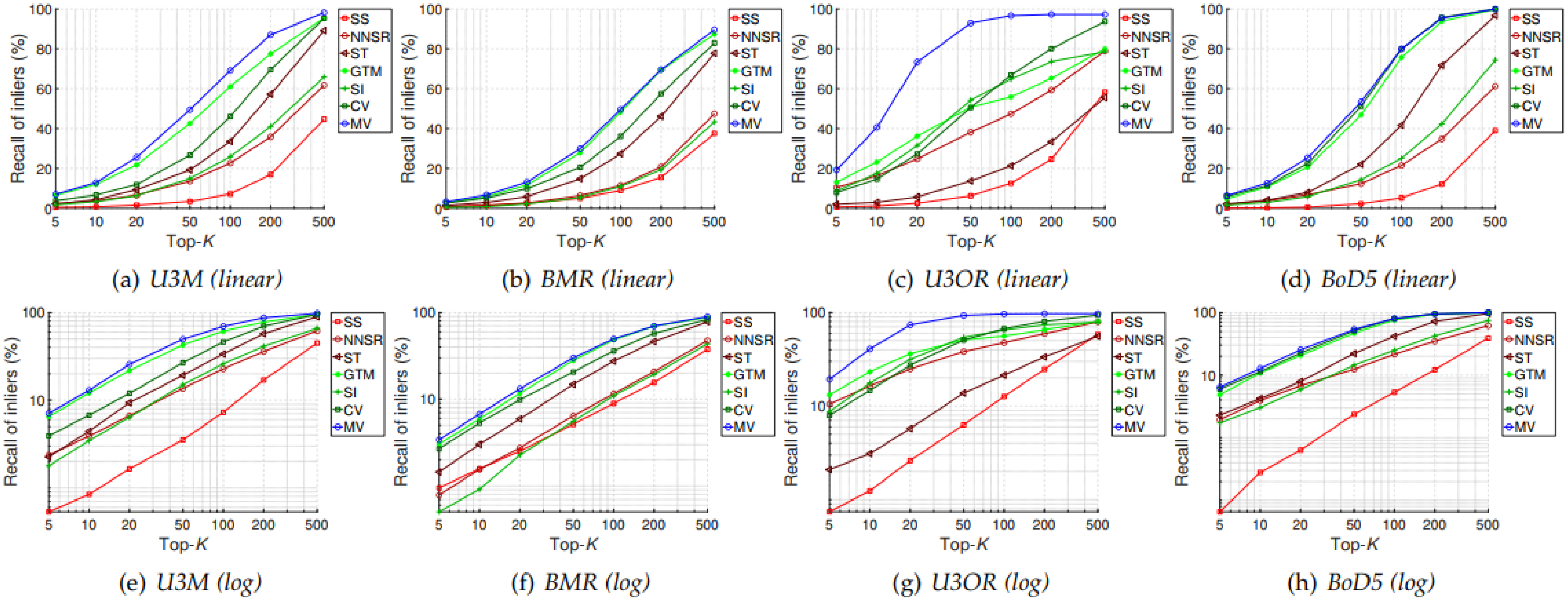


(a) *Chef* (b) *Chicken* (c) *Parasaurolophus* (d) *T-rex*

Analyze：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

Experiment- Feature Matching Experiments

Feature Matching Results



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

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

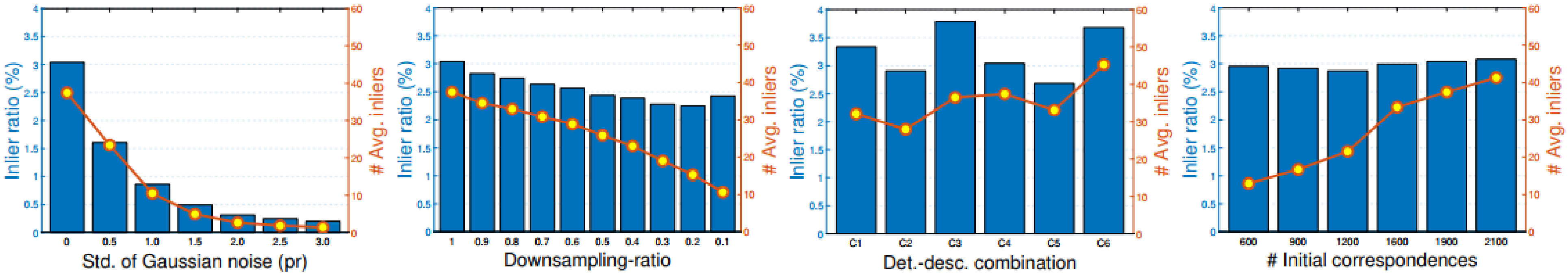
Fig. 5. Feature matching performance of tested methods on four feature matching datasets.

Analyze : 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.

Experiment- Feature Matching Experiments

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 detectordescriptor 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

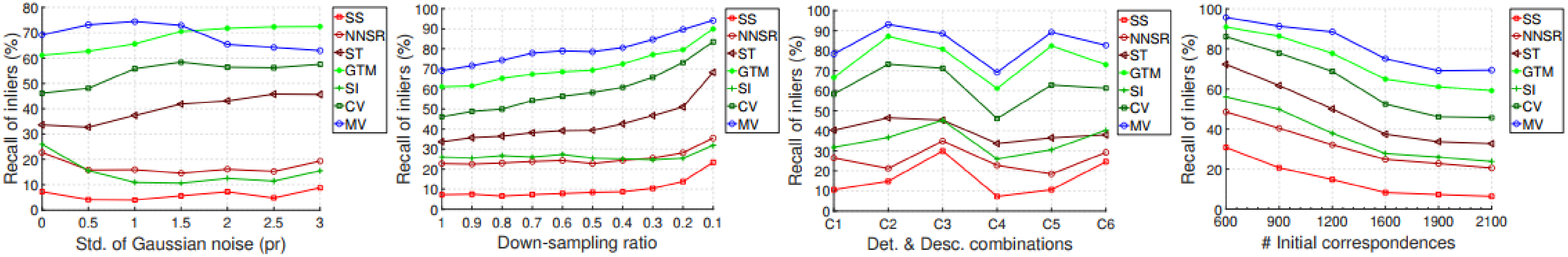


(a) *Gaussian noise* (b) *Down-sampling* (c) *Different det.-desc.* (d) *Diff. input scales*

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.

Experiment- Feature Matching Experiments

Robustness Results



(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.

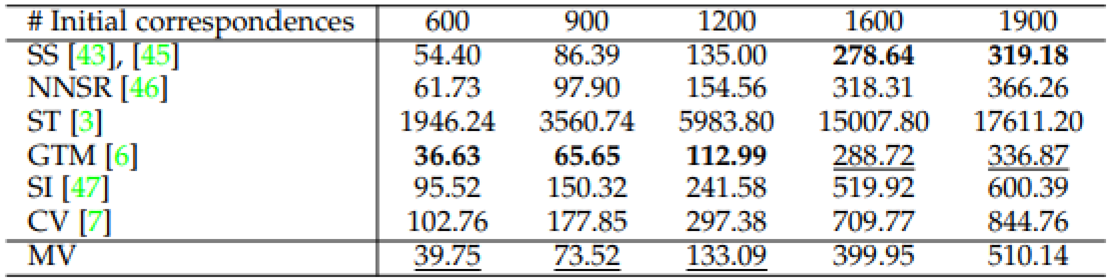
Analyze : When faced with inputs at different scales, the MV consistently achieves the best performance. The gap becomes more clear for inputs wither greater scales.

Experiment- Feature Matching Experiments

Time efficiency

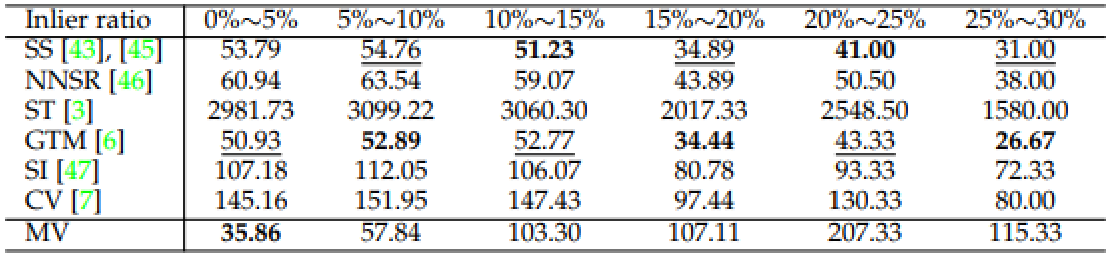
To compare the time efficiency of tested feature matching methods, two cases are analyzed here: inputs with different magnitudes and different inlier ratios. For the former, the number of initial correspondences is set to 600, 900, 1200, 1600, and 1900; for the latter, the input magnitude is fixed to 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.

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

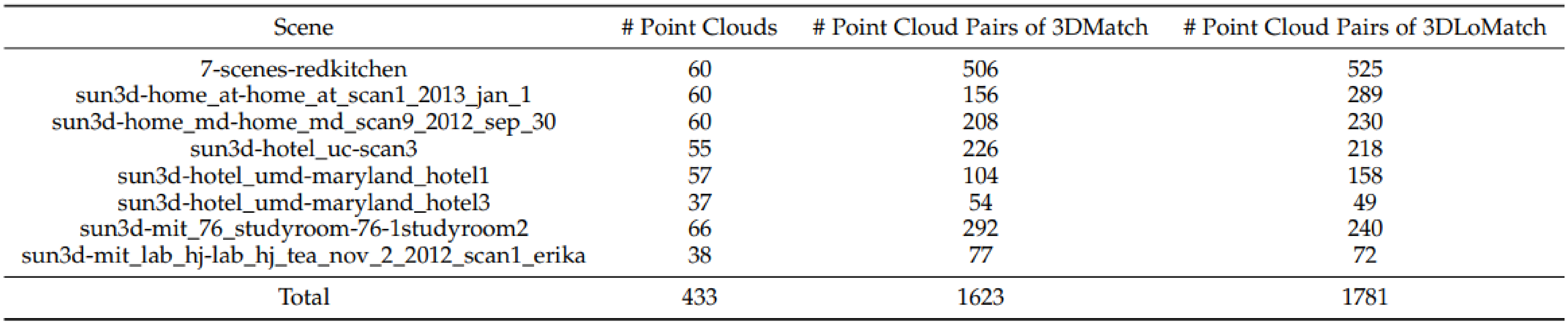
From Table 7, it can be observed that the time cost taken by each method tends to increase as the input correspondence magnitude increase. In particular, MV is a top-ranked performer when the number of input correspondences is smaller than 1600. From Table 8, it can be found that MV is the most efficient 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.

Experiment- Point Cloud Registration Experiments

Datasets

We consider four datasets, i.e., the object-scale dataset U3M, the scene-scale indoor datasets 3DMatch & 3DLoMatch , and scene-scale outdoor dataset KITTI. 3DLoMatch is the subset of 3DMatch, where the 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 , and obtain 555 pairs of point clouds for testing. We use both FPFH (handcrafted descriptor) and FCGF (learned descriptor) as feature descriptors for correspondence generation on scene-scale datasets.

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

Experiment- Point Cloud Registration Experiments

Evaluation metric

We follow 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 **R***est* and translation vector **t***est*, the point-wise error εp between two truly corresponding points **p***s* and **p***t* is defined as:

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

the definition of RMSE is:

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

When the RMSE of a registration is smaller than a threshold *trmse*, we judge it as success.

Experiment- Point Cloud Registration Experiments

Evaluation metric

we employ the rotation error (RE) and translation error (TE) to evaluate the registration results on scenescale dataset. Given the estimated rotation matrix **R***est* and ground-truth rotation matrix **R***gt*, estimated translation vector **t***est* and ground-truth translation vector **t***gt*, RE and TE can be defined as:

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

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

the registration is considered successful when the RE ≤ 15°, TE ≤ 30 cm on 3DMatch & 3DLoMatch datasets, and RE ≤ 5°, TE ≤ 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.

Experiment- Point Cloud Registration Experiments

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.

Results on U3M Dataset

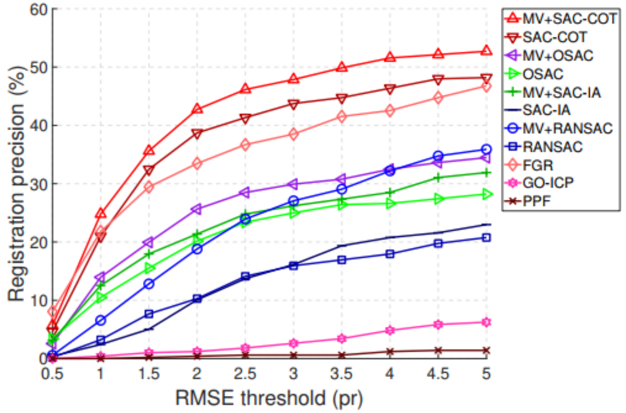


Fig. 10. Registration performance of tested point cloud registration methods on U3M.

Analyze : 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.

Experiment- Point Cloud Registration Experiments

Results on 3DMatch & 3DLoMatch Datasets

1k correspondences setting.

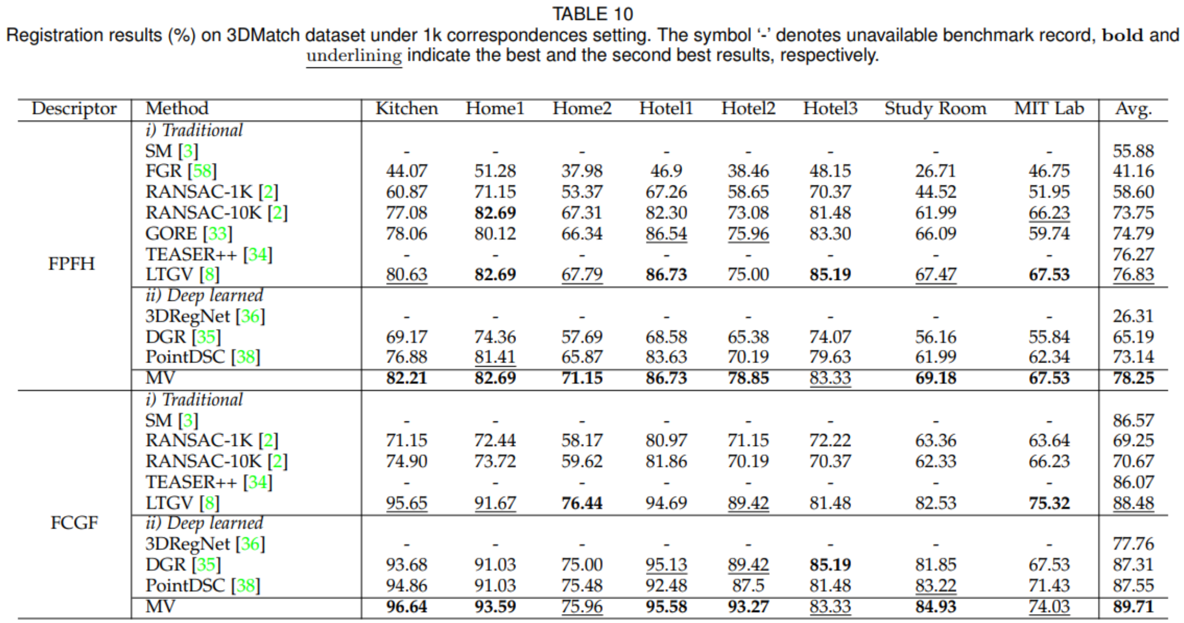


TABLE 10
Registration results (%) on 3DMatch dataset under 1k correspondences setting. The symbol '-' denotes unavailable benchmark record, **bold** and underlining indicate the best and the second best results, respectively.

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**

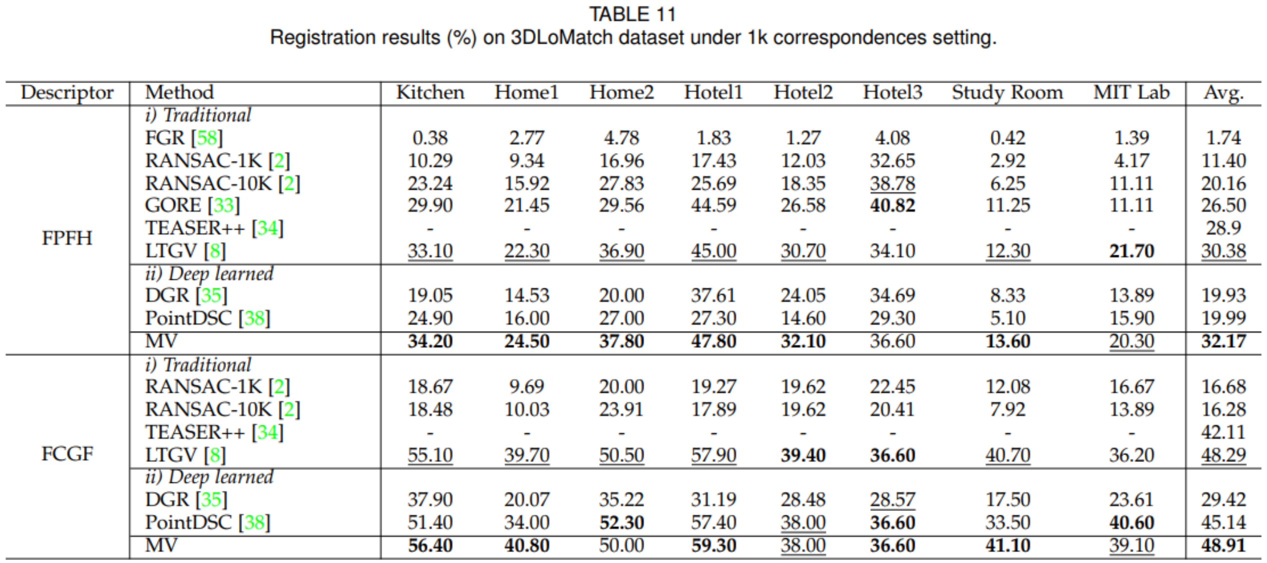


TABLE 11
Registration results (%) on 3DLoMatch dataset under 1k correspondences setting.

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**

Analyze : 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.

Experiment- Point Cloud Registration Experiments

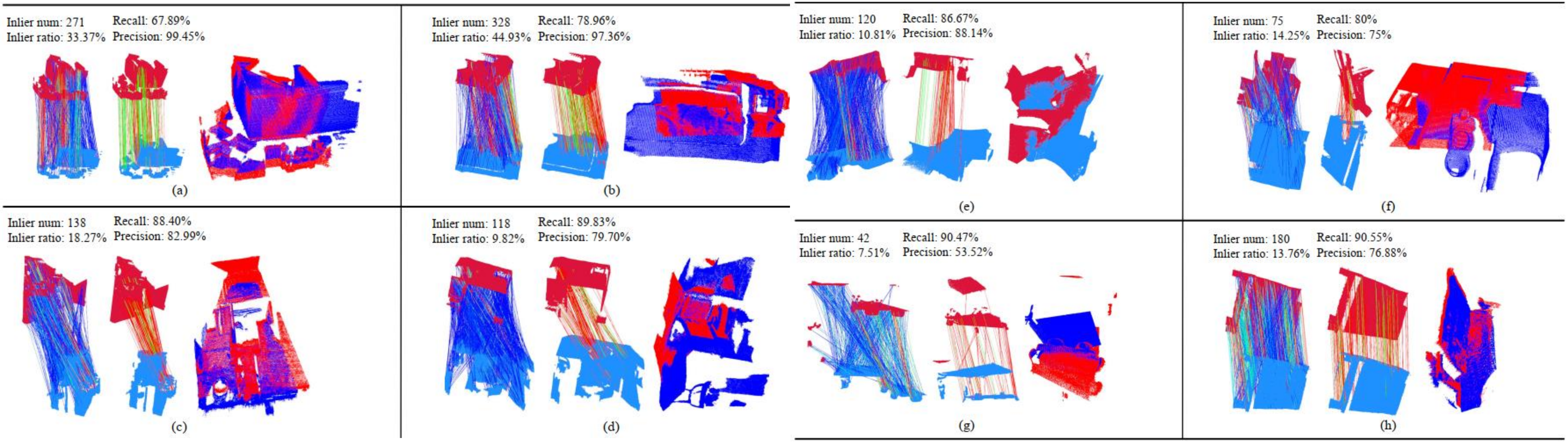


Fig. 11. Visual feature matching and registration results by MV on 3DMatch dataset. For each result, from left to right: initial correspondences rendered based on voting scores (red→ blue: high→low voting scores), selected correspondences by MV (green and red lines respectively denote correct and incorrect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

Experiment- Point Cloud Registration Experiments

Results on 3DMatch & 3DLoMatch Datasets

5k correspondences setting.

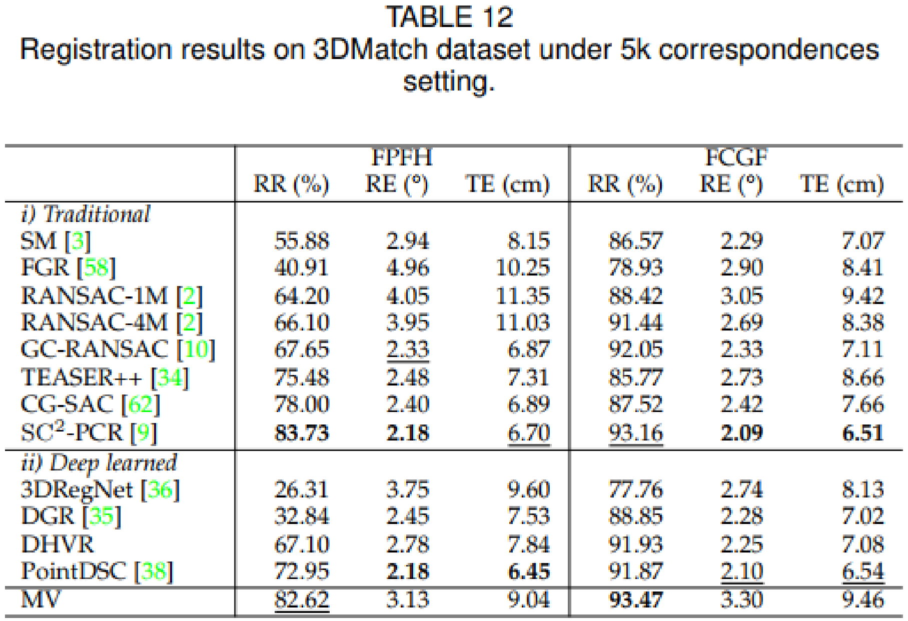


TABLE 12
Registration results on 3DMatch dataset under 5k correspondences setting.

	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

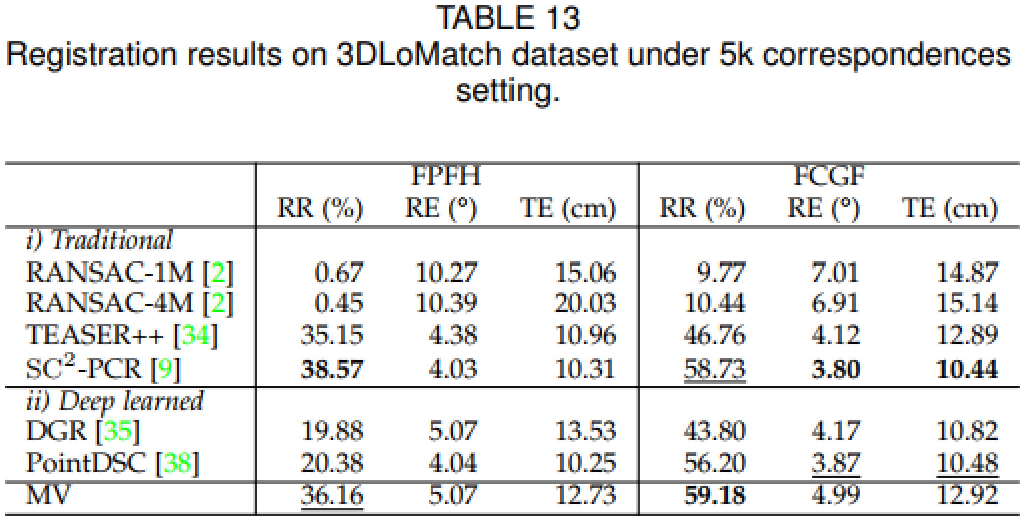


TABLE 13
Registration results on 3DLoMatch dataset under 5k correspondences setting.

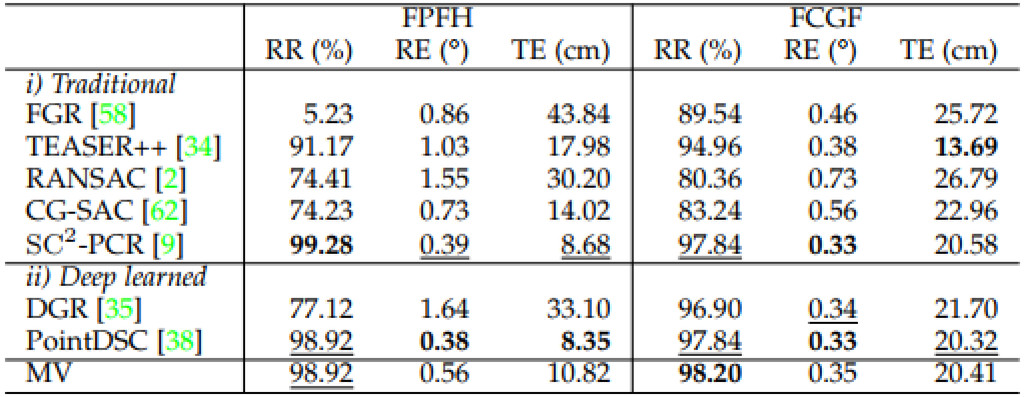
	FPFH			FCGF		
	RR (%)	RE (°)	TE (cm)	RR (%)	RE (°)	TE (cm)
i) Traditional						
RANSAC-1M [2]	0.67	10.27	15.06	9.77	7.01	14.87
RANSAC-4M [2]	0.45	10.39	20.03	10.44	6.91	15.14
TEASER++ [34]	35.15	4.38	10.96	46.76	4.12	12.89
SC^2-PCR [9]	**38.57**	4.03	10.31	58.73	**3.80**	**10.44**
ii) Deep learned						
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

Analyze : 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 SC2-PCR.

Experiment- Point Cloud Registration Experiments

Results on KITTI dataset
In Table 14, the results of DGR, PointDSC , TEASER++ , RANSAC , CG-SAC , SC2-PCR and MV are reported for comparison.

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

Analyze : 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 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.

Experiment- 3D Object Recognition Experiments

Datasets

The two datasets used for the experiments in this section are Queen and U3OR . 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 and compute the εres and rov metrics. Assume that the transformation estimation matrix Mi = (Ri, ti) is obtained in the i-th iteration of RANSAC, and the model point cloud Ps is transformed to obtain the point cloud Ptrans. Let pt i be the point of Ps, ptrans i be the point of Ptrans and the nearest neighbor to the point in the point cloud Pt. If the distance d(pt i, ptrans i) = ||pt i -ptrans i || is less than the threshold drec, which is set to 2 pr, ptrans i will be classified into the point set Ptrans overlap. The point cloud residual error εres and the point cloud overlap rate rov are defined as:

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

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

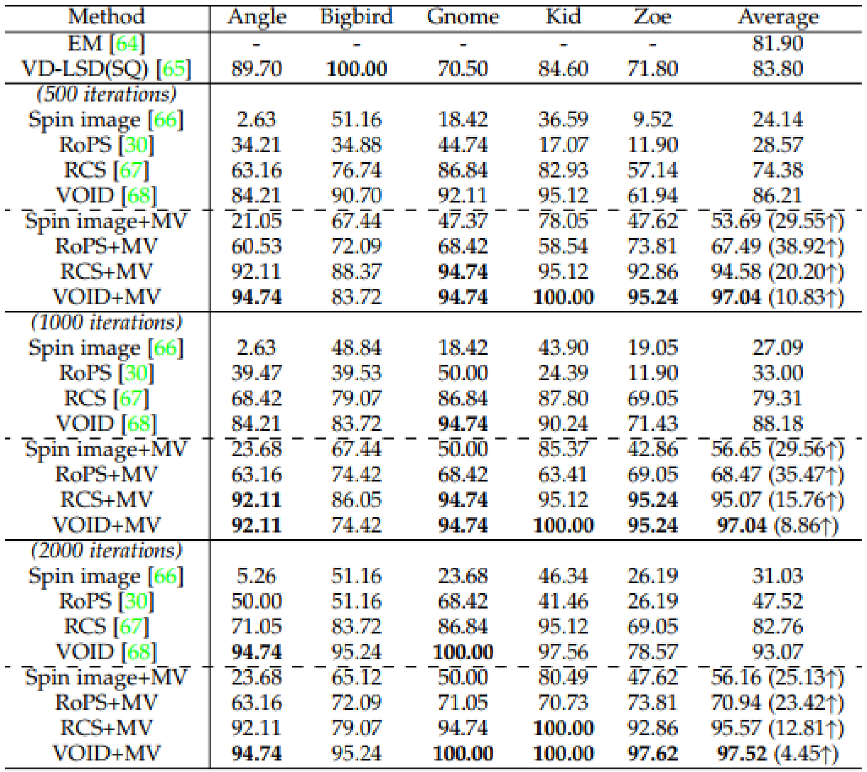
When the residual is less than the threshold *dres* and the overlap rate is greater than the threshold *toverlap*, the transformation estimation matrix **M***i* is considered to satisfy the condition and the iteration is stopped.

Experiment- Point Cloud Registration Experiments

Comparative Results

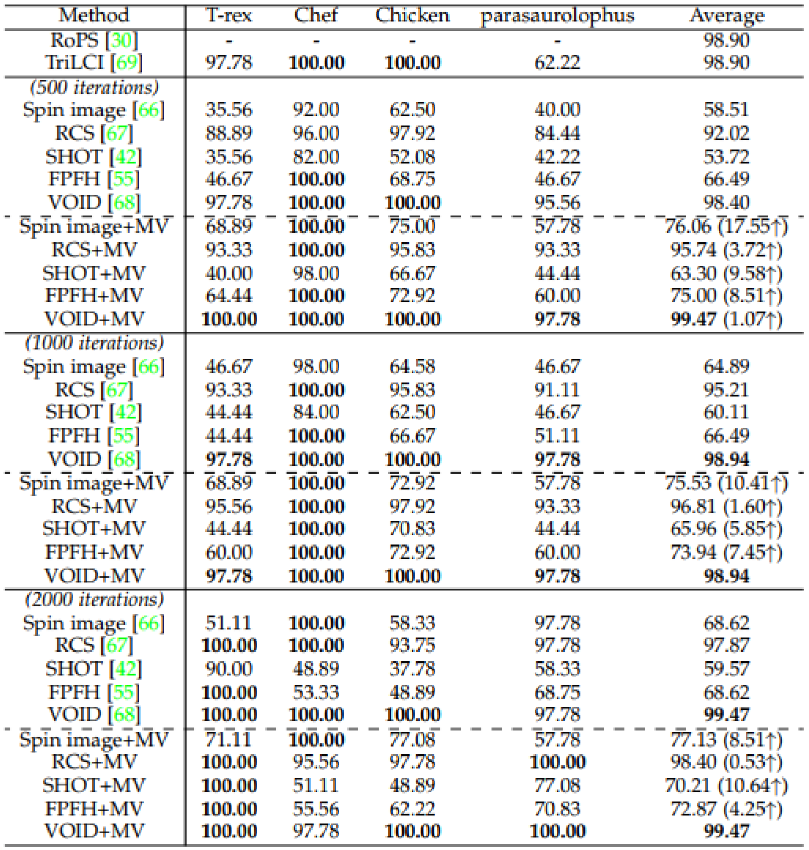
For 3D object recognition, descriptor-based methods are main solutions. We tested the recognition performance of several representative descriptors with and without MV selected correspondences under different RANSAC iterations. Results on Queen and U3OR datasets are shown in Tables 17 and 18, respectively.

TABLE 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↑)

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**

Experiment- Point Cloud Registration Experiments

Comparative Results

Analyze : 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

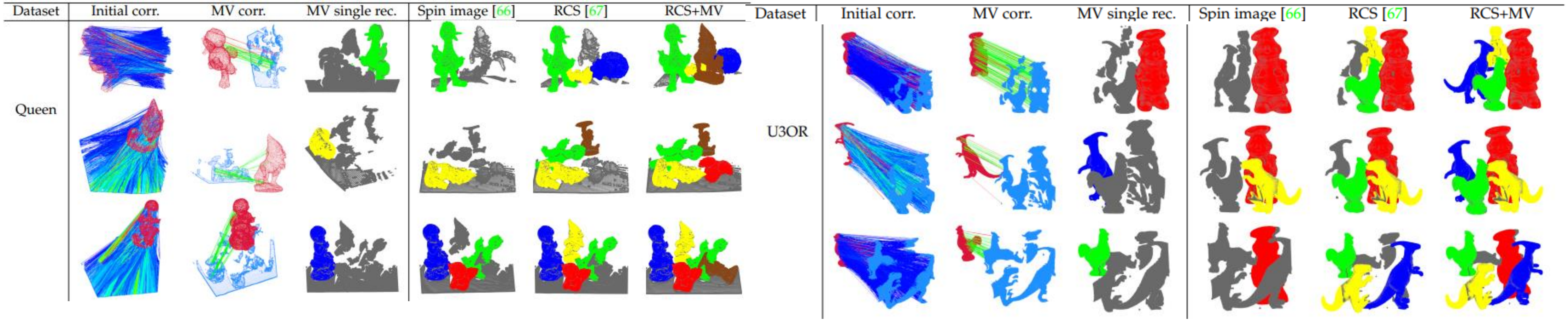


Fig. 12. Visualization of 3D object recognition results on Queen and U3OR datasets. From left to right: initial feature correspondences (red→blue: high→low voting scores), MV-selected correspondences (green and red lines respectively denote correct and incorrect correspondences), single-object recognition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

Conclusion

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.

Thank You!

Avenida da Universidade, Taipa, Macau, China
Tel : (853) 8822 8833 Fax : (853) 8822 8822
Email : info@um.edu.mo Website : www.um.edu.mo