RANSAC Back to SOTA: A Two-Stage Consensus Filtering for Real-Time 3D Registration

Pengcheng Shi, Shaocheng Yan, Yilin Xiao, Xinyi Liu, Yongjun Zhang, *Member, IEEE*, and Jiayuan Li

***Abstract*—Correspondence-based point cloud registration (PCR) plays a key role in robotics and computer vision. However, challenges like sensor noises, object occlusions, and descriptor limitations inevitably result in numerous outliers. RANSAC family is the most popular outlier removal solution. However, the requisite iterations escalate exponentially with the outlier ratio, rendering it far inferior to existing methods (SC2PCR [Chen et al., 2022], MAC [Zhang et al., 2023], etc.) in terms of accuracy or speed. Thus, we propose a two-stage consensus filtering (TCF) that elevates RANSAC to state-of-the-art (SOTA) speed and accuracy. Firstly, one-point RANSAC obtains a consensus set based on length consistency. Subsequently, two-point RANSAC refines the set via angle consistency. Then, three-point RANSAC computes a coarse pose and removes outliers based on transformed correspondence's distances. Drawing on optimizations from one-point and two-point RANSAC, three-point RANSAC requires only a few iterations. Eventually, an iterative reweighted least squares (IRLS) is applied to yield the optimal pose. Experiments on the large-scale KITTI and ETH datasets demonstrate our method achieves up to three-orders-of-magnitude speedup compared to MAC while maintaining registration accuracy and recall.**

***Index Terms*—Point cloud registration, correspondence, consensus filtering, RANSAC, iteratively reweighted least squares.**

I. Introduction

POINT cloud registration (PCR) seeks to estimate a six-degree-of-freedom (6-DOF) pose to align point clouds, which is crucial for 3D reconstruction [3], [4], [5], [6], robotic navigation [7], [8], [9], [10], and aerial photogrammetry [11], [12], [13]. Iterative closest point (ICP) [14] pioneers a local registration pipeline that iteratively searches correspondences and minimizes their distances. Unfortunately, it often converges to erroneous local optima under notable viewpoint changes [15], [16]. To overcome this, researchers derive coarse poses from feature correspondences [17], [18], known as correspondence-based PCR. However, sensor noises, object occlusions, and descriptor limitations inevitably lead to outliers, presenting great challenges for registration [19], [20].

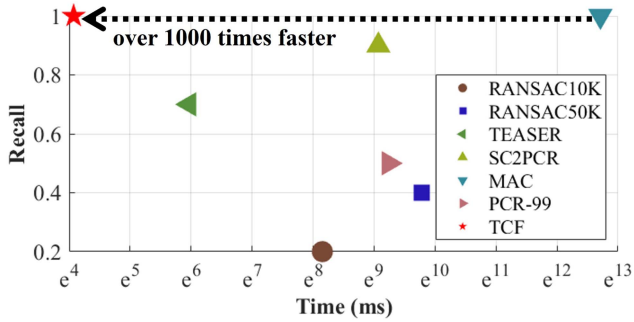


Fig. 1. Average registration recall and runtime for ETH's tree sequence. The runtime denotes the average time for a single registration. Both our method and MAC achieve a 100% recall. Remarkably, ours accomplishes this in just 59 ms, leading to a three-orders-of-magnitude improvement over MAC's 331587 ms.

Random sample consensus (RANSAC) [21], [22], [23] is the most popular solution to handle correspondences with outliers. It follows a *generation-and-selection* strategy that iteratively samples correspondences to generate and verify hypotheses. However, the requisite iterations escalate exponentially with the outlier ratio, which renders them far inferior to existing methods in accuracy and efficiency.

Thus, we propose a two-stage consensus filtering (TCF) incorporating one-point and two-point RANSAC steps before the three-point RANSAC to ensure accuracy and significantly boost efficiency. It begins with a one-point RANSAC to remove correspondence outliers via length constraints. Afterward, a two-point RANSAC evaluates the angle consistency to get a more reliable consensus. This design significantly reduces the iteration number by lowering the sampling dimension and also decreases the outlier ratio. As a result, the three-point RANSAC requires only a few iterations, enhancing registration performance. Following this, three-point RANSAC computes a coarse pose and removes outliers based on transformed correspondence's distances. Finally, a scale-adaptive Cauchy iteratively reweighted least squares (IRLS) is applied to calculate the optimal pose. As depicted in Fig. 1, our method demonstrates SOTA performance and achieves a speedup of up to three orders of magnitude. Our contributions are as follows:

Received 2 July 2024; accepted 8 November 2024. Date of publication 19 November 2024; date of current version 25 November 2024. This article was recommended for publication by Associate Editor Y. Zhou and Editor A. Banerjee upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 42271444 and Grant 42201474 and in part by the Wuhan University-Huawei Geoinformatics Innovation Laboratory Open Fund under Grant TC20210901025-2023-06. *(Corresponding authors: Yongjun Zhang; Jiayuan Li.)*

Pengcheng Shi is with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: shipc_2021@whu.edu.cn).

Shaocheng Yan, Xinyi Liu, Yongjun Zhang, and Jiayuan Li are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China (e-mail: shaochengyan@whu.edu.cn; liuxy0319@whu.edu.cn; zhangyj@whu.edu.cn; ljy_whu_2012@whu.edu.cn).

Yilin Xiao is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR 999077, China (e-mail: yilin.xiao@connect.polyu.hk).

Our code is available at https://github.com/ShiPC-AI/TCF.

Digital Object Identifier 10.1109/LRA.2024.3502056

- We devise a two-stage filtering method that swiftly eliminates the majority of correspondence outliers, facilitating RANSAC to reach SOTA levels.
- Based on our filtering method, we formulate a complete PCR pipeline. On large-scale outdoor datasets, it markedly enhances efficiency while ensuring accuracy.

II. Related Work

A. Classical PCR

RANSAC Family: This type of method follows a generation-and-selection strategy. GC-RANSAC [23] devises a locally optimized pipeline alternating between graph cut and model re-fitting. CG-SAC [18] evaluates correspondence compatibility by calculating distances between salient points and employs a compatibility-guided strategy to reduce sampling randomness. SC2-PCR [1] evaluates correspondence similarity using second-order spatial compatibility, applies a global spectral technique to identify reliable seeds, and uses a two-stage approach to expand each seed into a consensus set. To expedite processing, PCR-99 [24] devises an improved sample ordering guided by pairwise scale consistency and a triplet scale consistency-based outlier rejection scheme. Their performance depends on model parameter selection and requires tuning for specific problems. Runtime increases exponentially with large-scale data, especially when the outliers prevail.

Graph Theory: These methods model the correspondence set as a graph and use graph theories to achieve robust registration. MAC [2] loosens the maximum clique constraint to mine more local consensus information from a graph. It identifies maximal cliques and performs node-guided selection based on the clique with the highest graph weight. CLIPPER [25] ensures dense subgraph solutions with continuous relaxation and achieves efficiency using projected gradient ascent and backtracking line search. ROBIN [26] checks the compatibility of measurement subsets using invariants and prunes outliers by finding the maximum clique or k-core in the graph induced. TEASER [17] reformulates the registration problem using a truncated least squares (TLS) cost and introduces a graph-theoretic framework to decouple and sequentially solve scale, rotation, and translation estimation. Graph-based methods effectively leverage topology information and handle complex scenes but struggle with high algorithmic complexity.

Optimization-based: This category of methods progressively optimizes error functions to refine the pose. Iterative closest point (ICP) [14] pioneers a local registration pipeline that iteratively searches correspondences and minimizes Euclidean distances. Following this, several works seek to advance ICP from diverse perspectives, such as incorporating the point-to-plane metric [27], probabilistic models [28], and branch-and-bound (BnB) theories [29]. Despite robust results, poor initial guess tends to yield wrong results. Fast global registration (FGR) [4] formulates a dense objective featuring a scaled Geman-McClure estimator as the penalty function and alleviates the impact of local minima via graduated non-convexity. Yang et al. [30] propose a versatile approach for robust estimation that utilizes modern non-minimal solvers, extending the applicability of Black-Rangarajan duality and graduated non-convexity to various spatial perception tasks.

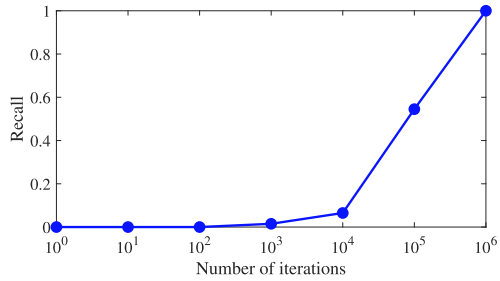


Fig. 2. Registration recall of RANSAC at different iterations. We create 200 pairs of point clouds containing 2% inliers to evaluate the registration recall.

B. Learning-Based PCR

End-to-end Direct Registration: End-to-end methods optimize all steps within a unified framework, directly producing registration results through the network. Deep Closest Point (DCP) [19] employs DGCNN [31] for correspondence identification and a pointer network [32] for soft matching, estimates the rigid transformation with a differentiable single value decomposition (SVD) layer. However, it struggles with partially overlapping point clouds. REGTR [11] employs a transformer network to estimate the probability of each point residing in the overlapping region, thereby supplanting explicit feature matching and RANSAC [21]. CCAG [5] employs cross-attention mechanisms and depth-wise separable convolutions to capture point cloud relationships. Additionally, it introduces an adaptive graph convolution multi-layer perceptron (MLP) to augment node expressiveness. Although proficient with synthetic datasets, these methods may falter on real-world data, limiting their accuracy and practical utility.

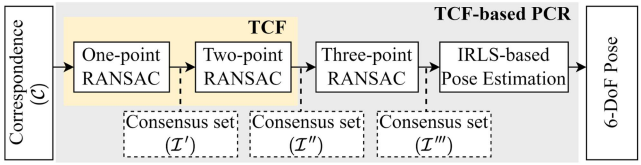
Descriptor Learning: Descriptor learning mainly aims to create sparse descriptions for local 3D patches using a weight-sharing network or produce dense descriptions for the entire point cloud in a single forward pass. The pioneering work 3DMatch [3] converts local 3D patches into volumetric representations and then derives descriptors via a siamese convolutional network. FCGF [33] adopts sparse tensor representation and employs Minkowski convolutional neural networks as the backbone to learn dense features. Several works successively boost the feature descriptiveness via N-tuple loss [34], folding-based encoder [35], and spherical CNNs [36]. While they eliminate manual feature design and bolster robustness, substantial labeling, training, and optimization remain time-consuming, particularly for low-cost robot platforms.

III. Method

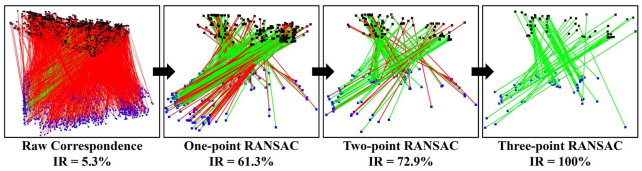
A. Motivation

The minimum iteration, N_{ransac}, required by RANSAC is computed as:

$$N_{ransac} = \left\lceil \frac{\log(1-\lambda)}{\log\left(1-(1-\vartheta)^{s}\right)} \right\rceil \tag{1}$$



(a) Overall framework



(b) Outlier removal

Fig. 3. Overall framework and outlier removal illustration. (a): Our method cascades one-point, two-point, and three-point RANSAC, followed by scale-adaptive Cauchy IRLS, each involving an iterative process. The raw correspondence $\mathcal{C}$ is progressively refined into subsets $\mathcal{I}'$, $\mathcal{I}''$, and $\mathcal{I}'''$, satisfying $\mathcal{I}''' \subseteq \mathcal{I}'' \subseteq \mathcal{I}' \subseteq \mathcal{C}$. The scale-adaptive Cauchy IRLS computes the final 6-DoF pose from $\mathcal{I}'''$. (b): The blue and black points represent the 3D correspondences. The green lines indicate inliers and red lines represent outliers. IR stands for the inlier ratio.

where ϑ is the outlier ratio. s denotes the sample size. $\lceil \cdot \rceil$ is a ceiling function. λ represents the probability of at least one perfect minimal subset, preset to a default value of 0.99. As shown in Fig. 2, we evaluate the registration recall of 200 pairs of point clouds at different iterations. As iterations increase from 10^3 to 10^5 and 10^6, registration recall improves from 2% to 5% and eventually reaches 100%. However, the outlier ratio in 3D correspondence is notably high, which leads to a substantial increase in RANSAC iterations based on (1). Carrying excessive iterations is impractical for real-world applications, necessitating a limit to balance efficiency and accuracy. Revisiting (1) reveals that the sample size impacts the iteration number besides the outlier ratio. Specifically, reducing the sample size can also exponentially decrease the required iterations. Therefore, we propose a two-stage consensus filtering (TCF) comprising one-point and two-point RANSAC. One-point RANSAC applies length constraints to eliminate outliers, followed by two-point RANSAC to enhance consensus reliability via angular consistency. This approach demands fewer samples than three-point RANSAC and reduces the outlier ratio. We then feed the refined correspondence to the three-point RANSAC. With a notably decreased outlier ratio, the required iterations decrease, thus leading to enhanced performance.

B. Problem Formulation

Given two point clouds $P = \{p_i \in \mathbb{R}^3, i = 1, \ldots, N_p\}$ and $Q = \{q_i \in \mathbb{R}^3, i = 1, \ldots, N_q\}$ to be aligned, we begin by extracting local features to establish initial correspondences $\mathcal{C} = \{c_k = (p_k, q_k), k = 1, \ldots, N_c\}$. We employ a two-stage filtering process, i.e., one-point and two-point RANSAC, to obtain the maximum consensus set. Eventually, we employ three-point RANSAC and IRLS to estimate the rigid transformation: a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$. Fig. 3 illustrates our registration pipeline.

Algorithm 1: One-Point RANSAC.

Input: Correspondences $\mathcal{C}$
Initialize: $\lambda = 0.99$, $|\mathcal{I}'| = 0, i = 0$, $N^{(1)}_{ransac} = 1$.
1: **while** $i \leq N^{(1)}_{ransac}$ **do**
2: Randomly select a point correspondence c_k from $\mathcal{C}$
3: Find a consensus set $\mathcal{I}_i$ including c_k via (2) and (3)
4: **if** $|\mathcal{I}_i| \geq |\mathcal{I}'|$ **then**
5: $|\mathcal{I}'| = |\mathcal{I}_i|, \mathcal{I}' = \mathcal{I}_i$
6: Update $N^{(1)}_{ransac}$ using (4)
7: **end if**
8: $i = i + 1$
9: **end while**
Output: Correspondences $\mathcal{I}'$

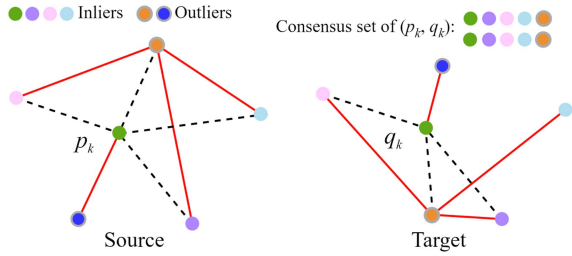


Fig. 4. One-point consensus generation. Same-colored points indicate a point correspondence. Dotted black lines show correct edge correspondences, while red indicates erroneous ones. An edge correspondence is correct if their length discrepancy $< 2\tau$. For (p_k, q_k), applying length consistency identifies a maximal consensus comprising four inliers and one outlier marked in yellow.

C. One-Point RANSAC

As described in Algorithm 1, one-point RANSAC takes the initial correspondence $\mathcal{C}$ as input and outputs a consensus set $\mathcal{I}'$ based on length consistency. The main steps are as follows:

One-point Consensus Generation: As depicted in Fig. 4, for a random point correspondence $c_k = (p_k, q_k)$, we compare it with others to gather a maximal consensus set $\mathcal{I}_i$.

$$\mathcal{I}_i = \{c_j \,|\, c_j \in \mathcal{C}, \forall j, d(c_j, c_k) < 2\tau\} \tag{2}$$

$$d(c_j, c_k) = \left| ||p_j - p_k||_2 - ||q_j - q_k||_2 \right| \tag{3}$$

where $\mathcal{I}_i \subseteq \mathcal{C}$ is a correspondence subset including c_k. i denotes the current iteration number. $||\cdot||_2$ means the L_2 norm. k and j are two indices of point correspondences. If c_j and c_k are inliers, they should satisfy $d(c_j, c_k) < 2\tau$. τ is a noise bound. Our method automatically adjusts the noise bounds as the noise level increases, ensuring the inliers continue to meet the criteria. Thus, it remains effective despite higher noise.

Consensus Verification: Upon identifying a larger consensus during the iteration, we update the minimal iteration requirement $N^{(1)}_{ransac}$.

$$N^{(1)}_{ransac} = \left\lceil \frac{\log(1-\lambda)}{\log\left(1 - \frac{|\mathcal{I}'|}{|\mathcal{C}|}\right)} \right\rceil \tag{4}$$

Algorithm 2: Two-Point RANSAC.

Input: Correspondences $\mathcal{I}'$

Initialize: $\lambda = 0.99$, $|\mathcal{I}''| = 0, i' = 0, N^{(2)}_{ransac} = 1$.

```
while i' ≤ N^(2)_ransac do
  Randomly select two point correspondences c_i, c_j ∈ I'
  Find a subset S_ij ⊆ I' using (5)
  Find a consensus set I_i' ⊆ S_ij using (6)–(8)
  if |I_i'| ≥ |I''| then
    |I''| = |I_i'|, I'' = I_i'
    Update N^(2)_ransac based on (9)
  end if
  i' = i' + 1
end while
```

Output: Correspondences $\mathcal{I}''$

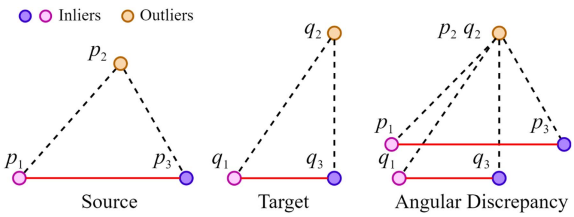


Fig. 5. Angular discrepancy. The circled letter signifies a point, and the same color denotes a point correspondence. Black dotted lines represent correct edge correspondences, i.e., $|p_1p_2 - q_1q_2| < 2\tau$ and $|p_3p_2 - q_3q_2| < 2\tau$. The yellow outlier produces triangles with two similar edge lengths but different shapes due to angle discrepancies.

where superscript (1) indicates one-point sampling. $\mathcal{I}'$ denotes the current optimal consensus set. When the iteration number i surpasses $N^{(1)}_{ransac}$, the algorithm terminates and outputs $\mathcal{I}'$.

D. Two-Point RANSAC

While the consensus set meets length requirements, outliers like the yellow correspondence still exist. Thus, we propose a two-point RANSAC to refine the consensus. As summarized in Algorithm 2, two-point RANSAC takes the result $\mathcal{I}'$ of one-point RANSAC as input and excludes outliers based on angle consistency. The main steps are as follows:

Two-point Subset Sampling: We begin by randomly selecting two different point correspondences, $c_i = (p_i, q_i)$ and $c_j = (p_j, q_j)$ from $\mathcal{I}'$. Then, we compare them with others to collect a subset $\mathcal{S}_{ij}$ that simultaneously satisfies two length consistencies:

$$\mathcal{S}_{ij} = \{c_k \,|\, c_k \in \mathcal{I}', \forall k, d(c_k, c_i) < 2\tau, d(c_k, c_j) < 2\tau\} \quad (5)$$

where $c_i, c_j \in \mathcal{S}_{ij}$ is a correspondence subset of $\mathcal{I}'$. $d(\cdot)$ is the length discrepancy in (3).

Consensus Generation As illustrated in Fig. 5, despite two edge correspondences maintaining length consistency, two triangles still exhibit distinct shapes due to angle discrepancy. Therefore, we propose angle consistency to encourage point correspondences to form roughly congruent triangles, yielding a more robust consensus set $\mathcal{I}_{i'}$.

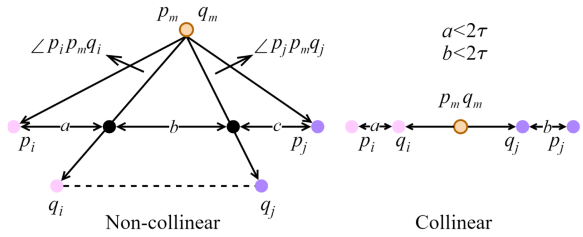


Fig. 6. β and two cases of correspondence distribution. We overlap the correspondence $c_m = (p_m, q_m)$ for clarity. Given that c_i and c_j are two inliers, they satisfy $||p_ip_j| - |q_iq_j|| \leq 2\tau$. If c_m is also an inlier, then $||p_mp_i| - |q_mq_i|| < 2\tau$ and $||p_mp_j| - |q_mq_j|| < 2\tau$. To ensure robust consensus, we further constrain the angle discrepancy $\alpha(c_m, c_i, c_j) = (\angle p_ip_mq_i + \angle p_jp_mq_j) < \beta$, where $\angle p_ip_mq_i \approx \arcsin \frac{a}{|p_mp_i|}$, $\angle p_jp_mq_j \approx \arcsin \frac{c}{|p_mp_j|}$, and $a, c < 2\tau$. We set β to $|\arcsin \frac{\tau}{|p_mp_i|} + \arcsin \frac{\tau}{|p_mp_j|}|$.

$$\mathcal{I}_{i'} = \underset{\mathcal{I} \subseteq \mathcal{S}_{ij}}{\operatorname{argmax}} |\mathcal{I}|, \forall\, c_m \in \mathcal{I}, \alpha(c_m, c_i, c_j) < \beta \quad (6)$$

$$\alpha(c_m, c_i, c_j) = \angle p_ip_mq_i + \angle p_jp_mq_j \quad (7)$$

$$\beta = \left| \arcsin \frac{\tau}{|p_mp_i|} + \arcsin \frac{\tau}{|p_mp_j|} \right| \quad (8)$$

where i' means the current iteration number. $\alpha(c_m, c_i, c_j)$ denotes the angle disparity. Fig. 6 depicts the computation of β and two cases of correspondence distribution.

Consensus Verification: This step is similar to that of the one-point RANSAC. Upon identifying a larger inlier set during the iteration, we immediately update the iteration's termination condition $N^{(2)}_{ransac}$.

$$N^{(2)}_{ransac} = \left\lceil \frac{\log(1-\lambda)}{\log\left(1 - \left(\frac{|\mathcal{I}''|}{|\mathcal{I}'|}\right)^2\right)} \right\rceil \quad (9)$$

This step is similar to that of the one-point RANSAC. Upon identifying a larger inlier set during the iteration, we immediately update the iteration's termination condition $N^{(2)}_{ransac}$.

E. Three-Point RANSAC

After one-point and two-point RANSAC, we obtain a correspondence subset satisfying length and angle constraints. Despite a notable increase in the inlier ratio, outliers may remain. Hence, we use three-point RANSAC to identify a consensus set $\mathcal{I}'''$ based on the distances of transformed correspondences.

$$\mathcal{I}''' = \underset{\mathbf{R}_k, \boldsymbol{t}_k}{\operatorname{argmax}} |\mathcal{I}|, \mathcal{I} \subseteq \mathcal{I}''$$
$$\text{s.t.}, \forall (p_i, q_i) \in \mathcal{I}, |\mathbf{R}_k p_i + \boldsymbol{t}_k - q_i| < \tau. \quad (10)$$

where k means the iteration number. $\mathcal{I}$ represents a correspondence subset, and (p_i, q_i) denotes a point correspondence. We employ singular value decomposition (SVD) to compute the rotation matrix $\mathbf{R}_k$ and translation vector $\boldsymbol{t}_k$ from three non-collinear correspondences.

Algorithm 3: Scale-Adaptive Cauchy IRLS.

Input: Correspondences $\mathcal{I}'''$
Initialize: $j = 1$, $N_j = 100$, $w_1 = \{1\}_{1:|\mathcal{I}'''|}$,
$|\mathcal{I}_1| = |\mathcal{I}'''|, \mu = 1.3, e_{\min} = 0.01, \gamma_{\min} = 1.0$
$\gamma_1 \xleftarrow{\max}$ residuals $e \xleftarrow{\mathbf{R},\boldsymbol{t}}$ SVD$(\hat{\mathcal{I}}_1')$
while $j \leq N_j$ **do**
  Estimate residuals $e_j \xleftarrow{\mathbf{R}_j,\boldsymbol{t}_j}$ IRLS$(w_j, \mathcal{I}_j)$
  Update inlier set $\mathcal{I}_{j+1}$ by $e_j < 3\gamma_j$
  Update inlier residuals $e'_{j+1} = e_j[\mathcal{I}_{j+1}] \subseteq e_j$
  Update inlier weights $w_{j+1} = \frac{\gamma_j^2}{\gamma_j^2 + e'^2_{j+1}}$
  Update scale factor $\gamma_{j+1} = \gamma_j / \mu$
  if $|w_{j+1} * {e'_{j+1}}^2 - w_j * {e'_j}^2| < e_{\min}$ or $\gamma_{j+1} < \gamma_{\min}$ **then**
    $\mathbf{R}^* = \mathbf{R}_j, \boldsymbol{t}^* = \boldsymbol{t}_j$
    break
  end if
  $j = j + 1$
end while
Output: Transformation $(\mathbf{R}^*, \boldsymbol{t}^*)$

F. Scale-Adaptive Cauchy IRLS

Traditional Cauchy estimation exhibits considerable deviation from the ground truth in the first iteration if the outliers exceed 50%. Correct inliers incur higher residuals and receive smaller weights. Thus, we introduce a scale-adaptive Cauchy IRLS [37] in Algorithm 3. We start by assigning equal weights (w) to all correspondences ($\mathcal{I}'''$). We then progressively lower the residual threshold based on a scale factor (γ) and preserve inliers ($\mathcal{I}$). γ decreases continuously via the constant $\mu = 1.3$. Upon iteration termination, we output the optimal transformation $(\mathbf{R}^*, \boldsymbol{t}^*)$. Unlike FGR [4], scale-adaptive Cauchy IRLS uses a novel Cauchy objective function and automatically excludes outliers with control parameters during optimization. Additionally, it provides the applications of space intersection, robust feature matching, and point cloud registration.

IV. Experiments

A. Experimental Setup

Datasets: As summarized in Table I, we select ETH [38] and KITTI [39] for validation. The ETH dataset comprises 32 scans from five scenes (office, facade, courtyard, arch, and trees), acquired using Z + F Imager 5006i and Faro Focus 3D scanners. The KITTI dataset provides 11 Velodyne HDL-64E LiDAR sequences across diverse scenarios. We choose sequences 01 (highway), 02 (urban+country), 03 (country), 06 (urban), and 09 (urban+country). We randomly sample 200 pairs of point clouds per sequence.

Evaluation Metrics: We evaluate all methods via registration recall $rr = \frac{N_{success}}{N_{all}}$, rotation error $e_r = \arccos \frac{tr(\mathbf{R}_e(\mathbf{R}_g)^{\mathrm{T}})-1}{2}$, and translation error $e_t = ||\boldsymbol{t}_e - \boldsymbol{t}_g||$. $tr(\cdot)$ is the matrix trace. $\boldsymbol{t}$ and $\mathbf{R}$ denote translation and rotation, respectively. Subscript g denotes ground-truth values while e represents estimated ones.

TABLE I
Dataset Details

Sequences		Type	D_p (m)	Pairs (/)	$\lvert\mathcal{C}\rvert$ (/)	ρ (/)
ETH	Arch	Outdoor	16.48	8	6989	1.1%
	Courtyard	Outdoor	16.63	28	9910	6.1%
	Facade	Outdoor	4.06	21	2250	12.2%
	Office	Indoor	7.49	8	2964	5.6%
	Trees	Outdoor	10.64	10	8242	1.2%
KITTI	01	Highway	16.97	200	3343	2.4%
	02	Urban+Country	16.97	200	3104	3.5%
	03	Country	17.02	200	3175	4.6%
	06	Urban	16.95	200	3655	3.6%
	09	Urban+Country	16.96	200	3042	3.8%

D_p means the average pairwise distance. $|\mathcal{C}|$ refers to the average number of correspondences. ρ is the average inlier ratio.

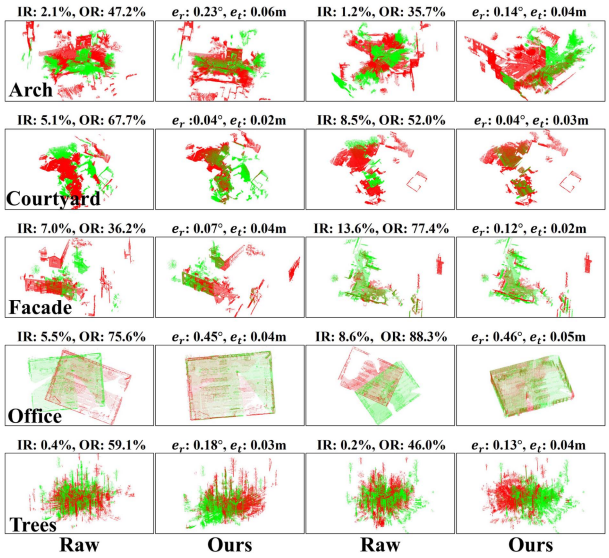


Fig. 7. Visualization of registration results on the ETH dataset. Red and green denote the source and target point clouds, respectively. IR refers to the inlier ratio, and OR denotes the overlap ratio. e_r refers to the rotation error, and e_t denotes the translation error.

$N_{success}$ is the count of successfully registered point cloud pairs while N_{all} is the total number. Registration is successful if $e_r \leq 5°, e_t \leq 0.5$ m on the ETH dataset, and $e_r \leq 5°, e_t \leq 1\ m$ on the KITTI dataset.

Implementation Details: We select RANSAC10K [21], RANSAC50K [21], TEASER [17], SC2-PCR [1], MAC [2], and PCR-99 [24] for comparisons. We implement our method in C++ on Ubuntu 20.04, running all experiments on an Intel Core i9-10850K CPU. For ETH, we downsample each point cloud with 0.1 m voxels, then extract ISS [40] keypoints and FPFH [22] descriptors. We apply 0.3 m voxels for KITTI, randomly sample 5,000 points, and extract FPFH [22] descriptors. Two points qualify as a correspondence if their descriptors rank in each other's top three. We set τ as the point cloud's resolution. The maximal distance for a valid correspondence is 0.3 m in the ETH dataset and 0.6 m for the KITTI dataset.

B. Results on ETH

Recall and Accuracy: Fig. 7 depicts our registration results on the ETH dataset and Table II outlines the registration results.

TABLE II
REGISTRATION RESULTS ON THE ETH DATASET

Methods	Arch				Courtyard				Facade				Office				Trees			
	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	t (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)
RANSAC10K	–	–	–	–	89	0.47	0.12	3863	**100**	0.41	0.09	1802	90	1.35	0.16	2022	20	1.11	0.20	3470
RANSAC50K	38	0.72	0.28	16263	100	0.21	0.10	18981	100	0.30	0.07	9047	90	1.23	0.22	10243	40	1.46	0.21	17752
TEASER	63	0.25	0.07	318	**100**	0.06	**0.04**	521	**100**	0.22	0.03	67	90	0.70	0.07	81	70	**0.12**	**0.03**	391
SC2PCR	**88**	0.18	0.06	6548	**100**	0.04	**0.04**	7383	**100**	**0.08**	**0.02**	693	**100**	**0.45**	**0.04**	1204	90	0.16	**0.03**	8699
MAC	**88**	**0.09**	0.05	316444	**100**	0.04	**0.04**	797857	**100**	0.13	**0.02**	14881	**100**	0.50	0.06	63080	**100**	0.25	0.07	331587
PCR-99	50	0.18	**0.04**	7718	**100**	0.05	**0.04**	13881	95	0.22	0.04	774	90	0.85	0.08	1362	50	0.15	0.04	10309
Ours	**88**	0.14	**0.04**	**35**	**100**	**0.02**	**0.04**	**96**	**100**	0.11	0.03	**35**	**100**	0.64	0.06	**27**	**100**	0.19	**0.03**	**58**

$|\mathcal{C}|$ refers to the average number of correspondences, while ρ is the average inlier ratio. Bold and underline denote the first and second place, respectively. – means that all trials have failed.

TABLE III
REGISTRATION RESULTS ON THE KITTI DATASET

Methods	01				02				03				06				09			
	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	t (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)	rr (%)	e_r (°)	e_t (m)	time (ms)
RANSAC10K	42	1.10	0.42	2206	69	1.16	0.42	2073	97	1.06	0.35	2098	88	0.99	0.38	2232	85	1.25	0.36	2063
RANSAC50K	44	0.74	0.30	11102	73	0.97	0.36	10268	**100**	0.78	0.28	10295	87	0.75	0.29	10854	88	1.06	0.32	9931
TEASER	52	0.65	0.26	166	74	1.00	0.34	117	99	0.86	0.25	115	97	0.65	0.22	134	89	1.05	0.28	106
SC2PCR	49	**0.36**	**0.21**	1801	**75**	**0.66**	**0.31**	1338	**100**	**0.55**	**0.23**	1387	92	**0.40**	**0.19**	1829	90	**0.66**	**0.23**	1279
MAC	54	0.40	0.26	35632	74	0.72	0.35	29158	**100**	0.56	0.27	75199	95	0.42	0.22	112980	**91**	0.69	0.28	26109
PCR-99	55	1.37	0.45	12219	60	1.59	0.43	5506	77	1.36	0.39	4355	65	1.45	0.43	10266	71	1.56	0.40	3078
Ours	**65**	0.54	0.25	**79**	**75**	0.73	0.34	**58**	**100**	0.65	0.25	**63**	**98**	0.48	0.21	**53**	90	0.73	0.25	**45**

$|\mathcal{C}|$ refers to the average number of correspondences. ρ is the average inlier ratio. Bold and underline denote the first and second place, respectively.

Courtyard, facade, and office have inlier ratios exceeding 5% while challenging arch and trees only have around 1%. Our method matches MAC in the recall, achieving top performance in all five scenarios. It delivers the best translation accuracy in the arch (0.04 m), courtyard (0.04 m), and trees (0.03 m), the second-best in the facade (0.03 m) and office (0.06 m).It also excels in rotational accuracy, leading in the courtyard (0.02°) and ranking second in the facade (0.11°). Our rotation error exceeds MAC, SC2PCR, and TEASER by 0.05°, 0.09°, and 0.07° in the arch, office, and trees, respectively. These results showcase that our method achieves comparable or superior registration accuracy and recall to mainstream methods.

Running Efficiency: Our method is generally two to three orders of magnitude faster than baseline methods. For example, on arch, it outperforms RANSAC50, SC2PCR, MAC, and PCR-99 by 465, 187, 9041, and 221 times, respectively. On the courtyard, it is 40, 198, 77, 8311, and 145 times faster than RANSAC10K, RANSAC50K, SC2PCR, MAC, and PCR-99, respectively. As MAC constructs a high-order graph and searches for the maximum clique, its time consumption significantly escalates with higher correspondence numbers and inlier ratios. In contrast, TCF shows minimal impact from correspondence quantity. It merely requires an additional 31ms and 69ms for trees and courtyard than the office, respectively, which underscores our high efficiency.

C. Results on KITTI

Recall and Accuracy: Table III summarizes the registration results on KITTI dataset. Fig. 8 depicts our registration results. The large offset (about 17m) between point clouds degrades performance for all methods. Sequence 01 is a high-speed scenario characterized by significant distortion and feature deterioration,

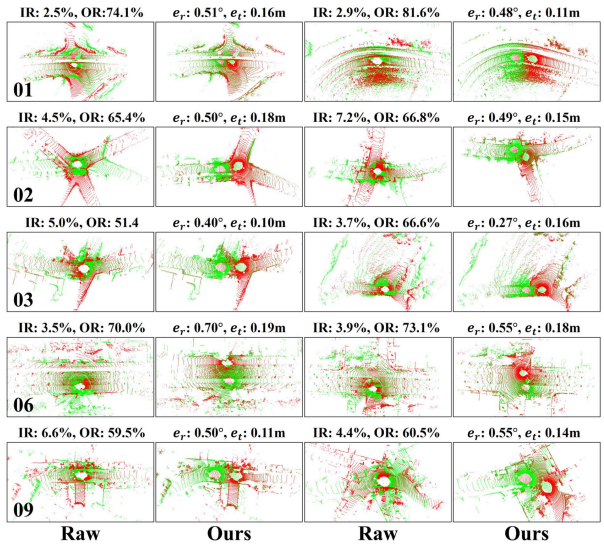


Fig. 8. Visualization of registration results on the KITTI dataset. Red and green denote the source and target point clouds, respectively. IR refers to the inlier ratio, and OR denotes the overlap ratio. e_r refers to the rotation error, and e_t denotes the translation error.

resulting in a low inlier ratio of 2.4%. Despite this, our method achieves the highest recall (65%). It attains the second-highest translation accuracy across all five sequences. Our rotation accuracy slightly lags behind SC2PCR and MAC but stays within a maximum deviation of 0.18°. These results underscore that our method excels in challenging environments like highways, exhibiting higher registration recall.

Running Efficiency: Our method still demonstrates superior efficiency. In sequence 01, it outperforms RANSAC10K, RANSAC50K, TEASER, SC2PCR, MAC, and PCR-99 by 28,

TABLE IV
PERFORMANCE IMPROVEMENT WITH OUR METHOD

Methods	Trees $\lvert\mathcal{C}_1\rvert = 8242, \rho_1 = 1.2\%$ $\lvert\mathcal{C}_2\rvert = 60, \rho_2 = 98\%$				01 $\lvert\mathcal{C}_1\rvert = 3343, \rho_1 = 2.4\%$ $\lvert\mathcal{C}_2\rvert = 121, \rho_2 = 48\%$			
	rr (%)	e_r (°)	e_t (cm)	time (ms)	rr (%)	e_r (°)	e_t (cm)	time (ms)
Ours+	100	0.24	0.05	59	64	0.54	0.25	96
TEASER	↑30	↓0.11	↓0.01	↓332	↑12	↑0.07	↓0.09	↓70
Ours+	100	0.18	0.02	63	64	0.50	0.24	87
SC2PCR	↑10	↑0.02	↓0.01	↓8636	↑15	↑0.14	↑0.03	↓1714
Ours+	100	0.22	0.04	1672	64	0.51	0.24	5203
MAC	–	↓0.03	↓0.03	↓329915	↑10	↑0.11	↓0.02	↓30,429
Ours+	100	0.27	0.04	59	63	0.94	0.33	121
PCR-99	↑50	↑0.03	↓0.02	↓10250	↑8	↓0.43	↓0.12	↓12098

The runtime includes our method and baseline methods. $\mathcal{C}_1$ and ρ_1 mean raw correspondences and inlier ratios, respectively, while $\mathcal{C}_2$ and ρ_2 refer to those generated by our method.

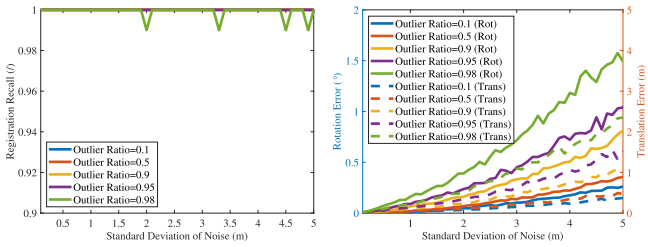
141, 2, 23, 451, and 155 times, respectively. Similarly, in sequence 06, our speeds surpass RANSAC10K, RANSAC50K, TEASER, SC2PCR, MAC, and PCR-99 by 42, 205, 3, 35, 465, 2132, and 194 times, respectively. In sequence 09, our method executes in 54 ms, significantly outperforming RANSAC10K, RANSAC50K, TEASER, SC2PCR, MAC, and PCR-99, being 46, 221, 2, 28, 580, and 68 times faster, respectively. Our approach offers comparable or superior performance to baseline methods while significantly enhancing speed. The results demonstrate our effectiveness and efficiency with low-overlap data. Furthermore, these results reaffirm that our method elevates the RANSAC family to SOTA performance.

D. Boosting Performance With TCF

As illustrated in Table IV, we feed our consensus set into other method. The total runtime in the table comprises ours and the baseline methods. Our approach boosts trees' inlier ratio from 1.2% to 98%. All pipelines achieve 100% recall, with deviations limited to 0.27° and 0.05 m. TEASER and PCR-99 enhance the recall by 30% and 50%, respectively. TEASER, SC2PCR, MAC, and PCR-99 are 332 ms, 9s, 330s, and 10s faster than their original methods, respectively. We boost the inlier ratio in sequence 01 jumps from 2.4% to 48%. TEASER, SC2PCR and MAC demonstrate 12%, 15%, and 10% recall improvement, respectively. TEASER, SC2PCR, MAC, and PCR-99 become faster by approximately 70 ms, 2s, 30s, and 12s, respectively. The results prove that cascading our method with other methods effectively boosts registration performance and facilitates a qualitative leap in speed.

E. Noise Study

As shown in Fig. 9, we evaluate the registration performance under various noise levels. In each, we randomly generate 3000 correspondences between −100 m and 100 m, satisfying the same transformation. Then, we replace 10%, 50%, 90%, 95%, and 98% correspondences with random values between −100 m and 100 m as outliers. Finally, we add 50 levels of Gaussian noise, with standard deviations σ_{noise} from 0.1 m to 5 m, to the remaining inliers. We run 100 tests to compute recall, average rotation error, and translation error. Registration is successful if the RMSE of true inliers is less than three times σ_{noise} . The



(a) Recall (b) Rotation and translation error

Fig. 9. Registration across various noise levels. (b): Rotation (left) and translation (right) errors are shown in one figure.

TABLE V
OUTLIER REMOVAL ANALYSIS FOR THREE RANSAC MODULES

RANSAC Module	Arch IR (%)	Arch IN (/)	Arch Time (ms)	01 IR (%)	01 IN (/)	01 Time (ms)
1R	21	**76**	**20**	8	**57**	**29**
2R	26	48	672	12	27	115
3R	27	16	3234	28	24	3139
1R+2R	30	35	22	19	21	34
1R+3R	63	57	25	35	47	43
2R+3R	62	35	680	36	25	118
1R+2R+3R	**85**	20	32	**38**	16	50

1R, 2R, and 3R denote one-point, two-point, and three-point RANSAC, respectively. IR is the inlier ratio. In means the inlier number.

TABLE VI
ABLATION EXPERIMENTS ON REGISTRATION PERFORMANCE FOR 3R AND IRLS

	3R	IRLS	Arch rr (%)	Arch e_r (°)	Arch e_t (m)	Arch time (ms)	06 rr (%)	06 e_r (°)	06 e_t (m)	06 time (ms)
1R+2R	✓		**88**	0.19	0.06	**30**	74	0.86	0.30	**39**
1R+2R		✓	**88**	0.17	0.05	31	74	0.63	0.25	40
1R+2R	✓	✓	**88**	**0.13**	**0.04**	32	**75**	**0.51**	**0.23**	43

Both 3R and IRLS can compute a 6-dof pose.

results show an overall recall of over 98%. With 98% outliers, the translation error stays below the noise level, and the rotation error remains under 2°, highlighting the method's robustness.

F. Ablation Study

Outlier Removal: Table V examines the contributions of our three RANSAC modules in outlier removal. IR is the inlier ratio while IN means the inlier number. In the arch, the slowest three-point RANSAC (3R) requires 3234 ms and achieves an inlier ratio of 27%. The fastest one-point RANSAC (1R) completes in 20 ms, delivering an inlier rate of 21%. Compared to 3R, 1R+2R+3R increases the inlier ratio by 58% and reduces runtime by 3202 ms. In sequence 01, 1R remains the fastest, taking 29 ms with 8% inliers. The slowest 3R requires 3139 ms and achieves 28% inliers. 1R+2R+3R attains the highest inlier ratio at 38%, reduces runtime by 3089 ms compared to 3R, and improves the inlier ratio by 30% relative to 1R. The results illustrate that our method decreases iterations for 3R and preserves a high inlier ratio.

PCR Performance: Table VI examines the contributions of 3R and IRLS to final registration performance. In Arch, 3R+IRLS reduces rotation errors by 0.06° and translation errors by 0.02 m compared to using only 3R. It shows improvements of 0.04° and 0.01 m upon IRLS. It takes only 1–2 ms longer than 3R and

IRLS. In sequence 06, 3R+IRLS enhances recall by 1% over other pipelines. Compared to 3R, it reduces angle error by 0.35° and translation error by 0.07 m. Compared to IRLS, it reduces errors by 0.12° and 0.02 m. It takes just an extra 4 ms than the fastest 3R. These results indicate: (1) with most outliers removed by 1R+2R, the subsequent registration converges quickly, and (2) our overall pipeline (1R+2R+3R+IRLS) balances accuracy and efficiency.

V. Conclusion

We propose a two-stage consensus filtering method for correspondence-based registration. Our approach restores the RANSAC family to SOTA performance. By combining length consistency with angle consistency, we eliminate the majority of outliers, thus reducing the requisite iterations by three-point RANSAC. We combine three RASNAC modules and iterative weighted least squares into a complete registration pipeline that accelerates registration without compromising accuracy. Experimental results demonstrate a three-orders-of-magnitude speed enhancement while maintaining high registration quality. Moreover, integrating our method with existing approaches can notably improve efficiency and recall.

References

[1] 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. 2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13211–13221.

[2] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3D registration with maximal cliques," in *Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17745–17754.

[3] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3Dmatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 199–208.

[4] Q. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 766–782.

[5] Y. Wang, P. Zhou, G. Geng, L. An, and Y. Liu, "CCAG: End-to-end point cloud registration," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 435–442, Jan. 2024.

[6] P. Shi, Q. Ye, and L. Zeng, "A novel indoor structure extraction based on dense point cloud," *ISPRS Int. J. Geo- Inf.*, vol. 9, no. 11, 2020, Art. no. 660.

[7] P. Shi, J. Li, and Y. Zhang, "LiDAR localization at 100 FPS: A map-aided and template descriptor-based global method," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 120, 2023, Art. no. 103336.

[8] Q. Ye, P. Shi, K. Xu, P. Gui, and S. Zhang, "A novel loop closure detection approach using simplified structure for low-cost LiDAR," *Sensors*, vol. 20, no. 8, 2020, Art. no. 2299.

[9] P. Shi, J. Li, and Y. Zhang, "A fast LiDAR place recognition and localization method by fusing local and global search," *ISPRS J. Photogrammetry Remote Sens.*, vol. 202, pp. 637–651, 2023.

[10] P. Shi, Y. Xiao, W. Chen, J. Li, and Y. Zhang, "A new Horizon: Employing map clustering similarity for LiDAR-based place recognition," *IEEE Trans. Intell. Veh.*, early access, Jan. 30, 2024, doi: 10.1109/TIV.2024.3360321.

[11] Z. J. Yew and G. H. Lee, "REGTR: End-to-end point cloud correspondences with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6677–6686.

[12] Y. Zhang, P. Shi, and J. Li, "3D lidar slam: A survey," *Photogrammetric Rec.*, vol. 39, pp. 457–517, 2024.

[13] P. Shi, Y. Zhang, and J. Li, "LiDAR-based place recognition for autonomous driving: A survey," 2023, *arXiv:2306.10561*.

[14] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[15] P. Shi, Q. Ye, Z. Shaoming, and D. Haifeng, "Localization initialization for multi-beam LiDAR considering indoor scene feature," *Acta Geodaetica et Cartographica Sinica*, vol. 50, pp. 1594–1604, 2021.

[16] P. Shi, J. Li, X. Liu, and Y. Zhang, "Indoor cylinders guided LiDAR global localization and loop closure detection," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 49, no. 7, pp. 1088–1099, 2024.

[17] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[18] S. Quan and J. Yang, "Compatibility-guided sampling consensus for 3-D point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7380–7392, Oct. 2020.

[19] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3523–3532.

[20] J. Li, P. Shi, Q. Hu, and Y. Zhang, "QGORE: Quadratic-time guaranteed outlier removal for point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11136–11151, Sep. 2023.

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

[22] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.

[23] D. Barath and J. Matas, "Graph-cut RANSAC," in *Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6733–6741.

[24] S. H. Lee, J. Civera, and P. Vandewalle, "PCR-99: A practical method for point cloud registration with 99% outliers," 2024, *arXiv:2402.16598*.

[25] P. C. Lusk, K. Fathian, and J. P. How, "Clipper: A graph-theoretic framework for robust data association," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13828–13834.

[26] J. Shi, H. Yang, and L. Carlone, "ROBIN: A graph-theoretic approach to reject outliers in robust estimation using invariants," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13820–13827.

[27] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[28] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, USA, 2009, vol. 2, no. 4, pp. 435–442.

[29] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[30] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[31] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[32] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. 28th Int.Conf. Neural Inf. Process. Syst.*, 2015, pp. 2692–2700.

[33] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[34] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 195–205.

[35] H. Deng, T. Birdal, and S. Ilic, "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 602–618.

[36] M. Marcon, R. Spezialetti, S. Salti, L. Silva, and L. Di Stefano, "Unsupervised learning of local equivariant descriptors for point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9687–9702, Dec. 2022.

[37] L. Jiayuan, Z. Yongjun, A. Mingyao, and H. Qingwu, "Scale-adaptive cauchy robust estimation based on progressive optimization and its applications," *Acta Geodaetica et Cartographica Sinica*, vol. 52, no. 1, pp. 61–70, 2023.

[38] P. W. Theiler, J. D. Wegner, and K. Schindler, "Globally consistent registration of terrestrial laser scans via graph optimization," *ISPRS J. Photogrammetry Remote Sens.*, vol. 109, pp. 126–138, 2015.

[39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The Kitti vision benchmark suite," in *Proc. 2012 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[40] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 689–696.