A Novel Sketch-Based Framework Utilizing Contour Cues for Efficient Point Cloud Registration

Gang Ma and Hui Wei

***Abstract*— Point cloud registration is a crucial part of 3-D computer vision. Existing point cloud registration methods primarily concentrate on utilizing features such as points, lines, and planes, disregarding the valuable contour cues inherent in the scene. In this article, we propose a novel sketch-based framework for point cloud registration that incorporates contour cues to enhance the point cloud registration task. To fully exploit the abundant information provided by contour cues in the scene, the point cloud is first abstracted into a sketch consisting of contour cues obtained through the utilization of planar features, which greatly preserves the inherent contour information. Subsequently, a local contour geometric descriptor is introduced to encode the contour cues in the sketch. Finally, a voting-based contour point pair feature (CPPF) framework is employed to fuse planar features, local contour geometric features, and point pair geometric features, enabling precise estimation of the pose transformation between pairwise point clouds. Extensive experiments conducted on two large-scale outdoor point cloud datasets and two indoor point cloud datasets validate the effectiveness of the proposed sketch-based method. Our proposed method successfully suppresses rotation and translation errors, ultimately achieving state-of-the-art performance.**

***Index Terms*— Contour point pair feature (CPPF), local contour geometric descriptor, point cloud registration, point cloud, sketch-based registration.**

I. Introduction

REGISTRATION of point clouds captured by 3-D scanner devices is an active research area, whose applications in 3-D computer vision and robotics [1] include 3-D scene reconstruction [2], simultaneous localization and mapping (SLAM) [3], indoor modeling [4], and virtual reality [5]. Given a source cloud $\mathcal{P}$ and target cloud $\mathcal{Q}$, point cloud registration is dedicated to estimating a 6-Degree-of-Freedom (DoF) transformation $\boldsymbol{T} = [\boldsymbol{R}|\boldsymbol{t}] \in \boldsymbol{SE}(3)$ to transform $\mathcal{P}$ into the target coordinate system of $\mathcal{Q}$ [6].

The iterative closest point (ICP) [7] is widely used to align point clouds, alternately conducting correspondence search and pose transformation estimation, and it has several variants [8], [9], [10]. ICP, a local point cloud registration method, is often trapped in local minima due to its sensitivity to initialization. Correspondence-based methods offer a global approach, can effectively reduce point cloud size [11], and have gained significant attention. They typically involve the steps of correspondence establishment and pose transformation estimation [12]. Two widely adopted approaches are based on point correspondence [13], [14] and primitive correspondence [15], [16]. Point correspondence methods establish correspondences through local point features [17], [18], while primitive correspondence methods consider higher level geometric features for alignment, such as lines [19], planes [15], and spheres [16]. Outlier removal-based methods, both guaranteed [20], [21], [22] and nonguaranteed [11], [23], [24], have been shown to effectively filter noise in correspondences based on point features.

Learning-based methods, which can be categorized as correspondence-based [25], [26], [27], [28] or end-to-end [29], [30], [31], have recently emerged for point cloud registration tasks and can achieve high-precision pose transformations. Learning-based methods rely on computing power and have not fully realized their potential for registration tasks in unseen scenes, leading to limited generalization ability, as typified by two recent methods: PointDSC [27] and RegTR [31].

Pose-estimation methods typically rely on features such as points, line segments, and planes, often disregarding the informative contour cues present in scenes. We overcome this limitation through a sketch-based point cloud registration framework that effectively incorporates contour cues, mitigating rotation, and translation errors with improved precision.

The main contributions of this work are summarized as follows.

1) We propose a preprocessing scheme for point cloud sketching and a planar local contour geometric feature descriptor, which is specifically designed for nonclosed contours while preserving the essential properties of rotation, translation, and scale-invariance.
2) A voting-based pairwise registration method is proposed, which effectively integrates planar features, local contour geometric features, and point pair geometric features, to enable comprehensive global modeling of point pair geometric information and optimize the registration pose in a locally reduced search space.

Manuscript received 29 May 2023; revised 4 August 2023; accepted 16 August 2023. Date of publication 21 August 2023; date of current version 5 September 2023. This work was supported by the National Science Foundation of China under Grant 61771146. *(Corresponding author: Hui Wei.)*

The authors are with the Laboratory of Algorithms for Cognitive Models, the Shanghai Key Laboratory of Data Science, and the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: mag20@fudan.edu.cn; weihui@fudan.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3307061

3) In experiments on several datasets, our method effectively suppressed rotation and translation errors, surpassing the performance of baselines.

The remainder of this article is structured as follows. Related work is discussed in Section II. Pairwise alignment is explained in Section III. Section IV discusses experiments using the proposed framework. We summarize this article in Section V.

II. Related Work

We review methods for point cloud registration from correspondence-based and correspondence-free approaches. We focus on rigid registration between two point clouds and do not consider registration in dynamic scenes.

A. Correspondence-Based Point Cloud Registration

1) Point Feature-Based Methods: Algorithms in this category usually have the stages of keypoint detection, feature descriptor, and correspondence selection.

Keypoint detection is accomplished using algorithms such as intrinsic shape signature (ISS) [32] and 3-D Harris [33]. Local point feature descriptors are then used to encode the local features of keypoints to distinguish them [16], [34], such as those based on signatures or histograms derived from statistical geometric properties [14]. The spin image algorithm [35] projects neighboring points around a keypoint onto a 2-D plane, encodes density information, and is sensitive to data resolution variation and noise. Fast point feature histogram (FPFH) [17] constructs reference frames from local shapes for selected point pairs, making it fast and discriminative. Rotation projection statistics (RoPS) [36] integrates the central moment and entropy of the projected 2-D maps of many rotated local surfaces into a 1-D vector. It is not descriptive for data with uneven point distributions and is time-consuming. Inspired by RoPS, Yang et al. [18] proposed the rotation contour signatures (RCSs), which have the advantages of strong robustness and a compact structure, but their complexity results in low computational efficiency. Features based on signatures or histograms have been widely used in point cloud registration tasks. In the correspondence selection stage, the encoded point feature descriptors are utilized to establish point pair matching between the keypoints in the source and target scenes. The correspondence selection method usually employs a feature similarity score [13], [37], the nearest neighbor similarity ratio (NNSR) [14], [36], and 3-D Hough voting (3DHV) [38] to determine the corresponding point pairs. Fast global registration (FGR) [13] is an optimization-based method that uses FPFH features for point cloud registration. Yang et al. [14] proposed a voting-based 2.5-D scene registration scheme, combining RCS features with NNSR matching.

Different from the above, the proposed method combines planar features and local contour geometric features and integrates them into a voting-based scheme.

2) Primitive Feature-Based Methods: In 3-D scenes, besides point features, there are various primitive features such as lines, planes, cylinders, cones, and spheres. A popular approach nowadays involves determining the registration pose of a point cloud by matching corresponding primitives [16], [19], [39], [40].

To obtain the coarse registration pose of urban point clouds, Xu et al. [19] introduced a framework that relies on two-line congruent sets. Similarly, Liu et al. [39] and Yu et al. [40] developed registration frameworks tailored for point clouds of rock mass, focusing on line and planar features, respectively. Chen et al. [15] proposed the PLADE method, which employs plane-line descriptors to establish correspondences between source and target point clouds and has enhanced resilience to point cloud resolution and noise. Unlike PLADE, Zhang et al. [16] took a different approach by extending the set of primitives from planes to include geometric primitives, such as cylinders, cones, and spheres. They constructed descriptors based on hybrid structures, ultimately achieving successful registration.

The combination of line and planar features is advantageous in computational geometry. Reyes-Aviles et al. [5] proposed a registration method that leverages geometric constraints extracted from parametric primitives within a 3-D parametric model, providing closed-form solutions for registration involving three types of correspondences: lines to lines, lines to planes, and planes to planes. Furthermore, line and planar features have gained popularity in the field of SLAM [41], [42], [43], [44], as evidenced by various works.

Primitive-based methods are generally more robust than point-based methods, particularly relating to noise and initial positioning. However, the accuracy of these methods relies heavily on the accurate extraction of primitive features and the precise estimation of corresponding primitive features.

3) Outlier Removal Methods: At present, point cloud registration methods that use outlier removal employ either nonguaranteed [11], [13], [23], [24] or guaranteed [20], [21], [22] mechanisms.

FGR [13] is a well-known nonguaranteed outlier removal method that utilizes a global approach combining straight line processing with robust estimation to enhance the optimization process. Despite being highly efficient, it can produce inaccurate results in complex registration scenarios. Similarly, Li et al. [23] proposed a robust, efficient method employing nonguaranteed outlier removal based on topological graphs and Cauchy weighted l_q-norm, which transforms point matching into graph edge matching and uses an edge voting strategy to differentiate between potential correct matches and mismatches. This method can still achieve accurate results even in scenarios with extremely high outlier rates.

The guaranteed outlier removal mechanism is designed to eliminate all of the outliers [22], which renders outlier optimization unnecessary. GORE [20], [21] uses a novel rotation search outlier removal technique based on simple geometric operations. Each correspondence is initially treated as an inlier, and upper and lower bounds on the consensus size are determined. If the bounds conflict, the correspondence is deemed a true outlier and removed [22], [24]. GORE is highly effective in removing true outliers, but its efficiency is somewhat limited due to the use of a Branch-and-Bound

(BnB) algorithm [45], [46]. Teaser++ [47] reformulates the registration problem through a truncated least squares (TLS) cost and uses a versatile graph-theoretic framework for outlier removal. It is highly efficient, but its performance may degrade significantly and require substantial memory resources when there are a large number of correspondences. Graph Reliability Outlier Removal (GROR) [22] utilizes an outlier removal strategy that depends on the reliability of correspondence graphs. It introduces the concept of reliability for graph nodes to facilitate optimal candidate selection and reliability for graph edges to determine the global maximum consensus set. GROR enables efficient and accurate outlier removal, even in scenes with high outlier ratios.

The abovementioned methods rely on point-corresponding outlier removal techniques. Integrating these methods with advanced primitive features present in the scene can potentially enhance the accuracy of the results.

4) Learning-Based Methods: The rapid progress of deep learning has led to the emergence of numerous learning-based local point cloud features [48], [49], [50], [51]. Notably, FCGF [49] has become one of the extensively employed learning-based features. The first step in correspondence-based point cloud registration methods involves utilizing precomputed features to establish point correspondences. Subsequently, these methods employ a learning-based outlier filtering module for correspondence filtering [25], [27], [52], [53], which enhances the accuracy of estimated poses. DGR [25] presented a novel approach incorporating a 6-D sparse convolutional network to effectively capture global context for correspondence classification and weighted Procrustes for accurate pose estimation. PointDSC [27] introduced deep spatial consistency to the registration framework, enhancing performance by pruning outlier correspondences. RoReg [52] is an innovative point cloud registration framework that maximizes the utilization of orientation descriptors and estimated local rotations. Through labeling and training processes, the aforementioned learning-based methods have achieved exceptional results on several point cloud registration datasets.

B. Correspondence-Free Point Cloud Registration

1) ICP-Based Methods: The ICP [7] and its variants are widely used for point cloud registration tasks. ICP directly performs the closest point correspondence search on two point sets, requiring proper initialization to avoid getting stuck in local extrema. Generalized-ICP [8] improves distance metric by modeling the local planar surface structure of source and target point clouds through a plane-to-plane approach that exhibits stronger robustness. Sparse ICP [54] preserves the simplicity of ICP while applying computationally efficient sparsity-inducing norms to registration optimization, demonstrating excellent performance with outliers and incomplete data. Fast ICP [10] employs an Anderson acceleration method and a robust error metric based on Welsch's function to improve accuracy and efficiency on noisy and partially overlapping datasets. VGICP [55] is a voxelization-based generalized ICP algorithm that avoids expensive nearest-neighbor searches. Despite their advantages, the aforementioned are local methods that are sensitive to initialization and prone to becoming stuck in local extrema. To address this issue, Go-ICP [9] uses the BnB method [45], [46], but this approach may introduce computational complexity, with the worst case being exponential.

2) Learning-Based Methods: Some recent learning-based approaches [26], [29], [30], [31] have introduced a fresh perspective on 3-D point cloud registration. Instead of establishing explicit correspondences between source and target point clouds, they compare their holistic representations to effectively solve rigid transformations. PointNetLK [30] is a pioneering method in this domain, utilizing PointNet [56] to extract global features from both the source and target point clouds. By analyzing the disparity between these global features, a modified Lucas–Kanade (LK) algorithm [57] is employed to estimate the rigid transformation. RegTR [31] is an end-to-end correspondence set prediction network that utilizes an attention mechanism and a network architecture comprising self-attention and cross-attention transformation layers to predict the probability of each point lying in an overlapping region and its corresponding position in other point clouds. Although learning-based methods have shown the potential for point cloud alignment, they usually require sufficient labeled data for training and may be ineffective on unseen scenes.

III. Methodology

The pipeline of the proposed voting-based registration method is depicted in Fig. 1. First, planar primitive features are extracted from the source and target point clouds, followed by the utilization of planar primitive contours to outline the sketch of each point cloud. Then, a local contour geometric descriptor, histogram entropy of angular distribution (HEAD), is employed to encode each planar contour in a sketch. Finally, a novel voting-based method, contour point pair feature (CPPF), leverages the integration of the local contour geometric feature HEAD with the point pair geometric feature to accurately estimate the transformation between the two sketches.

A. 3-D Scene Sketching

Sketching is a simple way for humans to describe and express information [58]. Consequently, we sketch point clouds by extracting planar primitives.

When abstracting scenes into a set of planar primitives, we employ an iterative "fit and remove" strategy based on RANSAC [59] to extract planar primitive features in each scene. The contours of all of the planar primitives are obtained, which plays a pivotal role in the proposed framework for pairwise registration. From a point cloud, the set of extracted planes is denoted as $\mathcal{P} = \{\pi_1, \ldots, \pi_N\}$, and the set of corresponding planar contours is $\mathcal{C} = \{C_1, \ldots, C_N\}$. N is the number of planar primitives in the scene, and $\mathcal{C}$ signifies the sketch corresponding to the given scene. As illustrated in Fig. 2, the process of abstracting a point cloud into a sketch using planar contours preserves the contour cues to a significant degree while reducing the overall size of the point cloud.

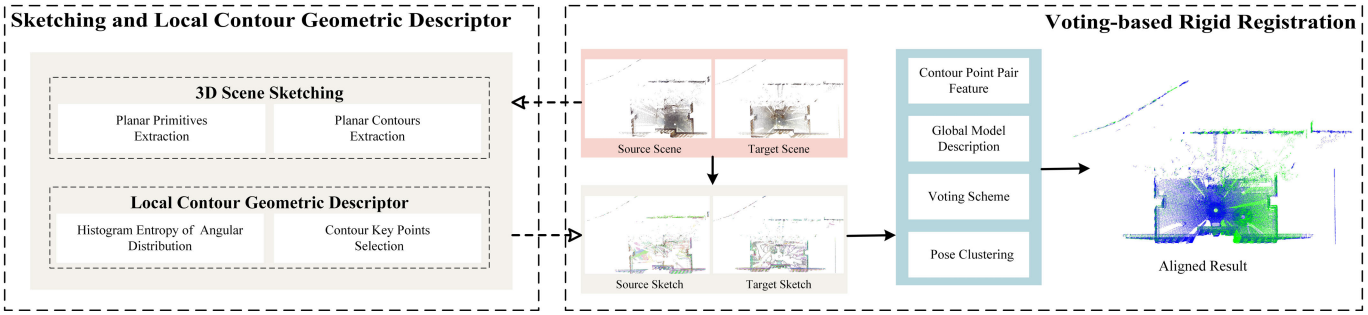


Fig. 1. Pipeline of the proposed sketch-based pairwise alignment method.

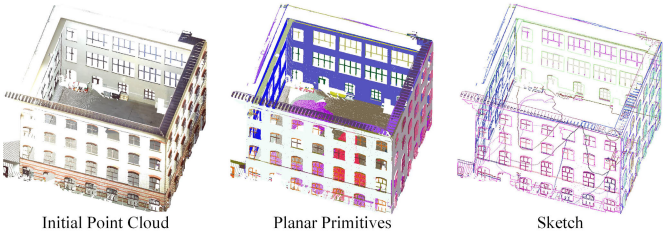


Fig. 2. Snapshots consist of an initial point cloud of a scene, along with the planes present and their corresponding sketch data. The initial point cloud comprises 18 958 883 points, while the sketch data consists of 260 461 points.

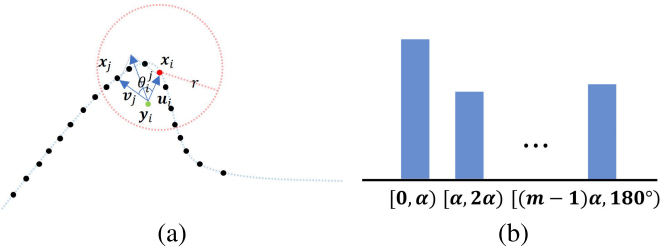


Fig. 3. Description of HEAD for nonclosed contours. (a) Red point $\boldsymbol{x}_i$ is a reference point, black points distributed in red-dashed circle are related points $\boldsymbol{x}_j (j \in [1, k])$, and green point $\boldsymbol{y}_i$ indicates barycenter of related points, $\boldsymbol{u}_i = \overrightarrow{\boldsymbol{y}_i \boldsymbol{x}_i}$ and $\boldsymbol{v}_j = \overrightarrow{\boldsymbol{y}_i \boldsymbol{x}_j}$. (b) Histogram of orientation angular distribution, accumulating $\| \boldsymbol{u}_i + \boldsymbol{v}_j \| (j \in [1, k])$ for each bin if $\| (\boldsymbol{u}_i, \boldsymbol{v}_j) \|$ is mapped to this bin.

B. Planar Contour Geometric Descriptor

Representing 3-D scenes using sketches poses a significant challenge, which we address by constructing a 2-D contour shape feature characterization from planar contours. Drawing inspiration from a 2-D contour descriptor [60], we propose the HEAD descriptor, which is well-suited for nonclosed contours, while ensuring rotation, translation, and scale-invariance. The entropy of the HEAD descriptor reflects the distribution of local contour points, and the magnitude of the entropy intuitively indicates the significance of points on the contour.

$C \in \mathcal{C}$ is a planar contour of a sketch $\mathcal{C}$. Let $X = \{\boldsymbol{x}_i\} (i \in [1, n])$ indicate a set of points from the planar contour C. $\boldsymbol{x}_i$ is the reference point, whose local geometric feature is related to neighbor points in the field of radius r, called related points, with size k. $\boldsymbol{y}_i$ indicates the barycenter of the related points for reference point $\boldsymbol{x}_i$.

As illustrated in Fig. 3(a), the constant vector $\boldsymbol{u}_i$ is from $\boldsymbol{y}_i$ to $\boldsymbol{x}_i$, and $\boldsymbol{v}_j (j \in [1, k])$ denotes the vector from $\boldsymbol{y}_i$ to $\boldsymbol{x}_j$. $\angle(\boldsymbol{u}_i, \boldsymbol{v}_j) (j \in [1, k])$ for the reference point $\boldsymbol{x}_i$ is defined as θ_i^j, and all of the angle values with respect to the point $\boldsymbol{x}_i$ are denoted as $\boldsymbol{\Theta}_i = (\theta_i^1, \ldots, \theta_i^k)^T$. In Fig. 3(b), the range $[0°, 180°)$, divided into m bins, has m disjoint intervals. θ_i^j is mapped to a bin, and each bin accumulates $\| \boldsymbol{u}_i + \boldsymbol{v}_j \|$ of related point $\boldsymbol{x}_j$ if θ_i^j is mapped to it. Thus, the value of the dth bin is calculated as

$$t_i^d = \sum_{j=1}^{k} \| \boldsymbol{u}_i + \boldsymbol{v}_j \| \cdot \Omega[\mathcal{F}(\theta_i^j) - d] \tag{1}$$

where $\mathcal{F}(\cdot)$ maps an angle into the bin index. $\Omega[\cdot]$ is an indicator function, taking the value 0 when $\mathcal{F}(\theta_i^j) \neq d$ and 1 otherwise. In this way, a new feature vector of the reference point $\boldsymbol{x}_i$ is obtained as $\boldsymbol{H}_i = (t_i^1, t_i^2, \ldots, t_i^m)^T$, whose each column is normalized to keep the shape scale-invariant

$$\boldsymbol{u}_i = \frac{\boldsymbol{u}_i}{\| \boldsymbol{u}_i \|}, \quad \boldsymbol{v}_i = \frac{\boldsymbol{v}_i}{d\boldsymbol{v}_i}, \quad \boldsymbol{H}_i = \frac{\boldsymbol{H}_i}{\| \boldsymbol{H}_i \|} \tag{2}$$

where $d\boldsymbol{v}_i$ is the maximum distance between $\boldsymbol{y}_i$ and $\boldsymbol{x}_j (j \in [1, k])$.

Furthermore, entropy of the feature vector $\boldsymbol{H}_i$, denoted as $\mathcal{E}$, is defined as

$$\mathcal{E}_i = -\sum_{j=1}^{m} \frac{\boldsymbol{H}_i^j}{\sum_{l=1}^{m} \boldsymbol{H}_i^l} \cdot \log_2 \left(\frac{\boldsymbol{H}_i^j}{\sum_{l=1}^{m} \boldsymbol{H}_i^l} \right). \tag{3}$$

$\mathcal{E} \in [0, \log_2^m]$, and m is the size of bins in Fig. 3(b). The entropy of feature vector $\boldsymbol{H}_i$ reflects the distribution of related points. If the related points of reference point $\boldsymbol{x}_i$ are distributed on a straight line, then $\mathcal{E}_i = 0$. The keypoints on a contour, inflection points, or points with large curvature tend to have large entropy values, but points in the smooth area on the contour do not.

C. Voting-Based Registration

The point pair feature (PPF) [61] is extensively applied in shape-based object recognition and combines global modeling and local matching. We present a voting-based method, the CPPF for sketch-based point cloud registration, which builds upon the PPF framework while incorporating the entropy of local contour geometric features. By introducing entropy channels, CPPF descriptors provide enhanced discriminability compared with traditional PPF, facilitating a more effective characterization of planar contour points.

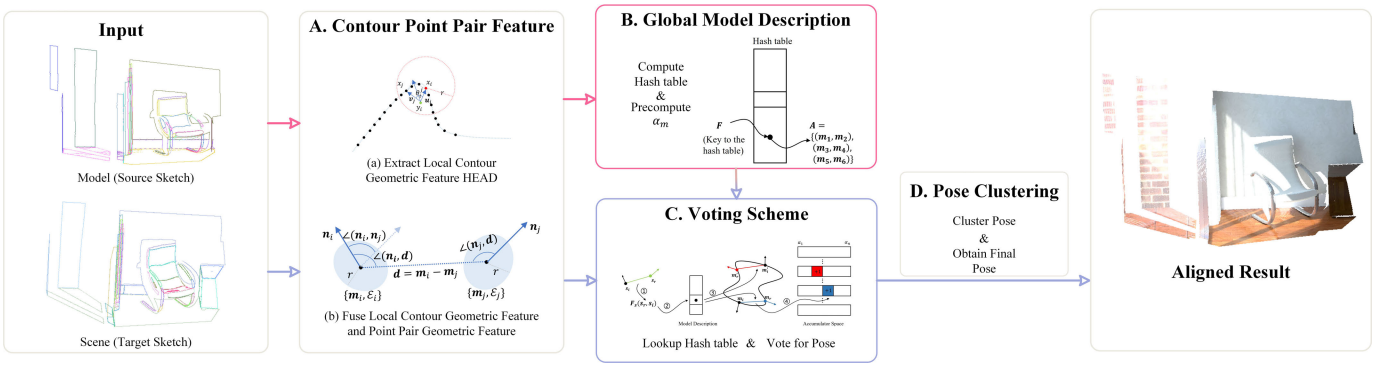


Fig. 4. Outline of voting mechanism for pairwise registration mainly containing four parts: CPPF, global model description, voting scheme, and pose clustering. (a) Extract local contour geometric feature HEAD. (b) Fuse local counter geometric feature and point pair geometric feature.

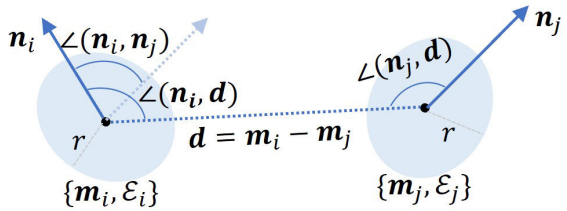


Fig. 5. CPPF is defined by contour orientated points $(\boldsymbol{m}_i, \boldsymbol{n}_i, \mathcal{E}_i)$ and $(\boldsymbol{m}_j, \boldsymbol{n}_j, \mathcal{E}_j)$. Each dimension is determined by the Euclidean distance between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, angles between point normal and distance vector, and two entropy channels of HEAD descriptors for two points. r is the domain radius associated with the local contour geometric feature HEAD.

We assume that the model and scene data come from the source sketch and the target sketch, respectively. They are both denoted as finite sets of oriented points with normal and entropy information. Points in the model are denoted as $\boldsymbol{m}_i \in \mathcal{M}$ and $\boldsymbol{s}_i \in \mathcal{S}$ for points in the scene. As shown in Fig. 4, the proposed voting-based method mainly contains four parts: CPPF, global model description, voting scheme, and pose clustering.

1) Contour Point Pair Feature: The CPPF vector $\boldsymbol{F}_{\text{CPPF}} \in \mathbb{R}^6$ is illustrated in Fig. 5 and is defined as

$$\begin{aligned}\boldsymbol{F}_{\text{CPPF}} &= \text{CPPF}(\boldsymbol{m}_i, \boldsymbol{m}_j, \boldsymbol{n}_i, \boldsymbol{n}_j, \mathcal{E}_i, \mathcal{E}_j) \\ &= \left(\| \boldsymbol{d} \|_2, \angle(\boldsymbol{n}_i, \boldsymbol{d}), \angle(\boldsymbol{n}_j, \boldsymbol{d}), \angle(\boldsymbol{n}_i, \boldsymbol{n}_j), \mathcal{E}_i, \mathcal{E}_j\right)^T \end{aligned} \quad (4)$$

where distance component $\| \boldsymbol{d} \|_2 = \| \boldsymbol{m}_i - \boldsymbol{m}_j \|_2 \in \mathbb{R}^+$ represents the distance between points $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$. Angle components are angles between the vector $\boldsymbol{d}$ and surface normal vectors $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$. Entropy component $(\mathcal{E}_i, \mathcal{E}_j)$ indicates the HEAD features for two points. As described by (4), the proposed CPPF effectively integrates planar primitive features, point pair geometric features, and local contour geometric features.

2) Global Model Description: The CPPF is used to build a global model description. When there are N_m points in the sketch point cloud $\mathcal{M}$, $N_m \times N_m$ CPPFs are computed and saved for later feature matching. The hash table technique is used due to its fast query operation [61], [62], [63]. To use the CPPF as the key for the hash table, we quantize the feature vector as follows:

$$\begin{aligned}\boldsymbol{K} &= \text{HashKey}(\boldsymbol{F}_{\text{CPPF}}, \sigma, \psi, \tau) \\ &= \begin{pmatrix} \left\lfloor \frac{\| \boldsymbol{d} \|_2}{\sigma} \right\rfloor \\ \left\lfloor \frac{\angle(\boldsymbol{n}_i, \boldsymbol{d})}{\psi} \right\rfloor \\ \left\lfloor \frac{\angle(\boldsymbol{n}_j, \boldsymbol{d})}{\psi} \right\rfloor \\ \left\lfloor \frac{\angle(\boldsymbol{n}_i, \boldsymbol{n}_j)}{\psi} \right\rfloor \\ \left\lfloor \frac{\mathcal{E}_i}{\tau} \right\rfloor \\ \left\lfloor \frac{\mathcal{E}_j}{\tau} \right\rfloor \end{pmatrix} \end{aligned} \quad (5)$$

where $\sigma \in \mathbb{R}$, $\psi \in \mathbb{R}$, and $\tau \in \mathbb{R}$ are respective quantization levels for the distance, angle, and entropy of contour feature vectors. The hash key $\boldsymbol{K} \in \mathbb{Z}^6$ is encoded by the BitEncode function defined as

$$\mathcal{K} = \text{BitEncode}(\boldsymbol{K}) \quad (6)$$

where $\mathcal{K} \in \mathbb{Z}$ is a 64-bit key, with the 64 bits (8 bytes) allocated as follows: the first dimension, the distance component, is two bytes; each dimension of angle and entropy components is one byte; and the last byte is reserved and zeroed.

Algorithm 1 describes the process of building the global model description. Given model data $\mathcal{M}$ with N_m points from the scene sketch point cloud as an input, the algorithm outputs the hash keys $\mathcal{H}$ with intermediate angle array $\mathcal{A}$. Given a model point cloud data $\mathcal{M}$, we first calculate HEAD features and CPPF features of all of the point pairs. Then, for each CPPF feature, we map it to a discrete space to obtain feature vector $\boldsymbol{k}$. BitEncode is used to encode $\boldsymbol{k}$ into a 64-bit numerical value to perform hash mapping. To quickly calculate the transformation pose of matching point pairs, the PlanarRotAngle function is adopted to calculate α_m, as described in (8)–(10), during hash table construction. Finally, the algorithm returns the hash table $\mathcal{H}$ and the matrix $\mathcal{A}$ of related angles.

3) Voting Scheme: With the hash table we have built, the CPPF correspondences between scene and model can be quickly established. Assuming that a correct match of CPPFs

Algorithm 1 Building Global Model Description

Input: $\mathcal{M}$
$\mathcal{H} \leftarrow \mathbf{0}_{N_m \times N_m}$
$\mathcal{A} \leftarrow \mathbf{0}_{N_m \times N_m}$
for $i \leftarrow 1$**toN**$_\mathbf{m}$ **do**
  for $j \leftarrow 1$**toN**$_\mathbf{m}$ **do**
    if $i \neq j$ **then**
      $\mathcal{E}_i^m \leftarrow \text{HEAD}(\boldsymbol{p}_i^m)$
      $\mathcal{E}_j^m \leftarrow \text{HEAD}(\boldsymbol{p}_j^m)$
      $\boldsymbol{F} \leftarrow \text{CPPF}(\boldsymbol{p}_i^m, \boldsymbol{p}_j^m, \boldsymbol{n}_i^m, \boldsymbol{n}_j^m, \mathcal{E}_i^m, \mathcal{E}_j^m)$
      $\boldsymbol{k} \leftarrow \text{HashKey}(\boldsymbol{F}, \sigma, \psi, \tau)$
      $\leftarrow \text{BitEncode}(\boldsymbol{k})$
      $\alpha_m \leftarrow \text{PlanarRotAngle}(\boldsymbol{n}_i^m, \boldsymbol{p}_i^m, \boldsymbol{p}_j^m)$
      $\mathcal{H}(i, j) \leftarrow$
      $\mathcal{A}(i, j) \leftarrow \alpha_m$
    end
  end
end
Output: $\mathcal{H}, \mathcal{A}$

between scene and model point clouds is found, as explained in Fig. 6, the alignment of matched points $\{\boldsymbol{s}_r, \boldsymbol{m}_r\}$ contains a 3-DoF translation, while the alignment of matched normals $\{\boldsymbol{n}_s^r, \boldsymbol{n}_m^r\}$ contains a 2-DoF rotation. Hence, only a 1-DoF rotation $\alpha \in \mathbb{R}$ around the $\boldsymbol{x}$-axis of the intermediate coordinate system must be determined. Once α is calculated by two vectors $\boldsymbol{m}_i - \boldsymbol{m}_r$ and $\boldsymbol{s}_i - \boldsymbol{s}_r$, the 6-DoF transformation from local model coordinate to scene coordinate [61] is defined by

$$\begin{aligned} \boldsymbol{s}_i &= \boldsymbol{T}_{s\to g}^{-1} \boldsymbol{R}(\alpha) \boldsymbol{T}_{m\to g} \boldsymbol{m}_i \\ \boldsymbol{T}_{s\to g} &\in \text{SE}(3) \\ \boldsymbol{T}_{m\to g} &\in \text{SE}(3) \\ \boldsymbol{R}(\alpha) &\in \text{SO}(3) \end{aligned} \tag{7}$$

where $\boldsymbol{R}_{\boldsymbol{x}}(\alpha)$ rotates around the $\boldsymbol{x}$-axis of the intermediate coordinate system with angle α. To evaluate (7) more quickly, α is split into two parts [61], α_m and α_s, depending only on the model and scene, respectively. Then, we obtain

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{x}}(\alpha) &= \boldsymbol{R}_{\boldsymbol{x}}(-\alpha_s) \boldsymbol{R}_{\boldsymbol{x}}(\alpha_m) \\ \boldsymbol{R}_{\boldsymbol{x}}{}^{-1}(-\alpha_s) &= \boldsymbol{R}_{\boldsymbol{x}}(\alpha_s) \end{aligned} \tag{8}$$

and

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{R}_{\boldsymbol{x}}(\alpha_s) \boldsymbol{T}_{s\to g} \boldsymbol{s}_i \\ &= \boldsymbol{R}_{\boldsymbol{x}}(\alpha_m) \boldsymbol{T}_{m\to g} \boldsymbol{m}_i \in \mathbb{R}\boldsymbol{x} + \mathbb{R}_0^+ \boldsymbol{y} \end{aligned} \tag{9}$$

where $\boldsymbol{w}$ lies on the half-plane defined by the $\boldsymbol{x}$-axis and the nonnegative part of the $\boldsymbol{y}$-axis in the intermediate coordinate system. Finally, α is simply defined as

$$\alpha = \alpha_m - \alpha_s \tag{10}$$

where α_m is precomputed and saved in the model descriptor (see Algorithm 1) and α_s is calculated only once when voting for the pose (see Algorithm 2).

Once α is computed, a vote is cast for the local coordinate $(\boldsymbol{m}_i, \alpha)$ to obtain a 6-DoF transformation, as described in Algorithm 2. Given model $\mathcal{M}$, scene $\mathcal{S}$, hash table $\mathcal{H}$, and

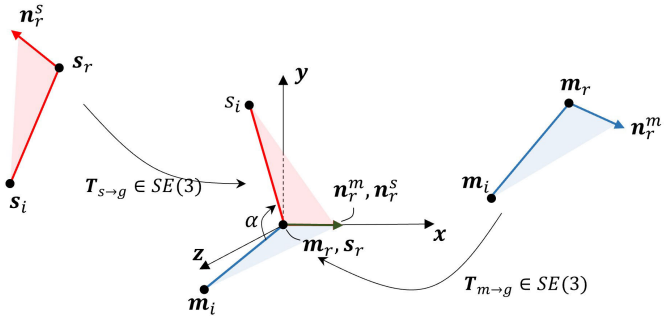


Fig. 6. Transformation between two coordinates from model and scene. $\boldsymbol{T}_{m\to g}$ translates $\boldsymbol{m}_r$ into the origin and rotates its normal $\boldsymbol{n}_r^m$ onto the $\boldsymbol{x}$-axis. $\boldsymbol{T}_{s\to g}$ does same for scene point pair. $\boldsymbol{R}(\alpha)$ around the $\boldsymbol{x}$-axis with angle α is required to match $\boldsymbol{s}_i$ and $\boldsymbol{m}_i$ [61].

angle matrix $\mathcal{A}$ for α_m, each point pair $(\boldsymbol{p}_i^s, \boldsymbol{p}_j^s)$ is from scene $\mathcal{S}$. The CPPF of the point pair is computed and searched in the hash table keys $\mathcal{H}(i_r)$. If i_r is not null, then the corresponding key is found, and N_h is the number of duplicated keys. Subsequently, the poses for each correspondence are computed by looking up $\mathcal{H}(r_i)$, culminating in the final pose set $\mathcal{P}$.

Algorithm 2 Voting for Pose

Input: $\mathcal{M}, \mathcal{S}, \mathcal{H}, \mathcal{A}$
for $i \leftarrow 1$**toN**$_\mathbf{s}$ **do**
  for $j \leftarrow 1$**toN**$_\mathbf{s}$ **do**
    if $i \neq j$ **then**
      $\mathcal{E}_i^s \leftarrow \text{HEAD}(\boldsymbol{p}_i^s)$
      $\mathcal{E}_j^s \leftarrow \text{HEAD}(\boldsymbol{p}_j^s)$
      $\boldsymbol{F} \leftarrow \text{CPPF}(\boldsymbol{p}_i^s, \boldsymbol{p}_j^s, \boldsymbol{n}_i^s, \boldsymbol{n}_j^s, \mathcal{E}_i^s, \mathcal{E}_j^s)$
      $\boldsymbol{k} \leftarrow \text{HashKey}(\boldsymbol{F}, \sigma, \psi, \tau)$
      $\leftarrow \text{BitEncode}(\boldsymbol{k})$
      $\alpha_s \leftarrow \text{PlanarRotAngle}(\boldsymbol{n}_i^s, \boldsymbol{p}_i^s, \boldsymbol{p}_j^s)$
      $i_r \leftarrow \text{Search}(\mathcal{H},)$
      if $i_r \neq null$ **then**
        $N_h \leftarrow |\mathcal{H}(i_r)|$
        for $h \leftarrow 1$**toN**$_\mathbf{h}$ **do**
          $\alpha_m \leftarrow \mathcal{A}(i_r)$
          $\alpha \leftarrow \alpha_m - \alpha_s$
          $P \leftarrow$
          $\text{ComputePose}(\boldsymbol{T}_{s\to g}, \boldsymbol{T}_{m\to g}, \alpha)$
          $\mathcal{P} \leftarrow \mathcal{P} \bigcup \{P\}$
        end
      end
    end
  end
end
Output: $\mathcal{P}$

4) Pose Clustering: The retrieved poses are clustered to eliminate incorrect poses by enforcing a predefined threshold on the differences in rotation and translation among all of the poses within a cluster. The score of each cluster is computed as the sum of scores for individual poses, whose scores correspond to the number of votes they receive. Once the cluster with the highest score is identified, the poses within it are averaged to produce the final pose. Pose clustering

generally enhances algorithm stability by filtering out isolated poses with low scores, and averaging improves the accuracy of the final pose [61], [62].

IV. Experimental Results

A. Experimental Setup

The proposed method was evaluated on benchmark datasets, including the synthetic indoor dataset ICL-NUIM [64], the real indoor dataset 3DMatch [48], and the outdoor datasets ETH3D [65] and ETH-TLS [66]. Experiments were carried out on a desktop PC equipped with an Intel Core i7-10700 CPU @ 2.90 GHz with 16-GB RAM.

1) Registration Metrics: The pairwise registration results are evaluated by adopting widely used metrics: rotation error δ_R and translation error δ_t [13], which are defined as

$$\begin{cases} \delta_R & = \arccos\dfrac{1}{2}\left(\mathrm{tr}\left(R_{\mathrm{gt}}^T \cdot R_{\mathrm{est}}\right) - 1\right) \\ \delta_t & = \| t_{\mathrm{gt}} - t_{\mathrm{est}} \| \end{cases} \tag{11}$$

where R and t are the rotation matrix and the translation vector, respectively.

2) Parameters Setup: The key parameters of our proposed method are the quantization steps of CPPF, as described in Section III-C2. These include the quantization steps for distance, angle, and entropy of the contour feature vector, denoted as σ, ψ, and τ, respectively. Only the distance step is affected by the resolution of the point cloud. In our experiments, we let $\psi = 0.20$ and $\tau = 0.20$, with $\sigma = 0.05$ m for indoor scenes and $\sigma = 0.25$ m for outdoor scenes.

B. Performance on Indoor Datasets: ICL-NUIM and 3DMatch

1) Datasets and Baselines: We first evaluated our method on two indoor datasets: ICL-NUIM [64] and 3DMatch [48]. ICL-NUIM has approximately 300 000 points per point cloud and is widely used in indoor scene reconstruction. We chose four scenarios: Livingroom0, Livingroom1, Office0, and Office1. We randomly selected ten groups of pairwise registration experiments from each. 3DMatch has around 500 000 points per point cloud and is popular in learning-based research. We selected four scenes: Kitchen, Home, Hotel, and Studyroom. Again, we randomly selected ten groups of data for each scenario.

In the challenging indoor experiment, we compared the registration performance of the proposed method against the FGR [13], VGICP [55], PLADE [15], Teaser++ [47], GROR [22], PointDSC [27], and RegTR [31] methods. FGR and VGICP are basic iterative optimization methods; Teaser++ and GROR are guaranteed outlier removal methods. PLADE leverages plane-line-based descriptors to establish structure-level correspondences between point clouds. In addition, PointDSC and RegTR are learning-based methods, whose weights were trained on the 3DMatch dataset. PointDSC and RegTR were executed on a server equipped with an Intel Xeon 4210R CPU @ 2.40 GHz with 32-GB RAM and a GeForce RTX3090 GPU with 24-GB RAM.

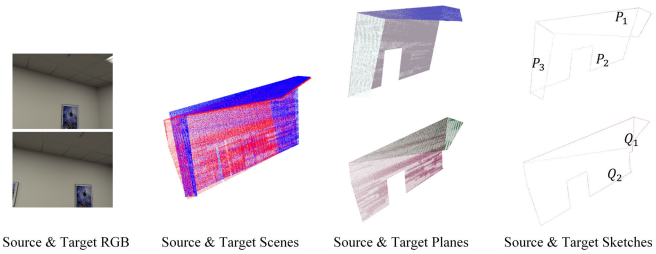


Fig. 7. Challenging example of point cloud registration. The matched planar pairs between target cloud $\mathcal{P}$ and source cloud $\mathcal{Q}$ are $\{P_1, Q_1\}$ and $\{P_2, Q_2\}$.

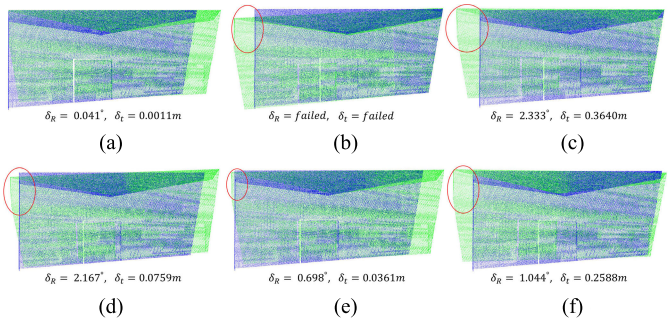


Fig. 8. Comparison of the proposed method with several baselines on example in Fig. 7. Errors in registration results are highlighted with red circles. (a) Proposed. (b) PLADE. (c) Teaser++. (d) FGR. (e) RegTR. (f) DGR.

TABLE I

Rotation Error δ_R (deg) and Translation Error δ_t (m) for the Challenging Example of Point Cloud Registration in Fig. 7

Method	δ_R	δ_t
FGR [13]	2.1666	0.0759
PLADE [15]	failed	failed
VGIP [55]	3.9909	0.7341
Teaser++ [47]	2.3328	0.3640
GROR [22]	0.4948	0.0055
DGR [25]	1.0441	0.2588
PointDSC [27]	0.1713	0.0093
RegTR [31]	0.6978	0.0361
Proposed	**0.0407**	**0.0011**

2) Qualitative Evaluation on Pairwise Registration: Fig. 7 shows a challenging experiment of pairwise registration, emphasizing the crucial role of sketch in the proposed method for 3-D registration. Fig. 8 and Table I present the experimental results obtained from the proposed algorithm and several comparative methods on this challenging example. In this experiment, the target point cloud is composed of three planes, while the source contains only two. Accordingly, a maximum of two pairs of nonparallel planar correspondences can be matched between the two point clouds. From these correspondences, only 3-DoF for rotation can be deduced theoretically, leaving 3-DoF for translation indeterminate. As shown in Fig. 8, it is clear that the primitive-based PLADE [15] fails to achieve accurate alignment directly between the two point clouds. Notably, Teaser++ [47] and DGR [25] exhibit the highest translation error. It is evident that RegTR [31] and FGR [13] demonstrate translation errors at the centimeter level. However, compared with baselines in Table I, our proposed method achieves a remarkably accurate 6-DoF pose, indicating that sketch contours are highly beneficial to

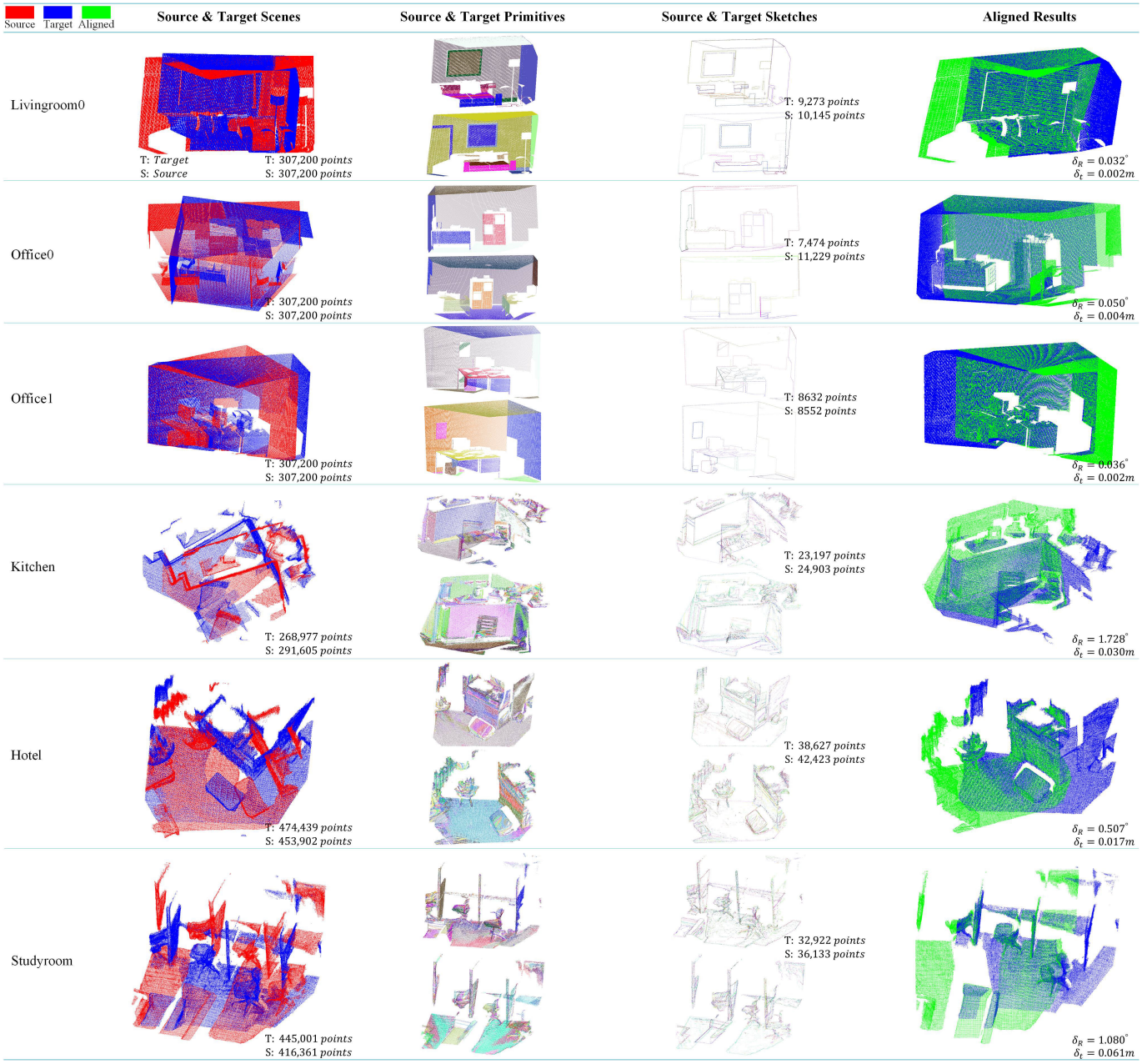


Fig. 9. Results of main steps of the proposed method. First column: initial point clouds of source and target scenes. Second to fourth columns: planar extraction results, sketches, and alignment results, respectively.

accurately determining 3-DoF for translation. The results of our proposed method demonstrate that point pair geometric information and the proposed local contour geometric feature HEAD in 3-D registration can effectively suppress translation errors.

Fig. 9 shows several experimental outcomes of each module of the proposed method, encompassing plane extraction, point cloud sketching, and voting-based point cloud registration. The first three rows show the results obtained from three groups of data from the ICL-NUIM dataset, while the last three correspond to 3DMatch. The remarkably low rotation and translation errors observed in the registration results provide compelling evidence for the effectiveness of our proposed sketch-based method for point cloud registration.

3) Quantitative Comparison With Baselines: Fig. 10 presents some registration results obtained by the proposed method and baselines on the ICL-NUIM and 3DMatch datasets. The first eight rows show the results obtained from ICL-NUIM, and the last eight rows correspond to 3DMatch. To quantitatively evaluate the registration performance, the rotation error δ_R (deg) and translation error δ_t (m) are counted. Fig. 11 compares rotation and translation errors for the proposed method and baselines on ICL-NUIM and 3DMatch, considering each group of data. Tables II and III present the error statistics of each scene in ICL-NUIM and 3DMatch, respectively. Table II demonstrates that the proposed method outperforms baselines, with the lowest rotation and translation errors on all of the scenes in ICL-NUIM, and the

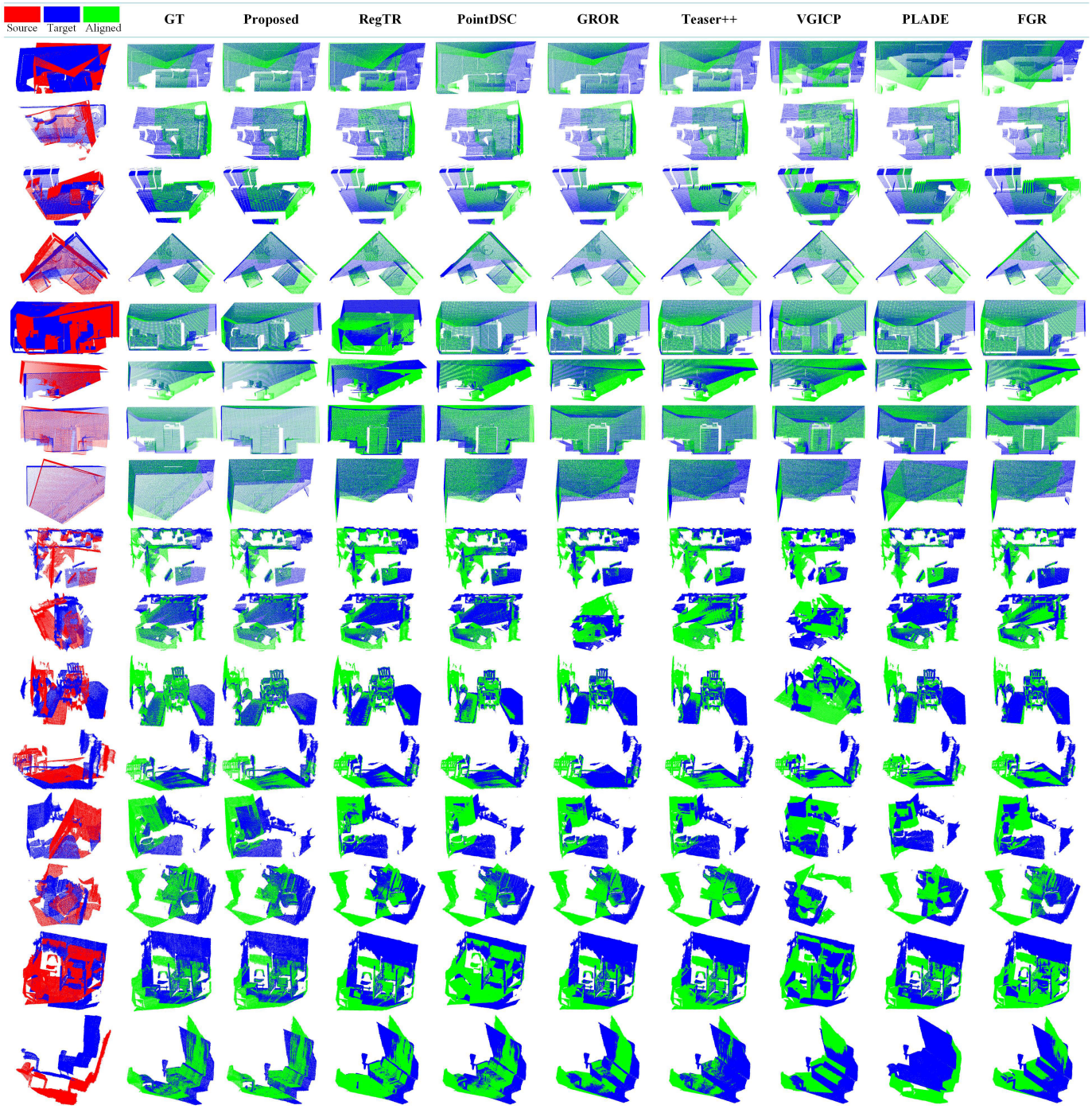


Fig. 10. Visualization of some alignment results obtained using the proposed method and baselines on ICL-NUIM and 3DMatch datasets. The experimental results of 16 groups of data represent eight scenes from the two datasets. For each scene, two groups of data are selected for visualization.

performance of GROR is second only to ours. In Table III, RegTR, a learning-based method, performs best on three scenes of 3DMatch, and the proposed method is the second best. Although PointDSC performs well on Redkitchen, it does not perform as well as the proposed method in the other three scenarios. Furthermore, PointDSC fails to produce satisfactory results for certain groups of data in the Studyroom scene.

Tables IV and V present the statistical results of rotation and translation errors for ICL-NUIM and 3DMatch, respectively, which corresponds to the box plots shown in Fig. 12. Examining Table IV and Fig. 12(a), (b), and (e), it is evident that the proposed method achieves the best results on ICL-NUIM, with remarkably low rotation and translation errors that significantly outperform the baseline methods. GROR delivers the second-best performance on this dataset. According to Table V, the proposed method performs the second best on 3DMatch, surpassed only by the learning-based method RegTR. However, it is worth noting that the contrast in error between the proposed method and RegTR is relatively small.

TABLE II

ROTATION ERROR δ_R (deg) AND TRANSLATION ERROR δ_t (m) FOR DIFFERENT REGISTRATION METHODS ON FOUR SCENES FROM THE ICL-NUIM DATASET. EACH COLUMN DISPLAYS THE TOP-RANKING ENTRY IN BOLD, AND THE SECOND-RANKING ENTRY IS UNDERLINED

Method	Livingroom0		Livingroom1		Office0		Office1	
	δ_R	δ_t	δ_R	δ_t	δ_R	δ_t	δ_R	δ_t
FGR [13]	1.4945	0.0477	2.9123	0.0691	0.8445	0.0249	1.8923	0.0563
PLADE [15]	2.3161	0.0228	9.2877	0.2211	6.7174	0.0536	8.2320	0.1419
VGICP [55]	3.5419	0.2359	16.133	0.3740	0.3083	0.1652	5.6291	0.5238
Teaser++ [47]	0.8671	0.0329	1.2296	0.0421	0.6129	0.0238	1.1671	0.0944
GROR [22]	0.1694	0.0060	0.2088	0.0058	0.1298	0.0069	0.2254	0.0060
PointDSC [27]	0.2620	0.0072	0.6407	0.0165	0.1500	0.0085	0.5386	0.0192
RegTR [31]	1.1506	0.0622	0.5730	0.0207	13.118	0.8005	1.4244	0.0514
Proposed	**0.0471**	**0.0017**	**0.0637**	**0.0028**	**0.0323**	**0.0019**	**0.0439**	**0.0018**

TABLE III

ROTATION ERROR δ_R (deg) AND TRANSLATION ERROR δ_t (m) FOR DIFFERENT REGISTRATION METHODS ON FOUR SCENES FROM THE 3DMATCH DATASET. EACH COLUMN DISPLAYS THE TOP-RANKING ENTRY IN BOLD, AND THE SECOND-RANKING ENTRY IS UNDERLINED

Method	Redkitchen		Home		Hotel		Studyroom	
	δ_R	δ_t	δ_R	δ_t	δ_R	δ_t	δ_R	δ_t
FGR [13]	2.2147	0.0469	2.9920	0.0741	7.3666	0.1687	8.0715	0.2360
PLADE [15]	11.155	0.4965	41.734	0.8742	41.174	1.0479	6.7353	0.1931
VGICP [55]	24.739	0.6467	42.018	1.2029	63.431	1.2822	22.831	0.7873
Teaser++ [47]	1.4207	0.0425	2.0516	0.0697	2.0669	0.0544	2.8408	0.1034
GROR [22]	7.1134	0.0653	4.5022	0.1130	2.1315	0.0537	2.2161	0.0983
PointDSC [27]	**0.9631**	**0.0221**	1.1370	0.0331	1.5348	0.0348	9.2753	0.1180
RegTR [31]	1.1144	0.0253	**0.8143**	**0.0257**	**1.3720**	**0.0331**	**0.7884**	**0.0326**
Proposed	1.0866	0.0283	0.8447	0.0293	1.4334	0.0363	1.0267	0.0367

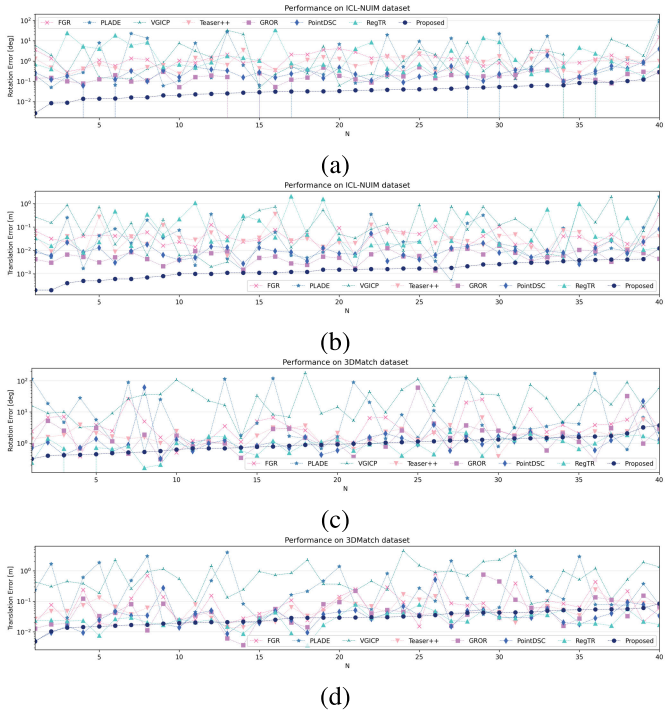


Fig. 11. Rotation and translation errors for the proposed method and baselines on two indoor datasets. (a) and (b) Rotation and translation errors on ICL-NUIM, respectively. (c) and (d) Rotation and translation errors on 3DMatch, respectively.

TABLE IV

ROTATION ERROR δ_R (deg) AND TRANSLATION ERROR δ_t (m) FOR REGISTRATION METHODS ON THE ICL-NUIM DATASET

Method	δ_R		δ_t	
	Mean	Median	Mean	Median
FGR [13]	1.7859	1.2718	0.0495	0.0392
PLADE [15]	6.6383	0.3891	0.1098	0.0171
VGICP [55]	6.4030	1.1147	0.3252	0.1016
Teaser++ [47]	0.9692	0.8231	0.0483	0.0300
GROR [22]	0.1833	0.1629	0.0062	0.0055
PointDSC [27]	0.3978	0.2325	0.0129	0.0084
RegTR [31]	4.0666	0.8294	0.2347	0.0328
Proposed	**0.0468**	**0.0347**	**0.0020**	**0.0015**

TABLE V

ROTATION ERROR δ_R (deg) AND TRANSLATION ERROR δ_t (m) FOR REGISTRATION METHODS ON THE 3DMATCH DATASET

Method	δ_R		δ_t	
	Mean	Median	Mean	Median
FGR [13]	5.1612	2.8635	0.1314	0.0685
PLADE [15]	38.255	20.179	0.9798	0.6165
VGICP [55]	6.4030	1.1147	0.3252	0.1016
Teaser++ [47]	2.0950	1.8082	0.0675	0.0572
GROR [22]	3.9908	1.4997	0.0826	0.0419
PointDSC [27]	3.2275	1.1173	0.0520	0.0337
RegTR [31]	**1.0223**	0.9784	**0.0292**	**0.0246**
Proposed	1.0979	**0.9424**	0.0326	0.0295

The difference in average rotation error between the two methods is no more than 0.08°, while the difference in average translation error is less than 0.006 m. It should be emphasized that the RegTR method, trained with network weights specifically on the 3DMatch dataset, achieves the lowest rotation and translation errors, thus attaining the highest accuracy on this dataset. However, when evaluated on the untrained ICL-NUIM dataset, RegTR exhibits significantly higher rotation and translation errors compared to our proposed method (see Table IV). Despite performing worse than RegTR on the training 3DMatch dataset, PointDSC, another learning-based method, shows superior performance on the untrained

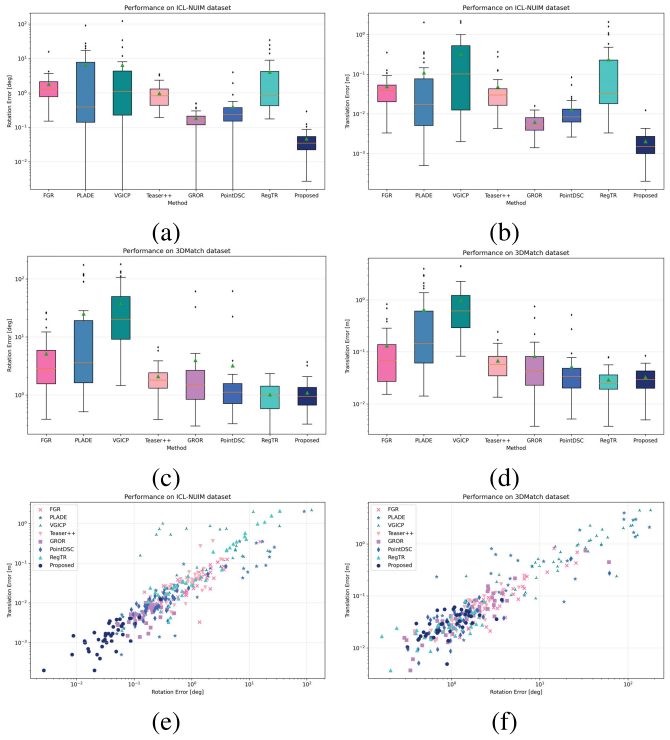


Fig. 12. Performance evaluation of algorithms on two indoor datasets. (a) and (b) Rotation and translation performance, respectively, on ICL-NUIM. (c) and (d) Rotation and translation performance, respectively, on 3DMatch. (e) and (f) Joint distributions of rotation and translation errors for ICL-NUIM and 3DMatch datasets, respectively. Algorithms with distributions closer to the lower left corner show better performance; those farther away indicate worse performance.

ICL-NUIM dataset and is surpassed only by our proposed method and GROR, with a notably lower error compared with RegTR. These findings suggest that PointDSC has better generalization performance than RegTR. GROR, an outlier-removal algorithm based on computational geometry, shows remarkable alignment capability for most scenes from both ICL-NUIM and 3DMatch. On the ICL-NUIM, it achieves the second-lowest rotation and translation errors, only after the proposed method. On 3DMatch, the performance of GROR is comparable to that of the proposed method, RegTR, and PointDSC, with only a few instances of alignment failures, as demonstrated in Fig. 12(c), (d), and (f).

In summary, our proposed method showcases its effectiveness in leveraging contour cues in the scene to significantly reduce rotation and translation errors, as illustrated in Fig. 7, and it performed well against baselines in experiments conducted on two indoor datasets. The outlier removal algorithm GROR ranks second only to our method overall. Notably, both the learning-based algorithms, PointDSC and RegTR, demonstrate superior performance on the trained 3DMatch dataset but struggle on the untrained ICL-NUIM dataset.

C. Performance on Outdoor Datasets: ETH3D and ETH-TLS

1) Dataset and Baselines: We also evaluated the proposed method on two large-scale outdoor point cloud registration datasets: ETH3D [65] and ETH-TLS [66]. We chose nine groups of data, six groups of data from ETH3D and three from ETH-TLS, each encompassing point clouds containing approximately 20–30 million points.

In this demanding task of large-scale outdoor point cloud registration, FGR [13], VGICP [55], PLADE [15], Teaser++ [47], and GROR [22] were again used as baselines. However, the learning-based PointDSC and RegTR fail to register on these datasets. PointDSC requires an allocation of over 3 TB of memory when calculating the distance between a set of feature point pairs, which far exceeds the capabilities of existing devices, and RegTR demands computing resources that exceed the 24-GB memory capacity of a single RTX3090 during registration.

2) Performance and Comparison With Baselines: Fig. 13 shows the experimental outcomes of each module of the proposed method on two ETH datasets, encompassing plane extraction, point cloud sketching, and voting-based point cloud registration. It has been observed that abstracting point clouds into sketches using planar primitives significantly reduces the number of points in the point cloud while remarkably

TABLE VI

ROTATION ERROR δ_R (deg) FOR REGISTRATION METHODS ON TWO ETH DATASETS. EACH ROW DISPLAYS THE TOP-RANKING ENTRY IN BOLD, AND THE SECOND-RANKING ENTRY IS UNDERLINED

ID	Proposed	GROR [22]	Teaser++ [47]	VGICP [55]	PLADE [15]	FGR [13]
1	**0**	**0**	5.3600	1.0930	0.1013	1.0434
2	0.0446	**0.0430**	0.3374	1.2680	170.62	1.2689
3	0.0771	**0.0610**	176.87	2.6030	0.1720	176.21
4	0.3090	**0.2010**	2.5698	95.900	0.2260	20.349
5	0.1500	1.2659	1.3126	4.6540	**0**	0.1855
6	**0.0530**	0.0926	1.2938	0.3220	49.533	2.3641
7	**0.0530**	0.1543	19.291	25.071	116.99	4.3685
8	0.0370	**0**	0.5718	2.7731	8.0657	0.2317
9	0.2457	179.48	0.5294	11.314	**0.1222**	177.15

TABLE VII

TRANSLATION ERROR δ_t (m) FOR REGISTRATION METHODS ON TWO ETH DATASETS. EACH ROW DISPLAYS THE TOP-RANKING ENTRY IN BOLD, AND THE SECOND-RANKING ENTRY IS UNDERLINED

ID	Proposed	GROR [22]	Teaser++ [47]	VGICP [55]	PLADE [15]	FGR [13]
1	**0.0055**	0.0172	1.8478	6.4223	0.0247	0.3036
2	0.0075	**0.0045**	0.2706	13.105	12.730	2.0614
3	0.0296	**0.0200**	26.174	26.063	0.0281	26.259
4	**0.0023**	0.0171	0.1083	4.0347	0.4693	2.4877
5	**0.0024**	0.0582	0.0150	0.7226	0.0082	0.0065
6	**0.0062**	0.0159	0.1027	2.5892	9.9881	1.0937
7	**0.0124**	0.0216	0.6615	14.109	21.529	0.4692
8	**0.0066**	0.0251	0.3786	17.887	12.381	0.4887
9	0.0161	11.942	0.0330	11.745	**0.0155**	12.004

TABLE VIII

ROTATION ERROR δ_R (deg), TRANSLATION ERROR δ_t (m), AND AVERAGE TIME (s) FOR REGISTRATION METHODS ON TWO ETH DATASETS. EACH COLUMN DISPLAYS THE TOP-RANKING ENTRY IN BOLD, AND THE SECOND-RANKING ENTRY IS UNDERLINED

Method	δ_R		δ_t		Avg Time
	Mean	Median	Mean	Median	
FGR [13]	42.57	2.364	5.019	1.094	351
PLADE [15]	38.42	0.226	6.353	0.469	276
VGICP [55]	16.11	2.773	10.74	11.74	43.8
Teaser++ [47]	23.13	1.313	3.288	0.271	**1.12**
GROR [22]	20.14	0.093	1.347	0.020	17.8
Proposed	**0.107**	**0.053**	**0.012**	**0.007**	64.7

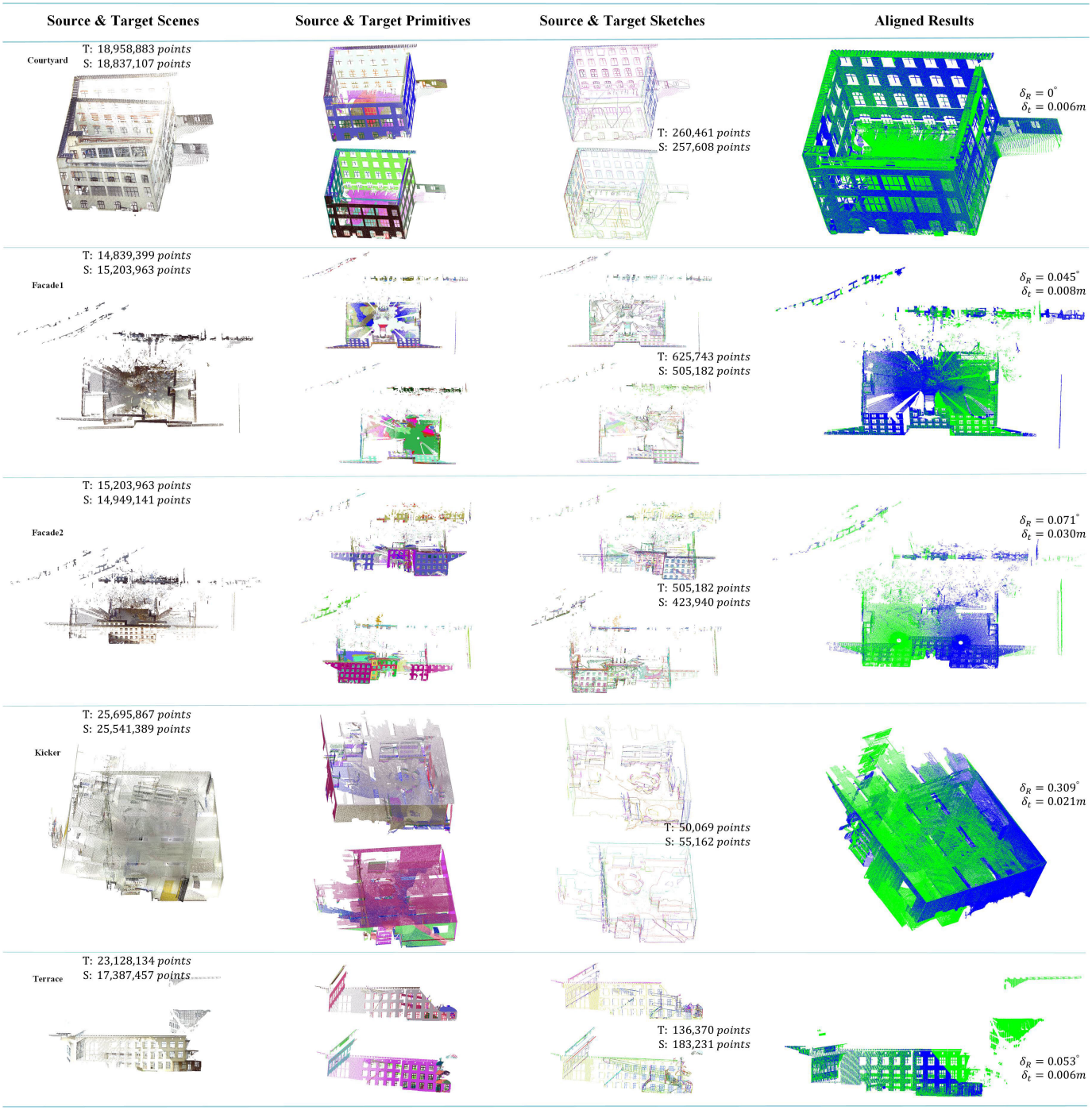


Fig. 13. Results of main steps of the proposed method on two ETH datasets.

preserving contour cues. This effectively validates the effectiveness of the proposed sketch module. Furthermore, we visualize the results obtained from nine groups of data on two ETH datasets using the proposed method and baselines in Fig. 14. To quantitatively assess the registration results obtained from all of the algorithms on two large-scale point cloud datasets, we count the rotation error δ_R and the translation error δ_t for each group of data, as depicted in Tables VI and VII, respectively. Fig. 15 shows boxplots illustrating the statistical results of the rotation and translation errors for each algorithm on the two ETH datasets, complementing the information provided in Table VIII. From the quantitative evaluation results, we observe the following.

1) The proposed method demonstrates superior performance, with the smallest average rotation and translation errors, as evidenced by the results presented in Tables VI–VIII and Fig. 15.
2) The GROR algorithm is second only to the proposed method. As shown in Tables VI and VII, GROR performs well on the other eight groups of data on ETH, with the last group of data registration failures.

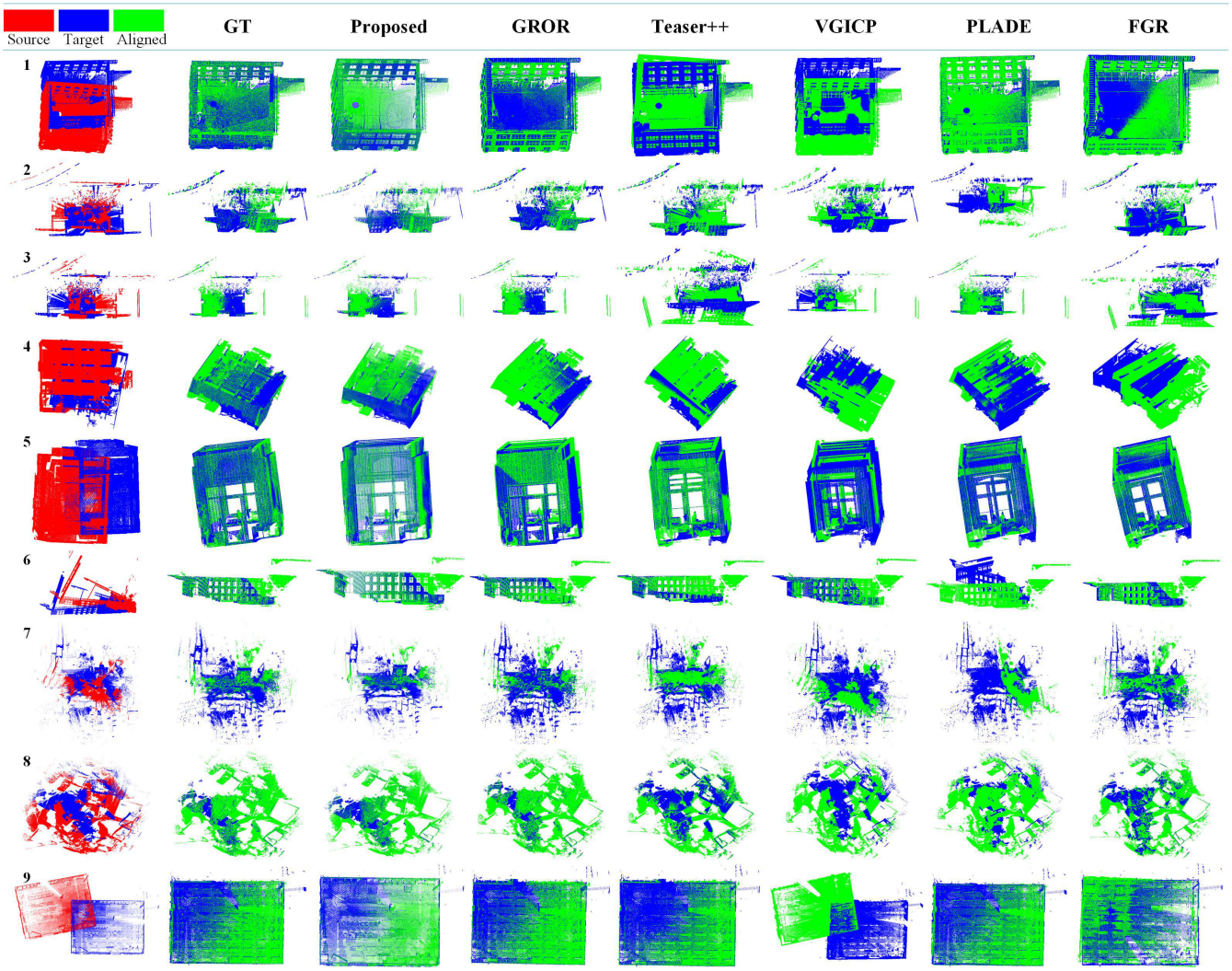
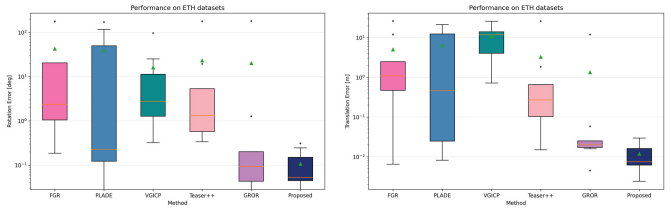


Fig. 14. Visualization of alignment results for the proposed method and baselines on two ETH datasets.



(a) (b)

Fig. 15. Quantitative evaluation results for the proposed method and baselines on two ETH datasets: (a) rotation performance and (b) translation performance.

3) Fig. 15 shows that the PLADE method performs better on rotation than on translation since correct planar primitive correspondences help suppress rotation errors [43]. The proposed algorithm shares similarities with PLADE in utilizing the normal information of the planes. However, our algorithm goes a step further by incorporating three channels in the point pair feature of the CPPF module to comprehensively integrate the plane's normal information, which is the potential reason why the proposed method suppresses rotation errors (as shown in Tables VI and VIII). Different from all of the baselines, the proposed method incorporates local contour geometric features to effectively suppress translation errors (as verified in Figs. 7 and 8) and achieves remarkably low translation errors (as shown in Tables VII and VIII).

In general, the proposed method outperforms the baselines in obtaining high-precision 6-DoF poses on the two large-scale outdoor datasets. The advantage of the proposed method lies in its independence from additional training processes and expensive hardware resources, such as GPU memory, ensuring compatibility with affordable devices.

D. Ablation Study on Components

We evaluated the effectiveness of the sketch, HEAD, and CPPF modules in the pairwise alignment pipeline through five sets of comparative experiments conducted on Kicker in the ETH dataset, Livingroom1 in ICL-NUIM, and Redkitchen in the 3DMatch. We combined the proposed sketch with the proposed CPPF, classic ICP [7], and outlier-removal-based GROR [22], and measured the rotation and translation errors

TABLE IX
ABLATION STUDY FOR THE PROPOSED PAIRWISE ALIGNMENT PIPELINE: SKETCH, HEAD, AND CPPF MODULES. IN EACH COLUMN, THE TOP-RANKING ENTRY IS BOLDED

Method	Kicker			Livingroom1			Redkitchen		
	δ_R (deg)	δ_t (m)	time (s)	δ_R (deg)	δ_t (m)	time (s)	δ_R (deg)	δ_t (m)	time (s)
ICP	96.99	2.854	1091	14.02	0.323	12.07	11.11	0.352	16.19
Sketch+ICP	96.17↓	3.149↑	2.272↓	12.07↓	0.359↑	0.241↓	10.17↓	0.343↓	0.454↓
GROR	**0.201**	0.017	3.044	0.209	0.006	3.538	7.113	0.065	3.647
Sketch+GROR	116.1↑	6.648↑	0.145↓	3.947↑	0.020↑	0.110↓	8.103↑	0.224↑	0.420↓
Sketch+CPPF(Ours)	0.309	**0.002**	9.690	**0.064**	**0.003**	2.190	**1.087**	**0.028**	6.241

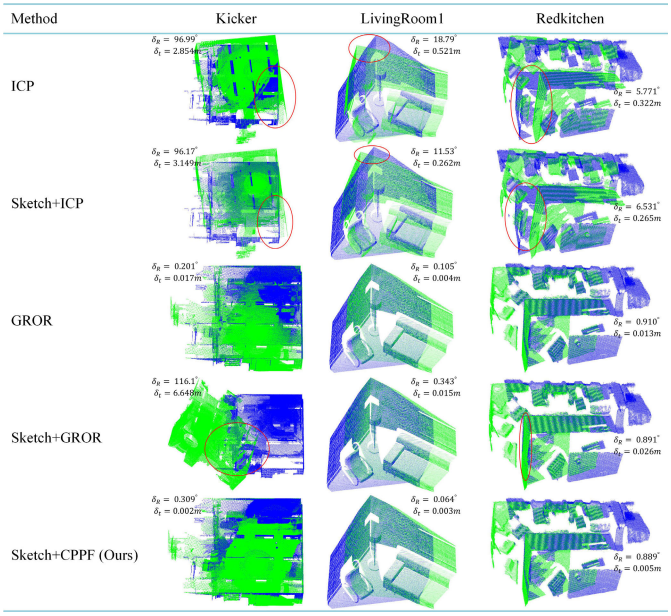


Fig. 16. Visualization of alignment results for ablation study on Kicker, Livingroom1, and Redkitchen. δ_R and δ_t denote rotation and translation errors, respectively. Some errors in alignment results are highlighted by red circles.

of each scheme on three scenes. According to the results in Table IX, the combination of the sketch with ICP shows a slight decrease in overall rotation and translation errors, while the combination of the sketch with GROR shows a significant increase in these errors. However, when CPPF is combined with sketch, our proposed method achieves significantly lower rotation and translation errors compared with the other schemes. Moreover, the combination of the sketch module with the ICP and GROR algorithms results in a significant reduction in the running time of the algorithm, which is attributed to the sketch module's ability to reduce the point cloud size while preserving important contour clues in the scene. Fig. 16 visualizes the alignment results for the ablation study, clearly demonstrating that our proposed pipeline significantly outperforms the other four comparison schemes. The ablation study highlights the significant role of each module in our pipeline and validates the effectiveness of our proposed method.

E. Failure Case

The abovementioned experiments confirm the performance of our proposed method on artificial scenes, which typically contain a large number of planar features. However, when applied to nonartificial scenes, it encounters challenges. For example, Fig. 17 presents a group of data involving two scenes abundant with trees, which makes it difficult to accurately extract planar features. As a result, the method encounters obstacles in achieving precise alignment between the two point clouds.

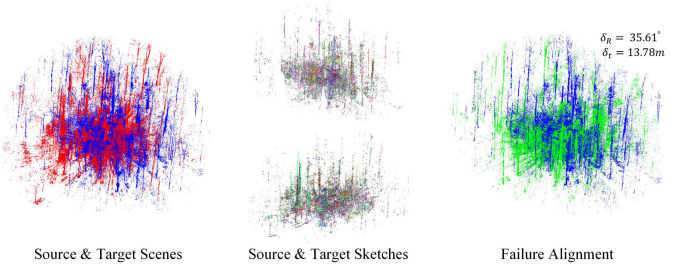


Fig. 17. Failure example of the proposed method. The rotation error δ_R is 35.61°; the translation error δ_t is 13.78 m.

V. CONCLUSION

We demonstrated a framework for sketch-based point cloud registration utilizing contour cues. The proposed method abstracts the scene into a sketch by planar features and incorporates local contour geometric feature descriptors into a voting-based pipeline, combining planar features, local contour geometric features, and point pair geometric features to estimate the 6-DoF transformation between the target point cloud $\mathcal{P}$ and the source point cloud $\mathcal{Q}$. Experimental results on multiple datasets validate the effectiveness of our approach in suppressing both rotation and translation errors. The proposed method demonstrates promising performance on artificial scenes; future research efforts will be directed toward enhancing its performance on nonartificial scenes. This article underscores the significance of contour cues in point cloud registration tasks, paving the way for the exploration of reinforcing the role of contour features in such tasks.

REFERENCES

[1] Y. Shu, Z. Liao, B. Xiao, W. Li, and X. Gao, "Registration-is-evaluation: Robust point set matching with multigranular prior assessment," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4701414.
[2] A. Rosinol et al., "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *Int. J. Robot. Res.*, vol. 40, nos. 12–14, pp. 1510–1546, Dec. 2021.
[3] J. A. Placed et al., "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1686–1705, Jun. 2023.

[4] M. Liu, K. Zhang, J. Zhu, J. Wang, J. Guo, and Y. Guo, "Data-driven indoor scene modeling from a single color image with iterative object segmentation and model retrieval," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 4, pp. 1702–1715, Apr. 2020.

[5] F. Reyes-Aviles, P. Fleck, D. Schmalstieg, and C. Arth, "Compact world anchors: Registration using parametric primitives as scene description," *IEEE Trans. Vis. Comput. Graphics*, early access, Jun. 15, 2022, doi: 10.1109/TVCG.2022.3183264.

[6] M. Zhao, X. Huang, J. Jiang, L. Mou, D.-M. Yan, and L. Ma, "Accurate registration of cross-modality geometry via consistent clustering," *IEEE Trans. Vis. Comput. Graphics*, early access, Feb. 22, 2023, doi: 10.1109/TVCG.2023.3247169.

[7] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.

[8] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot. Sci., Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.

[9] 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.

[10] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3450–3466, Jul. 2022.

[11] 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. 1113–11151, Sep. 2023.

[12] Z. Zhang, J. Sun, Y. Dai, B. Fan, and M. He, "VRNet: Learning the rectified virtual corresponding points for 3D point cloud registration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 4997–5010, Aug. 2022.

[13] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision—ECCV*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 766–782.

[14] J. Yang, Y. Xiao, and Z. Cao, "Aligning 2.5D scene fragments with distinctive local geometric features and voting-based correspondences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 714–729, Mar. 2019.

[15] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "PLADE: A plane-based descriptor for point cloud registration with small overlap," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2530–2540, Apr. 2020.

[16] L. Zhang, J. Guo, Z. Cheng, J. Xiao, and X. Zhang, "Efficient pairwise 3-D registration of urban scenes via hybrid structural descriptors," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5700717.

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

[18] J. Yang, Q. Zhang, K. Xian, Y. Xiao, and Z. Cao, "Rotational contour signatures for both real-valued and binary feature representations of 3D local shape," *Comput. Vis. Image Understand.*, vol. 160, pp. 133–147, Jul. 2017.

[19] E. Xu, Z. Xu, and K. Yang, "Using 2-lines congruent sets for coarse registration of terrestrial point clouds in urban scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5701618.

[20] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for rotation search," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2165–2173.

[21] Á. Parra Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.

[22] L. Yan, P. Wei, H. Xie, J. Dai, H. Wu, and M. Huang, "A new outlier removal strategy based on reliability of correspondence graph for fast point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 7986–8002, Jul. 2023.

[23] J. Li, P. Zhao, Q. Hu, and M. Ai, "Robust point cloud registration based on topological graph and Cauchy weighted Lq-norm," *ISPRS J. Photogramm. Remote Sens.*, vol. 160, pp. 244–259, Feb. 2020.

[24] J. Li, "A practical O(N^2)O(N2) outlier removal method for correspondence-based point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 3926–3939, Aug. 2022.

[25] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2511–2520.

[26] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3DRegNet: A deep neural network for 3D point registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7191–7201.

[27] X. Bai et al., "PointDSC: Robust point cloud registration using deep spatial consistency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15854–15864.

[28] J. Wang et al., "PG-net: Progressive guidance network via robust contextual embedding for efficient point cloud registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5701712.

[29] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3522–3531.

[30] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7156–7165.

[31] 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. (CVPR)*, Jun. 2022, pp. 6667–6676.

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

[33] I. Sipiran and B. Bustos, "Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes," *Vis. Comput.*, vol. 27, no. 11, pp. 963–976, Nov. 2011.

[34] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, Jan. 2016.

[35] A. E. Johnson, "Spin-images: A representation for 3-D surface matching," Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-97-47, 1997.

[36] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, Oct. 2013.

[37] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3D point cloud registration," *Inf. Sci.*, vols. 346–347, pp. 163–179, Jun. 2016.

[38] F. Tombari and L. Di Stefano, "Object recognition in 3D scenes with occlusions and clutter by Hough voting," in *Proc. 4th Pacific-Rim Symp. Image Video Technol.*, Nov. 2010, pp. 349–355.

[39] L. Liu, J. Xiao, Y. Wang, Z. Lu, and Y. Wang, "A novel rock-mass point cloud registration method based on feature line extraction and feature point matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5701117.

[40] D. Yu, J. Xiao, and Y. Wang, "Registration method for point clouds of complex rock mass based on dual structure information," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5703518.

[41] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[42] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-SLAM: Low-drift monocular SLAM in indoor environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6583–6590, Oct. 2020.

[43] R. Yunus, Y. Li, and F. Tombari, "ManhattanSLAM: Robust planar tracking and mapping leveraging mixture of Manhattan frames," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6687–6693.

[44] D. Chen et al., "VIP-SLAM: An efficient tightly-coupled RGB-D visual inertial planar SLAM," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 5615–5621.

[45] R. I. Hartley and F. Kahl, "Global optimization through rotation space search," *Int. J. Comput. Vis.*, vol. 82, no. 1, pp. 64–79, Apr. 2009.

[46] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*. Berlin, Germany: Springer, 2013.

[47] 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.

[48] A. Zeng et al., "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 199–208.

[49] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8957–8965.

[50] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3Feat: Joint learning of dense detection and description of 3D local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6358–6366.

[51] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "PREDATOR: Registration of 3D point clouds with low overlap," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4265–4274.

[52] H. Wang et al., "RoReg: Pairwise point cloud registration with oriented descriptors and local rotations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10376–10393, Aug. 2023.

[53] F. Lu et al., "Sparse-to-dense matching network for large-scale LiDAR point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11270–11282, Sep. 2023.

[54] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Comput. Graph. Forum*, vol. 32, no. 5, pp. 113–123, Aug. 2013.

[55] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11054–11059.

[56] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[57] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. J. Conf. Artif. Intell.*, vol. 2, Aug. 1981, pp. 674–679.

[58] A. Qi et al., "Toward fine-grained sketch-based 3D shape retrieval," *IEEE Trans. Image Process.*, vol. 30, pp. 8595–8606, 2021.

[59] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[60] F. Wang, S. Lin, X. Luo, H. Wu, R. Wang, and F. Zhou, "A data-driven approach for sketch-based 3D shape retrieval via similar drawing-style recommendation," *Comput. Graph. Forum*, vol. 36, no. 7, pp. 157–166, 2017.

[61] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.

[62] C. Choi and H. I. Christensen, "RGB-D object pose estimation in unstructured environments," *Robot. Auto. Syst.*, vol. 75, pp. 595–613, Jan. 2016.

[63] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3D sensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1724–1731.

[64] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.

[65] T. Schöps et al., "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2538–2547.

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

Gang Ma received the B.S. degree in computer science and technology from Suzhou University, Suzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Fudan University, Shanghai, China.

His research interests include 3-D computer vision, robotics, and artificial intelligence.

Hui Wei received the Ph.D. degree from the Department of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing, China, in 1998.

From 1998 to 2000, he was a Post-Doctoral Fellow with the Department of Computer Science and the Institute of Artificial Intelligence, Zhejiang University, Hangzhou, China. Since November 2000, he has been with the Department of Computer Science and Engineering, Fudan University, Shanghai, China. His research interests include artificial intelligence and cognitive science.