

PARTE: Plane-Assisted Robust Transformation Estimation for Point Cloud Registration

Anonymous Authors*

Abstract—Global point-cloud registration remains challenging when limited overlap, repetitive geometry, and sensor noise produce correspondence sets dominated by outliers. Planar regions are particularly difficult for conventional point descriptors and are therefore often suppressed or discarded before matching. We present PARTE (Plane-Assisted Robust Transformation Estimation), a global registration method that instead treats planar structure as complementary registration evidence. PARTE extracts planar patches and represents them using our novel Plane Context Histogram (PCH), a descriptor that encodes the geometry surrounding each patch, while a two-level matching procedure identifies reliable plane correspondences. Candidate point and plane correspondences are combined in a confidence-weighted compatibility graph for joint outlier rejection, followed by rigid transformation estimation. When no usable plane correspondences are available, PARTE naturally reduces to point-only registration. We evaluate PARTE on 8,097 registration pairs across six indoor and outdoor benchmarks spanning dense RGB-D and sparse LiDAR measurements. Evaluations show PARTE achieves the highest overall success rate against 13 standard and state-of-the-art methods while maintaining low runtime. An (anonymous) open-source C++ implementation with Python bindings is provided at <https://parte.pages.dev>.

I. INTRODUCTION

A critical bottleneck in developing advanced autonomous robots is the ability to create an accurate 3D map of their environment in the presence of sensor noise, dynamic objects, and limited computational resources. Robotic mapping systems typically represent each sensor observation as a point cloud, a set of 3D measurements that captures the geometry of the surrounding environment. As the robot captures successive point clouds while traveling through its environment, the core challenge is determining how the robot moved between scans in order to stitch them into a coherent map. This alignment process is known as *point cloud registration* and underpins many important applications, including SLAM, object recognition, augmented reality, and medical imaging.

Achieving fast, accurate, and robust point cloud registration remains challenging. As illustrated in Fig. 1, achieving both high registration accuracy and low runtime remains difficult for existing methods. Although globally optimal registration algorithms exist [1], their exponential runtime scaling limits their use in real-time applications. Practical registration methods therefore generally follow one of two strategies: local optimization or feature-based global registration. Local methods iteratively minimize an alignment error between two point clouds. They are fast and accurate when provided with

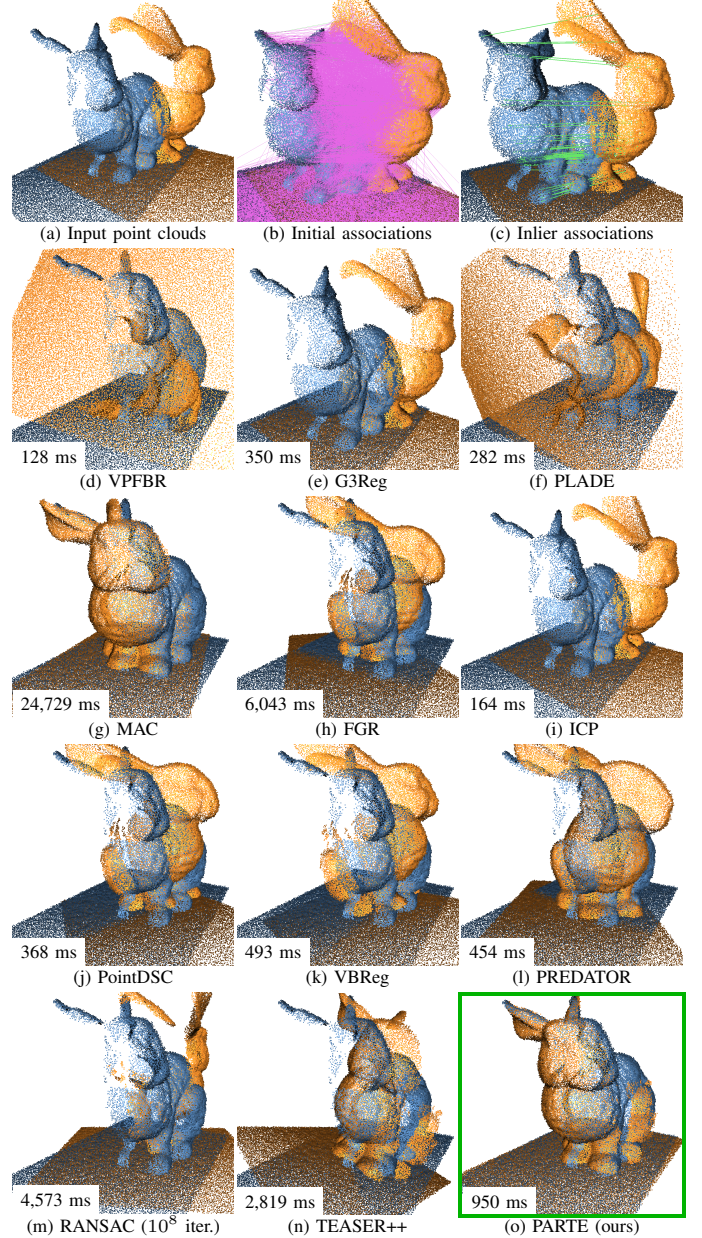


Fig. 1. Registration of noisy Stanford bunny point clouds augmented with ground-plane points using several state-of-the-art algorithms. (a) Input point clouds colored blue and orange, with Gaussian noise of $\mathcal{N}(0, 10^{-3})$ added to each point coordinate. (b) About 17,000 initial associations obtained by nearest-neighbor matching of FPFH point descriptors. (c) Ground-truth inlier associations, which constitute less than 0.2% of the initial associations. (d–n) Registered point clouds produced by various classical and state-of-the-art algorithms. (o) Registered point clouds produced by PARTE.

a good initial estimate, but may converge to an incorrect local minimum when the initial misalignment is large. Feature-based

*The authors used a Large Language Model (OpenAI, GPT-5.6) for minor language editing. All technical content and results were developed and verified by the authors, who take full responsibility.

global methods instead search for transformation-invariant features in the two point clouds, establish tentative correspondences, and estimate the transformation without requiring an initial alignment. This makes them more versatile and particularly useful for tasks such as loop closure, re-localization, and initialization-free scan matching. However, their reliability is often limited by incorrect correspondences, commonly referred to as outliers. These outliers arise from multiple sources, including dynamic objects, sensor noise, limited overlap, and feature-poor regions where distinctive geometric patterns are scarce. Planar regions are particularly challenging for point-based matching because points sampled from the same or similar planar surfaces often have similar local descriptors, making their correspondences highly ambiguous. Since planar surfaces are common in robotic environments, these ambiguities can substantially increase the outlier ratio and reduce registration reliability.

Existing global registration methods often treat planar structure as a source of ambiguous correspondences. Some remove specific planes, most commonly the ground plane, while others discard low-curvature planar points before feature matching. Although these strategies can reduce ambiguous point correspondences, they also discard large portions of the scene that provide useful structural cues. Moreover, removing planar points entirely can isolate nearby edge and corner points, leaving too few neighbors for reliable descriptor computation. The key observation behind PARTE is that, although individual points on a plane may be difficult to distinguish, the planar patch itself can provide a distinctive geometric feature and impose strong constraints on the relative pose.

Based on this observation, we formulate global registration over a mixed set of point and planar-patch correspondences. PARTE first uses a greedy region-growing algorithm to extract planar patches, while the remaining points are separated for point matching. For plane matching, we introduce a novel descriptor, named the Plane Context Histogram (PCH), alongside a two-level matching procedure for establishing reliable plane correspondences. During point-feature computation, planar points are retained as neighbors but excluded as standalone features. The resulting point and plane correspondences are jointly processed using a graph-theoretic outlier-rejection framework extended to support plane correspondences. To reduce sensitivity to manually tuned thresholds, PARTE uses geometry-dependent consistency tolerances that account for uncertainty in the matched planes. Plane correspondences are further weighted by their descriptor confidence, resulting in a weighted maximum-clique formulation. The remaining point and plane correspondences are then jointly optimized to recover the rigid transformation.

We evaluate PARTE on 8,097 registration pairs across six benchmarks spanning dense RGB-D and sparse LiDAR measurements, indoor and outdoor environments, and point clouds containing between 10^4 and 10^6 points. Across the four principal robotics benchmarks—3DMatch, 3DLoMatch, KITTI-10m, and KITTI-LC—PARTE achieves the highest mean success rate among the evaluated methods while maintaining low end-to-end runtime. PARTE also achieves the highest overall success rate on both additional benchmarks:

ETH and the partial-to-full RESSO benchmark.

In summary, the main contributions of this work are

- We present PARTE, a plane-assisted global registration pipeline that jointly uses point and planar-patch correspondences in graph-based outlier rejection and rigid-transformation estimation.
- We introduce the Plane Context Histogram, a descriptor that represents a planar patch using its surrounding geometry, outperforming 14 existing shape descriptor and plane matching algorithms.
- We formulate joint point-plane outlier rejection as a confidence-weighted maximum-clique problem, using geometry-dependent compatibility constraints that account for uncertainty in the extracted planes.
- We evaluate PARTE on 8,097 registration pairs across six indoor and outdoor benchmarks. PARTE achieves the highest mean success rate across the four principal benchmarks and the highest overall success rate on both ETH and partial-to-full RESSO.

II. RELATED WORK

Feature-based global registration. Most correspondence-based global registration methods represent a point cloud using a subset of distinctive point features. These features may be selected using hand-crafted keypoint detectors [2]–[4] or learned directly from data [5], [6]. Local descriptors are then computed around the selected points and matched between scans. Classical descriptors include FPFH [7], SHOT [8], Spin Images [9], and 3D Shape Context [10], while more recent methods use learned features and matching networks [6], [11]–[15]. Because repetitive or weakly structured regions can produce ambiguous point features, some methods additionally suppress such regions before matching. For example, QUATRO++ removes the ground plane before registration [16], while KISS-Matcher suppresses low-variation geometric regions that tend to produce ambiguous features [17].

Structure-aware registration. An alternative is to explicitly extract larger geometric or semantic structures and use them as registration primitives. The use of geometric primitives for 3D alignment dates back to early work on representing scenes using lines, planes, and other geometric entities [18], [19]. Later methods developed robust plane and line extraction for LiDAR data [20]–[22] and incorporated planar constraints directly into registration. Point-to-plane ICP [23], GP-ICP [24], and NAP-ICP [25], for example, use planar structure within local ICP-style alignment. More recent global methods employ richer geometric primitives: G3Reg [26] uses geometric structures, and QuadricsReg [27] represents scene elements using quadric primitives and optionally combines their semantic labels with intrinsic geometric attributes. Semantic registration methods similarly replace or augment individual point matches with higher-level scene information. Examples include SE-NDT [28], SegICP [29], SegReg [30], SemReg [31], and SarNet [32]. Semantic approaches can provide strong priors in repetitive environments, but typically require semantic prediction or learned representations in addition to the geometric registration pipeline.

Geometric descriptors and plane matching. Planes have been widely used as geometric features in registration and SLAM. In local registration, plane correspondences can often be established using a prior pose estimate or the small motion between consecutive scans, and simple plane properties such as orientation, position, extent, or overlap are often sufficient for association. This allows methods such as Plane-ICP, GP-ICP [24], and NAP-ICP [25], as well as plane-based SLAM systems such as Linear RGB-D SLAM [33], PlanarSLAM [34], and VIP-SLAM [35], to use planar features as additional constraints during pose estimation.

For global plane matching, simple geometric properties such as plane orientation, centroid, size, or boundary shape can be used to identify candidate correspondences, but these properties are often ambiguous when several similar planes are present in the same scene. Several methods therefore incorporate richer information. HIA-TCD [36] describes planar patches using their contour geometry, while PLAIDE [37] uses relationships between planes and lines to establish structure-level correspondences. Plane Pair Matching [38] similarly exploits relationships between pairs of planes rather than describing each patch independently. Other approaches learn the representation directly from data; PlaneMatch [39], for example, learns an RGB-D descriptor for matching coplanar patches across different views.

Generic 3D shape descriptors provide another way to represent planar patches. Descriptors such as ESF [40], Spin Images [9], 3D Shape Context [10], GASD [41], and the hand-crafted features used by SegMatch [42] encode the shape of an object or segment through distributions of point positions, surface orientations, or simple geometric statistics. Although these descriptors can be applied to planar patches, they were developed primarily for object, surface, or segment recognition rather than for distinguishing similar planes across different scenes. In contrast, PCH describes a planar patch using the geometry surrounding the plane, providing additional context when the shape of the patch itself is ambiguous.

Robust correspondence filtering. Feature matching often produces a large number of incorrect correspondences, making robust outlier rejection an important stage of registration. RANSAC [43] and its many variants [44]–[49] address this problem by repeatedly sampling small correspondence subsets and selecting transformations supported by a large set. Other approaches estimate the transformation using robust optimization [50]–[54]. Learning-based methods instead learn to identify or reweight reliable correspondences, including 3DRegNet [55], PointDSC [56], VBReg [57], and BUFFER [58].

Graph-based methods have recently become particularly attractive for registration with very high outlier ratios. They represent candidate correspondences as graph nodes and connect geometrically consistent pairs, allowing mutually compatible correspondences to be identified without first estimating the transformation. Methods such as ROBIN [59], TEASER++ [54], CLIPPER [60], SC²-PCR [61], and CLIPPER+ [62] build on this idea using compatibility graphs and clique-based formulations. A major advantage of this family, which PARTE also exploits, is its robustness to extreme outlier ratios, in some cases as high as 99%.

TABLE I. Notation used throughout the paper.

Symbol	Description
General	
$v \in \mathbb{R}_{>0}$	Voxel-downsampling size.
$R \in \text{SO}(3)$	Rotation between the two point clouds.
$t \in \mathbb{R}^3$	Translation between the two point clouds.
$\mathbb{S}^2$	Unit sphere in $\mathbb{R}^3$.
$\text{SO}(3)$	Group of 3D rotation matrices.
Point clouds	
$P, Q \subset \mathbb{R}^3$	Source and target point clouds.
$p_i \in P, q_i \in Q$	Source and target points.
$\nu_i \in \mathbb{S}^2$	Estimated unit normal of point p_i .
Plane segmentation	
τ_d	Plane-thickness threshold.
τ_θ	Plane-normal dispersion threshold.
Planar patches	
$\Pi = \{\pi_j\}$	Set of extracted planar patches.
$\mathcal{I}_j$	Indices of points belonging to patch π_j .
$\mu_j \in \mathbb{R}^3$	Centroid of patch π_j .
$u_j \in \mathbb{S}^2$	Estimated unit normal of patch π_j .
$d_j \in \mathbb{R}$	Plane offset, $d_j = u_j^\top \mu_j$.
Plane descriptors	
$r_{\text{PCH}} \in \mathbb{R}_{>0}$	PCH signed-distance half-range.
Correspondences	
$C_{\text{pt}} = \{(p_i, q_i)\}$	Candidate point correspondences.
$C_{\text{pl}} = \{(\pi_j^P, \pi_j^Q)\}$	Candidate plane correspondences.
$\hat{C}_{\text{pt}}$	Inlier point correspondences.
$\hat{C}_{\text{pl}}$	Inlier plane correspondences.
$w_j \in [0, 1]$	Confidence of plane correspondence j .
Outlier rejection	
ϵ_p	Base spatial consistency tolerance.
ϵ_θ	Base angular consistency tolerance.

PARTE builds on graph-based global registration by introducing plane correspondences as complementary evidence to conventional point matches. PARTE’s plane matching also targets planar regions in which local point descriptors are ambiguous by representing the geometry surrounding each planar patch. These correspondences are integrated with point matches throughout compatibility testing, confidence-weighted inlier selection, and transformation estimation.

III. NOTATION

The main notation used throughout the paper is summarized in Table I. For a finite set $\mathcal{X}$, $|\mathcal{X}|$ denotes its cardinality. For a vector $x \in \mathbb{R}^n$, we use $\|x\|_1 = \sum_i |x_i|$ and $\|x\|_2 = (\sum_i x_i^2)^{1/2}$ to denote the ℓ_1 and ℓ_2 norms, respectively. For nonnegative vectors $h, g \in \mathbb{R}^n$, we define their χ^2 -distance as

$$\chi^2(h, g) = \sum_{i: h_i + g_i > 0} \frac{(h_i - g_i)^2}{h_i + g_i}. \quad (1)$$

Superscripts P and Q identify quantities associated with the source and target point clouds, respectively. For example, π_j^P and π_j^Q denote a pair of matched planar patches from the two clouds. For notational simplicity, matched features are re-indexed by correspondence. Therefore, (p_i, q_i) denotes the i th point correspondence, while (π_j^P, π_j^Q) denotes the j th plane

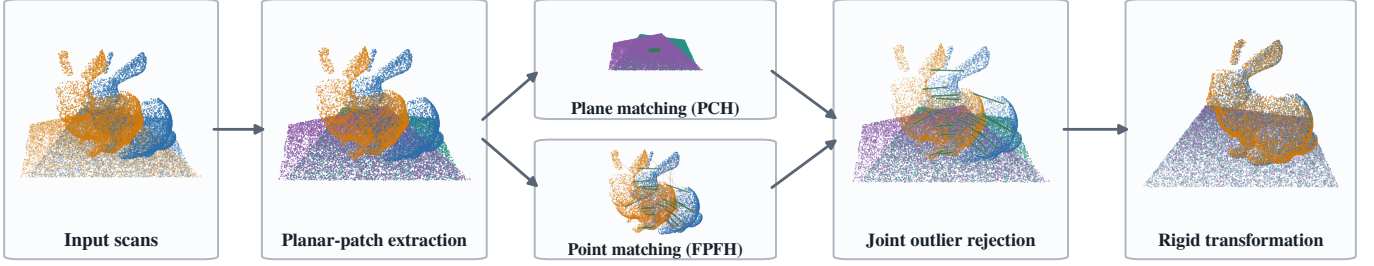


Fig. 2. Overview of the PARTE registration pipeline. Planar patches are extracted from the input scans and matched by PCH, while the remaining points are matched by FPFH. Candidate plane and point correspondences are combined for joint outlier rejection and rigid-transformation estimation.

correspondence. For a symmetric matrix $A \in \mathbb{R}^{3 \times 3}$, we denote its eigenvalues and corresponding unit eigenvectors by $\lambda_i(A)$ and $v_i(A)$, ordered as

$$\lambda_1(A) \geq \lambda_2(A) \geq \lambda_3(A), \quad Av_i(A) = \lambda_i(A)v_i(A). \quad (2)$$

IV. PROPOSED APPROACH

PARTE follows the standard feature-based registration stages of feature extraction, descriptor matching, outlier rejection, and transformation estimation, as illustrated in Fig. 2. Its distinction lies in how planar information is represented and integrated throughout these stages. PARTE describes each extracted planar patch through the surrounding scene geometry using PCH, while retaining conventional FPFH point features for non-planar regions. The resulting point and plane correspondences are combined in a single compatibility graph using mixed point–point, plane–plane, and plane–point consistency tests. Descriptor confidence controls the influence of each plane correspondence during maximum-clique selection, and the retained mixed correspondences jointly constrain the final rigid transformation. When no suitable planar patches are extracted, the same formulation reduces to point-only registration.

A. Preprocessing and Feature Extraction

Preprocessing starts by downsampling each point cloud to reduce data size and the computational cost of registration. Surface normals are then estimated for the downsampled points. The downsampled points are segmented into planar patches, while the remaining points are retained as point features. Together, the planar patches and point features are used for joint plane–point registration in the subsequent stages.

Downsampling. We use voxel-based downsampling with voxel size v . Three-dimensional space is divided into cubic voxels of edge length v , and the points within each occupied voxel are replaced by their centroid.

Normal estimation. For each downsampled point p_i , we estimate a unit surface normal ν_i from its k -nearest-neighbor neighborhood using local PCA. The normal is given by the eigenvector corresponding to the smallest eigenvalue of the neighborhood covariance matrix. We consistently orient the estimated normals toward the sensor viewpoint.

Planar-patch segmentation. We segment the input point cloud into planar regions using a region-growing strategy. The algorithm assumes that the input point cloud

Algorithm 1 Planar Patch Extraction

```

1: Input Point cloud  $P = \{p_i\}_{i=1}^N$  and point normals  $\{\nu_i\}_{i=1}^N$ .
2: Parameters Thickness threshold  $\tau_d$ , normal-dispersion threshold  $\tau_\theta$ , and minimum patch size  $\text{min\_pts}$ .
3: Output  $\Pi = \{\pi_j\}$  % Set of planar patches

4: % Planar seed initialization
5: for all  $p_i \in P$  do
6:    $\bar{C}_i, \bar{Q}_i \leftarrow$  neighborhood statistics of  $p_i$  from (5) and (6)
7:   if  $\lambda_3(\bar{C}_i) \leq \tau_d^2$  and  $1 - \lambda_1(\bar{Q}_i) \leq \tau_\theta^2$  then
8:      $\pi_i \leftarrow$  a patch initialized from  $(p_i, \nu_i)$ 
9:      $\Pi \leftarrow \Pi \cup \{\pi_i\}$ 

10: % Planar patch merging
11: repeat
12:   for all neighboring pairs  $(\pi_a, \pi_b) \in \Pi$  do
13:      $\mathcal{I}_c, \mu_c, C_c, Q_c \leftarrow$  merged patch statistics
14:      $u_c \leftarrow$  solution of (8)
15:     if  $u_c^\top C_c u_c \leq \tau_d^2$  and  $1 - u_c^\top Q_c u_c \leq \tau_\theta^2$  then
16:        $\pi_c \leftarrow (\mathcal{I}_c, \mu_c, C_c, Q_c, u_c)$ 
17:        $\Pi \leftarrow (\Pi \setminus \{\pi_a, \pi_b\}) \cup \{\pi_c\}$ 
18: until no patches are merged

19: % Remove patches with insufficient support
20:  $\Pi \leftarrow$  remove  $\pi_j$  with  $|\mathcal{I}_j| < \text{min\_pts}$ 
21: return  $\Pi$ 

```

$P = \{p_1, \dots, p_N\}$ has an estimated normal $\nu_i \in \mathbb{S}^2$ associated with each point p_i . At each stage, the algorithm maintains a set of planar patches $\Pi = \{\pi_j\}$, where each patch π_j has a support set $\mathcal{I}_j \subseteq \{1, \dots, N\}$. As shown in Lines 4–9 of Algorithm 1, the algorithm initializes a set of planar patches, each containing one point whose local neighborhood is planar. It then repeatedly merges neighboring patches whenever their union remains planar (Lines 10–18). The process terminates when no further valid merge exists, after which patches with insufficient support are discarded.

To determine whether a patch π_j remains planar, let μ_j and $u_j \in \mathbb{S}^2$ denote the centroid and estimated unit normal of its support points $\{p_i : i \in \mathcal{I}_j\}$. Given user-selected thresholds τ_d and τ_θ , we require every patch $\pi_j \in \Pi$ to satisfy

$$\frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} (u_j^\top (p_i - \mu_j))^2 \leq \tau_d^2, \quad (3)$$

$$\frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} [1 - (u_j^\top \nu_i)^2] \leq \tau_\theta^2. \quad (4)$$

The first quantity is the mean squared distance of the support points to the estimated plane. The second measures the average

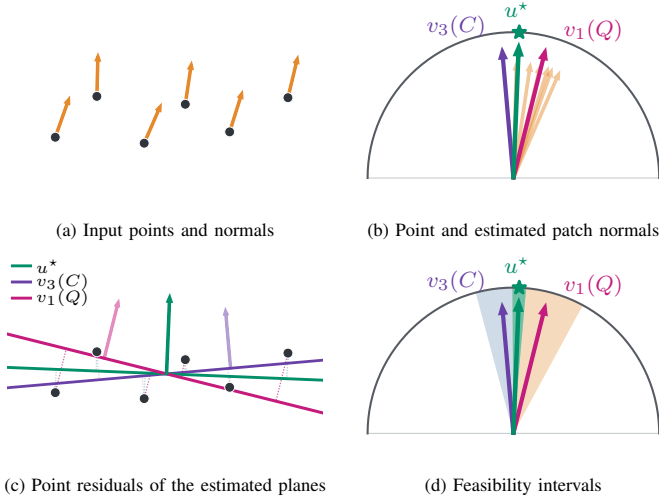


Fig. 3. Illustration of normal estimation. (a) Six noisy points and their normals. (b) The point-based estimate $v_3(C_j)$ can be inconsistent with the point normals. (c) Conversely, estimating the patch normal from the individual point normals using $v_1(Q_j)$ can produce large point-to-plane residuals. (d) The selected direction u^* satisfies both planarity conditions.

disagreement between the patch normal and the normals of its support points.

Directly evaluating these conditions would require iterating over all support points after each candidate merge. Instead, for each patch π_j , we maintain the centroid and covariance of its support points,

$$\mu_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} p_i, \quad C_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} (p_i - \mu_j)(p_i - \mu_j)^\top. \quad (5)$$

We also maintain the second-moment matrix of the corresponding point normals,

$$Q_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \nu_i \nu_i^\top. \quad (6)$$

Using (5) and (6), the planarity conditions reduce to

$$u_j^\top C_j u_j \leq \tau_d^2, \quad 1 - u_j^\top Q_j u_j \leq \tau_\theta^2. \quad (7)$$

The choice of patch normal u_j affects both planarity conditions. It can be estimated either from the covariance of the support points, using $v_3(C_j)$ to minimize the patch thickness, or from their point normals, using $v_1(Q_j)$ to maximize normal agreement. However, neither estimate alone guarantees that both planarity conditions are satisfied. Fig. 3 illustrates this through an extreme two-dimensional example, where the covariance-based estimate fits the points well but is not consistent with the point normals, while the normal-based estimate agrees with the point normals but is not consistent with the points. We therefore select, among all unit normals that are consistent with the points, the direction that best agrees with the point normals

$$u_j = \arg \min_{\|u\|_2=1} (1 - u^\top Q_j u) \quad \text{s.t.} \quad u^\top C_j u \leq \tau_d^2. \quad (8)$$

The resulting three-dimensional problem is solved efficiently through its one-dimensional KKT dual.

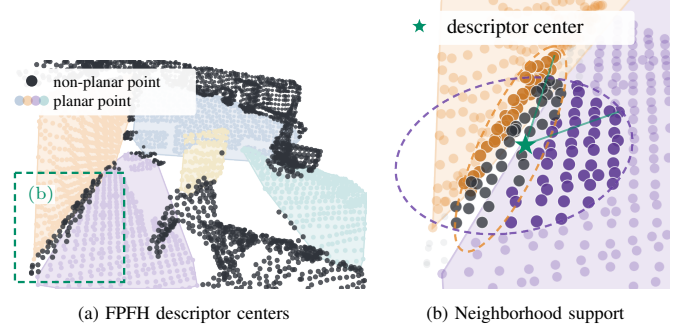


Fig. 4. FPFH descriptor computation after plane extraction. (a) Descriptors are calculated only at non-planar points. (b) Planar points are excluded as descriptor centers but kept in the neighborhood used for descriptor computation.

Algorithm 1 applies these planarity tests during both initialization and region growing. During initialization, each point whose k -nearest-neighbor neighborhood satisfies (7) initializes a singleton patch. All other points are excluded from the region-growing process. During region growing, neighboring patches π_a and π_b form a candidate patch π_c with support $\mathcal{I}_c = \mathcal{I}_a \cup \mathcal{I}_b$. Its statistics are computed from those of the two patches, and its normal is estimated using (8). The merge is finalized only if π_c satisfies (7). The process continues until no valid merge remains, after which patches with fewer than `min_pts` points are discarded.

B. Descriptor generation and matching

In this stage, we compute descriptors for the non-planar point features and extracted planar patches, and independently establish point and plane correspondences between the source and target point clouds.

Point descriptor and matching. We use FPFH [7] to compute a 33-dimensional descriptor for each non-planar point. FPFH captures the local geometric structure around a point using the relative positions and normals of its neighbors. As illustrated in Fig. 4, planar points are excluded as descriptor centers to avoid ambiguous matches, but remain in the neighborhoods of nearby non-planar points during descriptor computation. We match points whose descriptors are mutual nearest neighbors under the ℓ_2 distance, producing the candidate point-correspondence set $\mathcal{C}_{pt} = \{(p_i, q_i)\}_{i=1}^{N_{pt}}$. Although FPFH often produces a high fraction of incorrect correspondences, PARTE’s graph-based outlier-rejection stage is designed to tolerate such cases. This allows the pipeline to use an efficient CPU-based descriptor without relying on more computationally expensive learned alternatives. The point-descriptor stage is modular and can be replaced by other local descriptors without modifying the remaining pipeline.

Plane descriptor and matching. For each planar patch π_j , we encode the geometric information of nearby points into a compact descriptor named the Plane Context Histogram (PCH), illustrated in Fig. 5. For each point p_i , we calculate its signed distance $\delta_i = u_j^\top (p_i - \mu_j)$ and its normal agreement $a_i = u_j^\top \nu_i$ with the patch. We retain points satisfying $|\delta_i| \leq r_{PCH}$, where r_{PCH} is a parameter controlling the signed-distance support of the descriptor, as shown in Fig. 5(a). The resulting (δ_i, a_i) pairs are accumulated into a two-dimensional

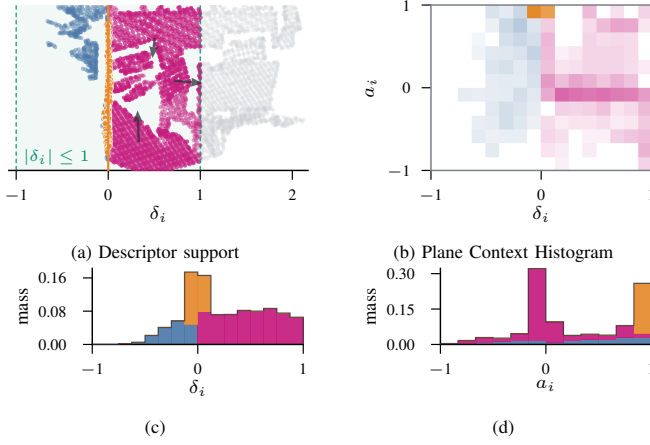


Fig. 5. Construction of the Plane Context Histogram. (a) Points within $|\delta_i| \leq r_{PCH}$ of the reference plane are marked. Orange denotes points of the patch, while blue and magenta denote nearby context points on opposite sides of the patch. (b) Each point contributes a signed-distance and normal-agreement pair (δ_i, a_i) , which is accumulated into the normalized two-dimensional PCH. (c,d) Signed-distance and normal-agreement distributions.

histogram with 16 distance bins and 12 normal-agreement bins, producing a 192-dimensional descriptor as illustrated in Fig. 5(b). The histogram is normalized to sum to one, reducing its sensitivity to the number of contributing points.

A point cloud may contain several disconnected patches that lie on the same infinite plane. Because these patches produce very similar descriptors, we group them and combine their histograms into a single plane-level descriptor, as illustrated in Fig. 6(a,b). The resulting plane-level descriptors are matched across the two point clouds using mutual nearest neighbors under the χ^2 -distance (Fig. 6(c)).

When a matched plane-level descriptor represents a single patch in each point cloud, the correspondence is obtained directly. If it represents multiple disconnected coplanar patches, we perform a second matching stage within the matched groups, as illustrated in Fig. 6(d). For each patch, we compute an oriented bounding box and construct a separate PCH using only points whose projections lie inside the box. This captures differences in the local surroundings of each patch, helping distinguish patches that lie on the same plane. We then apply the same matching procedure within each group. The direct and refined matches together form the candidate plane-correspondence set $\mathcal{C}_{pl}$.

For each correspondence, we also compute a confidence score by comparing the descriptor distance of the selected match with the descriptor distances to the next-best candidate for each patch. For a mutual nearest-neighbor match, let $d = d_P^{(1)} = d_Q^{(1)}$ denote the descriptor distance between the matched patches, and let $d_P^{(2)}$ and $d_Q^{(2)}$ denote the descriptor distances to their respective second-best candidate matches. We define the confidence score as

$$w = \sqrt{\left(1 - \frac{d}{d_P^{(2)}}\right) \left(1 - \frac{d}{d_Q^{(2)}}\right)}. \quad (9)$$

Because $d \leq d_P^{(2)}$ and $d \leq d_Q^{(2)}$ for a mutual nearest-neighbor match, $w \in [0, 1]$. The score approaches one when the selected match is substantially closer than the next-best alternatives for

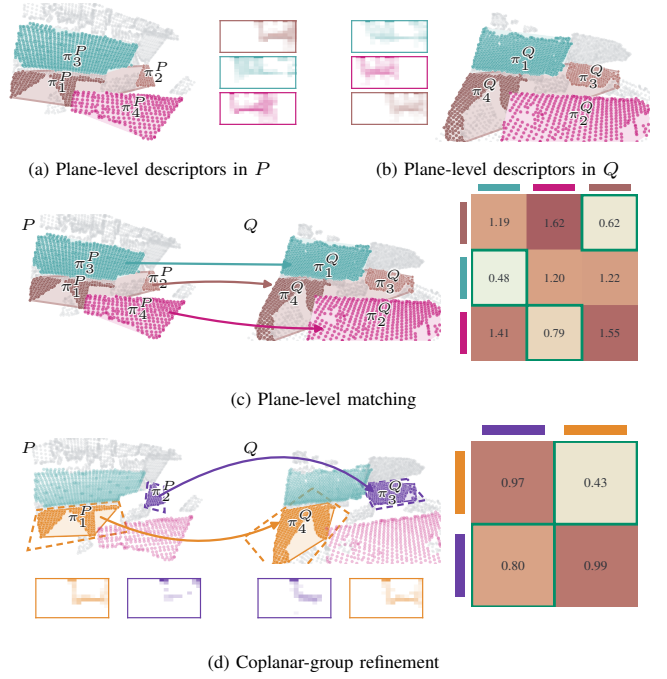


Fig. 6. Plane Context Histogram matching and coplanar-group refinement. (a,b) Individual patches are represented by their PCHs, while disconnected patches lying on the same plane are grouped and represented by a combined descriptor. (c) Plane-level descriptors are matched between P and Q using mutual nearest neighbors under the χ^2 -distance; the matrix shows the descriptor distances. (d) For a matched coplanar group, oriented bounding boxes restrict the support of each constituent patch to construct localized PCHs. These descriptors are then rematched within the group using the shown 2×2 distance matrix.

both patches, and approaches zero when either descriptor is ambiguous. This confidence score is later used to weight plane correspondences during graph-based outlier rejection.

C. Outlier Rejection

The candidate point and plane correspondences generally contain many incorrect matches. We reject them by finding a subset of correspondences that are mutually consistent under a rigid transformation. Let $\mathcal{C}_{pt} = \{(p_i, q_i)\}$ and $\mathcal{C}_{pl} = \{(\pi_j^P, \pi_j^Q)\}$ denote the candidate point and plane correspondences. An example of the resulting point and plane correspondences is shown in Fig. 7(a). Rigid transformations preserve several geometric relationships, which allow us to test the consistency of pairs of correspondences.

Point-point consistency. For two inlier point correspondences (p_i, q_i) and (p_k, q_k) , their pairwise distance is preserved:

$$\|p_i - p_k\|_2 - \|q_i - q_k\|_2 \leq 2\epsilon_p. \quad (10)$$

Plane-plane consistency. For two inlier plane correspondences (π_j^P, π_j^Q) and (π_k^P, π_k^Q) , the angle between their normals is preserved:

$$|\angle(u_j^P, u_k^P) - \angle(u_j^Q, u_k^Q)| \leq 2\epsilon_\theta. \quad (11)$$

Plane-point consistency. A rigid transformation also preserves the signed distance between a point and a plane. For

a plane correspondence (π_j^P, π_j^Q) and point correspondence (p_i, q_i) , we define

$$r_{ij} = \left| ((u_j^P)^\top p_i - d_j^P) - ((u_j^Q)^\top q_i - d_j^Q) \right|. \quad (12)$$

Normal uncertainty motivates the following bound:

$$r_{ij} \leq \epsilon_p + \epsilon_\theta \min(\|p_i - \mu_j^P\|_2, \|q_i - \mu_j^Q\|_2), \quad (13)$$

where ϵ_θ is expressed in radians. The first term accounts for the base distance error, while the second grows with the distance from the corresponding plane centroid. Therefore, normal uncertainty has a larger effect on points farther from the plane centroid. This effect is illustrated in Fig. 7(e), where uncertainty in the plane normal produces a larger bound farther from the plane centroid.

We construct a compatibility graph with one node for each candidate point or plane correspondence. Two nodes are connected when their correspondences satisfy the applicable consistency test above. Therefore, every clique represents a set of pairwise consistent correspondences.

For each plane correspondence, we convert the descriptor confidence w_j defined in (9) to the integer graph weight $\omega_j = \lceil 10w_j \rceil$. Point correspondences are assigned a unit graph weight, while plane correspondences are assigned weight ω_j . We then select a maximum-weight clique rather than simply the clique with the largest number of nodes. This favors sets containing distinctive plane matches without relaxing any of the geometric consistency tests. Fig. 7(b) shows an example in which the plane correspondences help select the correct joint clique, while Fig. 7(d) shows the larger but incorrect point-only clique obtained when correspondences are selected only by cardinality. A weighted clique can be represented in an unweighted graph by duplicating a node according to its integer weight. Rather than explicitly expanding the graph in this way, we extend the Parallel Maximum Clique (PMC) solver to operate directly on nonnegative integer node weights. This produces the same weighted-clique objective while avoiding the additional nodes and edges introduced by duplication. The correspondences retained by the joint clique are then used for transformation estimation, producing the alignment shown in Fig. 7(c).

D. Transformation Estimation

After outlier rejection, let $\hat{\mathcal{C}}_{\text{pt}} = \{(p_i, q_i)\}_{i=1}^{N_{\text{pt}}}$ and $\hat{\mathcal{C}}_{\text{pl}} = \{(\pi_j^P, \pi_j^Q)\}_{j=1}^{N_{\text{pl}}}$ denote the remaining correspondences. We estimate the rigid transformation (R, t) by solving

$$\begin{aligned} (R^*, t^*) = \arg \min_{\substack{R \in \text{SO}(3) \\ t \in \mathbb{R}^3}} & \sum_{i=1}^{N_{\text{pt}}} \|Rp_i + t - q_i\|_2^2 \\ & + \sum_{j=1}^{N_{\text{pl}}} w_j \|Ru_j^P - u_j^Q\|_2^2 \\ & + \sum_{j=1}^{N_{\text{pl}}} w_j \left(d_j^Q - d_j^P - (u_j^Q)^\top t \right)^2. \end{aligned} \quad (14)$$

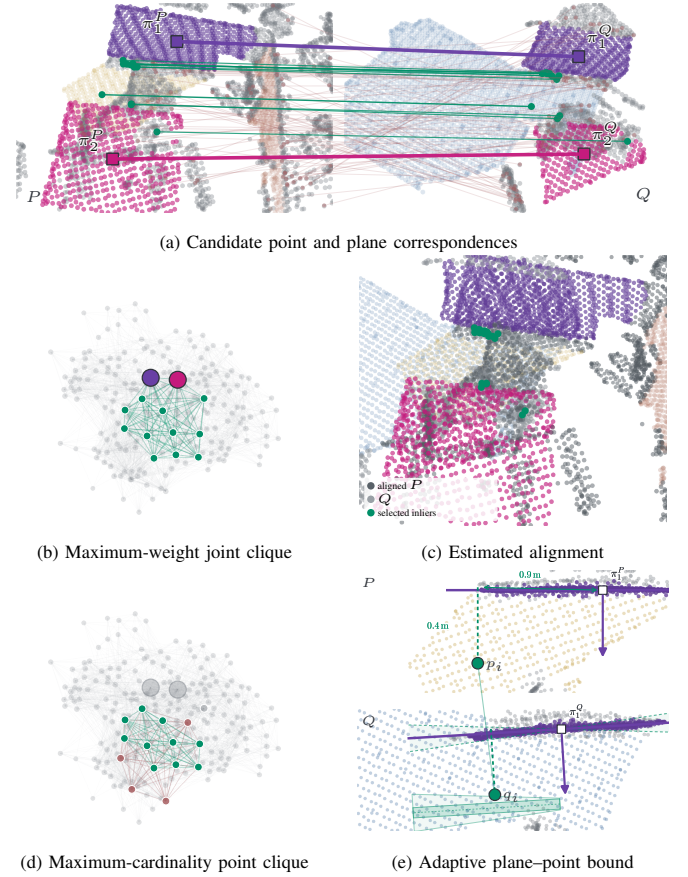


Fig. 7. Joint point-plane outlier rejection. (a) Descriptor matching produces candidate point and plane correspondences. (b) The maximum-weight clique combines the selected plane correspondences (large colored nodes) with a mutually consistent subset of point correspondences (green nodes). (c) The registration result from selected nodes. (d) In contrast, selecting the clique by size yields a larger set that includes incorrect correspondences. (e) The plane-point consistency threshold adapts to the plane-normal uncertainty and point-to-centroid distance.

The first term minimizes the Euclidean distance between corresponding points. The second aligns the normals of corresponding planes. The third aligns their offsets after applying the translation. Each plane correspondence is weighted by its confidence score w_j .

The joint problem in (14) does not admit a straightforward solution for R and t simultaneously. However, when t is fixed, optimizing R reduces to an orthogonal Procrustes problem involving the matched points and plane normals, which can be solved using a 3×3 SVD. Similarly, when R is fixed, optimizing t is a quadratic least-squares problem with a closed-form solution. We therefore minimize the objective using Block Coordinate Descent (BCD), alternating between these two updates. Since each update optimizes one variable while keeping the other fixed, the objective cannot increase between iterations.

To initialize BCD, we choose the initial translation so that the first rotation update reduces to an extended Arun solution [63]. Specifically, we set $t^{(0)} = \bar{q}$, where $\bar{q}$ is the centroid of the matched target points. This reduces the point contribution of the first rotation update to the centered formulation of the classical Arun method, while the matched plane normals

Algorithm 2 Joint Point–Plane Transformation Estimation

```

1: Input Point correspondences  $\hat{\mathcal{C}}_{\text{pt}} = \{(p_i, q_i)\}_{i=1}^{N_{\text{pt}}}$ , plane corre-
   spondences  $\hat{\mathcal{C}}_{\text{pl}} = \{(\pi_j^P, \pi_j^Q)\}_{j=1}^{N_{\text{pl}}}$ , and plane weights  $\{w_j\}$ 
2: Output Rigid transformation  $(R, t)$ 

3: % Initialization
4:  $t \leftarrow \frac{1}{N_{\text{pt}}} \sum_i q_i$  % Extended Arun initialization

5: % Block coordinate descent
6: repeat
7:   % Update  $R$  for fixed  $t$ 
8:    $H \leftarrow \sum_i (q_i - t)p_i^\top + \sum_j w_j u_j^Q (u_j^P)^\top$ 
9:    $H = U \Sigma V^\top$ 
10:   $R \leftarrow U \text{diag}(1, 1, \det(UV^\top)) V^\top$ 

11: % Update  $t$  for fixed  $R$ 
12:   $A \leftarrow N_{\text{pt}} I_3 + \sum_j w_j u_j^Q (u_j^Q)^\top$ 
13:   $b \leftarrow \sum_i (q_i - R p_i) + \sum_j w_j (d_j^Q - d_j^P) u_j^Q$ 
14:  Solve  $At = b$  for  $t$ 
15: until  $R$  and  $t$  have converged
16: return  $(R, t)$ 

```

provide additional rotation constraints. The resulting closed-form rotation provides the initialization for the subsequent BCD updates.

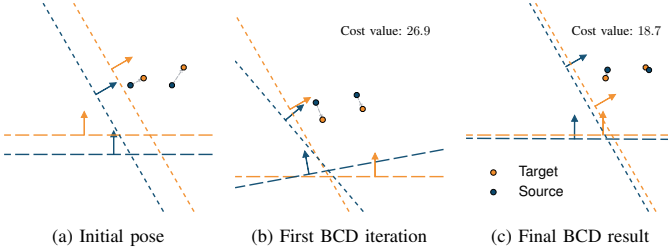


Fig. 8. Comparison of alignment stages in BCD optimization, Algorithm 2. Source points and planes (orange) are progressively aligned to the target configuration (blue). (a) shows the initial mismatch. (b) shows the pose obtained after the first rotation update of the BCD scheme with fixed translation $t = q_c$, which is equivalent to the classical Arun rotation estimate. (c) shows the final result after the full nonlinear BCD optimization, where both point correspondences and planes are jointly refined.

V. EXPERIMENTAL EVALUATION

In this section, we evaluate PARTE at three different levels. First, we evaluate the proposed plane descriptor and matching procedure directly by measuring the quality of the resulting plane correspondences. We then evaluate the complete registration pipeline across indoor RGB-D, outdoor LiDAR, and partial-to-full registration problems. Finally, we study the main design choices through ablation experiments and analyze the runtime of the complete pipeline.

A. Experimental Setup

Datasets. We organize the evaluation around established registration benchmarks and follow their standard pair definitions and success criteria whenever available. For indoor RGB-D registration, we use the eight test scenes of 3DMatch [11] and additionally report results on 3DLoMatch [13]. 3DMatch contains pairs with at least 30% overlap, while 3DLoMatch

Dataset	Sensor	#Scenes	#Pairs	v	#Pts/scan	#Planes/scan	Success criterion
3DMatch [11]	RGB-D	8	1,623	5 cm	4,838	4.6	$\leq 15^\circ, \leq 30$ cm
3DLoMatch [13]	RGB-D	8	1,781	5 cm	4,821	4.6	$\leq 15^\circ, \leq 30$ cm
KITTI-10m [26]	LiDAR	3	556	30 cm	18,238	5.1	$\leq 5^\circ, \leq 0.6$ m
KITTI-LC [26]	LiDAR	5	3,325	30 cm	18,354	7.9	$\leq 5^\circ, \leq 2$ m
RESSO P2F [37]	TLS + RGB-D	7	99	5 cm	10,143	7.4	$\leq 15^\circ, \leq 30$ cm
ETH [65]	LiDAR	4	713	10 cm	30,234	2.0	$\leq 15^\circ, \leq 30$ cm

TABLE II. Datasets and evaluation protocols used in our experiments. Point and plane counts are averages over unique scans appearing in the evaluation pairs after voxel downsampling.

contains the more challenging pairs with 10–30% overlap. For outdoor LiDAR registration, we use the KITTI odometry dataset [64] under two complementary protocols. The first is the standard KITTI-10m pairwise-registration benchmark commonly used in prior work [6], [13], while the second is the KITTI-LC loop-closure benchmark introduced by G3Reg [26]. We further include two additional settings that test registration outside these standard benchmarks. From RESSO [37], we select sequences in which a local hand-held scan is registered against a larger static scan, forming a challenging partial-to-full registration problem. Finally, we use the ETH benchmark [65] to evaluate full-scan registration on challenging terrestrial LiDAR data.

Compared methods. For the plane-matching evaluation, we compare PCH against a broad set of descriptors that represent planar patches using different geometric cues. As simple geometric baselines, we use patch area and perimeter. The local descriptor baselines include Spin Images [9], Angular Spin Images [66], USC [67], and 3D Shape Context [10]. We additionally evaluate global and segment-level representations, including VFH [68], CVFH [69], OUR-CVFH [70], GRSD [71], [72], ESF [40], GASD [41], the handcrafted segment features used by SegMatch [42], and the Gaussian Ellipsoid Model (GEM) representation used by G3Reg [26].

For the complete registration experiments, we compare PARTE with representative classical, graph-based, geometry-based, and learning-based methods. The classical baselines include RANSAC [43] and FGR [50]. We compare against robust correspondence-filtering methods TEASER++ [54], MAC [73], and CLIPPER+ [62], as well as the QUATRO estimator [74], evaluated with ground-filtered point correspondences following the QUATRO++ strategy [16]. G3Reg [26] provides a recent geometry-based comparison using higher-level geometric primitives. We additionally compare against the plane-based methods PLADE [37] and VPFB [75]. For datasets with publicly available trained models, we additionally compare against the learning-based methods PREDATOR [13], PointDSC [56], and VBReg [57].

Evaluation metrics. For plane matching, we report F1 score, average precision (AP), and the average number of true-positive correspondences per scan pair, $\overline{\text{TP}}$. F1 is computed from the mutual nearest-neighbor correspondences, while AP summarizes the precision–recall curve obtained by varying the descriptor distance threshold over all candidate plane pairs. For registration, given an estimated transformation (R, t) and ground truth (R^*, t^*) , we compute the rotation error (RE)

$$\text{RE} = \arccos\left(\frac{\text{tr}(R^\top R^*) - 1}{2}\right),$$

TABLE III. Parameter settings used throughout the experiments.

Stage	Symbol	Parameter	Value
Normal estimation	–	Radius / max neighbors	$2v$ / 30
Plane extraction	τ_d	Thickness	v
	τ_θ	Normal dispersion	0.2
	–	Minimum points	100
FPFH	–	Radius / max neighbors	$5v$ / 100
PCH	r_{PCH}	Signed-distance half-range	$20v$
Outlier rejection	ϵ_p	Base spatial tolerance	v
	ϵ_θ	Base angular tolerance	5°

and translation error (TE)

$$TE = \|t - t^*\|_2.$$

We report the success rate (SR), defined as the percentage of registration pairs whose RE and TE satisfy the dataset-specific criteria in Table II. We additionally report average rotation error (ARE), average translation error (ATE), and mean end-to-end runtime per pair. Unless otherwise stated, RE and ARE are reported in degrees, TE and ATE in centimeters, and runtime in milliseconds.

Implementation and parameter settings. Table III summarizes the parameters used throughout the experiments. We use the voxel size v as the common geometric scale of the pipeline. Apart from selecting v for each dataset as shown in Table II, the remaining parameters are fixed across all experiments or expressed directly as multiples of v . Surface normals are estimated within a radius of $2v$ using at most 30 neighbors, and the same neighborhoods are used during plane extraction. Plane segmentation uses $\tau_d = v$, $\tau_\theta = 0.2$, and a minimum patch size of 100 points. FPFH descriptors use a radius of $5v$ with at most 100 neighbors, while PCH uses $r_{PCH} = 20v$ and therefore covers signed distances in the range $[-20v, 20v]$. For outlier rejection, we use the base spatial tolerance $\epsilon_p = v$ and the base angular tolerance $\epsilon_\theta = 5^\circ$. The resulting point–point and plane–plane bounds are therefore $2v$ and 10° , respectively, while the plane–point bound adapts to the matched-plane geometry as described in Sec. IV-C.

B. Plane Descriptor Evaluation

We first evaluate the quality of the plane correspondences produced by PCH independently of the complete registration pipeline. We compare against a broad set of alternatives, including simple geometric properties, local point descriptors, and global and segment descriptors. All competing descriptors are computed from the same extracted planar patches. Patch-based descriptors use only the points belonging to the corresponding plane, whereas PCH additionally uses the surrounding scene geometry by design. For each descriptor, we construct the pairwise distance matrix between the planes extracted from the two scans and establish correspondences using mutual nearest-neighbor (MNN) matching. Different distance measures have been used with histogram-based geometric descriptors in prior work and reference implementations [10], [40], [68]–[70]. We therefore evaluate multiple reasonable choices where applicable and report only the distance that gives the highest overall MNN F1 score across the evaluated datasets. Specifically, we compare ℓ_1 and χ^2 distances for PCH, ESF, and VFH; minimum ℓ_1 and minimum χ^2 distances

between the signature sets produced by CVFH and OUR-CVFH; and ℓ_2 and normalized χ^2 distances for 3DSC. For Spin Image and Angular Spin, we use $1 - \rho$, where ρ denotes the Pearson correlation coefficient between the two descriptor vectors. For the simple geometric baselines, we similarly evaluate absolute difference, ratio, and log-ratio measures. The selected distance for each descriptor is reported in Table IV.

Table IV summarizes the resulting plane-matching performance. F1 and $\overline{TP}$ are computed from the MNN correspondences, while AP summarizes the precision–recall behavior of each descriptor. **Our one-level PCH matcher performs substantially better than the competing descriptors across most datasets, while our full two-level matcher, PCH-2L in Table IV, further improves performance.** In particular, the improvement is large on 3DLoMatch and the two KITTI benchmarks, where descriptors based primarily on the shape or extent of the planar patch provide relatively weak discrimination.

Fig. 9 shows the corresponding precision–recall curves and confidence analysis on 3DMatch and KITTI-LC. **PCH remains above the competing precision–recall curves throughout the plotted recall range on both datasets,** showing that its advantage is not limited to a particular matching threshold or method. What sets PCH apart is that it describes each plane using the geometry around it rather than the planar patch alone. Overall, these results support the main motivation for PCH: planar patches are often difficult to distinguish from their own geometry alone, while the geometry surrounding the plane provides a more distinctive matching cue.

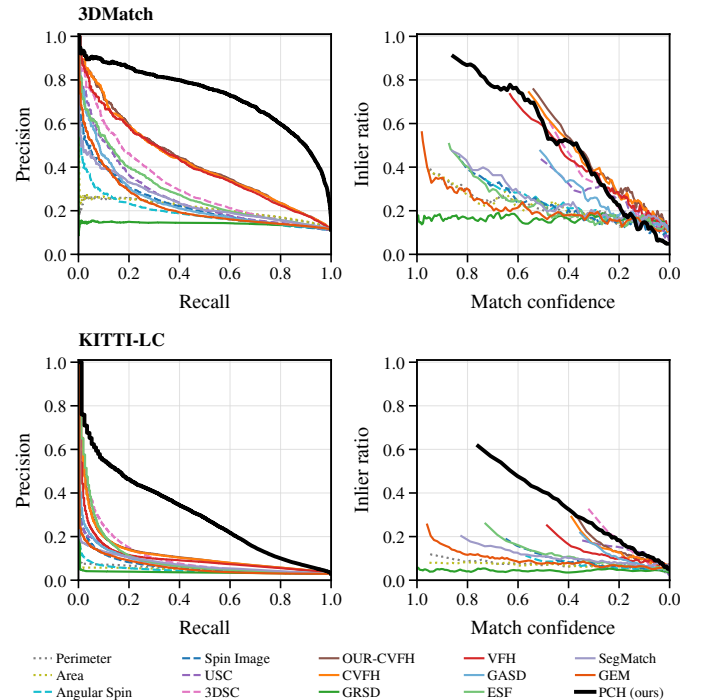


Fig. 9. Plane matching performance on 3DMatch (top) and KITTI-LC (bottom). Left column: precision–recall curves, where PCH maintains substantially higher precision across the recall range. Right column: descriptor-match confidence versus empirical inlier ratio, where higher PCH confidence corresponds strongly to a higher probability of a correct plane correspondence.

Name	Dist.	Size	3DLoMatch			3DMatch			KITTI-10m			KITTI-LC			ETH			RESSO		
			F1	AP	TP	F1	AP	TP	F1	AP	TP	F1	AP	TP	F1	AP	TP	F1	AP	TP
<i>Simple geometry</i>																				
Perimeter	$ \log(a/b) $	1	0.15	0.08	0.26	0.31	0.21	0.83	0.16	0.08	0.78	0.10	0.05	0.69	0.65	0.48	0.90	0.05	0.02	0.21
Area	$ \log(a/b) $	1	0.15	0.08	0.27	0.32	0.21	0.84	0.13	0.08	0.66	0.08	0.05	0.59	0.66	0.51	0.90	0.05	0.02	0.19
<i>Local point descriptors</i>																				
Angular Spin [66]	$1 - \rho$	153	0.13	0.08	0.19	0.24	0.19	0.54	0.12	0.09	0.50	0.07	0.05	0.42	0.25	0.14	0.22	0.06	0.03	0.20
Spin Image [9]	$1 - \rho$	153	0.12	0.09	0.19	0.27	0.23	0.64	0.17	0.12	0.75	0.11	0.07	0.66	0.69	0.68	0.91	0.08	0.03	0.27
USC [67]	ℓ_2	1960	0.17	0.10	0.19	0.29	0.27	0.55	0.14	0.13	0.42	0.09	0.08	0.36	0.64	0.60	0.68	0.06	0.04	0.15
3DSC [10]	χ^2	1980	0.15	0.12	0.19	0.36	0.32	0.74	0.25	0.19	0.99	0.16	0.11	0.86	0.72	0.77	0.91	0.05	0.04	0.14
<i>Global/segment descriptors</i>																				
OUR-CVfH [70]	$\min \ell_1$	$K \times 308$	0.22	0.14	0.36	0.47	0.42	1.21	0.26	0.20	1.23	0.17	0.12	1.13	0.46	0.35	0.40	0.12	0.05	0.41
CVfH [69]	$\min \ell_1$	$K \times 308$	0.22	0.15	0.36	0.47	0.42	1.21	0.26	0.19	1.22	0.16	0.12	1.08	0.76	0.82	1.02	0.13	0.06	0.48
GRSD [71], [72]	χ^2	21	0.09	0.07	0.15	0.16	0.14	0.39	0.09	0.06	0.41	0.05	0.04	0.34	0.57	0.40	0.72	0.03	0.02	0.11
VFH [68]	ℓ_1	308	0.22	0.14	0.38	0.46	0.41	1.20	0.23	0.17	1.12	0.15	0.10	1.02	0.77	0.82	1.02	0.11	0.05	0.42
GASD [41]	ℓ_1	512	0.11	0.08	0.18	0.29	0.24	0.69	0.20	0.13	0.81	0.13	0.08	0.71	0.68	0.65	0.90	0.09	0.04	0.31
ESF [40]	χ^2	640	0.15	0.10	0.26	0.32	0.29	0.82	0.21	0.15	1.01	0.15	0.10	0.98	0.64	0.63	0.82	0.10	0.04	0.35
SegMatch [42]	ℓ_2	8	0.15	0.09	0.24	0.32	0.24	0.83	0.22	0.14	1.05	0.14	0.07	0.97	0.66	0.56	0.83	0.08	0.04	0.28
GEM [26]	W_2^2	12	0.16	0.08	0.28	0.35	0.23	0.89	0.23	0.12	1.11	0.15	0.06	1.05	0.71	0.44	0.96	0.09	0.05	0.35
<i>Ours</i>																				
PCH	χ^2	192	0.43	0.26	0.74	0.79	0.72	2.23	0.59	0.44	2.91	0.42	0.30	2.76	0.76	0.66	1.04	0.31	0.18	1.07
PCH-2L	χ^2	192	0.46	0.34	0.77	0.80	0.77	2.24	0.66	0.54	3.17	0.48	0.33	3.00	0.76	0.72	1.04	0.32	0.18	1.09

TABLE IV. Plane-matching performance across the evaluated datasets. We report F1 score, average precision (AP), and the average number of true-positive correspondences per scan pair, TP.

Fig. 9 also shows how the empirical inlier ratio varies with the match confidence defined in (9). For PCH, the inlier ratio increases strongly with confidence on both datasets, with the highest-confidence correspondences being correct in the large majority of cases, making confidence a reliable basis for weighting plane matches.

C. Global Registration Performance

For the main registration benchmarks, methods that operate on a supplied correspondence set are given the same point correspondences, obtained using FPFH descriptors followed by mutual nearest-neighbor (MNN) matching. This includes RANSAC, FGR, TEASER++, MAC, CLIPPER+, PointDSC, and VBReg. The only exception is MAC: its input correspondence set is capped at 4,000 matches when necessary, as larger sets exceed 32 GB of memory. For methods requiring a noise-bound parameter, we use the voxel size v . For QUATRO, following QUATRO++ [16], we segment the ground on KITTI and ETH using TRAVEL [76], remove the identified ground points, and apply the same point-matching frontend to the remaining points. PARTE similarly uses TRAVEL for ground segmentation in KITTI, but it keeps the segmented ground for plane-matching. Methods that include their own feature extraction or correspondence-generation stage are evaluated using their native pipelines, including PREDATOR, G3Reg, KISS-Matcher, VPFBR, and PLADE. Unless otherwise stated, reported runtimes include the complete computational pipeline from feature extraction and matching through outlier rejection and transformation estimation, excluding file I/O.

Indoor RGB-D registration. We first evaluate global registration on 3DMatch and the more challenging 3DLoMatch benchmark. Table V reports the results using FPFH computed with camera-oriented normals. PARTE achieves success rates of 89.6% on 3DMatch and 55.8% on 3DLoMatch, ranking second only to PREDATOR on both benchmarks. PARTE also obtains the lowest rotation error on both datasets and

the lowest translation error on 3DMatch, while maintaining relatively low end-to-end runtime. The improvement is particularly notable on 3DLoMatch, where the low overlap produces a substantially more difficult point-matching problem.

TABLE V. Indoor registration performance on 3DMatch and 3DLoMatch using FPFH descriptors with mutual nearest-neighbor matching. Reported runtimes include the complete registration pipeline, from feature computation and matching through outlier rejection and transformation estimation.

Method	3DMatch				3DLoMatch			
	SR	ATE	ARE	Time	SR	ATE	ARE	Time
<i>Baseline</i>								
RANSAC-10 ³	59.8	9.5	3.4	63.4	15.8	13.5	5.0	58.7
RANSAC-10 ⁵	85.6	8.4	2.8	70.8	45.8	13.2	4.5	70.3
RANSAC-10 ⁷	87.7	8.4	2.8	120.2	50.9	13.0	4.4	282.7
FGR	80.3	6.8	2.2	92.3	22.6	10.7	3.7	67.2
<i>Robust point-based methods</i>								
KISS-Matcher	75.5	8.7	2.9	110.8	26.8	13.0	4.5	26.4
CLIPPER+	85.9	7.0	2.3	117.5	44.7	11.4	3.9	130.1
TEASER++	86.6	7.1	2.3	77.7	48.0	11.6	3.9	64.5
MAC	88.0	6.8	2.1	112.4	52.4	11.6	3.7	85.3
<i>Learning-based methods</i>								
PointDSC	85.5	6.6	2.1	72.2	41.5	11.0	3.6	65.4
VBReg	89.4	6.8	2.1	102.8	53.5	11.6	3.7	94.0
PREDATOR	92.1	7.4	2.3	95.7	61.0	11.0	3.7	82.0
<i>Segment- and plane-based methods</i>								
VPFBR	26.5	13.0	2.9	28.8	2.8	16.3	3.7	26.4
PLADE	34.4	8.2	2.0	1664.7	7.3	14.2	3.2	1292.1
G3Reg	4.8	14.2	6.2	35.1	0.0	—	—	42.9
<i>Ours</i>								
PORTE	89.6	6.2	1.8	59.4	55.8	10.8	3.4	56.8

The quality of the point correspondences has a substantial effect on registration performance. In particular, consistently orienting the point normals toward the camera considerably improves FPFH matching compared with independently estimated unoriented normals. Table VI compares several point-matching frontends to separate this effect from the registration method itself. PARTE achieves the highest success rate with both FPFH variants on 3DMatch and 3DLoMatch, and remains

TABLE VI. Success-rate comparison under different point-matching frontends on 3DMatch and 3DLoMatch. Oriented FPFH uses camera-oriented normals, while FPFH uses independently estimated unoriented normals. All frontends use mutual nearest-neighbor matching, except FCGF + 1K, which uses 1,000 sampled correspondences.

Method	Oriented FPFH		FPFH		FCGF		FCGF + 1K	
	3DM	3DLo	3DM	3DLo	3DM	3DLo	3DM	3DLo
<i>Baseline</i>								
RANSAC-10 ³	59.8	15.8	42.3	4.4	86.3	37.8	80.8	19.4
RANSAC-10 ⁵	85.6	45.8	73.4	25.2	88.6	44.6	90.0	47.5
RANSAC-10 ⁷	87.7	50.9	76.3	31.2	88.6	44.5	90.4	50.9
FGR	80.3	22.6	66.6	10.5	88.4	37.8	87.3	30.7
<i>Graph-based robust estimators</i>								
CLIPPER+	85.9	44.7	75.5	24.9	87.4	41.9	88.7	45.1
TEASER++	86.6	48.0	76.3	27.8	85.6	38.5	87.2	43.2
MAC	88.0	52.4	77.4	32.1	85.3	44.9	91.7	54.6
<i>Learning-based methods</i>								
PointDSC	85.5	41.5	71.8	22.0	88.7	43.9	91.1	47.1
VBReg	89.4	53.5	79.7	33.6	90.6	48.7	92.2	55.2
<i>Ours</i>								
PARTE	89.6	55.8	85.5	39.1	89.3	47.7	90.2	50.5

the highest-performing non-learned method when FCGF correspondences are matched using MNN. When 1,000 FCGF correspondences are retained, the gap narrows and MAC performs best among the geometric methods. This correspondence set contains more competing mutually consistent structures, a setting that MAC explicitly addresses by considering multiple maximal cliques rather than relying on a single maximum clique. Overall, PARTE remains among the top-performing methods across the different point frontends, indicating that its performance is not tied to a particular point descriptor or matching strategy.

The frontend comparison also illustrates the trade-off between correspondence quantity and robust-estimation cost. MNN produces a relatively compact correspondence set, whereas retaining a larger number of FCGF matches substantially increases the runtime of several robust estimators. This motivates the use of MNN in the main evaluation, while the consistent performance of PARTE across the different frontends indicates that its registration gains are largely complementary to the choice of point-matching method.

Outdoor LiDAR registration. We evaluate outdoor registration on the standard KITTI-10m benchmark and the more challenging KITTI-LC loop-closure benchmark.

Table VII reports the results on KITTI-10m. PARTE achieves a 99.8% overall success rate, matching PREDATOR for the highest performance and reaching near-perfect registration on all three sequences. G3Reg also performs strongly with a 99.1% success rate, while most modern robust registration methods exceed 97%. These results indicate that KITTI-10m is largely saturated by current global registration methods [26]. PARTE nevertheless reaches this performance with a mean runtime of 178 ms, substantially below PREDATOR, TEASER++, CLIPPER+, and MAC, although G3Reg and KISS-Matcher remain faster. The RANSAC results also show a clear convergence trend, increasing from 10.3% success at 10³ iterations to 91.5% at 10⁵ and 98.2% at 10⁷ iterations.

KITTI-LC provides a substantially more difficult setting by evaluating temporally separated loop closures over a wider range of relative motion. Table VIII therefore separates the results into 0–10 m, 10–20 m, and 20–30 m translation ranges

TABLE VII. Registration performance on the KITTI-10m benchmark. Success rate is reported for sequences 08–10 and overall, together with mean end-to-end runtime.

Method	Success Rate (%)				Runtime (ms)			
	08	09	10	All	08	09	10	All
<i>Baseline</i>								
RANSAC-10 ³	4.9	16.7	17.2	10.3	419.2	432.1	287.0	402.3
RANSAC-10 ⁵	89.3	94.4	94.3	91.5	434.2	446.4	300.0	416.7
RANSAC-10 ⁷	97.4	100.0	97.7	98.2	594.3	562.0	367.1	549.3
FGR	22.5	34.0	37.9	28.2	480.1	506.6	345.7	466.8
<i>Robust point-based methods</i>								
KISS-Matcher	97.1	98.8	94.3	97.1	98.8	107.1	78.4	98.0
CLIPPER+	98.0	99.4	100.0	98.7	800.1	1487.3	632.3	974.1
TEASER++	98.0	100.0	100.0	98.9	534.3	574.1	367.4	519.8
QUATRO	99.0	98.1	95.4	98.2	345.2	359.6	242.1	333.3
MAC	97.1	100.0	100.0	98.4	1967.2	2053.1	1433.8	1908.7
<i>Learning-based methods</i>								
PointDSC	96.1	100.0	98.9	97.7	441.6	457.0	303.5	424.5
VBReg	96.7	98.8	97.7	97.5	514.1	538.2	356.1	496.4
PREDATOR	99.7	100.0	100.0	99.8	564.5	579.2	465.7	553.3
<i>Segment- and plane-based methods</i>								
VPFBR	34.5	30.2	24.1	31.7	186.7	143.1	125.6	165.5
PLADE	2.0	0.0	0.0	1.1	799.9	730.6	691.1	762.7
G3Reg	99.0	98.8	100.0	99.1	90.1	85.1	85.9	88.0
<i>Ours</i>								
PARTE	99.7	100.0	100.0	99.8	182.9	188.3	141.5	178.0

in addition to the overall result.

While several methods perform nearly perfectly at short range, their success rates decrease substantially as the displacement increases. PARTE remains among the strongest methods across the sequences and distance ranges, with particularly strong performance in the short- and medium-range cases. At 20–30 m, G3Reg achieves the highest success rate on four of the five sequences. This is a favorable setting for G3Reg, whose frontend is specifically designed for outdoor LiDAR registration and exploits planes, lines, and object clusters that are common in structured urban environments [26].

Partial-to-full registration. We additionally evaluate registration on the RESSO partial-to-full benchmark, where a local RGB-D scan is aligned to a larger static reference scan. This setting is challenging because the two point clouds can have substantially lower overlap and different scene context.

Table IX reports the results across the seven evaluated scenes. PARTE achieves the highest overall success rate of 59.6%, followed by CLIPPER+ at 58.6%, MAC and VBReg at 57.6%, and TEASER++ at 56.6%. The smaller margin compared with the main indoor and outdoor benchmarks is consistent with the plane-matching results in Table IV. In partial-to-full registration, the context surrounding a corresponding plane can change substantially between scans, reducing the discriminative advantage of PCH as is evident in Table IV. PARTE fails on 40 of the 99 registrations, of which 28 are also missed by every other method in Table IX. Among these common failures, 25 contain fewer than five geometrically consistent FPFH correspondences and 13 contain none. Of the remaining PARTE failures, several result from an incorrect plane match that biases the selected clique toward outliers.

These cases expose a limitation of the joint point–plane formulation: confidence-weighted plane correspondences improve robustness when reliable, but incorrect associations

TABLE VIII. KITTI-LC registration performance across sequences and translation ranges. We report success rate under the $2 \text{ m}/5^\circ$ criterion and mean end-to-end runtime over all evaluated pairs.

Method	0–10 m					10–20 m					20–30 m					All					Runtime (ms)
	00	02	05	06	08	00	02	05	06	08	00	02	05	06	08	00	02	05	06	08	
<i>Baseline</i>																					
RANSAC-10 ³	45.9	44.8	31.8	38.8	17.2	3.8	3.5	2.0	1.6	0.6	0.0	0.0	0.0	0.0	0.0	14.8	14.2	10.1	8.5	5.0	360.1
RANSAC-10 ⁵	99.3	90.7	99.4	100.0	94.0	47.5	41.4	38.7	46.6	32.7	2.9	7.2	2.3	9.9	2.2	46.2	42.8	43.3	42.9	38.0	376.9
RANSAC-10 ⁷	100.0	90.7	100.0	100.0	100.0	71.0	60.1	71.4	95.2	80.2	16.5	26.9	13.7	47.6	9.2	59.3	56.5	58.6	77.2	58.4	1073.8
FGR	59.4	39.0	49.4	69.8	3.7	3.5	1.5	0.5	0.8	0.0	0.0	0.0	0.0	0.0	0.0	18.7	11.8	14.8	14.6	1.0	410.9
<i>Robust point-based methods</i>																					
KISS-Matcher	100.0	88.4	99.4	100.0	100.0	58.7	55.1	58.8	71.7	71.0	8.9	10.8	6.4	14.7	5.4	52.4	48.1	51.5	54.7	53.8	101.1
CLIPPER+	100.0	92.4	100.0	100.0	100.0	78.0	57.1	79.9	98.0	88.9	18.6	22.0	14.2	61.9	17.8	62.4	54.1	61.6	84.0	64.7	2574.9
TEASER++	100.0	93.0	100.0	100.0	100.0	81.5	61.6	86.4	99.2	95.7	25.7	30.9	21.0	75.4	26.5	66.2	59.2	66.3	89.9	70.3	494.7
QUATRO	99.0	100.0	100.0	100.0	100.0	93.8	81.8	95.5	98.8	98.1	52.2	46.6	52.1	94.0	61.1	79.9	73.9	80.8	97.2	84.4	304.3
MAC	100.0	91.3	100.0	100.0	100.0	77.1	60.1	78.9	96.8	88.3	21.5	31.8	19.2	57.9	17.3	63.2	58.5	63.1	82.0	64.2	1721.4
<i>Learning-based methods</i>																					
PointDSC	100.0	89.5	100.0	100.0	100.0	65.1	47.5	66.3	92.8	67.9	9.4	12.1	9.6	37.3	5.4	54.7	46.4	55.4	72.2	52.8	414.3
VBReg	100.0	87.8	100.0	100.0	100.0	78.3	54.0	81.4	96.4	81.5	21.8	24.7	15.5	40.9	13.0	63.7	52.8	62.6	75.0	60.3	465.1
PREDATOR	71.0	61.6	75.6	98.4	73.1	31.7	21.2	28.1	17.9	14.2	3.1	1.8	1.8	1.6	0.0	32.7	25.6	32.5	27.8	25.2	366.3
<i>Segment- and plane-based methods</i>																					
VPFBR	90.1	37.2	81.8	48.1	46.3	65.7	10.1	52.3	24.3	30.9	24.7	4.9	23.3	7.9	18.4	57.7	16.0	50.3	22.6	30.4	154.7
PLADE	17.8	0.6	4.5	7.8	0.7	2.9	0.5	0.5	1.2	0.6	2.4	0.0	0.0	0.4	0.0	7.1	0.3	1.5	2.2	0.4	878.0
G3Reg	100.0	91.9	100.0	97.7	100.0	96.8	67.2	97.5	100.0	98.8	58.3	52.0	73.5	99.2	76.8	83.4	68.6	89.4	99.2	90.6	96.5
<i>Ours</i>																					
PARTE	100.0	100.0	100.0	100.0	100.0	93.0	85.9	97.0	100.0	98.1	56.2	55.2	57.5	96.8	68.1	81.4	78.4	83.3	98.7	87.1	127.1

can reinforce an erroneous consensus. Nevertheless, PARTE achieves the highest overall success rate and succeeds on six pairs where both VBReg and TEASER++ fail.

TABLE IX. Partial-to-full registration performance on the seven evaluated RESSO scenes. We report success rate for each scene and overall, together with mean end-to-end runtime in milliseconds.

Method	Success Rate (%)								Time
	6a	6b	6c	6d	6f	6g	6h	All	
<i>Baseline</i>									
RANSAC-10 ³	0.0	15.8	0.0	13.3	10.0	0.0	0.0	6.1	348.8
RANSAC-10 ⁵	23.1	47.4	36.8	53.3	30.0	0.0	28.6	34.3	358.1
RANSAC-10 ⁷	76.9	47.4	42.1	73.3	80.0	0.0	57.1	54.5	727.1
FGR	23.1	21.1	31.6	53.3	20.0	0.0	28.6	27.3	362.7
<i>Robust point-based methods</i>									
KISS-Matcher	61.5	31.6	31.6	73.3	60.0	11.1	28.6	42.4	<u>213.5</u>
CLIPPER+	76.9	<u>52.6</u>	42.1	80.0	<u>70.0</u>	<u>33.3</u>	57.1	<u>58.6</u>	707.9
TEASER++	<u>69.2</u>	47.4	47.4	86.7	<u>70.0</u>	22.2	<u>50.0</u>	56.6	385.2
QUATRO	0.0	0.0	47.4	6.7	0.0	0.0	0.0	10.1	385.1
MAC	76.9	<u>52.6</u>	<u>42.1</u>	<u>80.0</u>	80.0	11.1	57.1	57.6	549.9
<i>Learning-based methods</i>									
PointDSC	53.8	47.4	36.8	73.3	60.0	11.1	42.9	47.5	380.3
VBReg	76.9	<u>52.6</u>	42.1	<u>80.0</u>	80.0	11.1	57.1	57.6	443.7
PREDATOR	0.0	21.1	26.3	6.7	0.0	0.0	0.0	10.1	392.3
<i>Segment- and plane-based methods</i>									
VPFB	30.8	0.0	5.3	20.0	0.0	11.1	7.1	10.1	166.3
PLADE	0.0	0.0	10.5	13.3	0.0	11.1	0.0	5.1	6617.7
<i>Ours</i>									
PARTE	76.9	57.9	<u>42.1</u>	66.7	<u>70.0</u>	55.6	57.1	59.6	315.8

Additional environments. We finally evaluate on ETH, which contains comparatively few planar patches and often only a dominant ground plane. As in the indoor experiments, we find that the orientation of the normals used by the FPFH frontend has a substantial effect on the reported registration rate. For example, TEASER++ improves from 72.7% to 94.0%

when FPFH is computed using consistently oriented normals. Similarly, MAC reports a registration recall of 36.1% with FPFH on ETH [73]; in our evaluation this increases to 57.7% with MNN matching and further to 87.2% when the FPFH normals are consistently oriented. This makes ETH considerably less challenging once a stronger point frontend is used.

TABLE X. Registration performance on the ETH benchmark. Success rate is reported for each sequence and overall, together with mean end-to-end runtime. MAC[†] encountered out-of-memory failures on 17 of the 713 registration pairs; these cases are counted as failures.

Method	Success Rate (%)					Time
	Gazebo Summer	Gazebo Winter	Wood Autumn	Wood Summer	All	
<i>Baseline</i>						
RANSAC-10 ³	26.6	9.0	3.5	7.2	12.3	872.4
RANSAC-10 ⁵	78.3	54.3	65.2	60.0	63.3	892.9
RANSAC-10 ⁷	98.4	77.5	95.7	96.0	89.1	1373.7
FGR	60.9	36.7	71.3	74.4	55.1	935.1
<i>Robust point-based methods</i>						
KISS-Matcher	95.1	82.4	95.7	93.6	89.8	226.7
CLIPPER+	99.5	89.3	99.1	95.2	94.5	4266.4
TEASER++	100.0	86.5	98.3	98.4	94.0	1078.3
QUATRO	87.5	95.2	68.7	81.6	86.5	1479.3
MAC [†]	98.4	76.5	93.9	89.6	87.2	7521.2
<i>Learning-based methods</i>						
PointDSC	91.8	79.2	81.7	76.8	82.5	905.0
VBReg	96.7	89.3	89.6	84.8	90.5	1015.4
PREDATOR	53.3	64.0	71.3	72.8	64.0	1657.8
<i>Segment- and plane-based methods</i>						
VPFBR	33.7	14.9	4.3	5.6	16.4	125.9
PLADE	4.9	5.9	2.6	3.2	4.6	708.9
G3Reg	29.9	20.1	47.8	52.0	32.7	161.9
<i>Ours</i>						
PARTE	100.0	99.0	100.0	100.0	99.6	437.6

Although ETH is largely unstructured and often contains only the ground plane as a reliable planar feature, the available plane correspondences still provide useful complementary constraints. PARTE recovers 40 of the 43 registrations missed by TEASER++, with plane correspondences participating in 39 of these recoveries. The ground plane is selected in all 39 cases, while only 12 contain an additional plane correspondence. This shows that even a single reliable plane can provide useful complementary evidence when the point correspondences alone are insufficient.

Gazebo Winter provides an especially interesting example. The relative motion is close to gravity-aligned, with a maximum tilt of 5.48° , making it well suited to the reduced rotational model exploited by QUATRO [16]. PARTE recovers 95 of the 96 registrations missed by QUATRO. In 92 of these cases, QUATRO’s point-only maximum clique contains incorrect correspondences, with outliers forming the majority in 36 cases. PARTE uses the matched ground plane directly in the compatibility graph, where plane–point consistency removes point correspondences that disagree with the shared plane before transformation estimation. Therefore, the ground plane provides information during correspondence selection, rather than only constraining the final pose.

Ablation studies of our algorithm for planar-patch extraction, plane weight selection, and runtime are presented in Appendices A and B. Fig. 10 summarizes performance across the four main benchmarks. **PARTE achieves the highest mean success rate while maintaining low runtime, showing consistently strong performance across both indoor and outdoor registration problems.**

D. Limitations

The experiments show that reliable plane correspondences can substantially improve global registration, particularly when the point correspondence set contains few inliers. At the same time, PARTE remains fundamentally a mixed point–plane method, and its behavior approaches that of the underlying point pipeline when few useful planes are available. As discussed in Appendix B, point descriptor computation and matching also remain the main computational bottleneck.

A second limitation is sensitivity to changes in the geometry and density surrounding a plane. This is most evident on RESSO in Table IV, where partial-to-full registration can substantially change the context observed around corresponding planes and reduce the effectiveness of PCH. Large density variations, particularly in LiDAR, can produce a similar effect.

VI. CONCLUSION

We presented PARTE, a global point-cloud registration method that complements conventional point features with explicitly extracted and matched planar patches. Our Plane Context Histogram (PCH) algorithm describes each plane using the surrounding geometry, enabling reliable plane correspondences when the patch itself is ambiguous. These correspondences are combined with point matches during outlier rejection and transformation estimation, while the method naturally reduces to point-only registration when useful planes

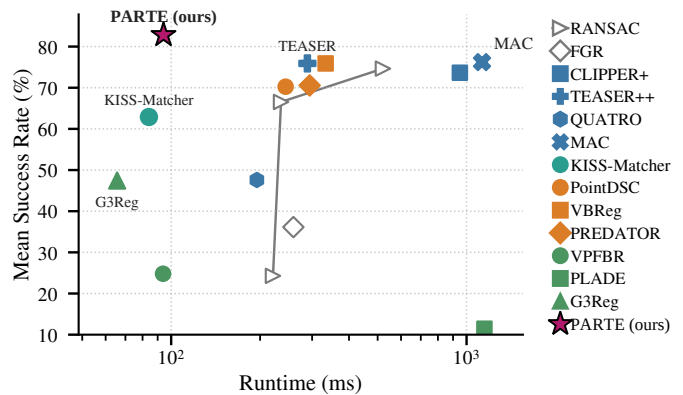


Fig. 10. Overall registration performance across 3DMatch, 3DLoMatch, KITTI-10m, and KITTI-LC. Each point reports the mean success rate and runtime across the four benchmarks, with each benchmark weighted equally.

are unavailable. Across diverse indoor RGB-D and outdoor LiDAR benchmarks, PARTE achieves consistently strong registration performance and the best overall success rate among the evaluated methods.

REFERENCES

- [1] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2241–2254, 2016.
- [2] I. Sipiran and B. Bustos, “Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes,” *The Visual Computer*, vol. 27, pp. 963–976, 2011.
- [3] Y. Zhong, “Intrinsic shape signatures: A shape descriptor for 3d object recognition,” in *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*. IEEE, 2009, pp. 689–696.
- [4] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “Narf: 3d range image features for object recognition,” in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [5] J. Li and G. H. Lee, “USIP: Unsupervised stable interest point detection from 3D point clouds,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 361–370.
- [6] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6359–6367.
- [7] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.
- [8] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in *Computer Vision—ECCV 2010*. Springer, 2010, pp. 356–369.
- [9] A. E. Johnson, “Spin-images: A representation for 3-D surface matching,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, August 1997.
- [10] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein, “3d shape matching with 3d shape contexts,” in *The 7th central European seminar on computer graphics*, vol. 3. Budmerice Slovakia, 2003, pp. 5–17.
- [11] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3DMatch: Learning local geometric descriptors from RGB-D reconstructions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1802–1811.
- [12] C. Choy, J. Park, and V. Koltun, “Fully convolutional geometric features,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8958–8966.
- [13] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, “PREDATOR: Registration of 3D point clouds with low overlap,” in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2021, pp. 4267–4276.

- [14] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "SpinNet: Learning a general surface descriptor for 3D point cloud registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 753–11 762.
- [15] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, S. Ilic, D. Hu, and K. Xu, "GeoTransformer: Fast and robust point cloud registration with geometric transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 9806–9821, 2023.
- [16] H. Lim, B. Kim, D. Kim, E. Mason Lee, and H. Myung, "Quatro++: Robust global registration exploiting ground segmentation for loop closing in LiDAR SLAM," *The International Journal of Robotics Research*, vol. 43, no. 5, pp. 685–715, 2024.
- [17] H. Lim, D. Kim, G. Shin, J. Shi, I. Vizzo, H. Myung, J. Park, and L. Carlone, "KISS-Matcher: Fast and robust point cloud registration revisited," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 11 104–11 111.
- [18] O. D. Faugeras and M. Hebert, "A 3-d recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 2*, 1983, pp. 996–1002.
- [19] —, "The representation, recognition, and locating of 3-d objects," *The international journal of robotics research*, vol. 5, no. 3, pp. 27–52, 1986.
- [20] C. Brenner and C. Dold, "Automatic relative orientation of terrestrial laser scans using planar structures and angle constraints," in *ISPRS Workshop on Laser Scanning*, vol. 200, 2007, pp. 84–89.
- [21] W. Von Hansen, H. Gross, and U. Thoennessen, "Line-based registration of terrestrial and airborne lidar data," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 37, no. B3b, pp. 161–166, 2008.
- [22] W. Tao, X. Hua, Z. Chen, and P. Tian, "Fast and automatic registration of terrestrial point clouds using 2d line features," *Remote Sensing*, vol. 12, no. 8, p. 1283, 2020.
- [23] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [24] H. Kim, S. Song, and H. Myung, "Gp-icp: Ground plane icp for mobile robots," *IEEE Access*, vol. 7, pp. 76 599–76 610, 2019.
- [25] K. Favre, M. Pressigout, E. Marchand, and L. Morin, "A plane-based approach for indoor point clouds registration," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 7072–7079.
- [26] Z. Qiao, Z. Yu, B. Jiang, H. Yin, and S. Shen, "G3Reg: Pyramid graph-based global registration using gaussian ellipsoid model," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 3416–3432, 2025.
- [27] J. Wu, H. Yu, S. Han, X.-M. Cai, M.-F. Wang, W. Yang, and G.-S. Xia, "QuadraticsReg: Large-scale point cloud registration using semantic quadric primitives," *IEEE Transactions on Robotics*, vol. 42, pp. 1961–1981, 2026.
- [28] A. Zaganidis, M. Magnusson, T. Duckett, and G. Cielniak, "Semantic-assisted 3d normal distributions transform for scan registration in environments with limited structure," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4064–4069.
- [29] J. M. Wong, V. Kee, T. Le, S. Wagner, G.-L. Mariottini, A. Schneider, L. Hamilton, R. Chipkatty, M. Hebert, D. M. S. Johnson, J. Wu, B. Zhou, and A. Torralba, "SegICP: Integrated deep semantic segmentation and pose estimation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5784–5789.
- [30] G. Mei, X. Huang, J. Zhang, and Q. Wu, "Partial point cloud registration via soft segmentation," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 681–685.
- [31] S. Fung, X. Lu, D. d. S. Edirimuni, W. Pan, X. Liu, and H. Li, "SemReg: Semantics constrained point cloud registration," in *European Conference on Computer Vision*. Springer, 2024, pp. 293–310.
- [32] H. Qin, Y. Zhou, C. Liu, X. Zhang, Z. Cheng, and J. Guo, "SarNet: Semantic augmented registration of large-scale urban point clouds," in *International Conference on Computational Visual Media*. Springer, 2024, pp. 152–174.
- [33] P. Kim, B. Coltin, and H. J. Kim, "Linear RGB-D SLAM for planar environments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 350–366.
- [34] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "RGB-D SLAM with structural regularities," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 581–11 587.
- [35] D. Chen, S. Wang, W. Xie, S. Zhai, N. Wang, H. Bao, and G. Zhang, "VIP-SLAM: An efficient tightly-coupled RGB-D visual inertial planar SLAM," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5615–5621.
- [36] P. Wei, L. Yan, H. Xie, and M. Huang, "Automatic coarse registration of point clouds using plane contour shape descriptor and topological graph voting," *Automation in Construction*, vol. 134, p. 104055, 2022.
- [37] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "PLADE: A plane-based descriptor for point cloud registration with small overlap," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 4, pp. 2530–2540, 2020.
- [38] A. Kaiser, J. A. Y. Zepeda, and T. Boubekeur, "Plane pair matching for efficient 3D view registration," *arXiv preprint arXiv:2001.07058*, 2020.
- [39] Y. Shi, K. Xu, M. Nießner, S. Rusinkiewicz, and T. Funkhouser, "PlaneMatch: Patch coplanarity prediction for robust RGB-D reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 767–784.
- [40] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2011, pp. 2987–2992.
- [41] J. P. Silva do Monte Lima and V. Teichrieb, "An efficient global point cloud descriptor for object recognition and pose estimation," in *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2016, pp. 56–63.
- [42] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 5266–5272.
- [43] 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, Jun. 1981. [Online]. Available: <https://doi.org/10.1145/358669.358692>
- [44] O. Chum and J. Matas, "Matching with PROSAC—progressive sample consensus," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 220–226.
- [45] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Joint pattern recognition symposium*. Springer, 2003, pp. 236–243.
- [46] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendiadua, and B. Sierra, "RANSAC for robotic applications: A survey," *Sensors*, vol. 23, no. 1, p. 327, 2023.
- [47] E. Hu and L. Sun, "Daniel: A fast and robust consensus maximization method for point cloud registration with high outlier ratios," *Information Sciences*, vol. 614, pp. 563–579, 2022.
- [48] L. Sun, "Ransic: Fast and highly robust estimation for rotation search and point cloud registration using invariant compatibility," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 143–150, 2022.
- [49] P. Shi, S. Yan, Y. Xiao, X. Liu, Y. Zhang, and J. Li, "Ransac back to sota: A two-stage consensus filtering for real-time 3d registration," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 881–11 888, 2024.
- [50] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 766–782.
- [51] N. Chebrolu, T. Labe, O. Vysotska, J. Behley, and C. Stachniss, "Adaptive robust kernels for non-linear least squares problems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2240–2247, 2021.
- [52] 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 Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.
- [53] B.-S. Chen, Y.-K. Lin, J.-Y. Chen, C.-W. Huang, J.-L. Chern, and C.-C. Sun, "Fracgm: A fast fractional programming technique for gemanclure robust estimator," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 666–11 673, 2024.
- [54] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2021.
- [55] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7193–7203.
- [56] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, and C.-L. Tai, "PointDSC: Robust point cloud registration using deep spatial consistency," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 859–15 869.

- [57] H. Jiang, Z. Dang, Z. Wei, J. Xie, J. Yang, and M. Salzmann, “Robust outlier rejection for 3D registration with variational bayes,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 1148–1157.
- [58] S. Ao, Q. Hu, H. Wang, K. Xu, and Y. Guo, “BUFFER: Balancing accuracy, efficiency, and generalizability in point cloud registration,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 1255–1264.
- [59] J. Shi, H. Yang, and L. Carlone, “Robin: a graph-theoretic approach to reject outliers in robust estimation using invariants,” in *2021 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 13 820–13 827.
- [60] P. C. Lusk, K. Fathian, and J. P. How, “CLIPPER: A graph-theoretic framework for robust data association,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 828–13 834.
- [61] Z. Chen, K. Sun, F. Yang, and W. Tao, “SC²-PCR: A second order spatial compatibility for efficient and robust point cloud registration,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 221–13 231.
- [62] K. Fathian and T. Summers, “CLIPPER+: A fast maximal clique algorithm for robust global registration,” *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3562–3569, 2024.
- [63] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-d point sets,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.
- [64] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.
- [65] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, “Challenging data sets for point cloud registration algorithms,” *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1705–1711, 2012.
- [66] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, “Unsupervised discovery of object classes from range data using latent dirichlet allocation,” in *Robotics: Science and Systems V*, 2009.
- [67] F. Tombari, S. Salti, and L. Di Stefano, “Unique shape context for 3d data description,” in *Proceedings of the ACM workshop on 3D object retrieval*, 2010, pp. 57–62.
- [68] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3D recognition and pose using the viewpoint feature histogram,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2155–2162.
- [69] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. R. Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues,” in *2011 IEEE International Conference on Computer Vision Workshops*. IEEE, 2011, pp. 585–592.
- [70] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, “OUR-CVFH: Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science, vol. 7476. Springer, 2012, pp. 113–122.
- [71] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, “Combined 2D–3D categorization and classification for multimodal perception systems,” *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1378–1402, 2011.
- [72] A. Kanezaki, Z.-C. Marton, D. Pangercic, T. Harada, Y. Kuniyoshi, and M. Beetz, “Voxelized shape and color histograms for RGB-D,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Active Semantic Perception and Object Search in the Real World*, San Francisco, CA, USA, 2011.
- [73] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, “3D registration with maximal cliques,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 17 745–17 754.
- [74] H. Lim, S. Yeon, S. Ryu, Y. Lee, Y. Kim, J. Yun, E. Jung, D. Lee, and H. Myung, “A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8010–8017.
- [75] J. Li, J. Zhan, T. Zhou, V. A. Bento, and Q. Wang, “Point cloud registration and localization based on voxel plane features,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 188, pp. 363–379, 2022.
- [76] M. Oh, E. Jung, H. Lim, W. Song, S. Hu, E. M. Lee, J. Park, J. Kim, J. Lee, and H. Myung, “TRAVEL: Traversable ground and above-ground object segmentation using graph representation of 3D LiDAR scans,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7255–7262, 2022.
- [77] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, “Segmentation of point clouds using smoothness constraint,” *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [78] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for point-cloud shape detection,” *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [79] A. M. C. Araújo and M. M. Oliveira, “A robust statistics approach for plane detection in unorganized point clouds,” *Pattern Recognition*, vol. 100, p. 107115, 2020.

APPENDIX A ABLATION STUDIES

Plane extraction. The plane-extraction method introduced in Sec. IV-A is designed specifically for the mixed point–plane registration pipeline rather than for maximizing segmentation quality. During initialization, points whose neighborhoods do not satisfy the planarity conditions are excluded from region growing and remain available as point features. As illustrated in Fig. 11, this produces conservative planar patches while preserving edges, corners, and other non-planar regions.

This early rejection also reduces the segmentation workload. On average, the initialization test excludes approximately 40% of the points from region growing on 3DMatch, 50% on KITTI-10m, and 65% on ETH. Fig. 12 shows that these rejected regions are also more useful for point matching: the PPFH inlier ratio generally increases with normal dispersion. Therefore, the same test both avoids absorbing distinctive point features into planes and reduces unnecessary region-growing operations. Table XI compares several alternative plane-extraction backends within the same registration pipeline.

TABLE XI. Effect of the plane-extraction backend on registration performance. SR reports the success rate of the complete PARTE pipeline, while Time reports the mean plane-extraction time per scan.

Plane extraction	3DMatch		3DLoMatch		KITTI-10m	
	SR (%)	Time (ms)	SR (%)	Time (ms)	SR (%)	Time (ms)
PCL RANSAC [43]	78.5	254.4	27.4	251.3	90.3	860.9
PCL region growing [77]	76.8	18.9	26.9	18.7	97.7	43.1
Efficient RANSAC [78]	85.8	13.3	43.2	13.3	90.6	54.4
Robust Planar Patches [79]	75.2	14.2	26.2	14.3	89.9	24.5
GEM frontend [26]	83.4	2.9	39.2	2.8	99.8	6.8
PARTE	89.4	7.5	56.0	7.4	99.8	12.0

PARTE provides the strongest registration performance while remaining faster than most reliable segmentation baselines. For completeness, the reported PARTE segmentation time includes KNN construction, although these neighborhoods are reused elsewhere in the pipeline.

Weighted correspondence selection. We next evaluate how plane correspondences should contribute during clique selection. We compare the standard unweighted maximum clique against two weighted variants. The first assigns the same weight to every plane correspondence, while the second scales each plane weight using its PCH confidence w_j . In both cases, a weight multiplier controls the overall influence of plane correspondences relative to unit-weight point matches.

Fig. 13 shows the effect of varying the weight multiplier on 3DMatch and KITTI-LC. Confidence-based weighting is considerably less sensitive to the multiplier, maintaining strong performance over a broader range. This supports using the PCH confidence to distinguish reliable plane matches from ambiguous ones during clique selection.

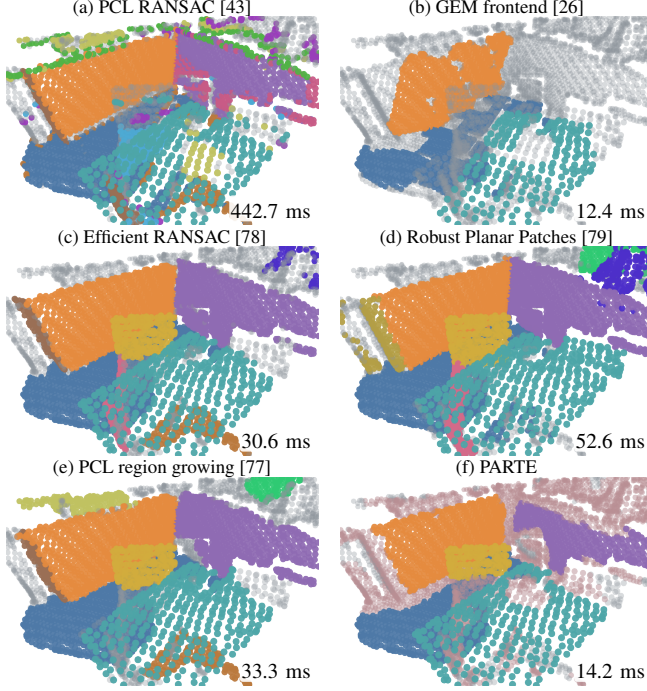


Fig. 11. Comparison of plane-extraction methods on a representative scene from the RedKitchen dataset. Colored regions denote extracted planar patches. In PARTE, red points fail the initial planarity test and are excluded from region growing, leaving them available for point-feature matching. Reported times show plane-extraction runtime; the PARTE timing also includes KNN neighborhood search.

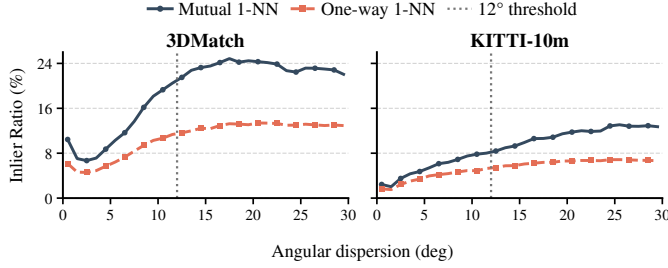


Fig. 12. FPFH inlier ratio as a function of local normal dispersion on 3DMatch and KITTI-10m. Points with larger normal dispersion are more likely to produce correct FPFH correspondences, supporting the 12° threshold used to exclude such regions from plane growing.

APPENDIX B RUNTIME ANALYSIS

To understand where PARTE spends computation, we compare its runtime breakdown against a point-only variant. The point-only variant uses the same downsampling, normal estimation, FPFH matching, and robust registration pipeline, but does not extract or match planes. Instead, FPFH descriptors are computed over the full downsampled point cloud. Fig. 14 shows the average runtime on 3DMatch and KITTI-10m under both multithreaded and sequential execution. Overall, separating planar points and matching them as patches can substantially reduce the total runtime by lowering the cost of point descriptor computation, matching, and outlier rejection. Nevertheless, PARTE’s runtime remains dominated by the conventional point-matching stage.

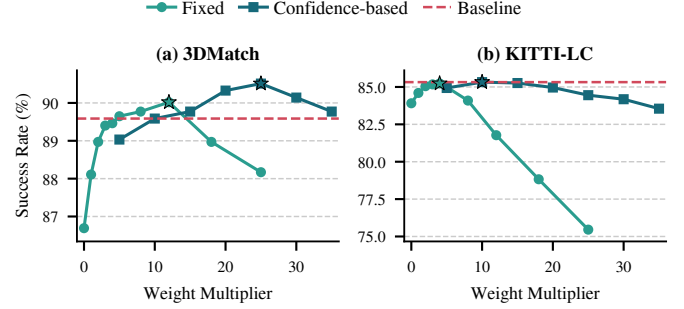


Fig. 13. Effect of plane-correspondence weighting on registration success. Fixed weighting assigns the same weight to all plane matches, while confidence-based weighting scales their contribution using PCH confidence.

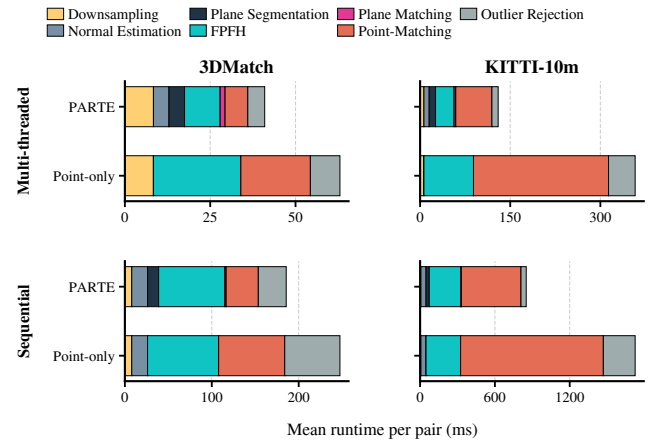


Fig. 14. Runtime breakdown of PARTE and the point-only variant on 3DMatch and KITTI-10m under multithreaded and sequential execution. Plane extraction and matching add overhead, which is offset by lower point-feature, matching, and outlier-rejection costs.