A New Guaranteed Outlier Removal Method Based on Plane Constraints for Large-Scale LiDAR Point Cloud Registration

Gang Ma, Hui Wei*, Runfeng Lin, and Jialiang Wu
School of Computer Science, Fudan University, China
{mag20,weihui}@fudan.edu.cn, {rflin22,wujl23}@m.fudan.edu.cn

Abstract

In this paper, we present a novel registration method based on plane constraints for large-scale LiDAR point clouds, effectively decoupling rotation estimation and translation estimation. For rotation estimation, we propose an outlier removal method that combines coarse filtering with rotation-invariant constraints and refined filtering based on computational geometric consistency checks, effectively pruning outliers and robustly estimating accurate relative rotations from plane normals. In translation estimation, we propose a component-wise method based on plane translation constraints to efficiently estimate relative translations. The robustness and effectiveness of our proposed method are empirically validated on three popular LiDAR point cloud datasets. The experimental results convincingly demonstrate that our approach achieves state-of-the-art performance.

1 Introduction

Rapid advancements in 3D laser scanning and computer technology have made 3D point cloud registration a pivotal endeavor in computer vision and robotics, with its significance extending across various domains such as photogrammetry [Li *et al.*, 2022], 3D scene reconstruction [Liu *et al.*, 2023b], autonomous navigation [Lu *et al.*, 2019], and virtual reality [Reyes-Aviles *et al.*, 2022]. At its essence, point cloud registration involves the computation of a precise 6-Degree-of-Freedom (DoF) transformation $\boldsymbol{T} = [\boldsymbol{R}|\boldsymbol{t}] \in SE(3)$, serving the vital function of meticulously aligning the source point cloud $\mathcal{P}$ with the coordinate system of the target point cloud $\mathcal{Q}$ [Yan *et al.*, 2022; Chen *et al.*, 2022].

Accordingly, point cloud registration methods are primarily categorized into two groups: local-based and global-based methods [Ma and Wei, 2023]. Among these, the Iterative Closest Point (ICP) [Besl and McKay, 1992], a well-known local-based method with several variants [Segal *et al.*, 2009; Bouaziz *et al.*, 2013; Zhang *et al.*, 2021; Koide *et al.*, 2021], has had a profound influence on subsequent research in this field. Nevertheless, ICP and its variants employ a greedy and exhaustive nearest neighbor (NN) search during each iteration to establish point correspondences [Lim *et al.*, 2022]. Consequently, ICP-variants heavily depend on the initial alignment and is susceptible to becoming trapped in local minima. Local registration methods are most effective when the source point cloud and the target point cloud are in close proximity, nearly overlapping [Pomerleau *et al.*, 2013].

In response to the limitations of ICP and its variants, global-based registration methods have gained attention for providing an initial alignment that aids subsequent local registration, guiding estimations towards the global minimum [Lim *et al.*, 2022]. This paper mainly focuses on correspondence-based global registration methods, categorizing them into point and primitive correspondence-based approaches. Point correspondence-based methods typically involve two essential stages: establishing correspondences based on local point features and estimating transformation from identified correspondences [Li *et al.*, 2023]. During establishing point correspondences, it is inevitable that some noise correspondences are introduced. When dealing with noisy correspondences, a commonly employed strategy is to utilize “correspondences + RANSAC (Random Sample Consensus) [Fischler and Bolles, 1981]” [Barath *et al.*, 2020; Barath and Matas, 2018]. Although RANSAC-based methods are renowned for their simplicity and efficiency, it is important to note that their runtime exponentially escalates as the outlier rates increase. Another widely recognized strategy is “correspondences + outlier removal” [Bustos and Chin, 2017; Yang *et al.*, 2020; Yan *et al.*, 2022], which effectively filters out outlier correspondences. Nonetheless, methods relying on point features encounter challenges such as longer computation times, sensitivity to point density, and reduced descriptor discriminability due to different viewpoints.

Recently, there has been growing interest in the application of learning-based techniques in point cloud registration. On one hand, specific deep learning approaches have been dedicated to enhancing distinct components of the registration process, including the development of more discriminative point feature descriptors [Huang *et al.*, 2021; Ao *et al.*, 2021; Wang *et al.*, 2022; Wang *et al.*, 2023] and more efficient correspondence selection methods [Aoki *et al.*, 2019; Bai *et al.*, 2021; Fu *et al.*, 2021; Xue *et al.*, 2024]. On the other hand, [Li and Harada, 2022; Yew and Lee, 2022; Qin *et al.*, 2022; Liu *et al.*, 2023a] have showcased end-to-end

*Corresponding author

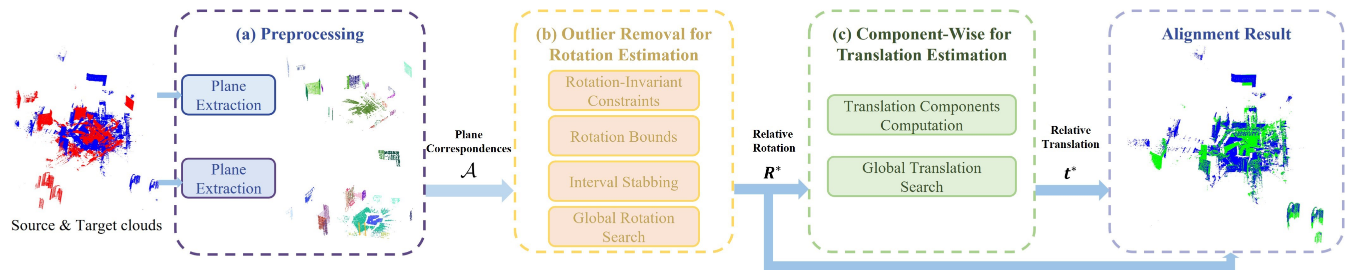


Figure 1: Pipeline of our novel framework for point cloud registration, decoupling rotation estimation and translation estimation effectively.

point cloud registration methods. However, learning-based approaches require substantial training datasets and continue to face challenges related to generalization, especially when dealing with outliers and cross-dataset conditions. When handling large-scale outdoor environments, learning-based methods face limitations due to small trainable datasets and the large-scale of point cloud data, hindering their full potential in point cloud registration tasks.

Unlike point-based methods, primitive-based registration approaches leverage higher-level geometric features in scenes, such as lines [Yan *et al.*, 2023; Liu *et al.*, 2023c], planes [Chen *et al.*, 2019; Li *et al.*, 2022; Ma *et al.*, 2023], and spheres [Zhang *et al.*, 2022], making them better suited to handle challenges like uneven point density, occlusions, and noise in point clouds. While existing primitive-based registration methods [Chen *et al.*, 2019; Li *et al.*, 2022] have showcased remarkable performance on large-scale LiDAR point clouds, it is important to note that primitive descriptors often exhibit lower discriminability compared to point descriptors. This reduced discriminability can lead to a higher likelihood of introducing outlier correspondences during the establishment of primitive correspondences. Unfortunately, current primitive-based algorithms lack effective mechanisms for filtering out these outlier correspondences, which in turn restricts their precision in point cloud registration tasks.

In this paper, we present an efficient framework that decouples rotation estimation and translation estimation. For rotation estimation, our method introduces a pioneering outlier removal method based on plane primitives. For translation estimation, we propose an innovative approach for estimating relative translation using plane translation components.

In summary, the contributions of our work are as follows:

1. We present a novel point cloud registration framework grounded in plane constraints, effectively decoupling rotation estimation and translation estimation.
2. We propose a guaranteed outlier removal method for rotation estimation. This method combines rotation-invariant constraints with computational geometric consistency checks for outlier removal and globally searches for the optimal relative rotation.
3. We propose a component-wise method for translation estimation. This approach efficiently estimates each translation component through plane translation constraints and globally estimates the optimal relative translation.
4. Experimental results on multiple datasets convincingly demonstrate the robustness, effectiveness, and efficiency of our proposed framework, achieving state-of-the-art performance.

2 Methodology

First, we represent the source and target point clouds captured by a 3D LiDAR sensor as $\mathcal{P}$ and $\mathcal{Q}$, respectively. Drawing inspiration from [Li *et al.*, 2022], we employ a voxel-based plane extraction method to obtain planes from $\mathcal{P}$ and $\mathcal{Q}$. Let us define $\mathcal{P} = \{P_1, P_2, \cdots, P_M\}$ and $\mathcal{Q} = \{Q_1, Q_2, \cdots, Q_N\}$, where each plane $P_i = (\boldsymbol{x}_i, \boldsymbol{p}_i)(1 \leq i \leq M)$ from $\mathcal{P}$ and each plane $Q_j = (\boldsymbol{y}_j, \boldsymbol{q}_j)(1 \leq j \leq N)$ from $\mathcal{Q}$ respectively contain normals $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, along with points $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$ on their respective planes. Next, let $(P_k, Q_k) \in \mathcal{A}(1 \leq k \leq M * N)$ represent candidate plane correspondences, where $P_k \in \mathcal{P}$ and $Q_k \in \mathcal{Q}$. While it is inevitable that the candidate set $\mathcal{A}$ contains some noisy correspondences, our goal is to minimize the impact of these outliers and accurately estimate the relative rotation $\boldsymbol{R} \in SO(3)$ and relative translation $\boldsymbol{t} \in \mathbb{R}^3$ from candidate set $\mathcal{A}$.

As shown in Figure 1, after extracting planes from the source cloud $\mathcal{P}$ and target cloud $\mathcal{Q}$, along with the establishment of plane correspondences $\mathcal{A}$, we propose a robust decoupled point cloud registration method that effectively decomposes rotation estimation and translation estimation.

2.1 Outlier Removal for Rotation Estimation

In this section, we present a guaranteed outlier removal method for estimating relative rotation from plane normals. For each candidate normal correspondence $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ from (P_k, Q_k), we initially apply coarse filtering to remove outliers based on rotation-invariant constraints. Subsequently, we propose a strategy based on computational geometric consistency checks to finely prune outliers while calculating rotation bounds for each inlier $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ from (P_i, Q_i) $(i \neq k)$ to the candidate $(\boldsymbol{x}_k, \boldsymbol{y}_k)$. We then utilize an interval stabbing approach to estimate the rotation $\boldsymbol{R}_k$ for the candidate $(\boldsymbol{x}_k, \boldsymbol{y}_k)$. Finally, we employ a global search to estimate the optimal relative rotation $\boldsymbol{R}^*$ from $\boldsymbol{R}_k$ for each $(\boldsymbol{x}_k, \boldsymbol{y}_k)$.

Rotation-Invariant Constraints

Given two pairs of non-parallel normal correspondences, the candidate $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ from (P_k, Q_k) and an inlier $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ from (P_i, Q_i), a rotation can be determined [Chen *et al.*, 2019]. In an ideal situation, discounting noise, and assuming that rotation $\hat{\boldsymbol{R}} = R(\hat{\theta}, \hat{\boldsymbol{a}})$ precisely aligns $\boldsymbol{x}_k$ to $\boldsymbol{y}_k$ and $\boldsymbol{x}_i$ to $\boldsymbol{y}_i$:

$$\boldsymbol{y}_k = R(\hat{\theta}, \hat{\boldsymbol{a}})\boldsymbol{x}_k, \quad \boldsymbol{y}_i = R(\hat{\theta}, \hat{\boldsymbol{a}})\boldsymbol{x}_i, \tag{1}$$

where $\hat{\theta}$ represents the rotation angle, while $\hat{\boldsymbol{a}}$ denotes the rotation axis. As stated in (1), it is derived that:

$$\angle(\boldsymbol{x}_k, \boldsymbol{y}_k) = \angle(\boldsymbol{x}_i, \boldsymbol{y}_i) = \hat{\theta}. \tag{2}$$

Assuming that the rotation $\tilde{\boldsymbol{R}} = R(\tilde{\theta}, \tilde{\boldsymbol{a}})$ aligns $\boldsymbol{y}_i$ to $\boldsymbol{y}_k$ precisely, it is inferred that:

$$\boldsymbol{y}_k = R(\tilde{\theta}, \tilde{\boldsymbol{a}})\boldsymbol{y}_i = R(\tilde{\theta}, \tilde{\boldsymbol{a}})R(\hat{\theta}, \hat{\boldsymbol{a}})\boldsymbol{x}_i = R(\hat{\theta}, \hat{\boldsymbol{a}})\boldsymbol{x}_k. \tag{3}$$

According to (3), one can infer that:

$$\angle(\boldsymbol{y}_k, \boldsymbol{y}_i) = \angle(\boldsymbol{x}_k, \boldsymbol{x}_i) = \tilde{\theta}. \tag{4}$$

In the usual case involving noise and outliers, derived from (2) and (4), the rotation-invariant constraints are defined as:

$$\begin{aligned} |\angle(\boldsymbol{y}_k, \boldsymbol{y}_i) - \angle(\boldsymbol{x}_k, \boldsymbol{x}_i)| \leq \xi, \\ |\angle(\boldsymbol{x}_k, \boldsymbol{y}_k) - \angle(\boldsymbol{x}_i, \boldsymbol{y}_i)| \leq \xi, \end{aligned} \tag{5}$$

where ξ represents the angular threshold for the constraints.

Moreover, to avoid parallelism between the candidate $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ and an inlier $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, we set $\angle(\boldsymbol{y}_k, \boldsymbol{y}_i) \in [\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $\angle(\boldsymbol{x}_k, \boldsymbol{x}_i) \in [\frac{1}{12}\pi, \frac{11}{12}\pi]$ [Li *et al.*, 2022].

Rotation Bounds

Rotation-invariant constraints serve as a preliminary method to filter out outlier correspondences. Further, a robust and refined outlier removal method is presented in this section, enhancing the accuracy in pruning outlier correspondences.

For each candidate normal correspondence $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ and an inlier $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, our goal is to estimate $\boldsymbol{R}_k$ which maps $\boldsymbol{x}_k$ to $\boldsymbol{y}_k$ and brings $\boldsymbol{x}_i$ within angular distance ε from $\boldsymbol{y}_i$:

$$\angle(\boldsymbol{y}_i, \boldsymbol{R}_k\boldsymbol{x}_i) \leq \varepsilon. \tag{6}$$

As illustrated in Figure 2, the rotation matrix $\boldsymbol{R}_k$ is decomposed into two rotations:

$$\boldsymbol{R}_k = \boldsymbol{A}_{\theta, \boldsymbol{y}_k}\boldsymbol{B}_k. \tag{7}$$

$\boldsymbol{B}_k$ is a rotation matrix that maps $\boldsymbol{x}_k$ to $\boldsymbol{y}_k$:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{B}_k\boldsymbol{x}_k, \\ \boldsymbol{B}_k &= \boldsymbol{I} + [\boldsymbol{x}_k \times \boldsymbol{y}_k]_\times + [\boldsymbol{x}_k \times \boldsymbol{y}_k]_\times^2 \frac{1}{1 + \boldsymbol{x}_k^\top \cdot \boldsymbol{y}_k}, \end{aligned} \tag{8}$$

where $[\cdot]_\times$ is a skew-symmetric cross-product matrix.

$\boldsymbol{A}_{\theta, \boldsymbol{y}_k}$ in (7) is a rotation about the axis $\boldsymbol{y}_k$ by an angle θ, maintaining $\boldsymbol{y}_k$ and $\boldsymbol{B}_k\boldsymbol{x}_k$ unchanged. It is denoted as:

$$\boldsymbol{A}_{\theta, \boldsymbol{y}_k} = exp(\theta[\boldsymbol{y}_k]_\times), \tag{9}$$

where $exp(\cdot)$ is the exponential map [Yan *et al.*, 2022]. Combining (8) and (9), the constraint (6) is rewritten as:

$$\angle(\boldsymbol{y}_i, \boldsymbol{A}_{\theta, \boldsymbol{y}_k}\boldsymbol{B}_k\boldsymbol{x}_i) \leq \varepsilon. \tag{10}$$

As illustrated in Figure 2, rotating by $\boldsymbol{A}_{\theta, \boldsymbol{y}_k}$ while maintaining (10) causes $\boldsymbol{B}_k\boldsymbol{x}_i$ to be situated within a spherical region with an angular radius ε centered at $\boldsymbol{y}_i$, denoted as:

$$\begin{aligned} \boldsymbol{A}_{\theta, \boldsymbol{y}_k}\boldsymbol{B}_k\boldsymbol{x}_i \in S_\varepsilon(\boldsymbol{y}_i), \\ S_\varepsilon(\boldsymbol{y}_i) = \{\boldsymbol{p} | \angle(\boldsymbol{y}_i, \boldsymbol{p}) \leq \varepsilon, \| \boldsymbol{p} \| = 1\}. \end{aligned} \tag{11}$$

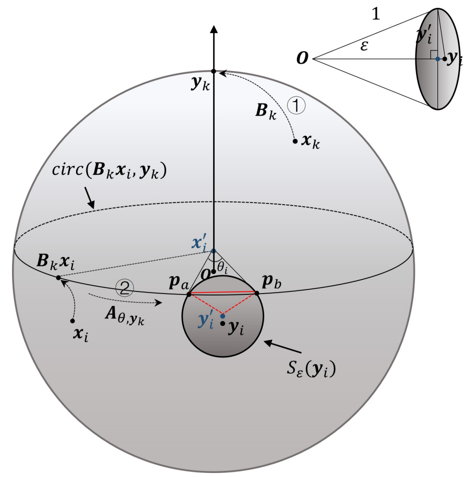


Figure 2: Explanation of rotation decomposition per as Equation (7). Rotation $\boldsymbol{B}_k \in SO(3)$ illustrates the process of mapping $\boldsymbol{x}_k$ to $\boldsymbol{y}_k$. Rotation $\boldsymbol{A}_{\theta, \boldsymbol{y}_k} \in SO(3)$ describes the rotation of $\boldsymbol{B}_k\boldsymbol{x}_i$ around the axis $\boldsymbol{y}_k$ by an angle θ, aligning it to $\boldsymbol{y}_i$. $S_\varepsilon(\boldsymbol{y}_i)$ denotes a spherical region centered at $\boldsymbol{y}_i$ with an angular radius ε.

The trace of $\boldsymbol{B}_k\boldsymbol{x}_i$ in Figure 2 rotating around the axis $\boldsymbol{y}_k$ forms a circle, which is denoted as:

$$circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k) = \{\boldsymbol{A}_{\theta, \boldsymbol{y}_k}\boldsymbol{B}_k\boldsymbol{x}_i | \theta \in [0, 2\pi]\}. \tag{12}$$

As depicted in Figure 2, if $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ and $S_\varepsilon(\boldsymbol{y}_i)$ do not intersect, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ cannot be aligned by any rotation $\boldsymbol{R}_k$ and it is removed safely. If they intersect, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is an inlier for $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, and points $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ are the bounds of rotation $\boldsymbol{A}_{\theta, \boldsymbol{y}_k}$. Further, The bounds are denoted as an angle interval $\theta \in [\angle(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{p}_a), \angle(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{p}_b)] = [\theta_i^a, \theta_i^b]$.

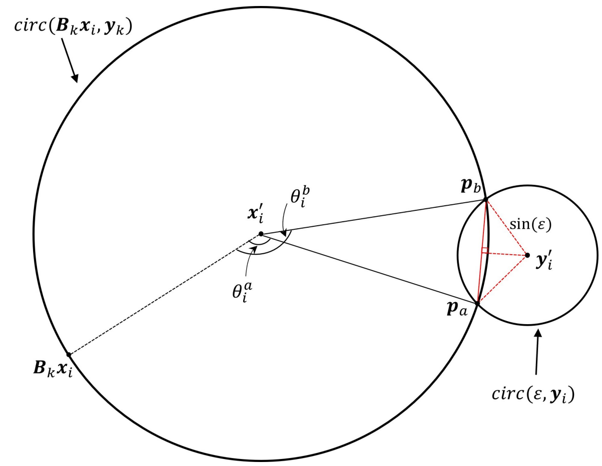


Figure 3: Illustration of rotation bounds $\theta_i \in [\theta_i^a, \theta_i^b]$. The circle $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ denotes the trace of $\boldsymbol{B}_k\boldsymbol{x}_i$ as it rotates around the axis $\boldsymbol{y}_k$, with its center point denoted as $\boldsymbol{x}_i^{'}$. Describing the boundary of $S_\varepsilon(\boldsymbol{y}_i)$, the circle $circ(\varepsilon, \boldsymbol{y}_i)$ has its center denoted as $\boldsymbol{y}_i^{'}$, as shown in the upper-right corner. Points $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ are the intersections of circles $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ and $circ(\varepsilon, \boldsymbol{y}_i)$.

As illustrated in Figure 3, the computation for the rotation bounds $[\theta_i^a, \theta_i^b]$ is demonstrated by projecting the sphere from Figure 2 onto a 2D plane. To simplify, we align $\boldsymbol{y}_k$ with the z-axis. Let $\boldsymbol{B}_k\boldsymbol{x}_i = [x_0, y_0, z_0]^\top$. The center of $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ is denoted as $\boldsymbol{x}_i^{'} = [0, 0, z_0]^\top$, with a radius $r_{\boldsymbol{x}_i^{'}} = \sqrt{x_0^2 + y_0^2}$. The equation for $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ is:

$$(\boldsymbol{p} - \boldsymbol{x}_i^{'})^\top(\boldsymbol{p} - \boldsymbol{x}_i^{'}) = r^2_{\boldsymbol{x}_i^{'}}, \tag{13}$$

where $\boldsymbol{p} = [x, y, z_0]^\top$ and z_0 is a constant value from $\boldsymbol{B}_k\boldsymbol{x}_i$.

Another circle, defining the boundary of $S_\varepsilon(\boldsymbol{y}_i)$ with an angular radius ε centered at $\boldsymbol{y}_i$, is denoted as:

$$circ(\varepsilon, \boldsymbol{y}_i) = \{\boldsymbol{p} | \angle(\boldsymbol{y}_i, \boldsymbol{p}) = \varepsilon, \| \boldsymbol{p} \| = 1\}. \tag{14}$$

The center of $circ(\varepsilon, \boldsymbol{y}_i)$ is $\boldsymbol{y}_i^{'} = cos(\varepsilon)\boldsymbol{y}_i = [x_1, y_1, z_1]^\top$, and its radius $r_{\boldsymbol{y}_i^{'}} = sin(\varepsilon)$. Point $\boldsymbol{p} = [x, y, z]^\top$ on the $circ(\varepsilon, \boldsymbol{y}_i)$ satisfies:

$$(\boldsymbol{p} - \boldsymbol{y}_i^{'})^\top(\boldsymbol{p} - \boldsymbol{y}_i^{'}) = r^2_{\boldsymbol{y}_i^{'}}. \tag{15}$$

To obtain the intersections between the two circles $circ(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{y}_k)$ and $circ(\varepsilon, \boldsymbol{y}_i)$, combining (13) and (15), and letting $\boldsymbol{p} = [x, y, z_0]^\top$, the equation for line $\boldsymbol{p}_a\boldsymbol{p}_b$ is:

$$\begin{aligned} y = ux + v, \quad u = -\frac{x_1}{y_1}(y_1 \neq 0), \\ v = -\frac{1}{2y_1}\{r^2_{\boldsymbol{x}_i^{'}} - r^2_{\boldsymbol{y}_i^{'}} + x_1^2 + y_1^2 + (z_0 - z_1)^2\}. \end{aligned} \tag{16}$$

Further, (13) and (16) are combined to obtain $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$:

$$(1 + u^2)x^2 + 2uvx + (v^2 - r^2_{\boldsymbol{x}_i^{'}}) = 0, \tag{17}$$

$$\Delta = 4u^2v^2 - 4(1 + u^2)(v^2 - r^2_{\boldsymbol{x}_i^{'}}). \tag{18}$$

If $\Delta < 0$, it indicates that the correspondence $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is an outlier and it is removed safely. Otherwise, let $\boldsymbol{p}_a = [\frac{-2uv-\sqrt{\Delta}}{2(1+u^2)}, \frac{-2uv-\sqrt{\Delta}}{2(1+u^2)}u + v, z_0]^\top$, and $\boldsymbol{p}_b = [\frac{-2uv+\sqrt{\Delta}}{2(1+u^2)}, \frac{-2uv+\sqrt{\Delta}}{2(1+u^2)}u + v, z_0]^\top$. The rotation bounds are obtained as $\theta_i \in [\theta_i^a, \theta_i^b](\theta_i^a \leq \theta_i^b)$, where $\theta_i^a = \angle(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{p}_a)$, and $\theta_i^b = \angle(\boldsymbol{B}_k\boldsymbol{x}_i, \boldsymbol{p}_b)$.

Interval Stabbing

After pruning outlier correspondences for candidate $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, a set of angle intervals $\Theta_k = \{[\theta_i^a, \theta_i^b]\}$ is obtained. An interval stabbing technique [Bustos and Chin, 2017; Cai *et al.*, 2019] is introduced to determine an angle θ_k which is encompassed by maximum number of angle intervals within Θ_k:

$$\theta_k = \underset{\theta_k \in [0, 2\pi], [\theta_i^a, \theta_i^b] \in \Theta_k}{argmax} \sum_i \mathbb{I}(\theta_k \in [\theta_i^a, \theta_i^b]) \tag{19}$$

where $\mathbb{I}(\cdot)$ is an indicator function, yielding 1 when the input is selected and 0 otherwise. Finally, $\boldsymbol{A}_{\theta_k, \boldsymbol{y}_k}\boldsymbol{B}_k = \boldsymbol{R}_k \in \mathcal{R}_c$, where $\mathcal{R}_c$ denotes the rotation set.

Global Rotation Search

After obtaining $\boldsymbol{R}_k \in \mathcal{R}_c$ for each $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, we globally search for the optimal relative rotation $\boldsymbol{R}^*$:

$$\boldsymbol{R}^* = \underset{\boldsymbol{R}_k \in \mathcal{R}_c}{argmin} \sum_{i=1}^{K} min\left(\angle(\boldsymbol{y}_i, \boldsymbol{R}_k\boldsymbol{x}_i), \varepsilon\right). \tag{20}$$

Moreover, with the optimal $\boldsymbol{R}^*$, for each correspondence (P_k, Q_k), if $\angle(\boldsymbol{y}_k, \boldsymbol{R}^*\boldsymbol{x}_k) > 2\varepsilon$, (P_k, Q_k) is treated as an outlier and subsequently removed from $\mathcal{A}$. The experiment shows that relaxing the threshold from ε to 2ε, without affecting $\boldsymbol{R}^*$, enhances the precision of relative translations.

2.2 Component-Wise for Translation Estimation

Finally, a component-wise method based on plane constraints, named as CWTE, has been proposed for estimating relative translations. We define that parallel planes represent an identical translation component, and each translation component determines 1-DoF. In theory, at least three non-parallel translation components are required to determine a 3-DoF relative translation [Li *et al.*, 2022].

We parameterize planes using the Hessian form $\boldsymbol{\pi} = [\boldsymbol{n}^\top, d]^\top$, where $\boldsymbol{n} = [n_x, n_y, n_z]^\top$ is the unit vector representing the plane orientation and d is the distance of the plane from the origin [Hartley and Zisserman, 2003]. A point $\boldsymbol{p}$ on the plane $\boldsymbol{\pi}$ gives $d = -\boldsymbol{n}^\top\boldsymbol{p}$. Distance between two parallel planes $\boldsymbol{\pi}_i = [\boldsymbol{n}^\top, d_i]^\top$ and $\boldsymbol{\pi}_j = [\boldsymbol{n}^\top, d_j]^\top$ is defined as:

$$dist(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \frac{|d_i - d_j|}{\sqrt{\boldsymbol{n}^\top\boldsymbol{n}}} = |d_i - d_j|. \tag{21}$$

With the optimal $\boldsymbol{R}^*$, for each correspondence $(P_k, Q_k) \in \mathcal{A}$, where $P_k \in \mathcal{P}$ and $Q_k \in \mathcal{Q}$, the normal $\boldsymbol{x}_k$ from P_k is aligned to the normal $\boldsymbol{y}_k$ from Q_k by $\boldsymbol{R}^*$, denoted as follows:

$$P_k^{'} = (\boldsymbol{x}_k^{'}, \boldsymbol{p}_k^{'}) = \boldsymbol{R}^*(\boldsymbol{x}_k, \boldsymbol{p}_k) = \boldsymbol{R}^*P_k, \tag{22}$$

where $\angle(\boldsymbol{x}_k^{'}, \boldsymbol{y}_k) \leq 2\varepsilon$, and $(P_k^{'}, Q_k) \in \mathcal{A}^{'}$.

Translation Components Computation

First, based on the parallel characteristics among planes in the source point cloud $\mathcal{S}$, the plane correspondences in $\mathcal{A}^{'}$ are divided into $\mathcal{L}$ component sets, each component set is defined as $\mathcal{C}^l$ $(1 \leq l \leq \mathcal{L})$. For the l-th component set $\mathcal{C}^l$, an optimal 1-DoF translation component $\hat{\boldsymbol{t}}^l$ will be estimated based on plane translation constraints.

Assuming that there are two parallel inlier plane correspondences $(P_k^{'}, Q_k), (P_i^{'}, Q_i) \in \mathcal{A}^{'}$, and $Q_k \parallel Q_i$. As shown in Figure 5(a), in the ideal case, the plane translation operator $[\boldsymbol{0}^\top, \hat{d}]^\top$ precisely aligns plane $P_k^{'}$ to plane Q_k, and plane $P_i^{'}$ to plane Q_i, denoted as:

$$\boldsymbol{\pi}_{Q_k} = \boldsymbol{\pi}_{P_k^{'}} + [\boldsymbol{0}^\top, \hat{d}]^\top, \quad \boldsymbol{\pi}_{Q_i} = \boldsymbol{\pi}_{P_i^{'}} + [\boldsymbol{0}^\top, \hat{d}]^\top. \tag{23}$$

As indicated by (23), it is derived that:

$$dist(\boldsymbol{\pi}_{P_k^{'}}, \boldsymbol{\pi}_{Q_k}) = dist(\boldsymbol{\pi}_{P_i^{'}}, \boldsymbol{\pi}_{Q_i}) = |\hat{d}|. \tag{24}$$

Moreover, assuming that plane Q_i is aligned to plane Q_k by the plane translation operator $[\boldsymbol{0}^\top, \tilde{d}]^\top$, denoted as:

$$\begin{aligned} \boldsymbol{\pi}_{Q_k} = \boldsymbol{\pi}_{Q_i} + [\boldsymbol{0}^\top, \tilde{d}]^\top = \boldsymbol{\pi}_{P_i^{'}} + [\boldsymbol{0}^\top, \hat{d}]^\top + [\boldsymbol{0}^\top, \tilde{d}]^\top \\ = \boldsymbol{\pi}_{P_k^{'}} + [\boldsymbol{0}^\top, \hat{d}]^\top. \end{aligned} \tag{25}$$

According to (25), one can infer that:

$$dist(\boldsymbol{\pi}_{Q_k}, \boldsymbol{\pi}_{Q_i}) = dist(\boldsymbol{\pi}_{P_k^{'}}, \boldsymbol{\pi}_{P_i^{'}}) = |\tilde{d}|. \tag{26}$$

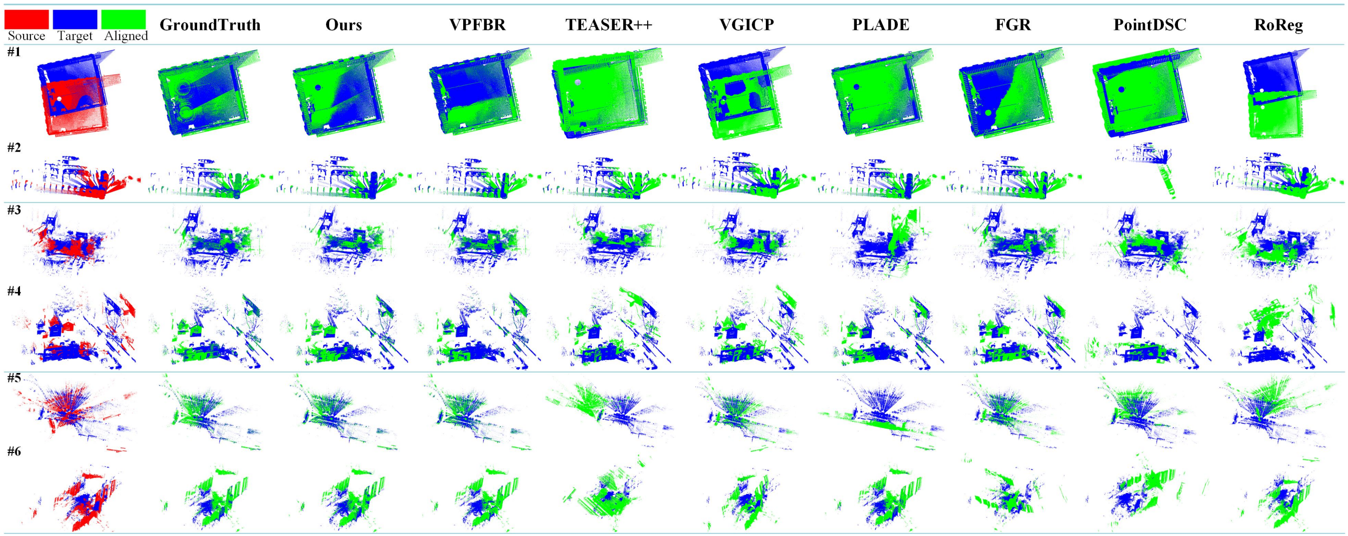


Figure 4: Visualization of some alignment results for our proposed method and baselines on three datasets: ETH3D (**#1** and **#2**), ETH-TLS (**#3** and **#4**), and WHU-TLS (**#5** and **#6**).

In the usual case involving noise and outliers, derived from (24) and (26), the plane translation constraints are defined as:

$$\begin{aligned} &|dist(\boldsymbol{\pi}_{P'_k}, \boldsymbol{\pi}_{Q_k}) - dist(\boldsymbol{\pi}_{P'_i}, \boldsymbol{\pi}_{Q_i})| \leq \eta, \\ &|dist(\boldsymbol{\pi}_{P'_k}, \boldsymbol{\pi}_{P'_i}) - dist(\boldsymbol{\pi}_{Q_k}, \boldsymbol{\pi}_{Q_i})| \leq \eta, \end{aligned} \tag{27}$$

where η represents the distance threshold for the constraints.

According to the plane translation constraints in (27), we estimate the l-th translation component $\hat{\boldsymbol{t}}^l$, defined as follows:

$$\begin{aligned} \hat{\boldsymbol{t}}^l &= \underset{\boldsymbol{t}^l = \boldsymbol{q}_k - Project(\boldsymbol{\pi}_{P'_k}, \boldsymbol{q}_k)}{argmin} \sum_{k=1}^{|\mathcal{C}^l|} \sum_{i=1}^{|\mathcal{C}^l|} (\mathcal{F}_1 + \mathcal{F}_2), \\ \mathcal{F}_1 &= min\left(|dist(\boldsymbol{\pi}_{P'_k}, \boldsymbol{\pi}_{Q_k}) - dist(\boldsymbol{\pi}_{P'_i}, \boldsymbol{\pi}_{Q_i})|, \eta \right), \\ \mathcal{F}_2 &= min\left(|dist(\boldsymbol{\pi}_{P'_k}, \boldsymbol{\pi}_{P'_i}) - dist(\boldsymbol{\pi}_{Q_k}, \boldsymbol{\pi}_{Q_i})|, \eta \right), \end{aligned} \tag{28}$$

where $Project(\boldsymbol{\pi}, \boldsymbol{p})$ denotes a function that projects the point $\boldsymbol{p}$ onto the plane $\boldsymbol{\pi}$.

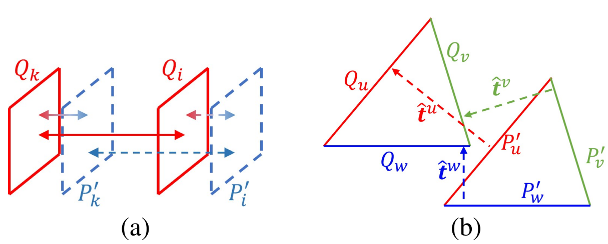


Figure 5: (a) Illustration of plane translation constraints derived from (24) and (26). (b) Computation of 3-DoF relative translation $\hat{\boldsymbol{t}} \in \mathbb{R}^3$ from three translation components: $\hat{\boldsymbol{t}}^u$, $\hat{\boldsymbol{t}}^v$, and $\hat{\boldsymbol{t}}^w$.

Global Translation Search

In accordance with (28), we have obtained the translation component set $\mathcal{T}_c = \{\hat{\boldsymbol{t}}^l\}_{l=1}^{\mathcal{L}}$, and each component $\hat{\boldsymbol{t}}^l \in \mathcal{T}_c$ is associated with a plane correspondence, denoted as (P'_l, Q_l). As depicted in Figure 5(b), three different translation components $\hat{\boldsymbol{t}}^u, \hat{\boldsymbol{t}}^v, \hat{\boldsymbol{t}}^w \in \mathcal{T}_c$ determine a 3-DoF relative translation $\hat{\boldsymbol{t}} \in \mathbb{R}^3$, denoted as follows:

$$\begin{aligned} &{}^1\hat{\boldsymbol{t}} = \hat{\boldsymbol{t}}^u, \quad {}^2\hat{\boldsymbol{t}} = \hat{\boldsymbol{t}}^v - \boldsymbol{y}_u^\top \hat{\boldsymbol{t}}^v \boldsymbol{y}_u, \\ &{}^3\hat{\boldsymbol{t}} = \hat{\boldsymbol{t}}^{w\top} \cdot \frac{\hat{\boldsymbol{t}}^u \times \hat{\boldsymbol{t}}^v}{|\hat{\boldsymbol{t}}^u \times \hat{\boldsymbol{t}}^v|} \cdot \frac{\hat{\boldsymbol{t}}^u \times \hat{\boldsymbol{t}}^v}{|\hat{\boldsymbol{t}}^u \times \hat{\boldsymbol{t}}^v|}, \\ &\hat{\boldsymbol{t}} = {}^1\hat{\boldsymbol{t}} + {}^2\hat{\boldsymbol{t}} + {}^3\hat{\boldsymbol{t}}, \end{aligned} \tag{29}$$

and we compute each candidate relative translation $\hat{\boldsymbol{t}} \in \mathcal{T}_r$ from $\mathcal{T}_c$. A global strategy is utilized to determine the optimal relative translation $\boldsymbol{t}^* \in \mathbb{R}^3$ by the following function:

$$\boldsymbol{t}^* = \underset{\hat{\boldsymbol{t}} \in \mathcal{T}_r, (P'_l, Q_l) \in \mathcal{A}'}{argmin} \sum_{l=1}^{\mathcal{L}} min(\left|dist(\boldsymbol{\pi}_{P'_l}, \boldsymbol{\pi}_{Q_l}) - |\hat{\boldsymbol{t}}|\right|, \eta). \tag{30}$$

3 Experimental Results

3.1 Experimental Environment

Our proposed method has been evaluated on three widely recognized LiDAR point cloud datasets, with each point cloud comprising approximately 5 to 30 million points: ETH3D [Schops *et al.*, 2017], ETH-TLS [Theiler *et al.*, 2015], and WHU-TLS [Dong *et al.*, 2020]. These experiments were conducted on a PC equipped with an Intel Core i7-10700 CPU @ 2.90 GHz and 16 GB RAM.

Baselines. We conducted a comprehensive performance comparison between our method and several well-known algorithms, including FGR [Zhou *et al.*, 2016], VGICP [Koide *et al.*, 2021], PLADE [Chen *et al.*, 2019], Teaser++ [Yang *et al.*, 2020], VPFBR [Li *et al.*, 2022], PointDSC [Bai *et*

Method	ETH3D						ETH-TLS				WHU-TLS				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FGR	1.043	1.269	176.2	0.186	1.483	2.364	4.369	**0.232**	2.947	177.1	13.31	4.536	64.11	60.37	177.3
PLADE	0.101	170.6	**0.172**	**0**	0.234	49.53	117.0	8.066	1.637	**0.122**	118.7	168.9	179.0	**1.104**	76.02
VGICP	1.093	1.268	2.603	4.654	6.629	0.322	25.07	2.773	14.07	11.31	**0.148**	0.897	48.32	21.91	18.29
Teaser++	5.360	0.337	176.9	1.313	0.198	1.294	19.29	0.572	18.01	0.529	179.6	180.0	**0.110**	166.4	**0.937**
VPFBR	0.180	**0.107**	0.302	179.4	**0.077**	0.435	2.020	5.564	**0.483**	1.125	0.686	0.872	0.321	1.588	0.966
PointDSC	9.378	31.85	51.86	1.182	79.66	0.559	175.0	1.425	168.1	156.8	146.1	31.81	112.2	177.9	21.35
RoReg	9.554	19.11	1.118	8.438	11.28	16.64	21.29	1.491	92.38	21.88	16.09	31.97	46.65	3.366	25.39
Ours	**0.093**	0.115	0.925	0.202	0.309	**0.277**	**0.998**	1.049	1.483	0.803	0.594	**0.603**	0.198	1.237	1.130

Table 1: Rotation errors $\delta_{\boldsymbol{R}}$ ($^\circ$) for baselines. Each column displays the top-ranking entry in bold, and the second-ranking entry is underlined.

Method	ETH3D						ETH-TLS				WHU-TLS				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FGR	0.304	2.061	26.26	**0.007**	0.598	1.094	0.469	0.489	3.242	12.00	65.86	16.46	33.37	114.9	26.09
PLADE	0.025	12.73	**0.028**	0.008	0.112	9.988	21.53	12.38	0.180	**0.016**	224.8	46.75	94.69	2.283	16.61
VGICP	6.422	13.11	26.06	0.723	2.258	2.589	14.11	17.88	5.822	11.74	70.76	18.73	46.86	46.10	19.93
Teaseer++	1.848	0.271	26.17	0.015	**0.010**	0.103	0.662	**0.379**	5.299	0.033	167.9	81.95	**0.044**	142.7	0.340
VPFBR	0.030	**0.034**	0.123	1.597	0.156	0.132	0.891	2.257	0.042	0.111	1.317	0.429	0.575	2.490	0.475
PointDSC	0.722	10.59	10.39	0.039	40.51	0.241	13.69	1.697	4.861	16.96	55.09	266.2	379.6	128.0	4.125
RoReg	15.56	24.24	21.56	3.140	4.194	11.43	29.57	31.44	24.93	23.98	90.60	61.37	46.61	45.89	20.11
Ours	**0.006**	**0.034**	0.091	0.011	0.043	**0.066**	**0.440**	0.888	**0.016**	0.063	**0.231**	**0.252**	0.783	**1.013**	**0.267**

Table 2: Translation errors $\delta_{\boldsymbol{t}}$ (m) for baselines. Each column displays the top-ranking entry in bold, and the second-ranking entry is underlined.

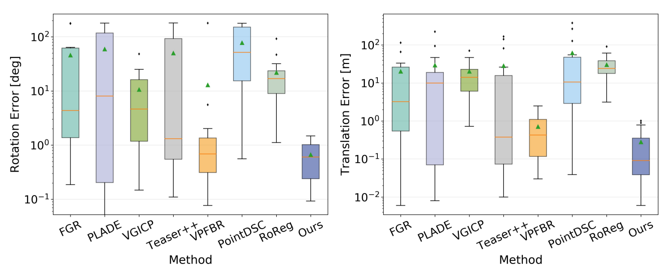
al., 2021], and RoReg [Wang *et al.*, 2023]. PLADE and VPFBR represent two prominent plane-based methods, while Teaser++ and FGR are point correspondence-based outlier removal methods. PointDSC and RoReg are popular learning-based methods, and VGICP is a recently proposed variant of ICP. Notably, the learning-based methods PointDSC and RoReg were executed on a server equipped with an Intel(R) Xeon(R) 4210R CPU @ 2.40 GHz, 32GB RAM, and a GeForce RTX3090 GPU with 24GB RAM.

Metrics. We have focused on two key evaluation metrics to assess the performance of our proposed method: computational efficiency and registration accuracy. Computational efficiency is gauged by measuring runtime, while registration accuracy is quantified through rotation error and translation error [Yang *et al.*, 2020], as expressed by the equation:

$$\begin{cases} \delta_R & = arccos\frac{1}{2}\left(tr(\boldsymbol{R}_{gt}^T\boldsymbol{R}_{est})-1\right) \\ \delta_t & = \parallel \boldsymbol{t}_{gt} - \boldsymbol{t}_{est} \parallel \end{cases} \tag{31}$$

where $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ denote rotation matrix and translation vector, respectively.

Parameter Configuration. When extracting planes, we choose the top 20 planes ($N = M = 20$) from each point cloud, based on [Li *et al.*, 2022]. In rotation estimation, the inlier threshold ε defaults to 1.5°, while the rotation-invariant constraints threshold ξ is set to 8°. In translation estimation, the plane translation constraints threshold η is $0.1m$.



(a) Rotation performance. (b) Translation performance.

Figure 6: The statistical results of the experimental rotation errors and translation errors from three datasets.

3.2 Performance Comparison

Registration Accuracy. We first evaluated the registration accuracy of our proposed algorithm. Tables 1 and 2 list the rotation and translation errors for each experiment of all algorithms on the three datasets, and Figure 6 provides a visual representation of the corresponding statistical outcomes. Evidently, our method outperformed others, exhibiting the lowest rotation and translation errors, a perspective further substantiated by the results in Table 3. VPFBR, a method based on plane pairs, ranks second in performance after our method. Conversely, another plane-based method, PLADE, despite performing well in several scenes, reveals instability, with rotation errors exceeding 5° in over 53% of cases. Teaser++ is a classic point correspondence-based outlier removal method, exhibiting commendable performance with rotation errors below 5° and translations under $1m$ across eight different scenes

Method	ETH3D			ETH-TLS			WHU-TLS		
	$\delta_{\boldsymbol{R}} \downarrow$	$\delta_{\boldsymbol{t}} \downarrow$	$time \downarrow$	$\delta_{\boldsymbol{R}} \downarrow$	$\delta_{\boldsymbol{t}} \downarrow$	$time \downarrow$	$\delta_{\boldsymbol{R}} \downarrow$	$\delta_{\boldsymbol{t}} \downarrow$	$time \downarrow$
FGR	30.42	5.054	183.6	46.16	4.050	327.7	63.93	51.34	407.3
PLADE	36.77	3.815	167.0	31.71	8.527	285.1	108.7	77.03	433.6
VGICP	2.762	8.527	11.42	13.31	12.39	19.38	17.91	40.48	27.62
Teaser++	30.90	4.736	20.90	9.600	1.593	31.27	105.4	78.59	69.84
VPFBR	30.08	0.345	7.593	2.298	0.825	11.47	0.887	1.057	21.03
PointDSC	29.08	10.42	**3.291**	125.3	9.302	**4.617**	97.87	166.6	**6.425**
RoReg	11.02	13.35	6.092	34.26	27.48	9.828	24.86	52.92	11.74
Ours	**0.320**	**0.042**	8.059	**1.083**	**0.352**	10.95	**0.752**	**0.509**	19.51

Table 3: Average of rotation errors δ_R ($^\circ$), translation errors δ_t (m) and runtime (s) for baselines on three datasets. Each column displays the top-ranking entry in bold, and the second-ranking entry is underlined.

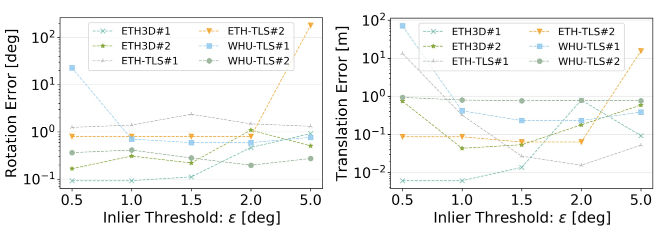
Method	Input Size (byte) $\downarrow$		Runtime (s) $\downarrow$	
	$Source$	$Target$	RE	TE
Ours	480	480	0.291	0.003

Table 4: Input size and average runtime of our method for rotation estimation (RE) and translation estimation (TE) on three datasets.

Method	Feature Size (byte)$\downarrow$		Runtime(s)$\downarrow$	$\delta_{\boldsymbol{R}}(^\circ)\downarrow$	$\delta_{\boldsymbol{t}}(\text{m})\downarrow$
	$Source$	$Target$			
Teaser++	6.8×10^5	1.1×10^6	47.33	19.29	0.662
Ours	480	480	9.670	0.998	0.440

Table 5: Detailed Comparison of our method and Teaser++.

from three datasets. However, its performance significantly varies in four other scenes, resulting in rotation errors of approximately 170° and translation errors exceeding $20m$. This notable inconsistency in results highlights the lack of robustness in the Teaser++ algorithm and underscores its varying reliability across different scenes. Unfortunately, as depicted in Figure 4, both learning-based methods, PointDSC and RoReg, exhibited poor performance in both rotation and translation. In rotation estimation, our method ensures accuracy by utilizing an outlier removal scheme, which ensures that the rotation error does not exceed ε (default $\varepsilon = 1.5^\circ$), as shown in Figure 2. In translation estimation, plane translation constraints facilitate efficient computation of each component, while a global search guarantees robust and accurate estimation.



(a) Rotation performance. (b) Translation performance.

Figure 7: Ablation study for inter threshold ε.

Computational Efficiency. In terms of computational efficiency, as depicted in Table 3, the two learning-based methods had the shortest runtime when training time was excluded. Our proposed method not only excelled in registration accuracy but also closely followed the learning-based methods in runtime performance. Significantly, our algorithm's runtime is primarily dedicated to preprocessing. As indicated in Table 4, without parallel optimization, the average runtime for the sum of rotation and translation estimations in our method stays below $0.3s$. Table 5 shows experimental results comparing our algorithm with Teaser++. Teaser++ demands a large input payload of about 1.8×10^6 bytes, including 5184 FPFH features from the source point cloud and 8584 FPFH features from the target point cloud, far exceeding our algorithm's input requirements. Conversely, our method efficiently obtains higher-precision rotation and translation from a mere 960 bytes of plane features, highlighting our algorithm's remarkable efficiency and accuracy.

3.3 Ablation Study

The inlier threshold ε is a critical parameter in our algorithm, as depicted in Figure 2, directly influencing both rotation and translation accuracy. We evaluated the impact of ε on our algorithm by selecting two scenes from each of the three datasets and varying $\varepsilon = [0.5^\circ, 1.0^\circ, 1.5^\circ, 2.0^\circ, 5.0^\circ]$. Figure 7 illustrates that with an inlier threshold ε between 1.0° and 2.0°, rotation error remains stable, generally below 1.0°, and translation error stays under $1m$. Notably, extremely small ε ($\leq 0.5^\circ$) may overly suppress outliers, potentially reducing rotation and translation accuracy. Conversely, excessively large ε ($\geq 5.0^\circ$) may compromise outlier removal, adversely affecting accuracy, too.

4 Conclusion

In this paper, we introduce an innovative plane constraints-based framework for decoupled point cloud registration in large-scale LiDAR point cloud scenes. For rotation estimation, we present a robust outlier removal scheme based on rotation-invariant constraints and computational geometric consistency checks to estimate relative rotations accurately. For translation estimation, we propose a component-wise method based on plane translation constraints to obtain high-precision relative translations. In the future, we will explore the application of our method in multi-view point cloud registration tasks.

Acknowledgments

This work was supported by the National Science Foundation of China under Grant 61771146.

References

[Ao *et al.*, 2021] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan 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*, pages 11753–11762, 2021.

[Aoki *et al.*, 2019] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7163–7172, 2019.

[Bai *et al.*, 2021] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021.

[Barath and Matas, 2018] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6733–6741, 2018.

[Barath *et al.*, 2020] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020.

[Besl and McKay, 1992] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[Bouaziz *et al.*, 2013] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013.

[Bustos and Chin, 2017] Alvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2868–2882, 2017.

[Cai *et al.*, 2019] Zhipeng Cai, Tat-Jun Chin, Alvaro Parra Bustos, and Konrad Schindler. Practical optimal registration of terrestrial lidar scan pairs. *ISPRS journal of photogrammetry and remote sensing*, 147:118–131, 2019.

[Chen *et al.*, 2019] Songlin Chen, Liangliang Nan, Renbo Xia, Jibin Zhao, and Peter Wonka. Plade: A plane-based descriptor for point cloud registration with small overlap. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2530–2540, 2019.

[Chen *et al.*, 2022] Zhi Chen, Kun Sun, Fan Yang, and Wenbing 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*, pages 13221–13231, 2022.

[Dong *et al.*, 2020] Zhen Dong, Fuxun Liang, Bisheng Yang, Yusheng Xu, Yufu Zang, Jianping Li, Yuan Wang, Wenxia Dai, Hongchao Fan, Juha Hyyppä, et al. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:327–342, 2020.

[Fischler and Bolles, 1981] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Fu *et al.*, 2021] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8893–8902, 2021.

[Hartley and Zisserman, 2003] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[Huang *et al.*, 2021] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021.

[Koide *et al.*, 2021] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized gicp for fast and accurate 3d point cloud registration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11054–11059. IEEE, 2021.

[Li and Harada, 2022] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564, 2022.

[Li *et al.*, 2022] Jianwei Li, Jiawang Zhan, Ting Zhou, Virgílio A Bento, and Qianfeng Wang. Point cloud registration and localization based on voxel plane features. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188:363–379, 2022.

[Li *et al.*, 2023] Jiayuan Li, Pengcheng Shi, Qingwu Hu, and Yongjun Zhang. Qgore: Quadratic-time guaranteed outlier removal for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[Lim *et al.*, 2022] Hyungtae Lim, Suyong Yeon, Soohyun Ryu, Yonghan Lee, Youngji Kim, Jaeseong Yun, Euigon Jung, Donghwan Lee, and Hyun Myung. A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8010–8017. IEEE, 2022.

[Liu *et al.*, 2023a] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang.

Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. *arXiv preprint arXiv:2303.12384*, 2023.

[Liu *et al.*, 2023b] Quan Liu, Yunsong Zhou, Hongzi Zhu, Shan Chang, and Minyi Guo. Apr: Online distant point cloud registration through aggregated point cloud reconstruction. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1204–1212. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Liu *et al.*, 2023c] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21445–21455, 2023.

[Lu *et al.*, 2019] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[Ma and Wei, 2023] Gang Ma and Hui Wei. A novel sketch-based framework utilizing contour cues for efficient point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[Ma *et al.*, 2023] Gang Ma, Siwei Tu, and Hui Wei. A novel sketch-based registration framework for point cloud frames. *IEEE Geoscience and Remote Sensing Letters*, 2023.

[Pomerleau *et al.*, 2013] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets: Open-source library and experimental protocol. *Autonomous robots*, 34:133–148, 2013.

[Qin *et al.*, 2022] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11143–11152, 2022.

[Reyes-Aviles *et al.*, 2022] Fernando Reyes-Aviles, Philipp Fleck, Dieter Schmalstieg, and Clemens Arth. Compact world anchors: Registration using parametric primitives as scene description. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[Schops *et al.*, 2017] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[Segal *et al.*, 2009] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[Theiler *et al.*, 2015] Pascal Willy Theiler, Jan Dirk Wegner, and Konrad Schindler. Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS journal of photogrammetry and remote sensing*, 109:126–138, 2015.

[Wang *et al.*, 2022] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1630–1641, 2022.

[Wang *et al.*, 2023] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[Xue *et al.*, 2024] Weiyi Xue, Fan Lu, and Guang Chen. Hdmnet: A hierarchical matching network with double attention for large-scale outdoor lidar point cloud registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3393–3403, 2024.

[Yan *et al.*, 2022] Li Yan, Pengcheng Wei, Hong Xie, Jicheng Dai, Hao Wu, and Ming Huang. A new outlier removal strategy based on reliability of correspondence graph for fast point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Yan *et al.*, 2023] Dayu Yan, Haitao Jiang, Tuan Li, and Chuang Shi. Efficient vanishing point estimation for accurate camera rotation estimation in indoor environments. *IEEE Robotics and Automation Letters*, 2023.

[Yang *et al.*, 2020] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.

[Yew and Lee, 2022] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022.

[Zhang *et al.*, 2021] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2021.

[Zhang *et al.*, 2022] Long Zhang, Jianwei Guo, Zhanglin Cheng, Jun Xiao, and Xiaopeng Zhang. Efficient pairwise 3-d registration of urban scenes via hybrid structural descriptors. *IEEE Trans. Geosci. Remote. Sens.*, 60:1–17, 2022.

[Zhou *et al.*, 2016] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016.