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

Gang Ma, Hui Wei∗, Runfeng Lin, and Jiali Yang
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 \( T = [R|t] \in SE(3) \), serving the vital function of meticulously aligning the source point cloud \( P \) with the coordinate system of the target point cloud \( Q \) [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 are 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
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 $P$ and $Q$, respectively. Drawing inspiration from [Li et al., 2022], we employ a voxel-based plane extraction method to obtain planes from $P$ and $Q$. Let us define $P = \{P_1, P_2, \cdots, P_M\}$ and $Q = \{Q_1, Q_2, \cdots, Q_N\}$, where each plane $P_i = (x_i, p_i)(1 \leq i \leq M)$ from $P$ and each plane $Q_j = (y_j, q_j)(1 \leq j \leq N)$ from $Q$ respectively contain normals $x_i$ and $y_j$, along with points $p_i$ and $q_j$ on their respective planes. Next, let $(P_k, Q_k) \in A(1 \leq k \leq M * N)$ represent candidate plane correspondences, where $P_k \in P$ and $Q_k \in Q$. While it is inevitable that the candidate set $A$ contains some noisy correspondences, our goal is to minimize the impact of these outliers and accurately estimate the relative rotation $R \in SO(3)$ and relative translation $t \in \mathbb{R}^3$ from candidate set $A$.

As shown in Figure 1, after extracting planes from the source cloud $P$ and target cloud $Q$, along with the establishment of plane correspondences $A$, 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 $(x_k, y_k)$ 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 $(x_i, y_i)$ from $(P_i, Q_i)$ ($i \neq k$) to the candidate $(x_k, y_k)$. We then utilize an interval stabling approach to estimate the rotation $R_k$ for the candidate $(x_k, y_k)$. Finally, we employ a global search to estimate the optimal relative rotation $R^* \in R_k$ for each $(x_k, y_k)$.

Rotation-Invariant Constraints

Given two pairs of non-parallel normal correspondences, the candidate $(x_k, y_k)$ from $(P_k, Q_k)$ and an inlier $(x_i, y_i)$ from $(P_i, Q_i)$, a rotation can be determined [Chen et al., 2019]. In an ideal situation, discounting noise, and assuming that...
rotation $\hat{R} = R(\hat{\theta}, \hat{a})$ precisely aligns $x_k$ to $y_k$, and $x_i$ to $y_i$:
\[ y_k = R(\hat{\theta}, \hat{a})x_k, \quad y_i = R(\hat{\theta}, \hat{a})x_i, \]  
where $\hat{\theta}$ represents the rotation angle, while $\hat{a}$ denotes the rotation axis. As stated in (1), it is derived that:
\[ \angle(x_k, y_k) = \angle(x_i, y_i) = \hat{\theta}. \]  

Assuming that the rotation $\hat{R} = R(\hat{\theta}, \hat{a})$ aligns $y_i$ to $y_k$ precisely, it is inferred that:
\[ y_k = R(\hat{\theta}, \hat{a})y_i = R(\hat{\theta}, \hat{a})R(\hat{\theta}, \hat{a})x_i = R(\hat{\theta}, \hat{a})x_k. \]  

According to (3), one can infer that:
\[ \angle(y_k, y_i) = \angle(x_k, x_i) = \hat{\theta}. \]  

In the usual case involving noise and outliers, derived from (2) and (4), the rotation-invariant constraints are defined as:
\[ |\angle(y_k, y_i) - \angle(x_k, x_i)| \leq \xi, \]  
\[ |\angle(x_k, y_k) - \angle(x_i, y_i)| \leq \xi, \]  
where $\xi$ represents the angular threshold for the constraints.

Moreover, to avoid parallelism between the candidate $(x_k, y_k)$ and an inlier $(x_i, y_i)$, we set $\angle(y_k, y_i) \in [\frac{\pi}{12}, \frac{\pi}{2}]$ and $\angle(x_k, x_i) \in [\frac{\pi}{12}, \frac{\pi}{2}]$ [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 $(x_k, y_k)$ and an inlier $(x_i, y_i)$, our goal is to estimate $R_k$ which maps $x_k$ to $y_k$ and brings $x_i$ within angular distance $\varepsilon$ from $y_i$:
\[ \angle(y_i, R_k x_i) \leq \varepsilon. \]  

As illustrated in Figure 2, the rotation matrix $R_k$ is decomposed into two rotations:
\[ R_k = A_{\theta, y_k} B_k. \]  

$B_k$ is a rotation matrix that maps $x_k$ to $y_k$:
\[ y_k = B_k x_k, \]  
\[ B_k = I + [x_k \times y_k]_x + [x_k \times y_k]^2 \frac{1}{1 + x_k \cdot y_k}, \]  
where $[\cdot]_x$ is a skew-symmetric cross-product matrix.

$A_{\theta, y_k}$ in (7) is a rotation about the axis $y_k$ by an angle $\theta$, maintaining $y_k$ and $B_k x_k$ unchanged. It is denoted as:
\[ A_{\theta, y_k} = exp(\theta [y_k]_x), \]  
where $exp(\cdot)$ is the exponential map [Yan et al., 2022]. Combining (8) and (9), the constraint (6) is rewritten as:
\[ \angle(y_i, A_{\theta, y_k} B_k x_i) \leq \varepsilon. \]  

As illustrated in Figure 2, rotating by $A_{\theta, y_k}$ while maintaining (10) causes $B_k x_i$ to be situated within a spherical region with an angular radius $\varepsilon$ centered at $y_i$, denoted as:
\[ A_{\theta, y_k} B_k x_i \in S_\varepsilon(y_i), \]  
\[ S_\varepsilon(y_i) = \{ p | \angle(y_i, p) \leq \varepsilon, \| p \| = 1 \}. \]  

Figure 2: Explanation of rotation decomposition per as Equation (7). Rotation $B_k \in SO(3)$ illustrates the process of mapping $x_k$ to $y_k$. Rotation $A_{\theta, y_k} \in SO(3)$ describes the rotation of $B_k x_i$ around the axis $y_k$ by an angle $\theta$, aligning it to $y_i$. $S_\varepsilon(y_i)$ denotes a spherical region centered at $y_i$ with an angular radius $\varepsilon$.

The trace of $B_k x_i$ in Figure 2 rotating around the axis $y_k$ forms a circle, which is denoted as:
\[ circ(B_k x_i, y_k) = \{ A_{\theta, y_k} B_k x_i | \theta \in [0, 2\pi] \}. \]  

As depicted in Figure 2, if $circ(B_k x_i, y_k)$ and $S_\varepsilon(y_i)$ do not intersect, $(x_i, y_i)$ cannot be aligned by any rotation $R_k$ and it is removed safely. If they intersect, $(x_i, y_i)$ is an inlier for $(x_k, y_k)$, and points $p_a$ and $p_b$ are the bounds of rotation $A_{\theta, y_k}$. Further, the bounds are denoted as an angle interval $\theta \in [\angle(B_k x_i, p_a), \angle(B_k x_i, p_b)] = [\theta^a, \theta^b]$.

Figure 3: Illustration of rotation bounds $\theta_i \in [\theta^a, \theta^b]$. The circle $circ(B_k x_i, y_k)$ denotes the trace of $B_k x_i$ as it rotates around the axis $y_k$, with its center point denoted as $x_i$. Describing the boundary of $S_\varepsilon(y_i)$, the circle $circ(\varepsilon, y_i)$ has its center denoted as $y_i$, as shown in the upper-right corner. Points $p_a$ and $p_b$ are the intersections of circles $circ(B_k x_i, y_k)$ and $circ(\varepsilon, y_i)$.

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

$$ (p - x_i)^T (p - x_i) = r_{x_i}^2, \quad (13) $$

where $p = [x, y, z_0]^T$ and $z_0$ is a constant value from $B_k x_i$.

Another circle, defining the boundary of $S_k(x_i)$ with an angular radius $\varepsilon$ centered at $y_k$, is denoted as:

$$ \text{circ}(\varepsilon, y_k) = \{ p | \angle(y_i, p) = \varepsilon, \| p \| = 1 \}. \quad (14) $$

The center of $\text{circ}(\varepsilon, y_k)$ is $y_i = \cos(\varepsilon) y_k = [x_1, y_1, z_1]^T$, and its radius $r_{y_i}' = \sin(\varepsilon)$. Point $p = [x, y, z]^T$ on the $\text{circ}(\varepsilon, y_k)$ satisfies:

$$ (p - y_i')^T (p - y_i') = r_{y_i}'^2. \quad (15) $$

To obtain the intersections between the two circles $\text{circ}(B_k x_i, y_k)$ and $\text{circ}(\varepsilon, y_k)$, combining (13) and (15), and letting $p = [x, y, z_0]^T$, the equation for line $p_a p_b$ is:

$$ y = ux + v, \quad u = -\frac{x_1}{y_1} (y_1 \neq 0), $$

$$ v = -\frac{1}{2y_1} (x_1^2 - r_{x_i}^2 + x_1^2 + y_1^2 + (z_0 - z_1)^2). \quad (16) $$

Further, (13) and (16) are combined to obtain $p_a$ and $p_b$:

$$ (1 + u^2) x^2 + 2uxv + (v^2 - r_{x_i}^2) = 0, \quad (17) $$

$$ \Delta = 4u^2v^2 - 4(1 + u^2)(v^2 - r_{x_i}^2). \quad (18) $$

If $\Delta < 0$, it indicates that the correspondence $(x_i, y_i)$ is an outlier and it is removed safely. Otherwise, let $p_a = \left[ \frac{-2u \pm \sqrt{\Delta}}{2(1 + u^2)}, \frac{-20u - \sqrt{\Delta}}{2(1 + u^2)}u + v + z_0 \right]^T$, and $p_b = \left[ \frac{-2u + \sqrt{\Delta}}{2(1 + u^2)}, \frac{-20u + \sqrt{\Delta}}{2(1 + u^2)}u + v + z_0 \right]^T$. The rotation bounds are obtained as $\theta_i = \{ \theta_i^a, \theta_i^b \}$ if $\theta_i^a \leq \theta_i^b$, where $\theta_i^a = \angle(B_k x_i, p_a)$, and $\theta_i^b = \angle(B_k x_i, p_b)$.

### Interval Stabbing

After pruning outlier correspondences for candidate $(x_k, y_k)$, 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 $\theta_k$ which is encompassed by maximum number of angle intervals within $\Theta_k$:

$$ \theta_k = \arg \max_{\theta_k \in [0, 2\pi]} \sum_i I(\theta_i \in [\theta_k^a, \theta_k^b]) \quad (19) $$

where $I(\cdot)$ is an indicator function, yielding 1 when the input is selected and 0 otherwise. Finally, $A_{\theta_k, y_k} B_k = R_k \in R_c$, where $R_c$ denotes the rotation set.

### Global Rotation Search

After obtaining $R_k \in R_c$ for each $(x_k, y_k)$, we globally search for the optimal relative rotation $R^*$:

$$ R^* = \arg \min_{R_k \in R_c} \sum_{i=1}^K \min \left( \angle(y_i, R_k x_i), \varepsilon \right). \quad (20) $$

Moreover, with the optimal $R^*$, for each correspondence $(P_k, Q_k)$, if $\angle(y_k, R^* x_k) > 2\varepsilon$, $(P_k, Q_k)$ is treated as an outlier and subsequently removed from $A$. The experiment shows that relaxing the threshold from $\varepsilon$ to $2\varepsilon$, without affecting $R^*$, 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 $\pi = [n', d']^T$, where $n = [n_x, n_y, n_z]$ 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 $p$ on the plane $\pi$ gives $d = -n^T p$. Distance between two parallel planes $\pi_i = [n_i', d_i]$ and $\pi_j = [n_i', d_j]$ is defined as:

$$ \text{dist}(\pi_i, \pi_j) = |d_i - d_j|. \quad (21) $$

With the optimal $R^*$, for each correspondence $(P_k, Q_k) \in A$, where $P_k \in P$ and $Q_k \in Q$, the normal $x_k$ from $P_k$ is aligned to the normal $y_k$ from $Q_k$ by $R^*$, denoted as follows:

$$ P_k' = (x_k', p_k') = R^* (x_k, p_k) = R^* P_k, \quad (22) $$

where $\angle(x_k, y_k) \leq 2\varepsilon$, and $(P_k', Q_k) \in A'$.

### Translation Components Computation

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

Assuming that there are two parallel inlier plane correspondences $(P_k', Q_k)$, $(P_i', Q_i) \in A'$, and $Q_k \parallel Q_i$. As shown in Figure 5(a), in the ideal case, the plane translation operator $[0^T, d]^T$ precisely aligns plane $P_k'$ to plane $Q_k$, and plane $P_i'$ to plane $Q_i$, denoted as:

$$ \pi_Q = \pi_{P_k'} + [0^T, d]^T, \quad \pi_Q = \pi_{P_i'} + [0^T, d]^T. \quad (23) $$

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

$$ \text{dist}(\pi_{P_k'}, \pi_Q) = \text{dist}(\pi_{P_i'}, \pi_Q) = |d|. \quad (24) $$

Moreover, assuming that plane $Q_i$ is aligned to plane $Q_k$ by the plane translation operator $[0^T, d]^T$, denoted as:

$$ \pi_{Q_k} = \pi_{Q_i} + [0^T, d]^T = \pi_{P_i'} + [0^T, d]^T + [0^T, d]^T = \pi_{P_i'} + [0^T, d]^T. \quad (25) $$

According to (25), one can infer that:

$$ \text{dist}(\pi_{Q_k}, \pi_{Q_i}) = \text{dist}(\pi_{P_i'}, \pi_{P_i'}) = |d|. \quad (26) $$
In the usual case involving noise and outliers, derived from (24) and (26), the plane translation constraints are defined as:

$$
\begin{align*}
|\text{dist}(\pi_{p''}, \pi_{Q_i}) - \text{dist}(\pi_{p'}, \pi_{Q_i})| &\leq \eta, \\
|\text{dist}(\pi_{p''}, \pi_{Q_i}) - \text{dist}(\pi_{Q_k}, \pi_{Q_i})| &\leq \eta,
\end{align*}
$$

(27)

where $\eta$ represents the distance threshold for the constraints.

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

$$
\hat{t}_l = \arg\min_{t_l = q_k - \text{Project}(\pi_{p''}, q_k)} |c_l| \sum_{k=1}^{c_l} (F_1 + F_2),
$$

$$
F_1 = \min \left( |\text{dist}(\pi_{p''}, \pi_{Q_k}) - \text{dist}(\pi_{p'}, \pi_{Q_i})|, \eta \right),
$$

$$
F_2 = \min \left( |\text{dist}(\pi_{p''}, \pi_{Q_k'}) - \text{dist}(\pi_{Q_k}, \pi_{Q_i})|, \eta \right),
$$

(28)

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

Global Translation Search

In accordance with (28), we have obtained the translation component set $T_c = \{ \hat{t}_l \}_{l=1}^{c}$, and each component $\hat{t}_l \in T_c$ is associated with a plane correspondence, denoted as $(\hat{p}_l, Q_l)$. As depicted in Figure 5(b), three different translation components $\hat{t}_u, \hat{t}_v, \hat{t}_w \in T_c$ determine a 3-DoF relative translation $\hat{t} \in \mathbb{R}^3$, denoted as follows:

$$
\begin{align*}
\hat{t} & = \hat{t}_u, \\
2\hat{t} & = \hat{t}_v - y_u \hat{t}_v y_u, \\
3\hat{t} & = \hat{t}_w^\top \cdot \frac{\hat{t}_u \times \hat{t}_v}{|\hat{t}_u \times \hat{t}_v|}, \\
\hat{t} & = 1\hat{t} + 2\hat{t} + 3\hat{t},
\end{align*}
$$

(29)

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

$$
\hat{t}^* = \arg\min_{\hat{t} \in T_c} \sum_{i=1}^{c} \min(\text{dist}(\pi_{p''}, \pi_{Q_i}) - |\hat{t}|, \eta).
$$

(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., 2022]. 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 al. 2016], and...
Table 1: Rotation errors $\delta_R$ (°) for baselines. Each column displays the top-ranking entry in bold, and the second-ranking entry is underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>ETH3D</th>
<th>ETH-TLS</th>
<th>WHU-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>FGR</td>
<td>0.104</td>
<td>0.264</td>
<td>0.598</td>
</tr>
<tr>
<td>PLADE</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>VGICP</td>
<td>4.121</td>
<td>22.59</td>
<td>6.652</td>
</tr>
<tr>
<td>Teaser++</td>
<td>0.104</td>
<td>0.264</td>
<td>0.598</td>
</tr>
<tr>
<td>VPFBR</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>PointDSC</td>
<td>0.304</td>
<td>0.007</td>
<td>0.598</td>
</tr>
<tr>
<td>RoReg</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>ETH3D</th>
<th>ETH-TLS</th>
<th>WHU-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>FGR</td>
<td>0.104</td>
<td>0.264</td>
<td>0.598</td>
</tr>
<tr>
<td>PLADE</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>VGICP</td>
<td>4.121</td>
<td>22.59</td>
<td>6.652</td>
</tr>
<tr>
<td>Teaser++</td>
<td>0.104</td>
<td>0.264</td>
<td>0.598</td>
</tr>
<tr>
<td>VPFBR</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>PointDSC</td>
<td>0.304</td>
<td>0.007</td>
<td>0.598</td>
</tr>
<tr>
<td>RoReg</td>
<td>0.220</td>
<td>0.171</td>
<td>0.014</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
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 $\varepsilon$ (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.

**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 estimation 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 $1.8 \times 10^8$ 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 $\varepsilon$ is a critical parameter in our algorithm, as depicted in Figure 2, directly influencing both rotation and translation accuracy. We evaluated the impact of $\varepsilon$ 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 $\varepsilon$ between 1.0° and 2.0°, rotation error remains stable, generally below 1.0°, and translation error stays under 1m. Notably, extremely small $\varepsilon (\leq 0.5^\circ)$ may overly suppress outliers, potentially reducing rotation and translation accuracy. Conversely, excessively large $\varepsilon (\geq 5.0^\circ)$ may compromise outlier removal, adversely affecting accuracy, too.

---

**Table 3: Average of rotation errors $\delta R$ (°), translation errors $\delta t$ (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.

<table>
<thead>
<tr>
<th>Method</th>
<th>ETH3D</th>
<th>ETH-TLS</th>
<th>WHU-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\delta R$</td>
<td>$\delta t$</td>
<td>time</td>
</tr>
<tr>
<td></td>
<td>$\delta R$</td>
<td>$\delta t$</td>
<td>time</td>
</tr>
<tr>
<td></td>
<td>$\delta R$</td>
<td>$\delta t$</td>
<td>time</td>
</tr>
<tr>
<td>FGR</td>
<td>30.42</td>
<td>5.054</td>
<td>183.6</td>
</tr>
<tr>
<td>PLADE</td>
<td>36.77</td>
<td>3.815</td>
<td>167.0</td>
</tr>
<tr>
<td>VGICP</td>
<td>2.762</td>
<td>8.527</td>
<td>11.42</td>
</tr>
<tr>
<td>Teaser++</td>
<td>30.90</td>
<td>4.736</td>
<td>20.90</td>
</tr>
<tr>
<td>VPFBR</td>
<td>30.08</td>
<td>0.345</td>
<td>7.593</td>
</tr>
<tr>
<td>PointDSC</td>
<td>29.08</td>
<td>10.42</td>
<td>3.291</td>
</tr>
<tr>
<td>RoReg</td>
<td>11.02</td>
<td>13.35</td>
<td>6.092</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.320</strong></td>
<td><strong>0.042</strong></td>
<td><strong>8.059</strong></td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Source</th>
<th>Target</th>
<th>RE</th>
<th>TE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input Size (byte)</td>
<td>Runtime (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>480</td>
<td>0.291</td>
<td>0.003</td>
<td></td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Feature Size (byte)</th>
<th>Runtime(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Source</td>
<td>Target</td>
</tr>
<tr>
<td>Teaser++</td>
<td>$6.8 \times 10^8$</td>
<td>$1.1 \times 10^8$</td>
</tr>
<tr>
<td>Ours</td>
<td>480</td>
<td>480</td>
</tr>
</tbody>
</table>

---

Figure 7: Ablation study for inter threshold $\varepsilon$. 

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

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

References


