Deterministic Point Cloud Registration via Novel Transformation Decomposition

Wen Chen[1,*] Haoang Li[1,*] Qiang Nie[1,2] Yun-Hui Liu[1,†]
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]Tencent Youtu Lab, Shenzhen, China

Abstract

Given a set of putative 3D-3D point correspondences, we aim to remove outliers and estimate rigid transformation with 6 degrees of freedom (DOF). Simultaneously estimating these 6 DOF is time-consuming due to high-dimensional parameter space. To solve this problem, it is common to decompose 6 DOF, i.e. independently compute 3-DOF rotation and 3-DOF translation. However, high non-linearity of 3-DOF rotation still limits the algorithm efficiency, especially when the number of correspondences is large. In contrast, we propose to decompose 6 DOF into (2+1) and (1+2) DOF. Specifically, (2+1) DOF represent 2-DOF rotation axis and 1-DOF displacement along this rotation axis. (1+2) DOF indicate 1-DOF rotation angle and 2-DOF displacement orthogonal to the above rotation axis. To compute these DOF, we design a novel two-stage strategy based on inlier set maximization. By leveraging branch and bound, we first search for (2+1) DOF, and then the remaining (1+2) DOF. Thanks to the proposed transformation decomposition and two-stage search strategy, our method is deterministic and leads to low computational complexity. We extensively compare our method with state-of-the-art approaches. Our method is more accurate and robust than the approaches that provide similar efficiency to ours. Our method is more efficient than the approaches whose accuracy and robustness are comparable to ours.

1. Introduction

Rigid point cloud registration is to align two 3D point clouds by an unknown-but-sought transformation with 6 degrees of freedom (DOF) [1]. It is a fundamental problem in computer vision and robotics. Existing methods can be classified into two categories in terms of whether the putative 3D-3D point correspondences (corrupted by outliers) are required. The methods that do not require correspondences typically rely on the initial value, which leads to low generality [6, 7], or provides low efficiency [36, 8]. To overcome these limitations, several methods requiring putative correspondences have been proposed [4, 24, 35, 38]. In practice, putative correspondences can be relatively easily obtained using the handcrafted or deep learning-based descriptors [11, 14, 29]. In this paper, we consider the cases that putative correspondences are known.

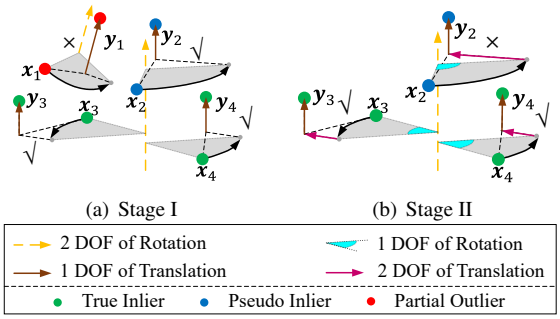


Figure 1. The proposed two-stage transformation estimation and outlier removal method. $(\mathbf{x}_i, \mathbf{y}_i)$ denotes a 3D-3D point correspondence. (a) In stage I, we search for 2 DOF of rotation, i.e. rotation axis, and 1 DOF of translation, i.e. displacement along the rotation axis. The optimal (2+1) DOF fit the most candidate inliers composed of both true and pseudo inliers. We save candidate inliers and discard partial outliers. (b) In stage II, we search for the remaining 1 DOF of rotation, i.e. rotation angle, and 2 DOF of translation, i.e. displacement orthogonal to rotation axis. The optimal (1+2) DOF fit the largest number of true inliers. We save true inliers and discard pseudo inliers.

For a set of putative correspondences, only inliers but not outliers can be fitted by the same transformation. Accordingly, point cloud registration with putative correspondences is commonly formulated as the inlier set maximization problem [20, 26]. Specifically, our problem aims to find the optimal 6-DOF transformation that fits the largest number of (inlier) correspondences. To tackle this problem, a straightforward way is to simultaneously search the 6 DOF [8, 27, 36]. However, their efficiency is unsatisfactory due to the high-dimensional search space. To overcome this limitation, a common strategy is to decompose the 6 DOF, i.e. independently compute 3-DOF rotation and 3-DOF translation [12, 16, 26, 31, 35]. However, existing decomposition methods have two main limitations. First, high non-linearity of 3 DOF of rotation still limits the al-

*Wen Chen and Haoang Li contributed equally to this work.
†Yun-Hui Liu is corresponding author.

gorithm efficiency [26, 31]. Second, in order to generate translation/rotation-invariant features, the size of input increases a lot [24, 35]. Another group of methods to solve the problem is based on sampling, such as RANSAC [12, 16]. But they are non-deterministic: outputs on independent trials with the same setup may be inconsistent. The inconsistency affects the algorithm robustness.

In contrast, we propose a deterministic and more efficient method based on a novel decomposition strategy. The proposed strategy first computes 2 DOF of rotation and 1 DOF of translation, followed by estimating the remaining 1 DOF of rotation and 2 DOF of translation. Specifically, as shown in Fig. 1(a), the 2 DOF of rotation and 1 DOF of translation refer to a 2-DOF rotation axis and a 1-DOF displacement along the rotation axis, respectively. We define "candidate" inliers as the observations fitted by the same 2 DOF of rotation and 1 DOF of translation. We search for these (2+1) DOF to fit as many candidate inliers (see blue and green correspondences) as possible. The search processing is based on branch and bound (BnB) [15]. Compared with searching for 3-DOF rotation with higher non-linearity [26, 31], the proposed search can provide tighter bound and thus improve the efficiency.

After obtaining 2 DOF of rotation and 1 DOF of translation, we compute the remaining 1 DOF of rotation and 2 DOF of translation. As shown in Fig. 1(b), the remaining 1 DOF of rotation and 2 DOF of translation represent 1-DOF rotation angle and 2-DOF displacement orthogonal to rotation axis, respectively. We define "true" inliers as a subset of candidate inliers that can be also fitted by the same 1 DOF of rotation and 2 DOF of translation. We search for these (1+2) DOF to fit as many true inliers (see green correspondences) as possible and discard pseudo-inliers (see blue correspondences). Our search in this stage is based on nested BnB. Attributed to relatively low computational complexity of nested BnB, our search achieves good efficiency.

Our main contributions are summarized as follows:

- We propose a novel strategy for transformation decomposition. It reduces the non-linearity of 3-DOF rotation, and thus improves the algorithm efficiency.
- We design a deterministic and efficient two-stage search strategy for parameter estimation. It leads to tight bounds and low computational complexity.

We extensively compare our method with state-of-the-art approaches. Our method is more accurate and robust than the approaches that provide similar efficiency to ours. Our method is more efficient than the approaches whose accuracy and robustness are comparable to ours.

2. Related Works

We classify existing works into two main categories in terms of whether decomposition of the 6-DOF transformation is conducted.

Methods without Decomposition. Several optimization algorithm-based methods simultaneously estimate 6-DOF transformation. They typically leverage BnB to search over a bounded 6D parameter space [15, 5, 8, 27, 36]. By computing the bounds of the objective functions, these methods continuously narrow down the search scope until the optimal parameters are found. While they achieve global optimality in terms of their objectives, their efficiency is unsatisfactory due to the high-dimensional parameter space. Instead of searching on the 6-DOF space, Zhou et al. [38] proposed to linearize the transformation as a 6D vector, and estimated this vector based on a robust cost function. This approach improves the efficiency, but can hardly handle the cases with high outlier ratios.

Some deep learning-based methods simultaneously estimate 6-DOF transformation in an end-to-end manner. They use different parameterization methods to represent the transformation. PointNetLK [2] represents the transformation by a 6D twist that can be regarded as a mapping in Lie group. DGR [10] expresses the rotation in a redundant 6D continuous space [40] and translation in a 3D space. Accordingly, transformation is encoded by a 9D vector. Lee et al. [19] proposed to map transformation into a 6D Hough space that facilities the cast voting-based parameter estimation. While these methods provide high accuracy on certain datasets, their generalization is relatively unsatisfactory.

Methods Based on Decomposition. Independently estimating 3-DOF rotation and 3-DOF translation is commonly used in the closed-form solvers [3, 16]. In presence of outliers, several methods integrate these solvers into RANSAC [12, 28] to conduct iterative sampling. At each iteration, they first use a solver and samples to hypothesize a transformation, and then verify this hypothesis. In addition, some approaches [24, 35] conduct exhaustive sampling to enumerate correspondence primitives. They search for the "invariant" primitives that satisfy the constraints with respect to only rotation or translation. For example, Liu et al. [24] first found a set of rotation-invariant primitives to compute the translation. Then they fixed the translation and computed the rotation. Yang et al. [35] first searched for a set of translation-invariant primitives to compute rotation, followed by computing translation. The decomposition-based methods are generally faster than the above approaches without decomposition. However, due to numerous samplings and high non-linearity of 3-DOF rotation, there is still room for efficiency improvement.

Different from the aforementioned methods that use sampling to identify inlier correspondences, some deep learning-based methods [22, 4, 25, 33] directly predict inliers using neural networks. Then based on these inliers, they employ the closed-form solvers to compute the transformation. A representative work of the deep learning-

based methods is PointDSC [4]. It considers the spatial consistency of correspondences to improve the accuracy of inlier prediction. These methods are not deterministic, which affects the algorithm robustness.

3. Problem Formulation

We formulate point cloud registration as an inlier set maximization problem in Section 3.1. To tackle this problem efficiently, we decompose it into two novel sub-problems based on transformation decomposition in Section 3.2. We discuss the optimality in Section 3.3.

3.1. Inlier Set Maximization

Similar to [4, 24], we assume that N putative 3D-3D point correspondences corrupted by outliers are known in our problem setting. In the noise-free case, an inlier correspondence $(\mathbf{x}_i, \mathbf{y}_i)$ $(1 \leq i \leq N)$ is strictly fitted by the unknown-but-sought transformation $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$, i.e. $\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i = \mathbf{0}$. In presence of noise, we rewrite this constraint as $\|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \xi$, where ξ is an inlier threshold. We aim to find the optimal transformation that fits the largest number of (inlier) correspondences, i.e.

$$\begin{aligned} \max_{\mathbf{R},\,\mathbf{t}} \quad & \lfloor \mathbb{I} \rfloor \\ \text{s.t.} \quad & \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \xi, \ \forall i \in \mathbb{I} \end{aligned} \tag{1}$$

where $\mathbb{I}$ is the index set of the inlier correspondences, "$\lfloor \cdot \rfloor$" denotes the cardinality of set, and "$\| \cdot \|$" denotes the L_2 norm. Following [15], we also represent the rotation using the angle-axis representation. We denote the rotation axis by $\mathbf{r}$ and the rotation angle by θ. Note that, we consider the purely rotational registration in the supplementary material.

3.2. New Transformation Decomposition

For problem in Eq. (1), simultaneously or independently estimating 3-DOF rotation $\mathbf{R}$ and 3-DOF translation $\mathbf{t}$ are both time-consuming due to high-dimensional parameter space or high non-linearity of rotation (see Section 1). To solve this issue, the original problem in Eq. (1) is decomposed into two new sub-problems based on transformation decomposition. Specifically, we first compute 2 DOF of rotation and 1 DOF of translation, followed by the remaining 1 DOF of rotation and 2 DOF of translation. We begin with introducing a lemma used in the following.

Lemma 3.1. *For any 3D point $\mathbf{p}$, it satisfies a constraint with respect to the rotation $\mathbf{R}$ and rotation axis $\mathbf{r}$ of this rotation, i.e.*

$$\mathbf{r}^\top (\mathbf{R}\mathbf{p}) = \mathbf{r}^\top \mathbf{p}. \tag{2}$$

Proof is available in the supplementary material. This lemma is the basis of our transformation decomposition.

First sub-problem. We reduce the inlier constraint $\|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \xi$ in Eq. (1) as

$$\begin{aligned} & \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \xi && \text{(3a)} \\ \Leftrightarrow \quad & \|\mathbf{r}\| \cdot \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \|\mathbf{r}\| \cdot \xi && \text{(3b)} \\ \Rightarrow \quad & |\mathbf{r}^\top (\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i)| \leq \xi && \text{(3c)} \\ \Leftrightarrow \quad & |\mathbf{r}^\top (\mathbf{x}_i - \mathbf{y}_i) + \underbrace{\mathbf{r}^\top \mathbf{t}}_{d}| \leq \xi && \text{(3d)} \end{aligned}$$

where Eq. (3c) is based on the Cauchy–Schwarz inequality[1] and $\|\mathbf{r}\| = 1$; Eq. (3d) is based on Lemma 3.1; "$|\cdot|$" denotes the absolute value; $d = \mathbf{r}^\top \mathbf{t}$ in Eq. (3d) geometrically represents the projection of translation onto the rotation axis, i.e. the 1D displacement along the rotation axis (see Fig. 1(a)).

The constraint definded by Eq. (3d) is with respect to 2 DOF of rotation $\mathbf{r}$ and 1 DOF of translation d. Based on this constraint, the first sub-problem is derived and defined. Namely, given the input correspondences, we aim to find the optimal $\mathbf{r}$ and d to maximize the number of inliers, i.e.

$$\begin{aligned} \max_{\mathbf{r},d} \quad & \lfloor \mathbb{I}_1 \rfloor \\ \text{s.t.} \quad & |\mathbf{r}^\top (\mathbf{x}_i - \mathbf{y}_i) + d| \leq \xi, \ \forall i \in \mathbb{I}_1 \end{aligned} \tag{4}$$

where $\mathbb{I}_1$ is the index set of the identified inliers. Note that Eq. (4) is only with respect to (2+1) DOF of the transformation. Accordingly, we treat the identified inliers in the first sub-problem as "candidate" inliers (see Fig. 1(a)). From another perspective, some outliers, which coincidentally satisfy this constraint, cannot be identified. We call these outliers "pseudo" inliers. These pseudo inliers will be removed in the second sub-problem introduced in the following.

Second sub-problem. By fixing the rotation axis $\mathbf{r}$ and displacement d estimated in the first sub-problem, the inlier constraint $\|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \leq \xi$ in Eq. (1) can be simplified as

$$\|\mathbf{R}_\theta \mathbf{x}_i + \mathbf{t}' - \mathbf{y}_i\| \leq \xi \tag{5}$$

where $\mathbf{R}_\theta$ is with respect to the remaining 1-DOF rotation angle θ around the known axis $\mathbf{r}$, and $\mathbf{t}'$ is a 2-DOF translation whose component of displacement along the rotation axis is known, but component of displacement orthogonal to the rotation axis is unknown. Based on this constraint, we define the second sub-problem. That is, given the candidate inliers, we aim to find the optimal θ and $\mathbf{t}'$ to maximize the number of true inliers, i.e.

$$\begin{aligned} \max_{\theta,\mathbf{t}'} \quad & \lfloor \mathbb{I}_2 \rfloor \\ \text{s.t.} \quad & \|\mathbf{R}_\theta \mathbf{x}_i + \mathbf{t}' - \mathbf{y}_i\| \leq \xi, \ \forall i \in \mathbb{I}_2 \end{aligned} \tag{6}$$

where $\mathbb{I}_2$ is the index set of true inliers. Eq. (6) is with respect to the remaining (1+2) DOF of the transformation. Pseudo inliers fail to satisfy this constraint. In another word, we can prune pseudo inliers from candidate inliers (see Fig. 1(b)), and thus retain only true inliers.

To solve the above two sub-problems, we propose a two-stage search strategy, which will be introduced in Section 4.

[1] $|\mathbf{a} \cdot \mathbf{b}| = \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot |\cos(\angle(\mathbf{a}, \mathbf{b}))| \leq \|\mathbf{a}\| \cdot \|\mathbf{b}\|$

3.3. Discussion about Optimality

In this section, we discuss the global optimality in terms of maximizing the number of inliers. Recall that at the stage I, we aim to find a model that fits the largest number of candidate inliers. Ideally, for these candidate inliers, all the true inliers are incorporated, and all the pseudo inliers can be pruned at the second stage. However, in practice, the identified candidate inliers may neglect partial true inliers, and also incorporate some pseudo inliers that cannot be pruned at the second stage. The reason is that the number of the neglected partial true inliers is smaller than the number of mistakenly incorporated pseudo inliers (our goal is to maximize the carnality of candidate inlier set). Therefore, our method indeed cannot theoretically guarantee global optimality in terms of maximizing the number of (true) inliers. Please note that this is also an inherent flaw shared by the other approaches based on transformation decomposition [24, 32, 35].

However, the above sub-optimal case only occurs when the true inliers and some outliers are not distinct enough. Otherwise, our method can (nearly) identify all the true inliers, which has been validated by the experiments in Section 5. In addition, to facilitate understanding of the above optimality discussion, we provide an analogy of line fitting in the supplementary material.

4. Two-stage Search Strategy

We begin with illustrating our parameter space of transformation in Section. 4.1. Then, we introduce our first stage search regarding 2 DOF of rotation and 1 DOF of translation in Section 4.2, and our second stage search regarding the remaining 1 DOF of rotation and 2 DOF of translation in Section 4.3.

4.1. Parameter Space of Rotation and Translation

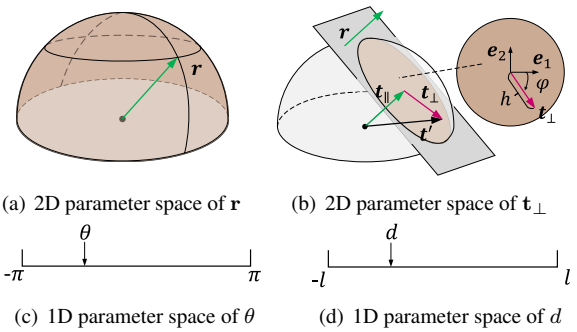
Rotation. As shown in Figs. 2(a) and 2(c), we follow [21, 23] to model the parameter space of the rotation axis $\mathbf{r}$ by a unit hemisphere, and the parameter space of rotation angle θ by the interval $[-\pi, \pi]$. To facilitate our search-based estimation of rotation axis (introduced in Section 4.2), we discretize the hemisphere. Our discretization is based on the Miller's method [30] which is a variant of common longitude-latitude discretization. Miller's method improves the efficiency and accuracy of our rotation axis estimation, as shown in the supplementary material.

Translation. Following [36], we assume that the norm of translation is smaller than a threshold l[2]. Accordingly, the original parameter space of $\mathbf{t}$ corresponds to a solid ball whose radius is l. Recall that we decompose the translation $\mathbf{t}$ into d in Eq. (3d) and $\mathbf{t}'$ in Eq. (5). We thus decompose the above ball-shaped parameter space, which is introduced as follows. As shown in Fig. 2(d), we model the parameter space of d by the 1D interval $[-l, l]$. The parameter space of $\mathbf{t}'$ is modeled based on two geometric constraints (see Fig. 2(b)). First, $\mathbf{t}'$ lies within the above ball-shaped space. Second, $\mathbf{t}'$ lies on the plane defined by $\mathbf{r}^\top \mathbf{t}' = d$ ($\mathbf{r}$ and d are known based on the first-stage search). Therefore, the intersection of the above ball-shaped and plane-shaped spaces corresponds to the parameter space of $\mathbf{t}'$, a disk-shaped space. We express $\mathbf{t}'$ by the component parallel to the rotation axis $\mathbf{t}_\parallel$ and the component orthogonal to the axis $\mathbf{t}_\perp$, i.e. $\mathbf{t}' = \mathbf{t}_\parallel + \mathbf{t}_\perp$. On the one hand, $\mathbf{t}_\parallel$ corresponds to the vector defined by the center of ball and the center of disk, i.e. $\mathbf{t}_\parallel = d\mathbf{r}$. On the other hand, $\mathbf{t}_\perp$ represents a vector whose initial point is the center of disk and terminal point lies on the disk. We express $\mathbf{t}_\perp$ by

$$\mathbf{t}_\perp = (h\cos\varphi)\cdot\mathbf{e}_1 + (h\sin\varphi)\cdot\mathbf{e}_2 \tag{7}$$

where $\{\mathbf{e}_1, \mathbf{e}_2\}$ is an arbitrary known orthogonal basis of the null space of the axis $\mathbf{r}$ and (φ, h) are unknown-but-sought polar coordinates in the disk-shaped space.

In the following, we first search for 2 DOF of rotation, i.e. $\mathbf{r}$ and 1 DOF of translation, i.e. d (we can obtain $\mathbf{t}_\parallel = d\mathbf{r}$ accordingly). Then we search for the remaining 1 DOF of rotation, i.e. θ and 2 DOF of translation, i.e. $\mathbf{t}_\perp$ (with respect to polar coordinates (φ, h)).



(a) 2D parameter space of $\mathbf{r}$ (b) 2D parameter space of $\mathbf{t}_\perp$

(c) 1D parameter space of θ (d) 1D parameter space of d

Figure 2. The parameter space of rotation and translation. (a) $\mathbf{r}$ denotes the 2 DOF of rotation, i.e. the rotation axis. We discretize the hemisphere based on Miller's method [30], i.e. a non-uniform latitude discretization. (b) $\mathbf{t}_\perp$ represents the 2-DOF displacement orthogonal to the rotation axis. (c) θ denotes the 1 DOF of rotation, i.e. the rotation angle. (d) d represents the 1-DOF displacement along the rotation axis.

4.2. Stage I: Searching for (2+1) DOF

We first search for 2 DOF of rotation, i.e. rotation axis $\mathbf{r}$, and 1 DOF of translation, i.e. displacement d along the rotation axis. Our search is based on BnB [15] that continuously divides the search space, and computes the upper and lower bounds of the cost function for each sub-space. A sub-space is pruned if its associated bounds prove it does not contain the optimal solution. BnB converges when the

[2]We follow [38] to normalize the point cloud. Accordingly, the variable l is equal to 1.

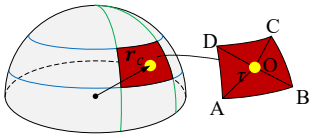


Figure 3. The bound computation for 2-DOF rotation axis. The red *trapezoid cap* on the hemi-shpere represents a sub-space of the parameter space of rotation axis $\mathbf{r}$. The maximum value of $\|\mathbf{r}-\mathbf{r}_c\|$ in the sub-space is $\|OA\|$ or $\|OB\|$.

upper and lower bounds are equal up to a threshold. The tightness of bounds determines the algorithm convergence. In our context, for the problem of inlier set maximization, lower bound can be easily computed based on any position in the parameter space [36]. We aim to derive tight upper bound to improve the algorithm efficiency as follows.

Let a trapezoid cap in Fig. 3 be a sub-space of the parameter space of the 2-DOF rotation axis $\mathbf{r}$. We denote the center of this cap by $\mathbf{r}_c$. Let an interval $\left[\underline{d}, \overline{d}\right]$ be a sub-space of the parameter space of the 1-DOF displacement d. We denote the center of this interval by $\dot{d}$. We take these sub-spaces and a correspondence $(\mathbf{x}_i, \mathbf{y}_i)$ for example to derive our upper bound of the constraint of candidate inlier in Eq. (4). For arbitrary $\mathbf{r}$ and d in their sub-spaces, we have

$$|\mathbf{r}^\top(\mathbf{x}_i-\mathbf{y}_i)+d| \tag{8a}$$
$$=|\mathbf{r}_c^\top(\mathbf{x}_i-\mathbf{y}_i)+\dot{d}+(\mathbf{r}-\mathbf{r}_c)^\top(\mathbf{x}_i-\mathbf{y}_i)+(d-\dot{d})| \tag{8b}$$
$$\le|\mathbf{r}_c^\top(\mathbf{x}_i-\mathbf{y}_i)+\dot{d}|+|(\mathbf{r}-\mathbf{r}_c)^\top(\mathbf{x}_i-\mathbf{y}_i)|+|d-\dot{d}| \tag{8c}$$
$$\le\xi+\|\mathbf{r}-\mathbf{r}_c\|\cdot\|\mathbf{x}_i-\mathbf{y}_i\|+|d-\dot{d}| \tag{8d}$$
$$\le\xi+\tau\|\mathbf{x}_i-\mathbf{y}_i\|+\delta_d \tag{8e}$$

where Eq. (8c) is owing to the triangle inequality; Eq. (8d) relies on the Cauchy–Schwarz inequality; In Eq. (8e), $\tau=\max(\|\mathbf{r}-\mathbf{r}_c\|)$ represents the maximum distance from center of cap $\mathbf{r}_c$ to all the points within this cap, and $\delta_d=\max(|d-\dot{d}|)$ is the maximum displacement from the center $\dot{d}$ to all points in the interval, i.e. δ_d is the half length of the interval of d. In the following, we introduce the geometric meaning of τ.

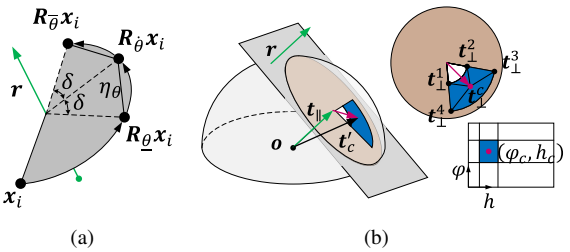
The specific value of τ only relies on the relative distance from the center to other points in the *trapezoid cap*. As illustrated in Fig. 3, O is the center of the *trapezoid cap*. We note that the shortest distances from the center O to the edge AB and CD are equal. Since the parallel lines on the unit sphere with higher latitudes have smaller perimeter, the length of AB is larger than the length of CD. Then, we can conclude that both $\|OA\|$ and $\|OB\|$ maximize the term $\|\mathbf{r}-\mathbf{r}_c\|$ in the *trapezoid cap*.

According to the derivation from (8a) to (8e), the residual of $|\mathbf{r}^\top(\mathbf{x}_i-\mathbf{y}_i)+d|$ (for any $\mathbf{r}$ in the cap and d in the interval) will be at most $\xi+\tau\|\mathbf{x}_i-\mathbf{y}_i\|+\delta_d$. In other words, the upper bound is valid if we choose the values of $\mathbf{r}$ and d as their centers $\mathbf{r}_c$ and $\dot{d}$ respectively. Thus, given the initial putative correspondences, we compute the upper bound $\overline{Q}_1$ of the number of candidate inliers as

$$\overline{Q}_1=\sum\mathbb{1}(|\mathbf{r}_c^\top(\mathbf{x}_i-\mathbf{y}_i)+\dot{d}|\le\xi+\tau\|\mathbf{x}_i-\mathbf{y}_i\|+\delta_d) \tag{9}$$

where $\mathbb{1}(\cdot)$ is an indicator function that returns 1 if the inside condition $(\cdot)$ is true and 0 otherwise. As mentioned above, we choose centers of sub-spaces to compute lower bound of the number of candidate inliers by

$$\underline{Q}_1=\sum\mathbb{1}(|\mathbf{r}_c^\top(\mathbf{x}_i-\mathbf{y}_i)+\dot{d}|\le\xi) \tag{10}$$



(a) (b)

Figure 4. The bound computation for the second sub-problem defined in Eq. (6). (a) Illustration of the point $\mathbf{R}_\theta\mathbf{x}_i$ which is generated by rotating $\mathbf{x}_i$ by the rotation angle θ around fixed rotation axis $\mathbf{r}$. θ is in a sub-space $\theta\in\left[\underline{\theta},\overline{\theta}\right]$ centered at $\dot{\theta}$ and with half-length of δ. The upper bound of $\|(\mathbf{R}_\theta\mathbf{x}_i-\mathbf{R}_{\dot{\theta}}\mathbf{x}_i)\|$ in this sub-space is the distance from $\mathbf{R}_{\dot{\theta}}\mathbf{x}_i$ to $\mathbf{R}_{\underline{\theta}}\mathbf{x}_i$ or $\mathbf{R}_{\overline{\theta}}\mathbf{x}_i$. (b) Illustration of a sub-space (blue sectors) of the parameter space of $\mathbf{t}_\perp$. The maximum value of $\|\mathbf{t}_\perp-\mathbf{t}_\perp^c\|$ in the sector-shaped space is the maximum distance from the center to four corners.

4.3. Stage II: Searching for Remaining (1+2) DOF

By fixing 2 DOF of rotation $\mathbf{r}$ and 1 DOF of translation d estimated in stage I, we search for the remaining 1 DOF of rotation θ and 2 DOF of translation $\mathbf{t}_\perp$. Similar with the stage I, we search for the remaining (1+2) DOF based on BnB. Specifically, inspired by [36], we design a nested BnB composed of the outer and inner BnB. We search for $\mathbf{t}_\perp$ by outer BnB, and θ by inner BnB. Outer BnB divides the search space with respect to $\mathbf{t}_\perp$ into four 2D sub-spaces, and passes each sub-space to inner BnB. Inner BnB searches on the whole 1D space with respect to θ, and returns the bounds on each sub-space to outer BnB. Compared with directly searching on a 3D non-linear space [15], our outer and inner BnB search on 2D and 1D linear spaces, respectively. Accordingly, our nested structure improves the efficiency. In the following, we first derive the bounds of parallel BnB (that divides the parameter spaces of $\mathbf{t}_\perp$ and θ simultaneously). Then based on these bounds, we compute the bounds of our nested BnB.

Bounds of Parallel BnB. Let an interval $\theta\in\left[\underline{\theta},\overline{\theta}\right]$ be a sub-space of rotation angle θ. We denote the center of the interval by $\dot{\theta}$. Let a local rectangular region (centered at (φ_c, h_c)) be a sub-space of displacement $\mathbf{t}_\perp$. The associated

local space in the plane $\mathbf{r}^\top \mathbf{t} = d$ is a sector (see Fig. 4(b)). We denote the center of the sector by $\mathbf{t}_\perp^c$. Accordingly, the center of $\mathbf{t}'$ is derived by $\mathbf{t}'_c = \mathbf{t}_\parallel + \mathbf{t}_\perp^c$. We take these sub-spaces and a correspondences $(\mathbf{x}_i, \mathbf{y}_i)$ for example to derive our upper bound of the constraint of true inlier in Eq. (6). For arbitrary θ and $\mathbf{t}_\perp$ in their sub-spaces, we have

$$\|\mathbf{R}_\theta \mathbf{x}_i + \mathbf{t}' - \mathbf{y}_i\| \tag{11a}$$

$$= \|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i + \left(\mathbf{R}_\theta \mathbf{x}_i - \mathbf{R}_{\dot{\theta}} \mathbf{x}_i\right) + (\mathbf{t}' - \mathbf{t}'_c)\| \tag{11b}$$

$$\leq \|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i - \mathbf{y}_i\| + \|\left(\mathbf{R}_\theta \mathbf{x}_i - \mathbf{R}_{\dot{\theta}} \mathbf{x}_i\right)\| + \|\mathbf{t}' - \mathbf{t}'_c\| \tag{11c}$$

$$= \|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i - \mathbf{y}_i\| + \|\left(\mathbf{R}_\theta \mathbf{x}_i - \mathbf{R}_{\dot{\theta}} \mathbf{x}_i\right)\| + \|\mathbf{t}_\perp - \mathbf{t}_\perp^c\| \tag{11d}$$

$$\leq \xi + \eta_\theta + \eta_t \tag{11e}$$

where Ineq. (11c) follows from the triangle inequality; Eq. (11d) is based on $\mathbf{t}' = \mathbf{t}_\parallel + \mathbf{t}_\perp$ and $\mathbf{t}'_c = \mathbf{t}_\parallel + \mathbf{t}_\perp^c$. In Ineq. (11e), $\eta_\theta = \max(\|\mathbf{R}_\theta \mathbf{x}_i - \mathbf{R}_{\dot{\theta}} \mathbf{x}_i\|)$ denotes the maximum distance from the center $\mathbf{R}_{\dot{\theta}} \mathbf{x}_i$ to the all point $\mathbf{R}_\theta \mathbf{x}_i$; and $\eta_t = \max(\|\mathbf{t}_\perp - \mathbf{t}_\perp^c\|)$ represents the maximum distance from the center to all the points in the sector.

We specify the values of η_θ and η_t in their sub-spaces as follows. As shown in Fig. 4(a), η_θ can be derived as:

$$\|\left(\mathbf{R}_\theta \mathbf{x}_i - \mathbf{R}_{\dot{\theta}} \mathbf{x}_i\right)\| \leq \sqrt{2 - 2\cos\delta}\|\mathbf{x}_i\| \doteq \eta_\theta \tag{12}$$

where $\delta = \dot{\theta} - \underline{\theta}$ is the half length of the interval of θ. As shown in Fig. 4(b), the sector is symmetrical. The relative distances from the center $\mathbf{r}_\perp^c$ to arbitrary point $\mathbf{t}_\perp$ and its symmetry point in the sector are equal, i.e. $\|\mathbf{t}_\perp^1 - \mathbf{t}_\perp^c\| = \|\mathbf{t}_\perp^2 - \mathbf{t}_\perp^c\|$ and $\|\mathbf{t}_\perp^3 - \mathbf{t}_\perp^c\| = \|\mathbf{t}_\perp^4 - \mathbf{t}_\perp^c\|$. Since the length of $\mathbf{t}_\perp^4$ is bigger than the length of $\mathbf{t}_\perp^1$, the maximum distance of $\|\mathbf{t}_\perp - \mathbf{t}_\perp^c\|$ is attained at $\mathbf{t}_\perp^3$ and $\mathbf{t}_\perp^4$. With the specific value of η_θ and η_t, we compute the upper bound $\overline{Q}_2$ and lower bound $\underline{Q}_2$ of the true inliers at the center by

$$\overline{Q}_2 = \sum \mathbb{1}(|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi + \eta_\theta + \eta_t) \tag{13a}$$

$$\underline{Q}_2 = \sum \mathbb{1}(|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi) \tag{13b}$$

Bounds of Our Nested BnB. We extend the above parallel BnB to our nested BnB as follows. Bounds in Eqs. (13a) and (13b) of parallel BnB are with respect to both $\mathbf{t}_\perp$ and θ. We extract the part only with respect to $\mathbf{t}_\perp$ for outer BnB, obtaining the bounds

$$\overline{Q}_{2,\mathbf{t}_\perp} = \sum \mathbb{1}(|\mathbf{R}_\theta \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi + \eta_t) \tag{14a}$$

$$\underline{Q}_{2,\mathbf{t}_\perp} = \sum \mathbb{1}(|\mathbf{R}_\theta \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi) \tag{14b}$$

We extract the part only with respect to θ for inner BnB, obtaining the bounds

$$\overline{Q}_{2,\theta} = \sum \mathbb{1}(|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi + \eta_\theta + \eta_t) \tag{15a}$$

$$\underline{Q}_{2,\theta} = \sum \mathbb{1}(|\mathbf{R}_{\dot{\theta}} \mathbf{x}_i + \mathbf{t}'_c - \mathbf{y}_i\| \leq \xi + \eta_t) \tag{15b}$$

5. Experiments

We first compare our method with state-of-the-art approaches for the *pairwise* registration task. We report experimental results on synthetic and real-world datasets in Sections 5.1 and 5.2, respectively. Then we evaluate various methods for the *multi-way* registration task in Section 5.3. We implement our method in C++, and conduct experiments on a PC with a 3.70GHz CPU and an NVIDIA GeForce RTX2060 GPU.

Methods to Compare. We denote our method based on TRansformation DEcomposition by *TR-DE*. We compare it with the state-of-the-art approaches introduced in Section 2:

- *RANSAC-xk* [4, 12][3]: A classical method based on RANSAC with *xk* iterations, including *RANSAC-1k*, *RANSAC-10k*, and *RANSAC-100k*;
- *PointDSC* [4]: A deep learning-based method leveraging the spatial consistency. We test the officially provided network pre-trained on 3DMatch dataset [37].
- *GORE* [26]: A guaranteed method that separately estimates rotation and translation based on BnB and exhaustive sampling, respectively.
- *FGR* [38]: A fast method that simultaneously computes rotation and translation by robust cost function;
- *TEASER* [35]: A certifiable method that first estimates rotation by translation-invariant feature and robust cost function, and then translation by adaptive voting.

RANSAC-xk and *PointDSC* are non-deterministic due to sampling uncertainty, i.e. their outputs on independent trials with the same setup may be inconsistent. In contrast, *FGR*, *TEASER* and *GORE* are deterministic.

Evaluation Metrics. Following [4, 10, 35], we evaluate the algorithm performance based on 1) *Rotation Error (RE)*, 2) *Translation Error (TE)*, 3) *Success Rate (SR)*[4], and 4) $F1$-score. The successful case with respect to *SR* represents that *RE* and *TE* are lower than thresholds defined in [4]. $F1$-score is the harmonic mean of precision and recall. We use it to evaluate the quality of the identified inliers.

5.1. Comparison on Synthetic Dataset

Data Generation. Similar to [34, 26], we synthesize several points within a unit cube, and treat them as the source point cloud. We add zero-mean Gaussian noise whose standard deviation is 0.005. By moving the source point cloud with a random transformation, we generate the target point cloud. A pair of the original and moved points defines an inlier correspondence. We contaminate inlier correspondences by outliers generated by random transformations. We vary the number of correspondences N and outlier ratio ρ. Specifically, we conduct robustness comparisons by varying ρ from 55% to 95% and fixing N as 3000. In ad-

[3] "k" represents 1000 in our context.

[4] *SR* is equivalent to Registration Recall (RR) used in [4, 10, 35]

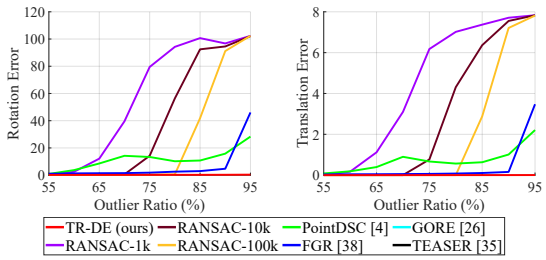


Figure 5. Robustness comparisons with respect to the outlier ratio. Left: mean of *RE*. Right: mean of *TE*. The curves of *TEASER*, *GORE* and our *TR-DE* are flat and difficult to visually distinguish, demonstrating that they are all robust to outliers.

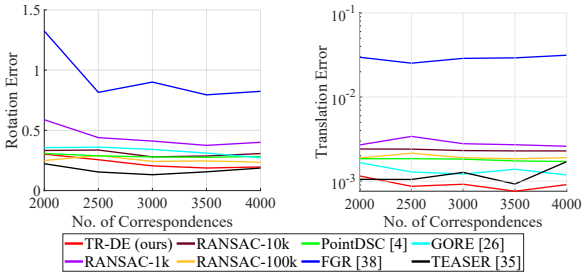


Figure 6. Accuracy comparisons with respect to the number of correspondences. Left: mean of *RE*. Right: mean of *TE*.

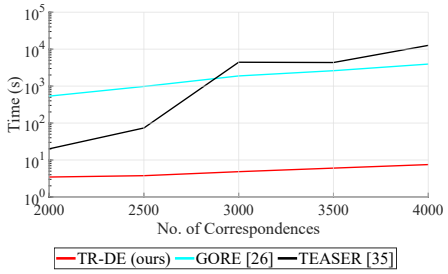
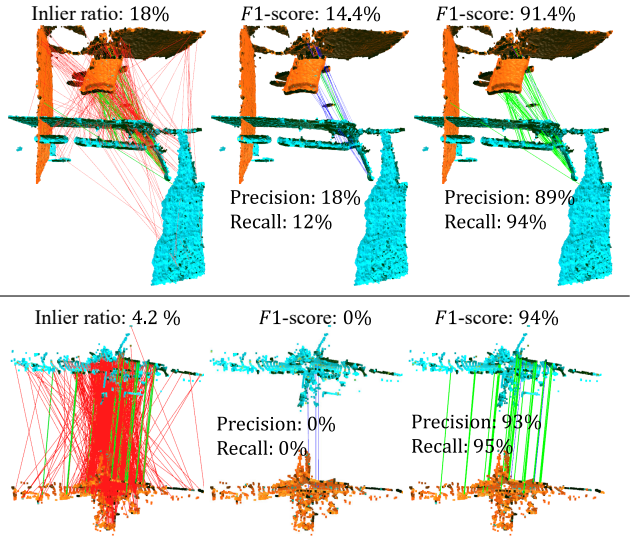


Figure 7. Efficiency comparison between our *TR-DE* and deterministic methods with respect to the number of correspondences.

dition, we conduct accuracy and efficiency comparisons by increasing N from 2000 to 4000 and fixing ρ as 55%.

Robustness. Under each outlier ratio ρ, we conduct 200 independent trials and reports the average result. As shown in Figs. 5(a) and 5(b), the errors of a series of *RANSAC* increase sharply as the outlier ratio grows. Even in the low outlier ratio such as 65%, *RANSAC-1k* may lead to registration failure. Similar to *RANSAC*, *PointDSC* follows the hypothesize-and-verify pipeline, and thus is prone to result in incorrect registration from the outlier ratio of 65%. In contrast, the deterministic methods such as *GORE*, *TEASER* and our *TR-DE* provide low error under high outlier ratios. An exception is that *FGR* fails to deal with the outlier ratios higher than 85%.

Accuracy and Efficiency. Under each data size N, we conduct 200 independent trials. As shown in Figs. 6(a) and 6(b), in terms of accuracy, a series of *RANSAC* lead to relatively low accuracy due to the noise of samples. The accuracy of *PointDSC* is unsatisfactory since our synthetic



(a) Ground truth (b) *RANSAC-1k* [12] (c) *TR-DE* (ours)

Figure 8. Comparisons between *RANSAC-1k* and our *TR-DE* on representative point cloud pairs from 3DMatch (first row) and KITTI (second row). True positive, true negative and false positive are shown in green, red and blue, respectively.

dataset is different from the dataset used to train *PointDSC*. For the deterministic methods, the accuracy of *GORE* and *FGR* is lower than that of *TEASER* and our *TR-DE*. Compared with *TEASER*, our *TR-DE* computes more accurate translation but less accurate rotation. In the following, for the methods with high accuracy, i.e. *TEASER*, *GORE* and our *TR-DE*, we compare their efficiency. Figure 7 shows that the run time of *GORE* and *TEASER* increase significantly as the number of correspondences grows. In contrast, the efficiency of our method is less affected by the number of correspondences. Particularly, when N grows to 4000, the run time of *TEASER* and *GORE* are about 10^3 times that of *TR-DE*. Overall, our *TR-DE* achieves the best efficiency among the deterministic methods while keeping competitive accuracy.

5.2. Comparison on Real-world Datasets

We conduct comparisons on the indoor RGB-D 3DMatch [37], indoor low-overlap 3DLoMatch [17], and outdoor large-scale KITTI [13] datasets. Figure 8 shows the qualitative results on two representative point clouds from 3DMatch and KITTI.

3DMatch Dataset [37]. The test set of the 3DMatch dataset which contains RGB-D scans from eight different scenes is utilized in our comparisons. Following [35, 4], we downsample the dense point clouds with a 5 cm voxel grid filter and set the inlier threshold to 10 cm. The thresholds for successful registration are $RE \leq 10^\circ$ and $TE \leq 30$ cm [35]. Average *RE* and *TE* are computed only on the successfully registered pairs [4, 10].

Following [35], we use 3DSmoothNet [14] to compute

Table 1. Accuracy and efficiency comparison on 3DMatch dataset [37] with 3DSmoothNet [14] descriptors following [35].

	SR (%)	*RE* (°)	*TE* (cm)	*F1* (%)	Time (s)
FGR [38]	73.26	2.51	7.45	-	0.03
TEASER [35]	91.99	2.01	6.29	87.59	0.13
RANSAC-1k [12]	68.02	3.17	9.28	70.97	0.05
RANSAC-10k	86.08	2.72	8.33	84.17	0.34
RANSAC-100k	92.05	2.50	7.71	87.62	2.59
PointDSC [4]	91.68	1.88	6.12	88.55	**0.02**
***TR-DE* (ours)**	**92.98**	**1.85**	**6.08**	**90.24**	0.16

Table 2. Success rate on the 3DLoMatch dataset [17] with different number of correspondences. "↑" represents the average increase with respect to RANSAC.

	5000	2500	1000	500	250	↑
FCGF [11]						
RANSAC [12]	35.7	34.9	33.4	31.3	24.4	-
PointDSC [4]	52.0	51.0	45.2	37.7	27.5	+10.74
***TR-DE* (ours)**	49.5	50.4	48.4	43.4	34.3	**+13.26**
Predator [17]						
RANSAC [12]	54.2	55.8	56.7	56.1	50.7	-
PointDSC [4]	61.5	60.20	58.5	55.4	50.4	+2.50
***TR-DE* (ours)**	**64.0**	**64.8**	**61.7**	**58.8**	**56.5**	**+6.46**

local descriptors. The putative correspondences are generated by the nearest-neighbour search in the feature space. As shown in Tab. 1, our method is compared with the state-of-the-art methods including *FGR*, *TEASER*, *RANSAC* and *PointDSC*. While *FGR* has the fastest computing speed, its success rate is low. The performance of *RANSAC*-based methods can be improved by increasing iteration times, but the computation time is also increasing sharply. Compare to these methods, our method attains the best results in terms of *SR*, *RE*, *TE* and *F1*-score with a low time consuming.

3DLoMatch Dataset [17]. We further report registration results on the more challenging 3DLoMatch dataset (overlap ratio $< 30\%$). Following [4, 17], we use FCGF [11] and Predator [17] to generate putative correspondences. We compare the success rates of registration by *RANSAC*, *PointDSC* and our *TR-DE*. As shown in Tab. 2, regardless of the registration methods, Predator has a better performance in the low-overlap cases than FCGF. No matter utilizing FCGF or Predator as descriptors, the proposed *TR-DE* improves the *SR* with a large margin, which is $3-4\%$ higher than the improvement brought by *PointDSC*.

Outdoor KITTI Dataset [13]. Using the same data preparation strategy in [4, 17, 19], we compare these methods on the outdoor LIDAR scans that are at least 10 m away from each other. The inlier threshold is set to 60 cm. The thresholds for successful registration are 5° for *RE* and 60 cm for *TE*. We report the registration results on the dataset with FCGF descriptors (see more results in the supplementary material). As shown in Table 3, almost all methods have satisfactory results. The reason is that the outlier ratio of the input correspondences is low (about 58.7% on average). Nonetheless, our *TR-DE* outperforms other methods in terms of *SR*, *TE* and $F1$-score. In addition, our method is more efficient than *TEASER*.

Table 3. Registration results on the outdoor KITTI [13] dataset using FCGF descriptors.

	SR (%)	*RE* (°)	*TE* (cm)	*F1* (%)	Time (s)
FGR [38]	96.22	0.37	22.03	-	2.61
TEASER [35]	95.5	**0.33**	22.38	85.77	31.46
RANSAC-1k [12]	97.12	0.48	23.37	84.26	**0.22**
RANSAC-10k	98.02	0.41	22.94	85.29	1.43
PointDSC [4]	97.84	**0.33**	20.99	85.29	0.31
***TR-DE* (ours)**	**98.2**	0.38	**18**	**85.99**	3.01

Table 4. Multi-way registration comparison in terms of absolute trajectory error (cm) on the Augmented ICL-NUIM [9] dataset.

	Living1	Living2	Office1	Ofiice2	AVG
FGR [38]	20.43	23.27	13.14	15.49	18.09
RANSAC [12]	19.46	20.21	17.07	13.7	17.61
PointDSC [4]	19.82	23.56	17.91	13.53	18.71
***TR-DE* (ours)**	**19.33**	**14.44**	**11.4**	**11.63**	**14.2**

5.3. Application in Multi-way Registration

The multi-way registration [38] is a task to obtain a model of scene or object by registering multiple fragments. It usually contains two steps: pairwise registration between fragments and global registration by pose graph optimization [18]. We show the application of our method in this task on the Augmented ICL-NUIM dataset [9]. Following [4, 10], we replace the pairwise registration steps in the multi-way pipeline implemented in Open3D [39] with our and some baseline methods. We first compute FCGF [11] descriptors from the down-sampled point cloud fragments. Then, we perform pairwise registration to compute aligned fragment poses and optimize the poses with pose graph optimization. We use the absolute trajectory error (ATE) as the evaluation metric. As shown in Tab. 4, our method achieves lower ATE than other methods in all scenes, which implies our better accuracy in practical application.

6. Conclusion

In this paper, we proposed a deterministic and efficient point cloud registration method based on a novel transformation decomposition. We decompose 6 DOF of transformation into (2+1) DOF and (1+2) DOF. This decomposition reduces the non-linearity of 3-DOF rotation, and thus improves the algorithm efficiency. Then based on branch and bound, we sequentially search for (2+1) DOF and (1+2) DOF. We extensively compare our method with state-of-the-art approaches. our method is more accurate and robust than the approaches that provide similar efficiency to ours. Our method is more efficient than the approaches whose accuracy and robustness are comparable to ours.

Acknowledgments. This work is supported in part by the Hong Kong Centre for Logistics Robotics, RGC via grant 14207119, the CUHK VC Discretionary Fund, and Shenzhen Municipal Government via the ShenzhenHK Collaboration Zone Project.

References

[1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, 2019.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *TPAMI*, 1987.

[4] 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 *CVPR*, 2021.

[5] Jean-Charles Bazin, Yongduek Seo, and Marc Pollefeys. Globally optimal consensus set maximization through rotation search. In *ACCV*, 2013.

[6] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *TPAMI*, 1992.

[7] Dylan Campbell and Lars Petersson. An adaptive data representation for robust point-set registration and merging. In *ICCV*, 2015.

[8] Dylan Campbell and Lars Petersson. Gogma: Globally-optimal gaussian mixture alignment. In *CVPR*, 2016.

[9] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *CVPR*, 2015.

[10] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020.

[11] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019.

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[14] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019.

[15] Richard I Hartley and Fredrik Kahl. Global optimization through rotation space search. *IJCV*, 2009.

[16] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 1987.

[17] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021.

[18] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *ICRA*, 2011.

[19] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. In *ICCV*, 2021.

[20] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, 2009.

[21] Haoang Li, Kai Chen, Pyojin Kim, Kuk-Jin Yoon, Zhe Liu, Kyungdon Joo, and Yun-Hui Liu. Learning icosahedral spherical probability map based on bingham mixture model for vanishing point estimation. In *ICCV*, 2021.

[22] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Pointnetlk revisited. In *CVPR*, 2021.

[23] Yinlong Liu, Guang Chen, and Alois Knoll. Globally optimal vertical direction estimation in atlanta world. *TPAMI*, 2020.

[24] Yinlong Liu, Chen Wang, Zhijian Song, and Manning Wang. Efficient global point cloud registration by matching rotation invariant features through translation search. In *ECCV*, 2018.

[25] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *CVPR*, 2020.

[26] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for point cloud registration with correspondences. *TPAMI*, 2018.

[27] Álvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, Hongdong Li, and David Suter. Fast rotation search with stereographic projections for 3d registration. *TPAMI*, 2016.

[28] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *ECCV*, 2008.

[29] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[30] John P. Snyder. Map projections - a working manual, 1987.

[31] Julian Straub, Trevor Campbell, Jonathan P How, and John W Fisher. Efficient global point cloud alignment using bayesian nonparametric mixtures. In *CVPR*, 2017.

[32] Chen Wang, Yinlong Liu, Yiru Wang, Xuechen Li, and Manning Wang. Efficient and outlier-robust simultaneous pose and correspondence determination by branch-and-bound and transformation decomposition. *TPAMI*, 2021.

[33] Yue Wang and Justin M. Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *NIPS*, 2019.

[34] Heng Yang, Jingnan Shi, and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. In *RSS*, 2019.

[35] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *T-RO*, 2021.

[36] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *TPAMI*, 2016.

[37] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[38] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, 2016.

[39] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

[40] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.