2021 6th IEEE International Conference on
Advanced Robotics and Mechatronics
July 3-5, 2021, Chongqing, China

2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM) | 978-1-6654-3909-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICARM52023.2021.9536185

Efficient and Robust Line-based Registration Algorithm for Robot Perception Under Large-scale Structural Scenes

Guang Chen[1], Yinlong Liu[2], Jinhu Dong[1], Lijun Zhang[1], Haotian Liu[1], Bo Zhang[3] and Alois Knoll[2]

***Abstract*— Point cloud registration is a classical problem in advanced robot perception. Despite having been widely studied, the registration of large-scale point clouds still remains challenging in terms of both efficiency and accuracy. In this paper, aiming at the registration in large-scale structural scenes that contains numerous line-features, we propose a line-based efficient and robust registration algorithm for robot perception. Concretely, we first extract lines from point clouds and use the line-features to perform the registration, which decreases the scale of algorithm's input and decouples the rotation and the translation sub-problems. Consequently, it reduces the complexity of registration problem. We then solve the rotation and translation sub-problems using the branch-and-bound algorithm, which ensures the accuracy and robustness of registration. In translation sub-problem, we propose two strategies to adapt to the registration problem in different scenes, the one is universal algorithm, the other is decoupled algorithm. Extensive experiments are performed on both synthetic and real-world data to demonstrate the advantages of our method.**

I. Introduction

Point cloud registration has many applications in robot and computer vision. Specifically, in robot navigation and localization, given two LiDAR points clouds, it can be applied to obtain the robot's odometry information [1]. Besides, the real-time loop closure in SLAM system can be achieved through matching the current scan points with the global map or sub-map [2]. In addition, in computer vision application, point cloud registration plays an important role in object recognition [3] and 3D reconstruction [4].

In the past few years, point cloud registration has been widely studied, and many algorithms have been proposed. Existing methods can be categorized into two main groups: non-feature-based algorithms and feature-based algorithms.

Non-feature-based algorithms, from the perspective of optimization, can be divided into two categories: local and global methods. Among them, Iterative Closest Point (ICP) algorithm [5] is one of the best-known classical local registration algorithms. Given an initial transformation, ICP algorithm alternates between finding the correspondence under the current transformation and calculating the best transformation according to the current correspondence until the algorithm converges to the optimal solution. On the basis of ICP, several improved algorithms have been proposed for the following years, such as LM-ICP [6] which widened the basin of convergence and trimmed ICP [7] based on the Least Trimmed Squares approach. Besides ICP and its improved algorithms, Normal Distribution Transform (NDT) [8] [9] and Gaussian Mixture Models (GMMs) [10] [11] [12] are also effective local registration algorithms. These local algorithms are widely used in solving the point cloud registration problem, due to their low time complexity and excellent performance. However, the accuracy of these approaches cannot be guaranteed, especially when considerable outliers exist in the input data, which is inevitable in many real applications. This is because that the point cloud registration problem is generally a non-convexity problem [13], and these algorithms all perform local optimization to get the transformation between two points sets. Therefore, the final solution is probably one of the numerous and dangerous local optimum, and the performance of the algorithms heavily rely on the proper initialization.

To avoid being stuck in local optimum, a series of global registration algorithms have been proposed. Most of these global methods are based on the branch-and-bound algorithm. Breuel et al. firstly apply the branch-and-bound algorithm for point cloud registration and use a match-list-based branch-and-bound algorithm to handle the geometric matching [14]. In addition, some researches based on Hartley and Kahl's theory [15] calculate the geometric bound for the branch-and-bound algorithm to achieve robust point cloud registration. Yang et al. in [13] propose a nested branch-and-bound algorithm, which is the first global 3D registration algorithm in $SE(3)$ space. Due to adopting global search, it is more robust than the local algorithms. However, in this algorithm, the dimensionality of searching space is six, while the time complexity of the branch-and-bound grows exponentially with the number of dimensions, which significantly deteriorates the efficiency. Recently, several works focus on speeding up the globally optimal registration method. Based on the nested algorithms, Chin et al. [16] introduce stereographic projection to increase the algorithm's efficiency. In [17], authors significantly reduce the fraction of outliers to make further optimization perform more quickly. But both of these algorithms do not decrease the dimensionality of the searching space, so they are still time-consuming. To decrease the dimensionality, Liu et al. use the rotation invariant feature to decouple the rotation and translation, which reduces the dimensionality of 3D points registration

This work was supported by the Key Technology Development and Application of Piloted Autonomous Driving Trucks Project, and by Shanghai Rising Star Program (No. 21QC1400900).

[1]G. Chen, J. Dong, L. Zhang, and H. Liu are with the School of Automotive Studies, Tongji University, Shanghai, 200092 China.

[2]Y. Liu and A. Knoll are with the Department of Informatics, Technical University of Munich, Munich 80333, Germany.

[3]B. Zhang is with Autonomous Driving Group, Shanghai Westwell Information and Technology Co. Ltd., 200050, China.

[†]Corresponding author is G. Chen. Email: guangchen@tongji.edu.cn.

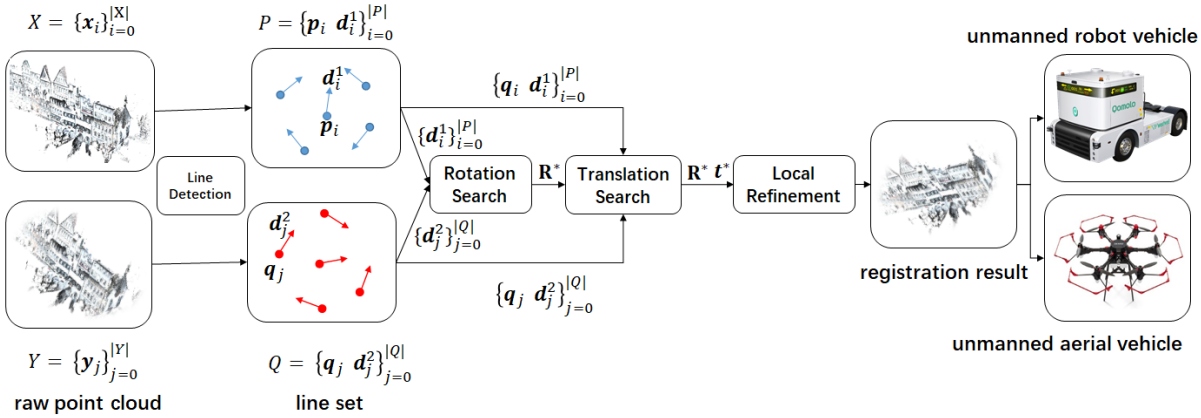


Fig. 1. The flow diagram of our proposed registration algorithm.

[18]. However, it increases the scale of input data and it takes some time to calculate the invariant rotation feature. Therefore, it cannot maintain the efficiency when solving the large scale points registration problem. In conclusion, all the aforementioned algorithms can not realize both efficient and robust high- scale points registration.

Existing feature-based algorithms mainly focus on non-structural features [19] [20] [21]. These descriptors can provide sparse correspondences which usually contains mismatches. After that, many robust correspondence-based registration techniques, for example, RANSAC (RANdom SAmple Consensus [22]), can be applied to remove outliers and estimate the transformation. The most commonly used descriptors can be divided into two classes: spatial distribution histogram-based descriptors and geometric attributes histogram-based descriptors [23], which are both non-structural features. These non-structural feature-based algorithms do not rely on the prior knowledge of points set, thus can be generally applied in many points registration problems. However, there do exist lots of structured scenes (e.g., Manhattan world [24] and Atlanta world [25]) that contain many special structural features in real applications.

In this paper, inspired by researches [26] [27], we propose an efficient and robust registration algorithm on the basis of branch-and-bound. In our algorithm, we choose lines instead of points as registration objects, aiming to study large-scale structured scene containing rich line features, which is commonly seen in urban environment. Firstly, we use the line detection algorithm [28] to exact the lines from the points sets, and then utilize these lines to implement registration.

Thanks to the line-features, we can decrease the input scale of algorithms, since the number of the line-features is significantly less than the number of points. More importantly, registering two point sets using line-feature can decouple estimating the rotation and estimating the translation, which significantly reduces the dimensionality of 3D points registration. Thus, the line-feature based registration algorithm is more efficient than non-feature based registration methods which exhaustively search $SE(3)$ space. Besides, compared with general feature-based registration methods, we respectively utilize the branch-and-bound optimizing the optimal rotation and translation, which ensures the algorithm's accuracy and increase the robustness to outliers.

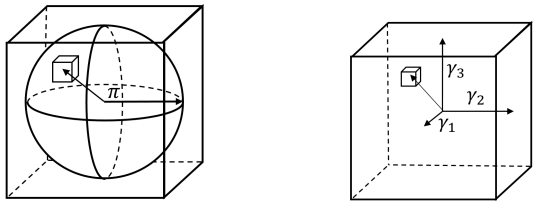
It is worth noting that we propose two strategies for solving the translation sub-problem, one is the universal algorithm, and the other is the decoupled algorithm. The universal algorithm performs translation search in three-dimension space, and the decoupled algorithm performs three translation searches in three one-dimension-spaces by coordinate system transformation. The decoupled algorithm is quite faster than the universal algorithm since its time complexity is $O(n)$, where n is the scale of the algorithm's input. Although, the application of the decoupled algorithm is limited. If we want to use the decoupled algorithm to do the translation search, there must be a certain number of parallel lines in the lines set obtaining from the points set and these parallel lines at least along with two directions. But the universal algorithm does not have this limitation, and it can be applied in every scene containing several lines. Therefore, we can choose different algorithms for translation searching according to the characteristic of the scenes.

The main contributions of our proposed method are as follows:

- We propose a new registration algorithm based on the line-based features, which can efficiently solve the registration problem of large scale structured scene for robust robot perception.
- We formulate a robust objective function for rotation and translation sub-problems, and use branch-and-bound to obtain the globally optimal solutions. Thus, our methods are more robust than general-feature based registration methods.
- We propose two strategies for solving the translator problem to adapt to the registration problem in different structured scenes.

II. Method

In this section, we introduce our proposed method (see flow diagram Fig. 1) in detail.



(a) (b)

Fig. 2. Rotation search space and translation search space. (a) is the rotation space $SO(3)$ which is parametrised by angle-axis vector in a solid ball whose radius is π. (b) is the translation search space which is parametrised by 3-vector in a cuboid whose half-widths are $[\lambda_1, \lambda_2, \lambda_3]$

We exact lines using the algorithm in paper [28] from the template points set $X = \{\boldsymbol{x}_i\}_{i=0}^{|X|}$ and target points set $Y = \{\boldsymbol{y}_j\}_{j=0}^{|Y|}$, where $|X|$ and $|Y|$ respectively represent the number of points in point set X and Y. We use a point and a unit direction vector to represent a line. Let $P = \{\boldsymbol{p}_i, \boldsymbol{d}_i^1\}_{i=0}^{|P|}$ and $Q = \{\boldsymbol{q}_j, \boldsymbol{d}_j^2\}_{j=0}^{|Q|}$ be the lines set detecting from the points set X and Y, where $\boldsymbol{d}_i^1$ and $\boldsymbol{d}_j^2$ represent the i-th and j-th unit direction vector and $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$ represent the i-th and j-th point.

A. Efficient Rotation Search

To obtain the optimal rotation, we define the maximum of inliers as the objective function as Eq. (1)

$$\begin{aligned} E_r(\mathbf{R}) &= \sum_{i=0}^{|P|} \max_{j\in[0,|Q|]} \lfloor \|\mathbf{R}\cdot\boldsymbol{d}_i^1 - \boldsymbol{d}_j^2\| \le \varepsilon_r \rfloor \\ \mathbf{R}^* &= \arg\max_{\mathbf{R}\in SO(3)} E_r(\mathbf{R}) \end{aligned} \tag{1}$$

where $\lfloor\cdot\rfloor$ is a binary function, if its inner condition is true, it will return 1 and 0 otherwise, $\|\cdot\|$ represents the Euclidean distance, ε_r is the inlier threshold, $\mathbf{R}^*$ is the optimal rotation. Our objective function Eq. (1) is robust to outliers because the points out of a distance threshold will not be counted as a inlier [16].

We use the branch-and-bound algorithm to solve the above problem obtaining the globally optimal rotation. The rotation search space of the branch-and-bound algorithm is parameterized with angle-axis vector $\boldsymbol{r}$, where $\|\boldsymbol{r}\|$ is the rotation angle and $\boldsymbol{r}/\|\boldsymbol{r}\|$ is the rotation axis. There is a exponential map between rotation matrix $\mathbf{R}$ and rotation vector $\boldsymbol{r}$.

$$\mathbf{R} = \exp(\boldsymbol{r}^\times) \tag{2}$$

where $\boldsymbol{r}^\times$ is the skew-symmetric matrix induced by $\boldsymbol{r}$. Thus, we can use a solid ball shown in Fig.2 (a) whose radius is π to represent all the 3D rotations. The sub-space of each branch is a cube in it. We refer to the method in [16] to calculate the bound of the objective function (1) and use the stereographic projection to speed up the rotation search.

The upper bound is

$$\bar{E}(\mathbb{B}) = \sum_{i=0}^{|P|} \max_{j\in[0,|Q|]} \lfloor S_{\alpha_{\mathbb{B}}}(\mathbf{R_c}\boldsymbol{d}_i^1) \cap l_{\varepsilon_r}(\boldsymbol{d}_j^2) \neq \emptyset \rfloor \tag{3}$$

Where $\mathbb{B}$ is the current branch, $\mathbb{R}_c$ is the rotation parameterized by the center of $\mathbb{B}$, $S_{\alpha_{\mathbb{B}}}(\mathbf{R}\cdot\boldsymbol{d}_i^1) = \{\boldsymbol{d} \mid \|\boldsymbol{d}\| = \|\mathbf{R}_c\cdot\boldsymbol{d}_i^1\|, \angle(\boldsymbol{d}, \mathbf{R}_c\cdot\boldsymbol{d}_i^1) = \alpha_{\mathbb{B}}\}$, $\alpha_{\mathbb{B}}$ is the half diagonal length of the cube $\mathbb{B}$ and $l_{\varepsilon_r}(\boldsymbol{d}_j^2) = \{\boldsymbol{d} \mid \|\boldsymbol{d} - \boldsymbol{d}_j^2\| \le \varepsilon_r\}$. The lower bound is

$$\underline{E}_r(\mathbb{B}) = \sum_{i=0}^{|P|} \max_{j\in[0,|Q|]} \lfloor \|\mathbf{R_c}\cdot\boldsymbol{d}_i^1 - \boldsymbol{d}_j^2\| \le \varepsilon_r \rfloor \tag{4}$$

For the rigorous mathematical proofs about the upper bound and lower bound, please refer to [16].

B. Universal Translation Search

We introduce our universal translation search method in this sub-section. After the rotation search process, we can obtain the optimal rotation. Simultaneously, we can also obtain line-to-line correspondence, which may include mismatches since only rotation is considered. We then introduce that given the optimal rotation, how to obtain the optimal translation.

First, we consider a simple case where there are no parallel lines in the scene, which means each line has its own unique direction. In this case, we can directly obtain the lines correspondence only by rotation, since there are no two lines that have the same direction. Let $P_1 = \{\boldsymbol{p}_i, \boldsymbol{d}_i\}_{i=0}^{K}$ be the lines set transformed from P under the optimal rotation. Let $Q_1 = \{\boldsymbol{q}_i, \boldsymbol{d}_i\}_{i=0}^{K}$ be the correspondence lines set of P_1. The element $\{\boldsymbol{p}_i, \boldsymbol{d}_i\}$ in P_1 and the element $\{\boldsymbol{q}_i, \boldsymbol{d}_i\}$ in Q_1 are correspondence lines. Therefore, the equation $\angle(\boldsymbol{p}_i + \boldsymbol{t}^* - \boldsymbol{q}_i, \boldsymbol{d}_i) \le \varepsilon_t$ holds as shown in Fig.3, where $\boldsymbol{t}^*$ is the optimal translation and ε_t is the t-inlier threshold. We can get the optimal translation by solving the following problem:

$$\begin{aligned} E_t(\boldsymbol{t}) &= \sum_{i=0}^{K} \lfloor \angle(\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_i, \boldsymbol{d}_i) \le \varepsilon_t \rfloor \\ \boldsymbol{t}^* &= \arg\max_{\mathbf{t}\in\mathbb{R}^3} E_t(\boldsymbol{t}) \end{aligned} \tag{5}$$

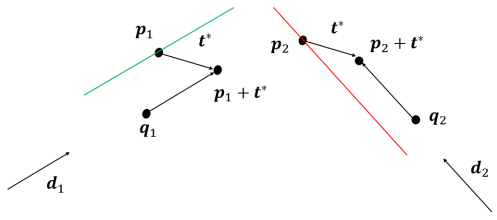


Fig. 3. $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ are the points falling on the same line whose direction vector is $\boldsymbol{d}_1$, and $\boldsymbol{p}_2$ and $\boldsymbol{q}_2$ are the points falling on the same line whose direction vector is $\boldsymbol{d}_2$. Let $\boldsymbol{t}^*$ be the optimal translation, the vector from point $\boldsymbol{q}_1$ to point $\boldsymbol{p}_1 + \boldsymbol{t}^*$ is parallel to the direction $\boldsymbol{d}_1$. And $\boldsymbol{p}_2$, $\boldsymbol{q}_2$, $\boldsymbol{t}^*$, $\boldsymbol{d}_2$ are also have this relationship

However, in the real applications, the simple scenes are usually uncommon. In contrast, more complex scenes where lines are parallel are more common. In these cases, there may be mismatches caused by parallel line-features.

Let $P_2 = \{U_i\}_{i=0}^{M}$ be a line set containing the lines transformed from P under the the motion of optimal rotation R^*, where the i-th element U_i of P_2 is a sub line-set

whose elements have the same direction (parallel lines): $U_i \triangleq \{\boldsymbol{p}_k^i,\ \boldsymbol{d}_i\}_{k=1}^{|U_i|}$. Let $Q_2 = \{V_i\}_{i=0}^{M}$ be the correspondence lines set of P_2, and similarly, $V_i \triangleq \{\boldsymbol{q}_l^i,\ \boldsymbol{d}_i\}_{l=0}^{|V_i|}$. Obviously, in theses scenes, the parallel lines lead to the ambiguity that we cannot obtain the correct correspondences only by the direction of lines. In other words, if two lines in P_2 and Q_2 have the same direction, we can not determine if they are the correspondence lines without translation. In these cases, even given the optimal rotation, we still need to solve the following chicken-and-egg problem to obtain the optimal translation and exact correspondence simultaneously:

$$\begin{aligned} E_t(\boldsymbol{t}) &= \sum_{i=0}^{M}\sum_{k=0}^{|U_i|}\max_{l\in[0,|V_i|]} \lfloor \angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \le \varepsilon_t \rfloor \\ \boldsymbol{t}^* &= \arg\max_{\mathbf{t}\in\mathbb{R}^3} E_t(\boldsymbol{t}) \end{aligned} \tag{6}$$

Similarly, we use the branch-and-bound algorithm solving Eq. (6). The translation space is parameterized with 3-dim vector $\boldsymbol{t}$. One of the differences between rotation search space and translation search space is that translation search space is not closed, so we should determine the maximum translations along three dimensions according to the real applications, which is practical. Given the translation domain, we can represent all translations with a cuboid as shown in Fig.2 (b) and the sub-space of each branch is a cuboid in it.

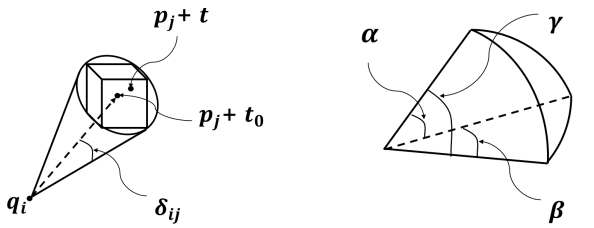


Fig. 4. (a) Given a sub-space cuboid, the translation of $\boldsymbol{p}_j$ may be anywhere within the cuboidal region. Let $\boldsymbol{t}$ be the random translation in the cuboid and $\boldsymbol{t}_0$ is the center of cuboid. Given two vectors, one is the vector from $\boldsymbol{q}_i$ to $\boldsymbol{p}_j+\boldsymbol{t}$ and the other is the vector from $\boldsymbol{q}_i$ to $\boldsymbol{p}_j+\boldsymbol{t}_0$. The maximum angle between the two vector is δ_{ij}. (b) It demonstrates the triangle angle relationship: $\gamma \le \alpha+\beta$.

Given a current branch $\mathbb{C}$ that is a cuboid space, $\boldsymbol{t}_0$ is the center of $\mathbb{C}$ and $\mathbb{S}$ is the surface of $\mathbb{C}$. the upper bound of objective function (6) can be:

$$\bar{E}_t(\mathbb{C}) = \sum_{i=0}^{M}\sum_{k=0}^{|U_i|}\max_{l\in[0,|V_i|]} \lfloor \angle(\boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \le \varepsilon_t+\delta_{lk} \rfloor \tag{7}$$

$$\delta_{lk} = \max_{\boldsymbol{t}\in\mathbb{S}} \angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i) \tag{8}$$

The lower bound of objective function Eq. (6) can be:

$$\underline{E_t}(\mathbb{C}) = \sum_{i=0}^{M}\sum_{k=0}^{|U_i|}\max_{l\in[0,|V_i|]} \lfloor \angle(\boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \le \varepsilon_t \rfloor \tag{9}$$

We then give the solid mathematical proof to confirm Eq.(7) is the upper bound.

$$\begin{aligned} &\angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i) \\ \le &\max_{\boldsymbol{t}\in\mathbb{C}} \angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i) \\ = &\max_{\boldsymbol{t}\in\mathbb{S}} \angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i) \end{aligned} \tag{10}$$

According to the triangle angle relationship, we have

$$\begin{aligned} &\angle(\boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \\ \le &\angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i)+\angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i) \\ = &\angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i)+\delta_{lk} \end{aligned} \tag{11}$$

From (11), we know that, given a branch sub-space $\mathbb{C}$, if $\boldsymbol{p}_k^i,\ \boldsymbol{q}_l^i,\ \boldsymbol{t} \in \mathbb{C}$ satisfy $\angle(\boldsymbol{p}_k^i+\boldsymbol{t}-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \le \varepsilon_t$, $\angle(\boldsymbol{p}_k^i+\boldsymbol{t}_0-\boldsymbol{q}_l^i,\ \boldsymbol{d}_i) \le \varepsilon_t+\delta_{lk}$ must hold. Therefore, it can be deduced $E_t(\boldsymbol{t}) \le \bar{E}_t(\mathbb{C})$, in other words, $\bar{E}_t(\mathbb{C})$ is the upper bound of the objective.

C. Efficiently Decoupled Translation Search

In this sub-section, aiming at the scenes that contain several parallel lines and these parallel lines are along at least two directions, we propose a fast decoupled translation search algorithm based on branch-and-bound. In contrast to universal translation search, the search space of this algorithm is three one-dimension spaces.

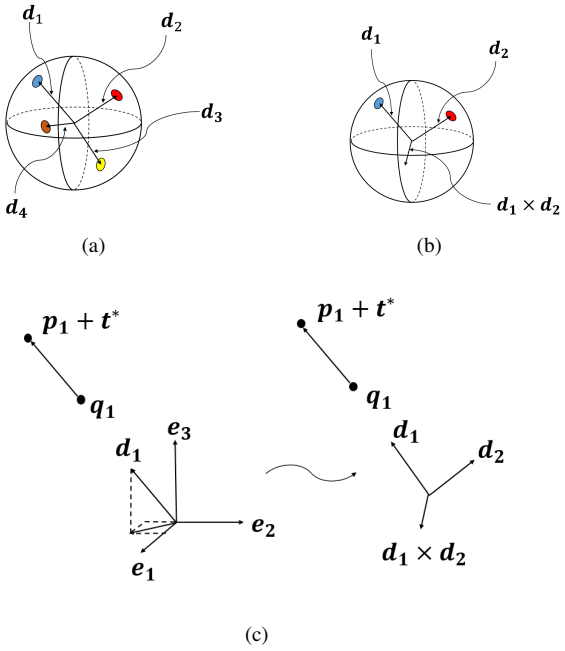


Fig. 5. (a) Illustrate the result of dbscan cluster. There are four cluster centers, $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$. (b) We select two directions $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ containing the most lines from them and calculate the cross product of two vectors. (c) Illustrate the basis transformation.

First, we use the dbscan cluster algorithm [29] to cluster the direction vectors of the lines, as shown in Fig.5 (a). The lines, along with the same direction, will be clustered in the same class. We choose two directions which contain the most lines from them, let $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ be the direction vectors. In this case, $\boldsymbol{d}_1$, $\boldsymbol{d}_2$ and $\boldsymbol{d}_3 = \boldsymbol{d}_1 \times \boldsymbol{d}_2$ form a group of basis in the vector space. We perform the basis transformation for

the lines, as shown in Fig.5 (c) and make the lines along with $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ as the registration objects.

Let $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ be the unit orthogonal basis of initial coordinate system. Let $M_1 = \{\boldsymbol{p}_i^1, \boldsymbol{d}_1\}_{i=0}^{|M_1|}$, $M_2 = \{\boldsymbol{p}_j^2, \boldsymbol{d}_2\}_{j=0}^{|M_2|}$ be the two template line sets and $N_1 = \{\boldsymbol{q}_k^1, \boldsymbol{d}_1\}_{k=0}^{|N_1|}$, $N_2 = \{\boldsymbol{q}_l^2, \boldsymbol{d}_2\}_{l=0}^{|N_2|}$ be the two target line sets.

$$\begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} = \begin{bmatrix} d_1^x & d_1^y & d_1^z \\ d_2^x & d_2^y & d_2^z \\ d_3^x & d_3^y & d_3^z \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \end{bmatrix} \tag{12}$$

Let $[x_i^1, y_i^1, z_i^1]$ be the coordinate of $\boldsymbol{p}_i^1$ in the coordinate system $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $[u_i^1, v_i^1, w_i^1]$ be the coordinate of $\boldsymbol{p}_i^1$ in the coordinate system $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, $[x_j^2, y_j^2, z_j^2]$ be the coordinate of $\boldsymbol{p}_j^2$ in the coordinate system $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $[u_j^2, v_j^2, w_j^2]$ be the coordinate of $\boldsymbol{p}_j^2$ in the coordinate system $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, $[a_k^1, b_k^1, c_k^1]$ be the coordinate of $\boldsymbol{q}_k^1$ in the coordinate system $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $[f_k^1, g_k^1, h_k^1]$ be the coordinate of $\boldsymbol{q}_k^1$ in the coordinate system $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, $[a_l^2, b_l^2, c_l^2]$ be the coordinate of $\boldsymbol{q}_l^2$ in the coordinate system $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $[f_l^2, g_l^2, h_l^2]$ be the coordinate of $\boldsymbol{p}_l^2$ in the coordinate system $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, $[t_1, t_2, t_3]$ be the coordinate of $\boldsymbol{t}^*$ in the coordinate system $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $[t_1', t_2', t_3']$ be the coordinate of $\boldsymbol{t}^*$ in the coordinate system $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$.

There is the following relationship between the coordinates. We use the coordinates of $\boldsymbol{p}_i^1$ as the example.

$$\begin{bmatrix} x_i^1 \\ y_i^1 \\ z_i^1 \end{bmatrix} = \begin{bmatrix} d_1^x & d_1^y & d_1^z \\ d_2^x & d_2^y & d_2^z \\ d_3^x & d_3^y & d_3^z \end{bmatrix} \cdot \begin{bmatrix} u_i^1 \\ v_i^1 \\ w_i^1 \end{bmatrix} \tag{13}$$

According to the above sub-section, we known that if $\boldsymbol{p}_i^1$ and $\boldsymbol{q}_k^1$ fall on the same line, $\boldsymbol{p}_i^1$, $\boldsymbol{q}_k^1$, $\boldsymbol{d}_1$ and $\boldsymbol{t}^*$ must satisfy the following relationship.

$$\boldsymbol{p}_i^1 + \boldsymbol{t}^* - \boldsymbol{q}_k^1 = \lambda \boldsymbol{d}_1 \tag{14}$$

where λ is a constant. We can also write the above equation as the following.

$$\begin{aligned} &\begin{bmatrix} u_i^1 & v_i^1 & w_i^1 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} + \begin{bmatrix} t_1' & t_2' & t_3' \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} \\ &- \begin{bmatrix} f_k^1 & g_k^1 & h_k^1 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \boldsymbol{d}_3 \end{bmatrix} \end{aligned} \tag{15}$$

Then, we can formulate the following equation. $u_i^1 + t_1' - f_k^1 = \lambda$, $v_i^1 + t_2' - g_k^1 = 0$, $w_i^1 + t_3' - h_k^1 = 0$. Similarly, for the lines along with $\boldsymbol{d}_2$, we can determine similar equations. $u_i^2 + t_1' - f_k^2 = 0$, $v_i^2 + t_2' - g_k^2 = \gamma$, $w_i^2 + t_3' - h_k^2 = 0$. In this case, we achieve decoupling the translation along the three direction $\boldsymbol{d}_1$, $\boldsymbol{d}_2$ and $\boldsymbol{d}_3$ and we can respectively get the optimal translations along three directions by solving the following problems.

$$\begin{aligned} E_t^1(t_1) &= \sum_{j=0}^{|M_2|} \max_{l \in [0,|N_2|]} \lfloor |u_j^2 + t_1 - f_l^2| \le \varepsilon_{t1} \rfloor \\ t_1' &= \arg\max_{t_1 \in \mathbb{R}} E_t^1(t_1) \end{aligned} \tag{16}$$

$$\begin{aligned} E_t^2(t_2) &= \sum_{i=0}^{|M_1|} \max_{k \in [0,|N_1|]} \lfloor |v_i^1 + t_2 - g_k^1| \le \varepsilon_{t2} \rfloor \\ t_2' &= \arg\max_{t_2 \in \mathbb{R}} E_t^2(t_2) \end{aligned} \tag{17}$$

$$\begin{aligned} E_t^3(t_3) &= \sum_{i=0}^{|M_1|} \max_{k \in [0,|N_1|]} \lfloor |w_i^1 + t_3 - h_k^1| \le \varepsilon_{t3} \rfloor \\ &+ \sum_{j=0}^{|M_2|} \max_{l \in [0,|N_2|]} \lfloor |w_j^2 + t_3 - h_l^2| \le \varepsilon_{t3} \rfloor \\ t_3' &= \arg\max_{t_3 \in \mathbb{R}} E_t^3(t_3) \end{aligned} \tag{18}$$

We can use the branch-and-bound algorithm to obtain the optimal translations along three dimensions,respectively. The search space of each sub-problem is a line segment.

The upper bound of (16) is

$$\bar{E_t^1}(\mathbb{L}) = \sum_{j=0}^{|M_2|} \max_{l \in [0,|N_2|]} \lfloor |u_j^2 + \bar{t}_1 - f_l^2| \le \varepsilon_{t1} + \delta_t \rfloor \tag{19}$$

where δ_t is the half length of $\mathbb{L}$ and $\bar{t}_1$ is the center of $\mathbb{L}$. The lower bound of (16) is

$$\underline{E_t^1}(\mathbb{L}) = \sum_{j=0}^{|M_2|} \max_{l \in [0,|N_2|]} \lfloor |u_j^2 + \bar{t}_1 - f_l^2| \le \varepsilon_{t1} \rfloor \tag{20}$$

The upper bound and lower bound of Eq. (17) and Eq. (18) are calculated in the same way as Eq. (19) and Eq. (20).

III. Experiments

To validate the performance of the proposed algorithm, we conduct experiments using both synthetic data and real data. And in real data experiments, we compare our algorithm with the state-of-the-art approaches, ICP, NDT, FPFH [30], GOICP [13] and GORE [17]. Since the synthetic data are random lines and the registration object of these state-of-art approaches is point, we only verify the performance of our algorithm in the synthetic experiments and don't compare with these approaches. Our algorithm is implemented in C++, ICP, NDT and FPFH use the functions in PCL and GOICP, GORE use the code released by the authors. All the experiment are performed on the notebook with intel i5 2.8 GHz CPU and 4G RAM.

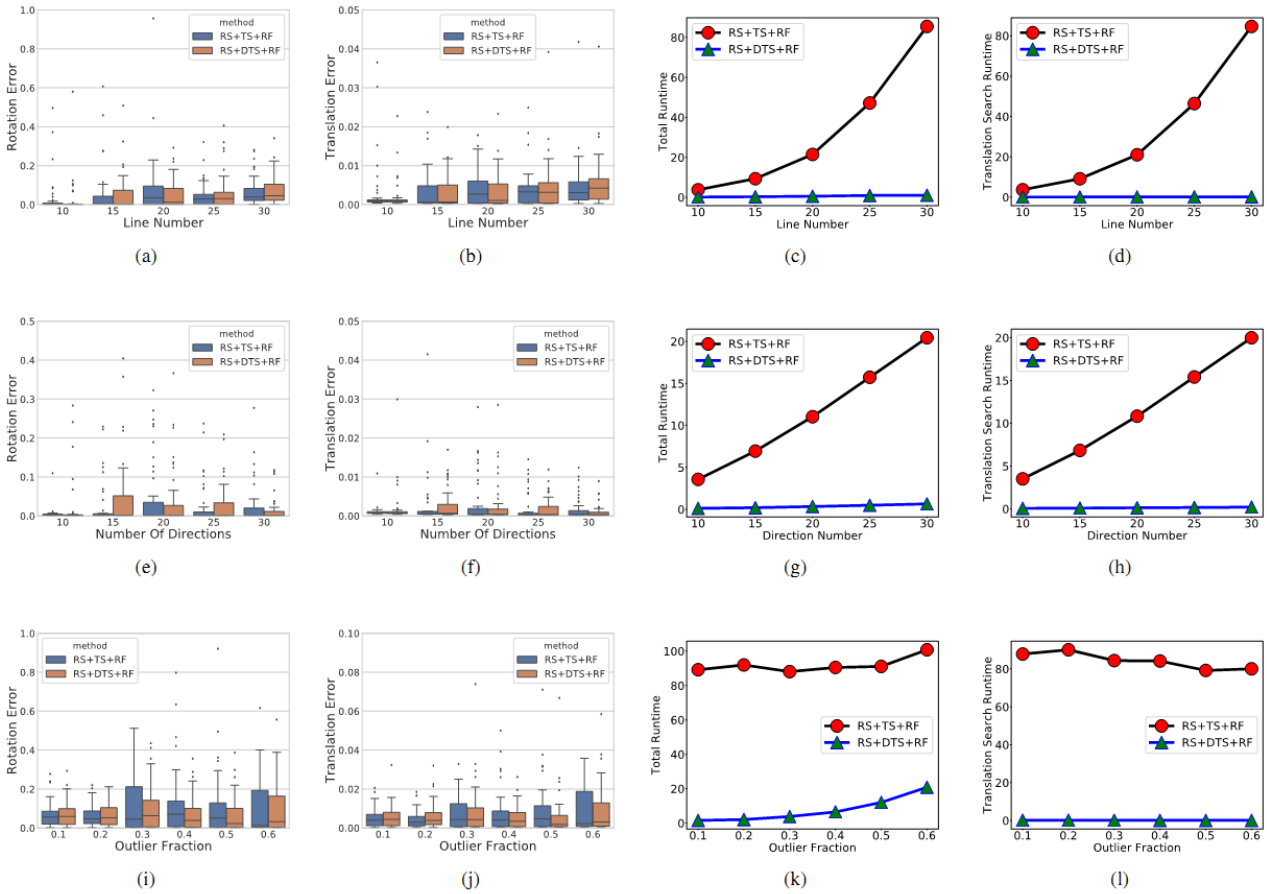


Fig. 6. RS+TS+RF represent the algorithm utilizing rotation search, universal translation search and local refinement. RS+DTS+RF represent the algorithm using rotation search, decoupled translation search and local refinement. From the first row to the third row, they are respective the number of lines experiments, the number of directions experiments and outlier fraction experiments. The box charts illustrate rotation error and translation error. The line charts illustrate the mediums of total runtime and translation search runtime. The unit of rotation error is degree, the unit of translation error is meter and the unit of runtime is second.

A. Synthetic Data

In this sub-section, we use synthetic data to verify the performance of our method. We mainly concern three experiment terms, the accuracy and efficiency concerning the number of directions, the accuracy and efficiency concerning the number of lines, the accuracy and efficiency concerning the outlier fraction. In number of directions experiment, we create five directions groups, the number of directions in these group are 10, 15, 20, 25, 30. And in each direction, there are ten different lines. For every number of direction, we perform fifty Monte Carlo experiments, where the transformation in each experiment are randomly generated from the space $SE(3) = SO(3) \times [-5, 5]^3$, directions and lines are randomly created. In number of experiments, given the number of directions is ten, we successively create five lines groups, in each lines group, we create the same number of lines for each direction. The number of lines successively are 10, 15, 20, 25, 30. For each number of lines, we also do fifty Monte Carlo experiments. In outlier fraction experiments, given ten directions and thirty lines in each direction, the outlier fractions are 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. The outliers randomly distribute in the lines set. We perform fifty Monte Carlo experiments for each fraction. Besides, we add noises with $\sigma = 0.02$ for all the direction vectors.

The result is shown in Fig.6. It is worth noting that all the synthetic data satisfy the condition using the decoupled translation search. In the view of accuracy, both methods proposed in this paper, RS+TS+RF(using rotation search, universal translation search and local refinement) and RS+DTS+RF(using rotation search, decoupled translation search), have excellent accuracy and robustness to outliers. In the view of efficiency, decoupled translation search method is faster than universal translation search method. The time complexity of universal translation search method is $O(kn^3)$, where k is the number of directions and n is the number of lines. The time complexity of the decoupled translation search method is $O(kn)$. The trend of the line charts illustrating the translation search runtime Fig. 6 (d) (h) verify this.

B. Real Data

In this sub-section, we compare our method with the state-of-art approaches using real data. We select 5 scenes

TABLE I

RE, TE, RT REPRESENT ROTATION ERROR, TRANSLATION ERROR, AND THE RUNTIME OF REGISTRATION.

Scene	Metric	ICP	NDT	FPFH	Our
Marketsquarefeldkirch4	RE (°)	20.22	41.13	0.57	**0.29**
	TE (m)	5.07	4.82	0.31	**0.06**
	RT (s)	26.15	136.91	63.42	**3.05**
Marketsquarefeldkirch7	RE (°)	17.16	43.58	0.48	**0.31**
	TE (m)	4.10	4.67	0.32	**0.13**
	RT (s)	17.04	116.99	61.21	**1.36**
Stgallencathedral1	RE (°)	11.81	40.69	0.58	**0.31**
	TE (m)	3.48	4.65	0.54	**0.19**
	RT (s)	25.46	41.57	58.93	**2.05**
Stgallencathedral3	RE (°)	17.37	40.24	0.71	**0.35**
	TE (m)	3.51	4.19	0.49	**0.18**
	RT (s)	18.37	36.12	59.69	**1.78**
Stgallencathedral6	RE (°)	11.71	40.45	0.57	**0.28**
	TE (m)	5.47	4.41	0.45	**0.13**
	RT (s)	13.11	39.81	50.13	**1.86**

from semantic 3D dataset [31]. We downsample the initial points cloud, and make the number of points in each scene be one million. For each scene, we perform fifty Mento Carlo experiments and the transformation are generated from the space $SE(3) = SO(3) \times [-5,5]^3$. From the synthetic experiments, we know the accuracy and robustness of decoupled translation search and universal translation search method are similar, and these scenes satisfy the application conditions of the decoupled translation search method. In these experiments, we utilize the decoupled translation search method, and we stipulate that if an algorithm can't obtain the solution within thirty minutes, we will terminate the algorithm earlier. We select the best parameters for the start-of-art approaches considering both accuracy and efficiency.

The results are shown in Table I and the data processing method is that we remove the five best results and the five worst results then calculate the mean of the following results. Both global algorithms, GOICP and GORE, didn't get the solution within thirty minutes. They are unpractical in solving the large scale points registration problem. Thus we don't write the result of both algorithms. From the tables, we can know that our method performs better than other algorithms in all scenes. . The errors of ICP and NDT are so large that it is considered that they are failed. FPFH achieved similar results to those of our method, but our method is faster and more accurate than it. To further compare the performance of our algorithm and FPFH, we choose 8 scenes from bremen city dataset [32]. The registration results are shown in Table II, table III and Fig.7

IV. CONCLUSION

In this paper, we introduce a line-based efficient and robust algorithm for the registration of large-scale point clouds. This method uses the lines detected from the point clouds as registration objects, which decouples the rotation and translation sub-problems. We sequentially use branch-and-bound algorithm to solve both sub-problems. In the translation sub-problem, we propose two strategies. The first one is a universal method without any limitations, and the second one is a fast decoupled method that can only be applied in the scenes containing a certain number of parallel lines, extend along with at least two directions. The proposed algorithm outperforms state-of-the-art local and global methods, while finds the robust pose solutions more reliably.

TABLE II

RE, TE, RTE, RT REPRESENT ROTATION ERROR, TRANSLATION ERROR, RELATIVE TRANSLATION ERROR AND THE RUNTIME OF REGISTRATION.

Algorithm	Metric	Scene1	Scene2	Scene3	Scene4
EPFH	RE (°)	2.75	16.94	2.01	14.33
	TE (m)	2.04	1.72	1.54	1.05
	RTE	0.11	0.09	0.05	0.04
	RT (s)	80.11	79.83	149.23	158.02
Our	RE (°)	0.32	1.19	1.18	1.61
	TE (m)	0.43	0.07	0.95	0.35
	RTE	0.03	0.004	0.03	0.01
	RT (s)	14.77	11.18	35.91	44.54

TABLE III

RE, TE, RTE, RT REPRESENT ROTATION ERROR, TRANSLATION ERROR, RELATIVE TRANSLATION ERROR AND THE RUNTIME OF REGISTRATION.

Algorithm	Metric	Scene5	Scene6	Scene7	Scene8
FPFH	RE (°)	42.97	5.66	6.19	13.76
	TE (m)	3.11	2.16	4.43	9.87
	RTE	0.19	0.11	0.19	0.24
	RT (s)	106.03	75.27	71.34	79.19
Our	RE (°)	0.43	0.57	2.64	0.92
	TE (m)	0.05	0.31	1.26	0.54
	RTE	0.003	0.01	0.05	.001
	RT (s)	100.28	11.84	44.54	70.78

REFERENCES

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems Conference*, July 2014.

[2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1271–1278.

[3] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, May 1999.

[4] A. S. Mian, M. Bennamoun, and R. A. Owens, "From unordered range images to 3d models: a fully automatic multiview correspondence algorithm," in *Proceedings Theory and Practice of Computer Graphics, 2004.*, June 2004, pp. 162–166.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[6] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image & Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2001.

[7] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *International Conference on Pattern Recognition*, 2002.

[8] M. Magnusson, A. J. Lilienthal, and T. Duckett, *Scan registration for autonomous mining vehicles using 3D-NDT*, 2007.

[9] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3d registration reliability and speed - a comparison of icp and ndt," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3907–3912.

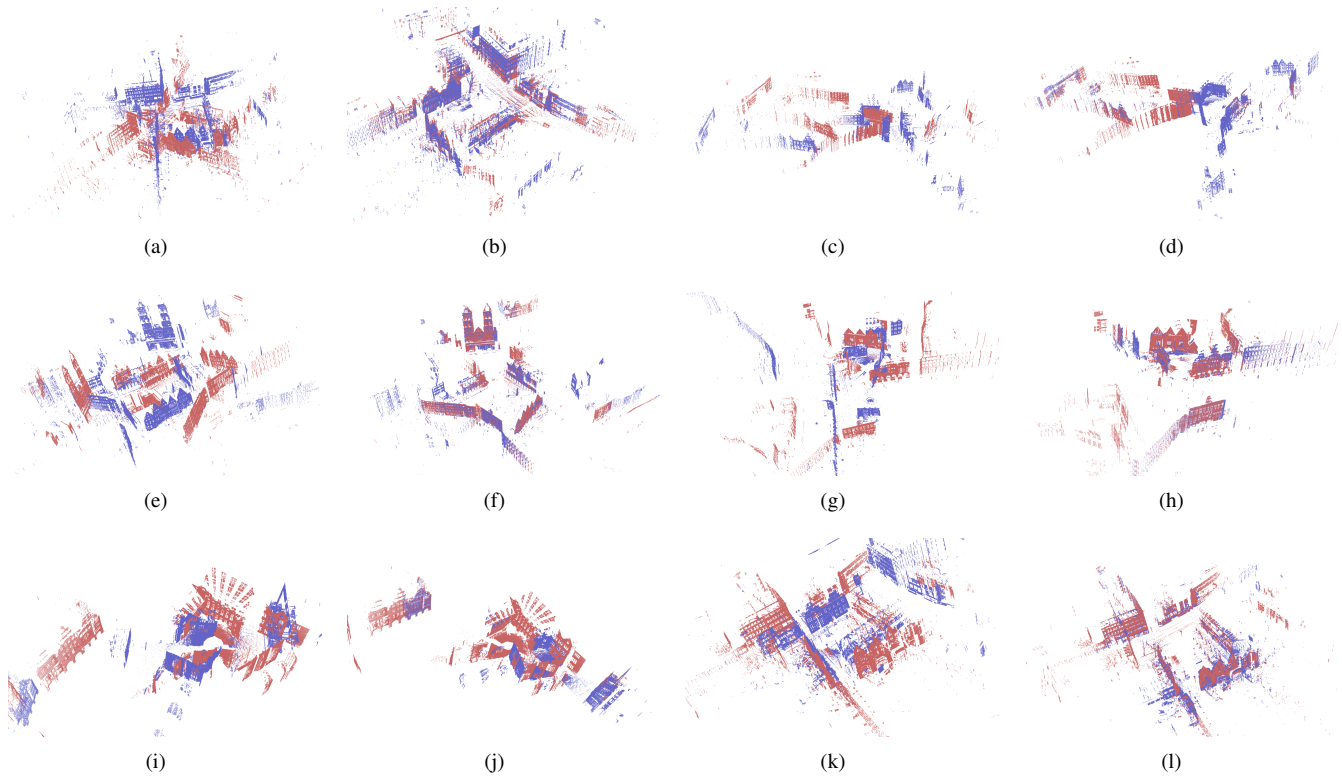


Fig. 7. Schematic diagrams of the registration results. (a) (c) (e) (g) (i) (k) are the point clouds before registration. And (b) (d) (f) (h) (j) (l) are the point clouds after registration.

[10] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of gaussians," *Proceedings. IEEE International Conference on Computer Vision*, vol. 2, p. 1246—1251, October 2005. [Online]. Available: http://europepmc.org/articles/PMC2630186

[11] ——, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[12] M. Andriy and S. Xubo, "Point set registration: coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[13] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1457–1464.

[14] T. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Computer Vision and Image Understanding*, vol. 90, no. 3, pp. 258–294, 2003.

[15] R. I. Hartley and F. Kahl, "Global optimization through rotation space search," *International Journal of Computer Vision*, vol. 82, no. 1, pp. 64–79, 2009.

[16] A. P. Bustos, T.-J. Chin, A. Eriksson, H. Li, and D. Suter, "Fast rotation search with stereographic projections for 3d registration," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 2227–2240, 2016.

[17] A. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2868–2882, 2018.

[18] Y. Liu, C. Wang, Z. Song, and M. Wang, "Efficient global point cloud registration by matching rotation invariant features through translation search," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 460–474.

[19] A. L. Kleppe, L. Tingelstad, and O. Egeland, "Coarse alignment for model fitting of point clouds using a curvature-based descriptor," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 811–824, 2019.

[20] SALTI, Samuele, TOMBARI, Federico, D. I. Stefano, and Luigi, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, no. 8, pp. 251–264, 2014.

[21] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3d data description," 01 2010.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Readings in Computer Vision*, pp. 726–740, 1987.

[23] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 66–89, 2016.

[24] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, "The manhattan frame model—manhattan world inference in the space of surface normals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 235–249, 2017.

[25] K. Joo, T.-H. Oh, I. S. Kweon, and J.-C. Bazin, "Globally optimal inlier set maximization for atlanta world understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[26] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "Plade: A plane-based descriptor for point cloud registration with small overlap," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 4, pp. 2530–2540, 2020.

[27] W. Forstner and K. Khoshelham, "Efficient and accurate registration of point clouds with plane to plane correspondences," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[28] X. Lu, Y. Liu, and K. Li, "Fast 3d line segment detection from unorganized point cloud," 2019.

[29] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, 1996, pp. 226–231.

[30] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms

(fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3212–3217.
[31] http://semantic3d.net/ .
[32] http://kos.informatik.uni-osnabrueck.de/3Dscans/ .