2020 IEEE International Symposium on
Safety, Security, and Rescue Robotics (SSRR)
November 4-6, Abu Dhabi, UAE

A Fast and Robust 2D LiDAR Alignment Method by Motion Decoupling

Jinhu Dong[1], Yinlong Liu[2], Lixuan Tang[1], Guang Chen[1*] and Alois Knoll[2]

Abstract— **Aligning the 2D LiDAR point clouds and estimating their relative poses are the fundamental components of the 2D robotic SLAM (Simultaneous Localization and Mapping) system. Although there are lots of works addressing this problem, existing methods often fail to get a balance between robustness and efficiency. Methods depending on local optimization run fast while sometimes get stuck in local optimum resulting in the wrong pose estimations, which weaken the robustness of the methods. Additionally, methods depending on global optimization always return globally optimal poses by searching the entire given domain for a long period of time, which sacrifice the efficiency of the methods. In this work, we propose a decoupled and globally optimal 2D LiDAR aligning method, which differs from existing methods by achieving the robustness and efficiency of the 2D LiDAR pose estimation simultaneously. Concretely, we use the invariant vectors features decoupling the motion of rotation and translation to reduce the dimensionality of the alignment problem. Consequently, the branch-and-bound algorithm runs in a low dimension to obtain a robust relative pose. Moreover, we propose a feature-selecting strategy to speed up our method. The proposed method is verified in various synthetic and real-world data and shows great performance.**

I. INTRODUCTION

2D points cloud alignment technique is a critical component for Simultaneous Localization and Mapping. Given two sets of points, the process of alignment is to find the best transformation to align the two points sets. The relative pose from aligning point clouds can be used for LIDAR odometry and robots localizing itself on the map [1].

In the past few years, many algorithms have been proposed for 2D LIDAR point clouds alignment, and some of them obtain great success in application. For example, [2] proposed the Iterative Closest Point(ICP) algorithm, which obtains the transformation by alternating between building closest-point correspondences and get the transformation with the current correspondences, until convergence. In addtion, Bliber et al. proposed the Normal Distributions Transform(NDT) algorithm, which maximizes the point set's probability density function with the mixture distribution to estimate the transformation [3]. These algorithms have been widely used in several real-world scenes [4] because of good performance and simple concepts. However the accuracy of these approaches cannot be guaranteed. Especially, there are many outliers in the input data, which is a common case in real application. The reason is that the point set alignment problem is generally a non-convexity problem [5], and these algorithms adopt the local optimization methods to obtain the transformation. Therefore, there is full of the dangerous possibility of local minima, and their performance heavily rely on the proper initialization.

To address this problem, global optimization, which mostly are based on branch-and-bound algorithm to search the entire given domain, is applied in point set alignment. [6] reviews and summarizes the methods for globally optimal point set alignment, especially 2D LIDAR alignment. However, when the initial domain cannot properly be estimated, the global searching strategy is by no mean efficient. Thus there is a demand for methods which is not only efficient but also robust.

In this paper, we propose a fast and robust algorithm for the registration of 2D LIDAR point clouds based on the branch-and-bound algorithm. The main contributions of our proposed method are as follows. (1) Our method is more efficient than other global methods that are based on branch-and-bound. In this paper, we have two strategies to get high efficiency. Firstly, we decouple the motion into rotation and translation, and solve the subproblems sequentially, which reduce the dimensionality of the original alignment problem. Secondly, we design an invariant-features selecting strategy, which significantly reduce the number of invariant-features. It is worth noting that the idea of motion decoupling appeared in related work [7] [8]. However, we propose a original invariant-features selecting strategy for 2D LIDAR alinment in this paper, which can significantly reduce the computational burden. (2) Our method is robust to outlier. It is well known that outlier is the most common enemy of 2D LIDAR alignment in real application, because outliers are always distributed irregularly and lead to the non-convexity of the alignment problem. To obtain a robust result, we formulate a robust objective function and apply the branch-and-bound algorithm to get the global optimum.

II. Related Works

The point set alignment problem has been studied for a long time. Recently, with the success of deep learning in other applications [9], the learning methods are being applied in the point set alignment problem. Li et al. proposed a end-to-end deep learning method in [10] for loop closure detection. In [11], authors defined a point-set kernel as a set of learnable 3D points for point cloud registration. Besides, there are also researches on learning descriptors [12] [13]. However, in this section, we focus on the most relevant non-learning algorithms, which, from the perspective of optimization, can be divided into two categories: local and global methods.

*Guang Chen is the corresponding author.

[1]Tongji University, Shanghai, China; [2]Technical University of Munich, Munich, Germany.

A. Local Methods

The Iterative Close Point (ICP) [14] is one of the most popular point set matching algorithms. It obtains the solution by alternately finding the closest correspondences and solving the optimal translation with the current correspondences. Due to its conceptual simplicity and excellent performance, it has been used in many tasks. But it can only obtain a local optimum and have risks of failure. Over the years, its several variants, such as, LM-ICP [15] which widened the basin of convergence, trimmed ICP [16] based on the Least Trimmed Squares approach and sparse ICP [17], have been proposed to improve the ICP algorithm. However, they are all have possibility to be stuck in local minima. Besides the ICP and its variants, the NDT algorithm is also widely used in the points set matching, which was proposed by Biber et al. in [3]. In addition, NDT has been extended to 3D point clouds [18] [19]. In [19], authors have evaluated the performance of ICP and NDT for 3D mapping, in their experiments, NDT depends less on the proper initialization than ICP. Other popular local methods were based on GMMs (Gaussian Mixture Models) [20] [21]. These approaches use a mixture of Gaussians to represent each point set and transform the point set matching problem into aligning probability distribution.

B. Global Methods

The optimal global methods can guarantee optimality without initialization. Most of the deterministic Methods are based on branch-and-bound(BnB) algorithm. [22] was one of the earliest work applying the branch-and-bound algorithm for point sets alignment, which used the matchlist-based branch-and-bound algorithm for geometric matching. Subsequently, several types of research based on Hartley and Kahl's theory [23] which use geometric bound were proposed. Li et al. formulated an objective function based on Lipschitz optimization theory and used the branch-and-bound algorithm to obtain the optimal solution [24]. However, this method assumes that there is no outlier in the point sets, and the translation has been determined in advance. It is almost unrealistic in the real-world scenes. Yang et al. in [5] proposed a nested branch-and-bound algorithm, which is the first global 3D points alignment algorithm in $SE(3)$. Based on the nested method, [25] used the stereographic projection to increase the algorithm's efficiency. These nested methods need search the optimal rotation and translation in the entire given domain, therefore, the dimension of searching space is six. Unfortunately, the complexity of BnB algorithm is exponential in the dimension, thus these nested methods are relatively time-consuming. To speed up the global searching process, [7] proposed a decoupling method to separate the motion and reduce the dimensionality of 3D point set alignment.

III. Method

A. Problem Formulation

There are two 2D LIDAR point sets, $X = \{\boldsymbol{x}_i\}_{i=1}^{M}$ and $Y = \left\{\boldsymbol{y}_j\right\}_{j=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^2$ and $\boldsymbol{y}_j \in \mathbb{R}^2$, which are related by a 2D transformation. M may be not equal to N because of the existence of outliers. The objective of aligning two point sets is to find a optimal transformation T^*

, which is composed of a rotation $\mathbf{R}^*$ and a translation $\boldsymbol{t}^*$. E is defined as the energy function evaluating the alignment between two point sets. In this paper, we define the maximum of inliers as the objective function as Eq.(1).

$$E(T) = \sum_{i=1}^{M} \max_{j\in[1,N]} \lfloor \|\mathbf{R}\cdot\boldsymbol{x}_i + \boldsymbol{t} - \boldsymbol{y}_j\| \leq \varepsilon \rfloor \tag{1}$$

where $\lfloor . \rfloor$ is a binary function, if its inner condition is true, it will return 1 and 0 otherwise, $\|.\|$ represents the Euclidean distance, ε is the inlier threshold. If the distance between $\mathbf{R}\cdot\boldsymbol{x}_k + \boldsymbol{t}$ and $\boldsymbol{y}_k$ is less than ε, we will consider $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ is an inlier pair under the current transformation. Our objective function Eq.(1) is robust to outliers because the points out of a distance threshold will not be counted.

B. Decoupling Rotation and Translation Search

As defined in the above section, the transformation is three- DoF(Degree of Freedom). Unfortunately, to obtain the global optimum, the branch-and-bound algorithm need considerable time to search all the 3D domain, which is exponential in the dimension. In this paper, instead of optimizing over the 3D transformation domain directly, we construct the feature vectors which are invariant to translation and has the same relative rotation as the original point sets. Concretely, we first align these invariant vectors to obtain the optimal rotation between two invariant vector sets, then recover the optimal translation by aligning the original point sets under the optimal rotation. In this case, our algorithm decomposes the transformation optimization problem into two sub-problems through decoupling the rotation and translation searching, which decreases the algorithm's dimensionality and thus improve the algorithm's efficiency. We then describe the decoupling method in detail.

Given two points $\boldsymbol{x}_{i1}$, $\boldsymbol{x}_{i2}$ in X, their corresponding points $\boldsymbol{y}_{j1}$, $\boldsymbol{y}_{j2}$ in Y and optimal rotation $\mathbf{R}^*$ and translation $\boldsymbol{t}^*$ from X to Y. We have

$$\boldsymbol{y}_{j1} = \mathbf{R}^*\boldsymbol{x}_{i1} + \boldsymbol{t}^* \tag{2}$$

$$\boldsymbol{y}_{j2} = \mathbf{R}^*\boldsymbol{x}_{i2} + \boldsymbol{t}^* \tag{3}$$

Using equation (3) minus the equation (2), we have

$$\boldsymbol{y}_{j1} - \boldsymbol{y}_{j2} = \mathbf{R}^*\cdot(\boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}) \tag{4}$$

where $\boldsymbol{y}_{j1} - \boldsymbol{y}_{j2}$ is the vector from $\boldsymbol{y}_{j1}$ to $\boldsymbol{y}_{j2}$ and $\boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}$ is the vector from $\boldsymbol{x}_{i1}$ to $\boldsymbol{x}_{i2}$.

From the Eq.(4) we can see that both sides vectors are invariant to the translation and the vector $\boldsymbol{y}_{j1} - \boldsymbol{y}_{j2}$ can be aligned with the vector $\boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}$ by the optimal rotation $\mathbf{R}^*$. We call these vectors as the translation invariant vectors

(TIVs). Using the characteristics of the TIVs, we can obtain the optimal rotation through aligning the TIVs. Given $P = \{\boldsymbol{p}_i\}_{i=1}^{W}$ are the TIVs' set constructed from original point set X, $Q = \left\{\boldsymbol{q}_j\right\}_{j=1}^{Z}$ are the TIVs' set constructed from original point set Y. Optimizing rotation can be described as the following problem

$$\begin{gathered}\mathbf{R}^* = \arg\max E(\mathbf{R}) \\ E(\mathbf{R}) = \sum_{i=1}^{W} \max_{j\in[1,Z]} \lfloor \|\mathbf{R}\boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r \rfloor\end{gathered} \tag{5}$$

Obviously, there is only rotation $\mathbf{R}$ to be solved in Eq.(5). It is also worth noting that solving the problem (5), we can not only recover the optimal rotation but also determine the TIVs' correspondence relationship which can be used to find the corresponding point pairs from the original point sets. For example, let $\left\{\boldsymbol{p}_i, \boldsymbol{q}_j\right\}$ be a corresponding TIVs pair, where $\boldsymbol{p}_i = \boldsymbol{x}_{i1} - \boldsymbol{x}_{i2}$ and $\boldsymbol{q}_i = \boldsymbol{y}_{j1} - \boldsymbol{y}_{j2}$. According to equation, it is easy to determine that both $\left\{\boldsymbol{x}_{i1}, \boldsymbol{y}_{j1}\right\}$ and $\left\{\boldsymbol{x}_{i2}, \boldsymbol{y}_{j2}\right\}$ are corresponding point pairs. These corresponding point pairs could provide a important prior condition for the translation searching.

After obtaining the optimal rotation $\mathbf{R}^*$, give $C = \left\{\left\{\boldsymbol{x}_k, \boldsymbol{y}_k\right\}\right\}_{k=1}^{K}$, where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ is a pair of corresponding points, then we have

$$\boldsymbol{t}^* = \boldsymbol{y}_k - \mathbf{R}^* \cdot \boldsymbol{x}_k \tag{6}$$

We denote $\boldsymbol{x}'_k = \mathbf{R}^*\boldsymbol{x}_k$ and $\boldsymbol{x}'_k = [x'_{k1}, x'_{k2}]^{\mathrm{T}}$, $\boldsymbol{y}_k = [y_{k1}, y_{k2}]^{\mathrm{T}}$, $\boldsymbol{t}^* = [t_1^*, t_2^*]^{\mathrm{T}}$ for element representation. Therefore,

$$\begin{gathered} t_1^* = y_{k1} - x'_{k1} \\ t_2^* = y_{k2} - x'_{k2} \end{gathered} \tag{7}$$

From the equations (7), we can see that the translation has two independent variables. Thus we can respectively search the variables, which decomposes the translation searching problem into two one-dimensional sub-problems. It can further decrease the complexity of the algorithm. We formulate one-dimensional translation searching problem as followings

$$\begin{gathered} t_n^* = \arg\max_{t_n\in\mathbb{R}} E(t_n),\ n \in \{1,2\} \\ E(t_n) = \sum_{k=1}^{K} \lfloor \|\ x_{kn} + t_n - y_{kn}\| \leq \varepsilon_t \rfloor \end{gathered} \tag{8}$$

C. Selecting Invariant-features

Using the above decoupling strategy, we decompose the 3-DoF transformation recovering problem into three 1-DoF sub-problems. It guarantees the algorithm's efficiency. Unfortunately, this strategy decreases the algorithm's complexity while increasing the scale of point set. If the original points' number is n, the number of TIVs constructing from the original points will grow to $n(n-1)/2$. Downsampling the TIVs can decrease the algorithm's input scale, improving its efficiency. However, downsampling contributes to losing geometric information . Thus, formulating a selecting-strategy which can get a balance between the downsampling and information loss is the key to guarantee the algorithm's accuracy and efficiency simultaneously. In this paper, we propose a robust feature-selecting strategy to downsample the TIVs without affecting the algorithm's accuracy. Our feature-selecting strategy adopts an equal interval sampling approach which is based on the property that the paired TIVs have equal norms. Concretely, let $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ be a pair of consensus vectors. According to eq.(4), we have

$$\boldsymbol{q}_k = \mathbf{R}^* \cdot \boldsymbol{p}_k \tag{9}$$

Calculating the norm of the equation left and right sides, we can get the conclusion the paired TIVs have equal norm.

$$\|\boldsymbol{q}_k\| = \|\mathbf{R}^* \cdot \boldsymbol{p}_k\| = \|\boldsymbol{p}_k\| \tag{10}$$

Combining this property with equal interval sampling method, we can decompose the subset selecting from TIVs' into several parts. In each part, the norm difference among TIVs would not be larger than twice the sampling threshold. Through the equal interval downsampling, the overall geometric of the TIVs' distribution hardly change, thus under a constant sampling scale, it can loss less geometric information. Besides, the selected TIVs' coarse correspondence can be determined according to the norm relationship. Here, we describe our feature-selecting strategy in detail.

Given the norm selecting sequence $S = \{s_i\}_{i=1}^{J}$ and the selecting threshold ε_s. Let V_{max} , V_{min} be the max and min norm of the selected TIVs. We have

$$\begin{gathered} s_1 = V_{min}\ , s_J = V_{max} \\ s_k - s_{k-1} = (s_J - s_1)/J\ , k \in [2, J] \end{gathered} \tag{11}$$

Let $P' = \{P'_i\}_{i=1}^{J}$ be the subset of P and $Q' = \{Q'_i\}_{i=1}^{J}$ be the subset of Q, We call P'_k and Q'_k as the sampling sub-sets.

After constructing the TIVs, we choose the sampling scale and the extreme norms to formulate the selecting sequence. For each selecting length s_k, we put all TIVs whose norm is in range of $[s_k - \varepsilon_s, s_k + \varepsilon_s]$ into the sampling sub-set P'_k or Q'_k. Due to the equal norm relationship and $\varepsilon_s \ll s_k - s_{k-1}$, the TIVs in P'_l only can be aligned with the TIVs in Q'_l. That is to say, the selected subsets have been pre-aligned and coarse correspondence can be determined. Thus using the pre-aligned subsets to recover optimal rotation reduces the problem's difficulty. It is useful for improving the algorithm's speed and accuracy.

D. Efficient Rotation Search Based on Branch-and-bound Algorithm

Since we utilize the subsets choosed according to the feature-selecting strategy rather than the whole TIVs sets to optimize the rotation, the objective function (5) can be changed as follows

$$E(R) = \sum_{k=1}^{J} \sum_{\boldsymbol{p}_i\in P'_k} \max_{\boldsymbol{q}_j\in Q'_k} \lfloor \|\mathbf{R} \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r \rfloor \tag{12}$$

Optimizing rotation is a one-dimension searching problem, so we use the angle space of a circle to represent the rotation domain, and the range is $[-\pi, \pi]$. When using the BnB algorithm to search optimal rotation, the algorithm explores iteratively the branches of a tree and each branch represents a sub-space. In the searching process, the algorithm choose a branch whose priority is the highest and check the upper bound of the selected branch against the best solution before creating candidate solutions with it. The branch will be cut, if there is no better solution it. We summarize the rotation searching based on BnB in **Algorithm 1**.

```
Algorithm 1: BnB rotation search to maximise (12)
Input: the subset of TIVs $P'$, $Q'$ and rotation
       threshold $\varepsilon_r$
Output: the optimal rotation R$^*$ with quality $E^*$
Initialize priority queue $q$, $E^* \leftarrow 0$, R$^* \leftarrow I$, $\mathbb{L} \leftarrow$ line
  segment of length $2\pi$;
while $q$ is not empty do
    Obtain highest priority line segment $\mathbb{L}$ from $q$;
    if $\bar{E} = E^*$ then
        Terminate;
    $R_c \leftarrow$ the center of $\mathbb{L}$;
    if $E(R_c) > E^*$ then
        $E^* \leftarrow E(R_c)$;
        $R^* \leftarrow R_c$;
    Divide the $\mathbb{L}$ into two sub-lines $\{\mathbb{L}_i\}_{i=1}^{2}$;
    for each $\mathbb{L}_i$ do
        if $\bar{E}(\mathbb{L}_i) \geq E^*$ then
            Add the $\mathbb{L}_i$ with its priority $\bar{E}(\mathbb{L}_i)$ into $q$;
        else
            Cut the $\mathbb{L}_i$ ;
return $R^*, E^*$
```

In the **Algorithm 1**, $\bar{E}_r(\mathbb{L})$ is the upper bound of the current branch, which satisfies:

$$\bar{E}(\mathbb{L}) \geq \max_{\mathbf{R}\in\mathbb{L}} E(\mathbf{R}) \tag{13}$$

where $\mathbb{L}$ is a branch of rotation domain. In this paper, we calculate the bound as followings.

$$\bar{E}(\mathbb{L}) = \sum_{k=1}^{J} \sum_{\boldsymbol{p}_i \in P'_k} \max_{\boldsymbol{q}_j \in Q'_k} \lfloor \|\mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r + \delta_r \rfloor \tag{14}$$

where $\delta_r = 2\|\boldsymbol{p_i}\| \sin(L_r/4)$; L_r is the length of $\mathbb{L}$ and ε_r is the threshold for rotation optimizing; $\mathbf{R}_c$ is the rotation correspondence to the center of $\mathbb{L}$.

We then give the solid mathematical proof to confirm Eq.(14) is the upper bound. Given $\|\mathbf{R} \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r$, we have:

$$\|\mathbf{R} \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| = \|\mathbf{R} \cdot \boldsymbol{p}_i - \mathbf{R}_c \cdot \boldsymbol{p}_i + \mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r \tag{15}$$

where $\angle(\mathbf{R}, \mathbf{R}_c)$ is the angle between the rotation $\mathbf{R}$ and $\mathbf{R}_c$.

$$\begin{aligned} &\|\mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| - \|(\mathbf{R} - \mathbf{R}_c) \cdot \boldsymbol{p}_i\| \leq \\ &\|\mathbf{R} \cdot \boldsymbol{p}_i - \mathbf{R}_c \cdot \boldsymbol{p}_i + \mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \end{aligned} \tag{16}$$

According to (15) and (16), we have

$$\|\mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| - \|(\mathbf{R} - \mathbf{R}_c) \cdot \boldsymbol{p}_i\| \leq \varepsilon_r \tag{17}$$

$$\|(\mathbf{R} - \mathbf{R}_c) \cdot \boldsymbol{p}_i\| = 2\|\boldsymbol{p}_i\| \sin\angle(\mathbf{R}, \mathbf{R}_c)/2 \tag{18}$$

Because $\angle(\mathbf{R}, \mathbf{R}_c) \leq L_r$

$$\|(\mathbf{R} - \mathbf{R}_c) \cdot \boldsymbol{p}_i\| \leq 2\|\boldsymbol{p}_i\| \sin L_r/4 \Longrightarrow \|(\mathbf{R} - \mathbf{R}_c) \cdot \boldsymbol{p}_i\| \leq \delta_r \tag{19}$$

Substituting (19) into (17):

$$\|\mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r + \delta_r \tag{20}$$

According to the proof, we can know if a pair of TIVs $\left\{\boldsymbol{p}_i, \boldsymbol{q}_j\right\}$ satisfies $\|\mathbf{R} \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r$, it must satisfy the $\|\mathbf{R}_c \cdot \boldsymbol{p}_i - \boldsymbol{q}_j\| \leq \varepsilon_r + \delta_r$. Thus $\bar{E}(\mathbb{L}) \geq \max_{\mathbf{R}\in\mathbb{L}} E(\mathbf{R})$.

E. Efficient Translation Search

Searching the translations of both coordinates are also the one-dimension searching problem. Thus, we use a line segment to represent the translation searching space and choose a fixed length for the line segment which is enough to search the optimal translation.

The objective function has been defined in (8). The upper bound of the objective function is

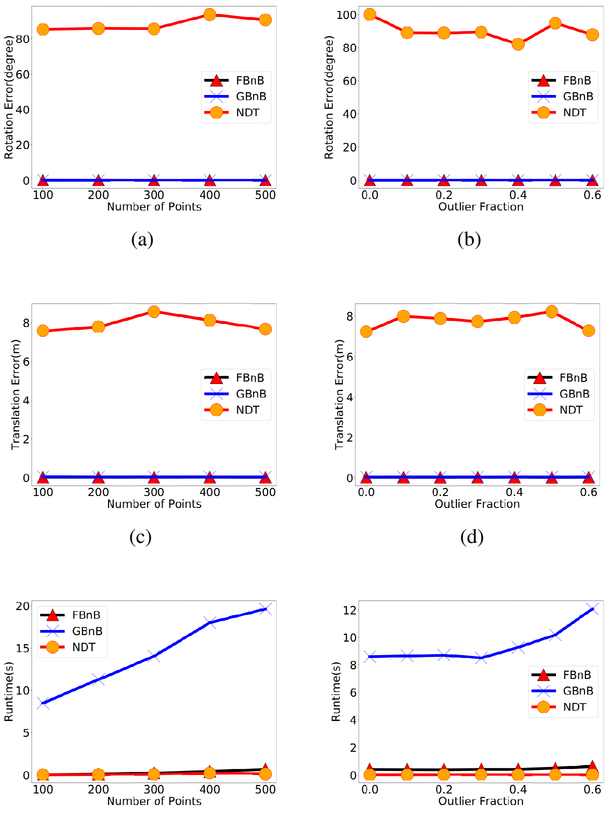
$$\bar{E}(\mathbb{T}) = \sum_{k=1}^{K} \lfloor \| \; x_{kn} + t_{nc} - y_{kn}\| \leq \varepsilon_t + \delta_t \rfloor \tag{21}$$

where $\mathbb{T}$ represent a branch of translation domain; ε_t is the threshold for translation searching, t_{nc} is the center of the $\mathbb{T}$, and δ_t is the half-length of the $\mathbb{T}$.

The essence of decoupling the translation search is projecting the points to both coordinates axis then aligning the projected points respectively. Although it can decrease the time complexity of the algorithm, it increases the density of points(the number of points in the unit area) and lose the points' geometric information. It increases the probability of false aligning and the error of the translations may be relatively large. Thus a local refinement is a demand for increasing accuracy.

IV. Experiments

To verify the effect of the proposed algorithm, denoted FBnB, we compare the performance of it with the state-of-the-art algorithms proposed in [3] [26] , denoted NDT and GBnB respectively, with both synthetic and real data. NDT is a local method and GBnB is a global method based on BnB. They have been integrated in the Matlab (function "matchScans" and "matchScansGrid"). We implemented the FBnB algorithm in python. All experiments were performed on a PC with the Intel Core i7 CPU.



(a) (b)

(c) (d)

(e) (f)

Fig. 1. Median rotation error, median translation error and median runtime with respect to outlier fraction and point number. (a) (c) (e): points number experiment. (b) (d) (f):outlier experiment

A. *Synthetic Data*

In this section, we verify the FBnB's effect using the synthetic data. To evaluate algorithm's accuracy and efficiency, we choose the following metrics.

The experiment about the synthetic data mainly includes two items. On the one hand, we verify FBnB algorithm's accuracy and efficiency through comparing against NDT and GBnB. On the other hand, we show FBnB algorithm's robustness to outliers using the synthetic dataset with a range of outlier fraction.

1) Accuracy and Efficiency Comparison with State-of-art Methods: In this experiment, the synthetic data are formulated based on a frame of point set in German Museum dataset. We choose a frame of point set from the German Museum dataset and downsample it according to different points number. We call it as the first point set. After that, we randomly select a rotation and translations in the range of $[-\pi,\pi]$ and $[-10,10]$ to transform the first point set obtaining a new point set which are called as the second point set.

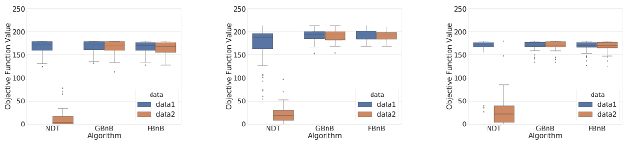
We respectively utilize NDT, GBnB and FBnB to align the point sets in the synthetic dataset. For FBnB, the inlier threshold of rotation search is 0.05 and both the threshold of translation search and local refinement are 0.1. Its searching space is the same as the GBnBs. For each experiment under different points number, 100 registrations are performed to ensure the experiment's reliability and the median rotation error, median translation error and median runtime are recorded. The result are plotted in Fig.1.

Both FBnB and GBnB algorithm have great accuracy in all points number experiment. However, the NDT's rotation and translation error are so large that we judge the NDT algorithm being invalid in this experiment. Fig.1e show FBnB has a similar efficiency to the NDT algorithm and FBnB' efficiency change less under different points number. The median runtime of GBnB are all more than 7s and it rises rapidly with the increase of points number. FBnB is quite faster than GBnB especially in the situation the points number is large.

2) Robustness to Outlier: The synthetic data in this experiment is formulated similar to the above way and the points number is set at 200. We create a random points set. These random points are in the region $[-10,10]\times[-10,10]$. Their number depends on the outlier fraction(outlier fraction is defined as $f_{outlier}=\frac{N_{oulier}}{N_{outlier}+N_{inlier}}$, where $N_{outlier}$ and N_{inlier} are the number of outliers and inliers). We use them randomly replacing the points in the second point set. Then we perform experiments with different outlier fractions and for each outlier fraction, 100 registrations are performed. The parameters of three algorithms are the same as the above experiment. We plot the result in Fig.1.

The NDT algorithm is also invalid in this experiment. For the accuracy under different outlier fraction, the performance of GBnB and FBnB are similar. And it is worth noting that FBnB has greater accuracy and stability than GBnB. For the efficiency under different outlier fraction, the median runtime of GBnB are all more than 8s and the FBnB's median runtime are all less than 1s. Besides, we can see the runtime of GBnB is exponential in the outlier fraction and FBnB's runtime doesn't have this rising trend. Thus regardless of accuracy or efficiency, FBnB is more robust than GBnB under the existence of outliers.

B. *Real data*



(a) (b) (c)

Fig. 2. Objective function value of NDT, GBnB ane FBnB on the three datasets. Data1 represents original data, and data2 represents the data being transformed. These three pictures respective the results of "German Museum," "Aces" and "Intel" datasets.

In this section, we use real data to evaluate the performance of FBnB in real-world tasks. The metric in real data experiments is the value of objective function (1). Give $\mathbf{R}'$ and $\boldsymbol{t}'$, which are obtained by one of the algorithms(NDT, GBnB and FBnB). The objective function value $E(\mathbf{R}',\boldsymbol{t}')$ can reflect the overlapping degree between two point sets under the transformation. The larger it is, the more precise it is for $\mathbf{R}'$, $\boldsymbol{t}'$. The inlier threshold in (1) is set at 0.3.

In this experiment, we choose three datasets(”Intel”, ”Aces” and ”German Museum”) as the testing object. For each dataset, we randomly choose fifty scenes from it, and in each scene, the time between the first and second point sets is about 1s. We respectively utilize three algorithms to align these point sets. The results are shown in Fig.2 and represented by ”data1”.

Comparing FBnB with NDT, although NDT is faster than FBnB, the NDT is not stable. As shown in Fig.2a, in several scenes, NDT is not as precise as GBnB and FBnB, even invalid. Comparing FBnB with GBnB, they have similar accuracy in the 150 scenes and FBnB is quite faster than it in all the scenes.

To evaluate the algorithms' performance under larger transformation. We transform the second point set in each scene according to the rotation and translation which are randomly selected in the range of $[-\pi,\pi]$ and $[-5,5]$. Then we do the same experiment using the transformed datasets. The result are plotted in Fig.2 and repesented by ”data2”.

Since NDT is a local method, it depends on proper initialization to ensure its accuracy. When the transformation expands, its performance rapidly becomes worse. As for GBnB, its accuracy is not affected by the transformation expanding, but its runtime becomes longer than before. In contrast, FBnB's accuracy and efficiency has not changed.

V. Conclusion

In this work, we propose a new fast algorithm for global matching of 2D LIDAR points cloud. We use the invariant vectors to decouple the rotation and translation searching which accelerates our BnB based optimization algorithm. Besides we propose a downsampling method to improve the effectiveness of our algorithm further and ensure the robustness of it at the same time. To evaluate the performance of our algorithm, we conduct the experiments with both synthetic and real data compared with the state-of-the-art. The result of the experiments proves that our algorithm is an efficient and accurate global algorithm.

VI. Acknowledgement

This research has received funding from the National Natural Science Foundation of China (No. 61906138), and from the Shanghai AI Innovative Development Project 2018.

References

[1] J. Zhang and S. Singh, “Loam : Lidar odometry and mapping in real-time,” *Robotics: Science and Systems Conference (RSS)*, pp. 109–111, 01.

[2] 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.

[3] P. Biber and W. Strasser, “The normal distributions transform: a new approach to laser scan matching,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, Oct 2003, pp. 2743–2748 vol.3.

[4] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *IEEE International Symposium on Mixed Augmented Reality*, 2012.

[5] 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.

[6] L. Consolini, M. Laurini, M. Locatelli, and D. Lodi Rizzini, “Globally optimal registration based on fast branch and bound,” 01 2019.

[7] 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.

[8] X. Li, Y. Liu, Y. Wang, C. Wang, M. Wang, and Z. Song, “Fast and globally optimal rigid registration of 3d point sets by transformation decomposition,” 12 2018.

[9] G. Chen, H. Cao, J. Conradt, H. Tang, F. Röhrbein, and A. Knoll, “Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception,” *IEEE Signal Processing Magazine*, 2020.

[10] J. Li, H. Zhan, B. M. Chen, I. Reid, and G. H. Lee, “Deep learning for 2d scan matching and loop closure,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 763–768.

[11] Y. Shen, C. Feng, Y. Yang, and D. Tian, “Mining point cloud local structures by kernel correlation and graph pooling,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4548–4557, 2018.

[12] M. Khoury, Q.-Y. Zhou, and V. Koltun, “Learning compact geometric features,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 153–161, 2017.

[13] J. Y. Zi and G. H. Lee, “3dfeat-net: Weakly supervised local 3d features for point cloud registration,” 2018.

[14] 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.

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

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

[17] S. Bouaziz, A. Tagliasacchi, and M. Pauly, “Sparse Iterative Closest Point,” *Computer Graphics Forum*, 2013.

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

[19] 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.

[20] 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

[21] J. Bing and B. C. Vemuri, “Robust point set registration using gaussian mixture models,” *IEEE Trans Pattern Anal Mach Intell*, vol. 33, no. 8, pp. 1633–1645, 2011.

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

[23] 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.

[24] H. Li and R. Hartley, “The 3d-3d registration problem revisited,” *Proc Iccv*, pp. 1–8, 2007.

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

[26] 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.