Image and Vision Computing 67 (2017) 52–66

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Efficient rotation estimation for 3D registration and global localization in structured point clouds ☆

Yanxin Ma [a], Yulan Guo [a,c,*], Yinjie Lei [b], Min Lu [a], Jun Zhang [a]

[a] College of Electronic Science and Engineering, National University of Defense Technology, China
[b] College of Electronic and information Engineering, Sichuan University, China
[c] Institute of Computing Technology, Chinese Academy of Sciences, China

A R T I C L E I N F O

Article history:
Received 23 June 2016
Received in revised form 12 July 2017
Accepted 9 September 2017
Available online 20 September 2017

Keywords:
Structured point clouds
Small overlaps
Registration
Global localization
ICP
Direction angle

A B S T R A C T

Fully automatic 3D point cloud registration for structured scenes is a highly challenging task. In this paper, an efficient rotation estimation algorithm is proposed for point clouds of structured scenes. This algorithm fully employs the geometric information of structured environment. For rotation estimation, a direction angle is defined for a point cloud and then the rotation matrix is obtained by comparing the difference between the distributions of angles. The proposed rotation estimation algorithm is used for both 3D registration and global localization. To conduct a full 3D registration, the translation parameters are estimated by aligning the centers of the corresponding points while the rotation parameters are estimated by the proposed algorithm. For global localization, a translation estimation algorithm is proposed using projection information. The point clouds are projected onto the orthogonal plane and template matching is performed on the projection images to calculate the translation vector. Experiments have been conducted on two datasets. Experimental results demonstrate that the proposed algorithm outperforms the state-of-the-art approaches in terms of both accuracy, computational efficiency and robustness.

1. Introduction

With the rapid development of 3D scanners (e.g., LiDAR, Microsoft Kinect, ASUS Xtion), 3D modeling and mapping has been extensively investigated for the last few years [1–5]. A key problem for 3D modeling and mapping is the registration of 3D data (e.g., point clouds and range data). Point clouds of a scene are commonly acquired from different view points, registration is therefore required to transform these point clouds into a common coordinate system. The task of point cloud registration is to minimize the alignment error by calculating the rotation matrix and translation vector between these point clouds. Point cloud registration plays an important role in a number of applications including 3D Simultaneous Localization And Mapping (SLAM) [6–8], 3D modeling [9–12], and object detection/recognition [10,13,14]. Although remarkable progress has been achieved, 3D point cloud registration remains a challenging problem, especially for point clouds with small overlaps.

A classic algorithm to address the registration problem is the Iterative Closest Point (ICP) [15] algorithm. A key step for the ICP algorithm and its variants is to determine point correspondence, which requires a large overlap between two point clouds [16]. It is challenging to efficiently generate corresponding points without any prior information. Moreover, the ICP algorithm and its variants are sensitive to initial alignment. That is, these algorithms are more suitable for the registration of point clouds with a small rotation transformation. Besides, point cloud registration can also be achieved by the probabilistic algorithms [17–20]. It is shown that probabilistic algorithms outperform the ICP algorithm in the presence of noise and outliers. A popularly used probabilistic model is the Gaussian Mixture Model (GMM) [19], which considers the point clouds as GMMs and estimates the transformation between point clouds by registering the GMMs. However, it is time-consuming to generate GMM models for large-scale point clouds [17].

Most of the aforementioned algorithms require an initial alignment, which can only work on point clouds with a large overlap and small rotation. Recently, a number of local feature based algorithms [21–23] have been proposed to address the local optimization problem encountered by the ICP and probabilistic based

☆ This paper has been recommended for acceptance by Ben Glocke.

* Corresponding author at: College of Electronic Science and Engineering, National University of Defense Technology, China.
E-mail addresses: mayanxin@nudt.edu.cn (Y. Ma), yulan.guo@nudt.edu.cn (Y. Guo), yinjie@scu.edu.cn (Y. Lei), lumin@nudt.edu.cn (M. Lu), zj67068@sina.com (J. Zhang).

https://doi.org/10.1016/j.imavis.2017.09.003
0262-8856/

algorithms, e.g., the Binary Robust Appearance and Normal Descriptor (BRAND) [24], Rotational Projection Statistics (RoPS) [13], and Depth-Interpolated Image Features (DIFT) [21]. However, feature extraction and matching is computationally expensive [22]. Moreover, point clouds can be partially overlapped and acquired from textureless scenes. Developing a fast and accurate registration algorithm for point clouds with small overlaps and large rotation is highly demanded. Estimation of the rotation parameters is a key step in 3D point cloud registration. Therefore, this paper focuses on the rotation estimation problem and proposes a novel algorithm using the geometric information of structured environment (see Fig. 1). Fig. 1 shows two point clouds for registration (as shown in Fig. 1 (a)) and their direction angle histograms (as shown in Fig. 1 (b) and (c)). There is only a rotation around the *Z* axis between point clouds *A* and *B*. The direction angle is defined as the angle between the projection of the normal vector of each point and the Cartesian coordinate axis. From left to right in Fig. 1 (b) and (c), the histograms represent the direction angles around the *X*, *Y*, and *Z* axes, respectively. From Fig. 1 (b) and (c), we can observe that only the histogram around the *Z* axis is different and the others are almost the same. Therefore, the direction

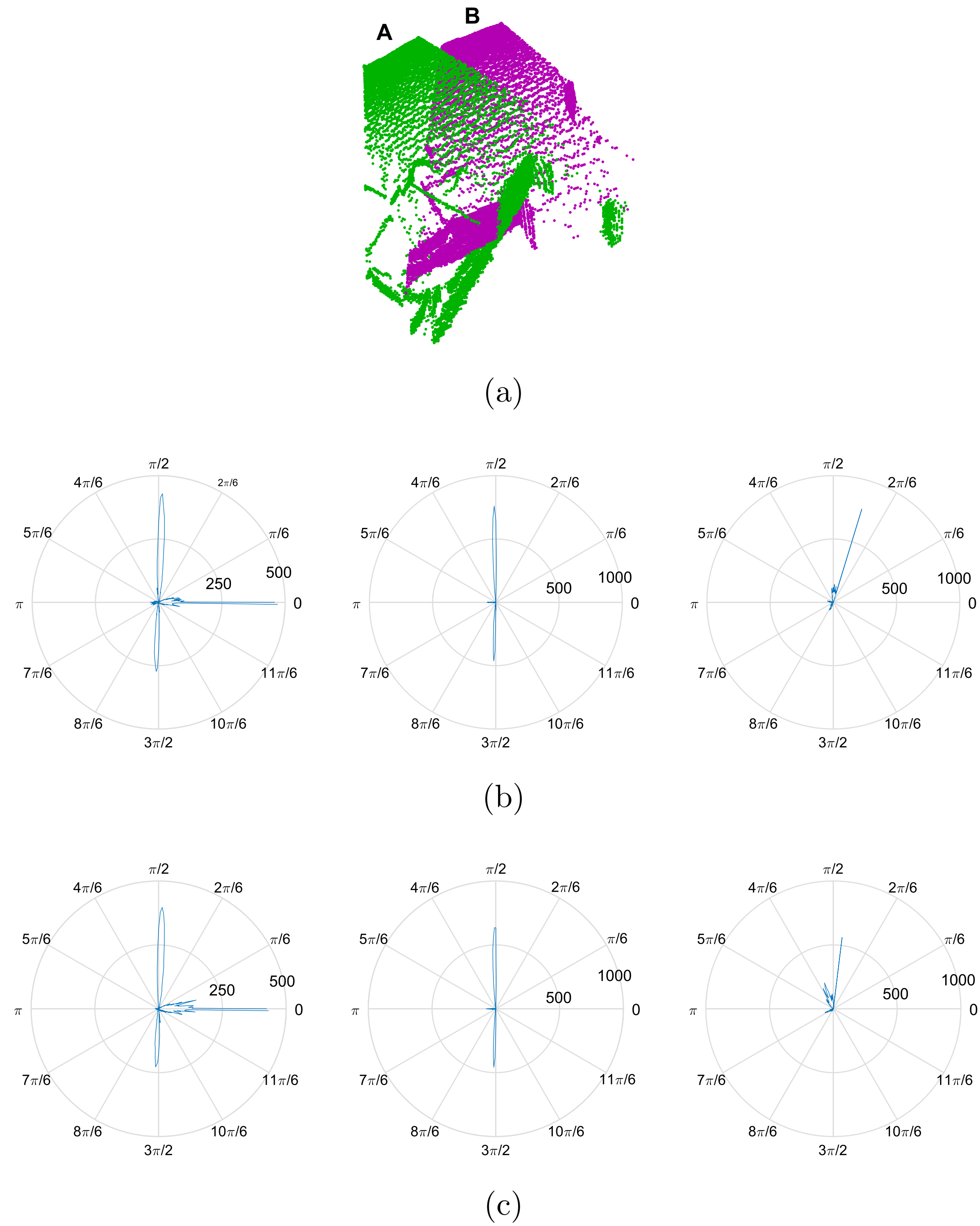


Fig. 1. Direction angle histograms of two point clouds. (a) Two point clouds for registration. (b–c) Direction angle histograms of point cloud *A* and *B* in (a).

angles can reflect the rotation between two point clouds. Based on this observation, a direction angle is first defined and the statistics information of the direction angles is used to obtain the rotation matrix.

To perform 3D registration, our registration algorithm is decomposed into two parts: rotation estimation and translation estimation. For rotation estimation, a novel estimation algorithm is proposed using the direction angle histograms. For translation estimation, the centers of the rotated point clouds are aligned using a variant of ICP. Specifically, we only conduct translation estimation without considering the rotation. Since the rotation angles are precisely estimated, we can obtain an accurate translation estimation after a few iterations (10 in our case). The proposed rotation estimation algorithm does not require any initial alignment, and can be implemented efficiently as it does not have to generate point correspondences. A set of experiments have been conducted on indoor point clouds. Experimental results show that the proposed algorithm outperforms the ICP algorithm [15,25], the Coherent Point Drift (CPD) algorithm [18], the Accelerated Coherent Point Drift (ACPD) algorithm [17] and the algorithm proposed in [26] on point clouds with small overlaps. Moreover, the proposed rotation estimation algorithm is used to address the global localization problem in 3D maps. For translation estimation, we use the projection information of each point cloud. The point cloud is projected onto the coordinate plane and the translation vector is then extracted by template matching between the two corresponding projection images. Experiments have been conducted on indoor point clouds. Experimental results show that the proposed algorithm achieves a high localization accuracy.

The rest of this paper is organized as follows. Section 2 reviews existing work on point cloud registration. Section 3 defines our problem. Section 4 introduces our direction angle distribution based rotation estimation algorithm. Section 5 presents the comparative experimental results and analyses in point clouds registration and global localization. Section 6 concludes this paper.

2. Related work

2.1. 3D point cloud registration

Despite of the recent research progress, 3D point cloud registration remains a challenging task. The task of 3D point cloud registration is to estimate the rotation and translation parameters to align the 3D point clouds acquired from different viewpoints. Many existing 3D registration algorithms consider the rotation and translation estimation task as an iterative optimization problem. The ICP algorithm [15] and its variants [16,25,27-33] have been frequently used to perform 3D point cloud registration. ICP is one of the earliest point cloud registration techniques, it uses a least square method to achieve optimal matching between two point clouds. At each step, the algorithm estimates the rotation and translation parameters by minimizing the Euclidean distance between corresponding points. Subsequently, many ICP variants have been proposed. Granger and Pennec [31] formulated the registration problem as a general Maximum-Likelihood (ML) estimation of the transformation and proposed an improved ICP using Expectation-Maximization (EM) principles, namely EM-ICP. To improve the computational efficiency of EM-ICP, Tamaki et al. [32] implemented the EM-ICP using CUDA and achieved a faster performance than the CPU based implementation. Chen and Medioni [30] used a more robust point-to-plane distance to replace the point-to-point distance in the original ICP. This variant improves the robustness and applicability of ICP. Segal et al. [27] developed a probabilistic version of ICP, namely Generalized-ICP (GICP). GICP also uses point-to-plane distance to generate point correspondences, and assigns a weight to each pairs of corresponding points in the objective function using their surface normals. Serafin and Grisetti [29] used normal features to obtain more accurate point correspondences, resulting in an improved registration accuracy for large-scale point clouds. The classical ICP algorithm and its variants require an accurate initial alignment and a large percent of overlaps. Moreover, only a local optima can be achieved by these algorithms.

Besides, several registration algorithms [17–20] consider a point cloud as a probabilistic model (e.g., GMM [19]). Evangelidis et al. [20] proposed a generative model based on multiple GMMs for joint registration of multiple point clouds. Myronenko et al. [18] proposed a Coherent Point Drift (CPD) algorithm to achieve high registration accuracy by maximizing the likelihood between the two GMMs of point clouds. The original CPD algorithm is time-consuming. Myronenko and Song [18] then proposed a Fast Gauss Transform based CPD (FGT-CPD) algorithm. To achieve a robust and faster registration, Lu et al. [17] proposed an Accelerated Coherent Point Drift (ACPD) algorithm. Experimental results show that the computational complexity of these probabilistic algorithms is very high, especially for large-scale point clouds.

Generally, the ICP variants and the probabilistic algorithms are prone to local optima. Yang et al. [28] proposed a Global Optimal ICP (GO-ICP) algorithm for point cloud registration. The GO-ICP algorithm combines the traditional ICP with a branch-and-bound (bnb) scheme to perform rotation estimation. It achieves a global accurate registration in a small-scale scene. Recently, several algorithms have been proposed to estimate the rotation and translation parameters [26,34-37]. Generally, these algorithms are able to obtain global optima at the cost of high computational complexity. A similar technique is used to estimate the rotation between two 3D point clouds in [34,35,37]. First, keypoints are extracted from the given point clouds, followed by the generation of keypoint matches. The keypoint matches [35,37] or the original points [34] can then be used to solve the point correspondence problem. The point correspondence searching process in [34,35,37] is slow, which requires at least three correct keypoint matches to guarantee reliable feature correspondence. As a major component in these algorithms, keypoint extraction and matching is computationally expensive. Furthermore, it is challenging for keypoint extraction algorithms to work in textureless environments. Makadia et al. [26] used the orientation histogram in spherical coordinates to estimate the rotation. Once a rough rotation and translation estimation is completed, ICP is performed to achieve fine alignment. Similar to [26], the global direction information of the normal vectors of all points is used for rotation estimation and a variant of ICP is used for translation estimation in this paper. Different from [26], we define a measure named direction angle and use the peaks of the direction angle histograms to obtain the rotation angles. No Fourier transform or Inverse Fourier transform (which is the key step in [26]) is needed in our algorithm. For automatic 3D point cloud registration in structural environments, the proposed algorithm is simple but effective. No keypoint or local feature extraction is required by our algorithm. Furthermore, our algorithm is more robust to outliers.

2.2. Global localization

Real-time localization forms the basis for outdoor and indoor navigation [38–41]. It can be used for many applications including robotics, and unmanned vehicles. The task of global localization is to estimate its position without any initial knowledge about its location. It also requires the platform (e.g., a vehicle or robot) to be able to relocate itself if its position is lost.

Indoor robot localization is an extensively investigated topic in the area of global localization. A large number of approaches have been proposed for mobile robot global localization using RGB/RGB-D cameras. Global navigation is usually achieved by matching a local map to the global map using global or local features extracted from the maps. Schiele and Crowley [38] proposed a Hough transform based localization algorithm. It uses Hough transform to extract environmental information from the grid map obtained by a sensor (local map) and a given grid map (global map). The information from both grid maps are then compared in the Euclidean space to obtain the location of the robot. Markov localization [39] is a typical feature matching based localization algorithm. It matches all the points acquired by the sensor to the reference data to perform localization. Since it fully uses the acquired data for matching, it can work in any environment. However, this algorithm is computationally expensive. Fang et al. [41] transformed both the local and global maps to the Hough space to achieve global localization. Compared to Markov localization [39], its computational complexity is reduced, but its localization accuracy is relatively poor. Considering geometric information of the indoor environment, Biswas and Veloso [40] extracted walls from the acquired point cloud and matched the walls to the global 2D map to achieve global localization. This algorithm achieves high location accuracy. However, its localization performance highly depends on a complete global map. To address the limitations of existing algorithms, this paper considers localization as a registration problem between the local and global map. The proposed algorithm is based on our previous work [42,43] and fully employs the geometric information of the indoor environment.

3. Problem definition

Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ to be two point clouds for registration, $\boldsymbol{p}_i = \left(x_i^p, y_i^p, z_i^p\right) \in \boldsymbol{P}$, $\boldsymbol{q}_j = \left(x_j^q, y_j^q, z_j^q\right) \in \boldsymbol{Q}$, are the points in $\boldsymbol{P}$ and $\boldsymbol{Q}$, $\left\{\boldsymbol{n}_i^p\right\}$ and $\left\{\boldsymbol{n}_j^q\right\}$ are the normal vectors of all the points in the point clouds, where $i = 1, 2, \cdots, N, j = 1, 2, \cdots, M$. The task of a registration algorithm is to estimate the rotation matrix $\mathbf{R} \in SO(3)$ and the translation vector $\boldsymbol{t} \in \mathbb{R}^3$ between the two given point clouds to minimize the alignment error e:

$$e(\mathbf{R}, \boldsymbol{t}) = \sum_i \left\|\mathbf{R}\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_{j*}\right\|^2 \quad (1)$$

Point $\boldsymbol{q}_{j^*} \in \boldsymbol{Q}$ denotes the corresponding point of $\boldsymbol{p}_i$.

Assuming that the translation vector $\boldsymbol{t} \in \mathbb{R}^3$ is known as $\boldsymbol{t}^*$, the alignment error e turns to be:

$$e\left(\mathbf{R}, \boldsymbol{t}^*\right) = \sum_i \left\|\mathbf{R}\boldsymbol{p}_i + \boldsymbol{t}^* - \boldsymbol{q}_{j*}\right\|^2 \quad (2)$$

Assuming that the rotation matrix $\mathbf{R} \in SO(3)$ is known as $\mathbf{R}^*$, the alignment error e turns to be:

$$e\left(\mathbf{R}^*, \boldsymbol{t}\right) = \sum_i \left\|\mathbf{R}^*\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_{j*}\right\|^2 \quad (3)$$

A registration algorithm is usually designed to minimize the error defined by Eq. (1). To obtain the optimal results, the point correspondences between two point clouds have to be determined. However, this process is difficult and time-consuming. It is observed that point clouds of structured scenes usually have several planes perpendicular to each other, e.g., the floor and walls (as shown in Fig. 1 (a)). The angle between the corresponding planes before and after a certain rotation in the scene contains the information for global rotation transformation. As a result, the angular relationship between the corresponding planes can be used to estimate the rotation between two point clouds. Consequently, this clue can be used for point cloud registration in structured scenes. By using the plane correspondence information for registration, no point correspondence is required anymore, and the efficiency can be significantly improved. Since normal vectors explicitly give the planar information of a scene, the direction angle histogram of normals can be used to estimate the rotation between two point clouds (see Fig. 1 (b) and (c)). Since planes cannot be completely occluded in a scene, the proposed algorithm is expected to be robust to small overlaps and large occlusion.

4. Fast and accurate rotation estimation

The rotation between two point clouds can be decomposed into three rotation angles around three orthogonal axes. The angular transformation between two planes can be calculated from their normal vectors. A rotation estimation problem defined in the Cartesian coordinate system can be converted to a translation estimation problem of angle parameters in the polar coordinate system. Consequently, we first extract the angles of surface normals and then achieve rotation estimation using these angles.

4.1. 2D direction angle

Given a 2D directional vector $\boldsymbol{p} = (x, y) \in \mathbb{R}^2$ where $x^2 + y^2 \neq 0$, and a reference direction vector $\boldsymbol{n}_x = (1, 0)$ on the XY plane, the 2D direction angle d_a of $\boldsymbol{p}$ is defined as:

$$d_a(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right), & \text{if } y > 0 \text{ and } x \geq 0 \\ \arctan\left(\frac{y}{x}\right) + \pi, & \text{if } y < 0 \\ \arctan\left(\frac{y}{x}\right) + 2\pi, & \text{if } y < 0 \text{ and } x \geq 0 \end{cases} \quad (4)$$

It is clear that the direction angle is within the range of $[0, 2\pi)$ and can be repeated with a period of 2π. Actually, the direction angle on the XY plane represents the rotation angle around the Z axis. Fig. 2 shows the d_a values over a region on the XY plane where x is within $[-1, 1]$ and y is within $[-1, 1]$. It can be observed from Fig. 2 that the angle function has a monotonous property in a certain region, which guarantees the uniqueness of the direction angle within the range of $[0, 2\pi)$.

4.2. 3D direction angle

Let $\boldsymbol{n} = (n_x, n_y, n_z)$ be the normal vector of a 3D point $\boldsymbol{q} = (x, y, z) \in \mathbb{R}^3$, $\boldsymbol{n}$ uniquely determines the tangent plane of $\boldsymbol{q}$. In

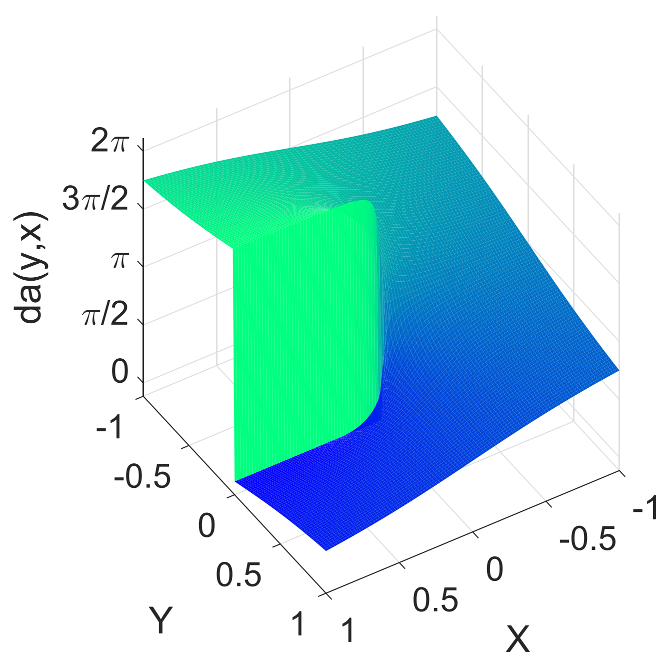


Fig. 2. The 2D direction angle of points located on the XY plane.

our work, the normal vectors are determined by performing Principal Component Analysis (PCA) on a local patch consisting of 10 nearest points of $\boldsymbol{q}$ [44]. Before PCA operation, a simple denoising process [45] is conducted on the point clouds to eliminate outliers. To avoid the ambiguity of normal vectors produced by PCA, the direction of each normal vector is further normalized after performing PCA, i.e., the normal vector $\boldsymbol{n}$ is oriented to the direction of the positive Z axis, as shown in Fig. 3. That is,

$$\boldsymbol{n} = \begin{cases} \boldsymbol{n}, & \text{if} \quad n_z \geq 0 \\ -\boldsymbol{n}, & \text{if} \quad n_z < 0 \end{cases} \tag{5}$$

Then, $\boldsymbol{n}$ is projected onto three coordinate planes, resulting in three projection directions $\boldsymbol{n_{xy}} = (n_x, n_y)$, $\boldsymbol{n_{yz}} = (n_y, n_z)$, $\boldsymbol{n_{xz}} = (n_x, n_z)$. Next, a direction angle d_a can be obtained on each coordinate plane. We define the direction angle of normal $\boldsymbol{n}$ on the XY, YZ, XZ planes as d_Z, d_X, d_Y, respectively. If point $\boldsymbol{q}$ is rotated around the Z axis by an angle θ, the new direction angle on the XY plane is calculated as $\tilde{d}_Z$. All the direction angles are calculated according to Eq. (4). Considering the periodicity property of direction angles, we have:

$$\theta = \text{mod}\left(\tilde{d}_Z - d_Z, 2\pi\right) \tag{6}$$

where mod$(\cdot)$ denotes the modulo operation.

Similarly, if point $\boldsymbol{q}$ is rotated around the X axis by angle α, the new direction angle on the YZ plane is $\tilde{d}_X$. We then have:

$$\alpha = \text{mod}\left(\tilde{d}_X - d_X, 2\pi\right) \tag{7}$$

If point $\boldsymbol{q}$ is rotated around the Y axis by angle β, the new direction angle on the XZ plane is $\tilde{d}_Y$. We then have:

$$\beta = \text{mod}\left(\tilde{d}_Y - d_Y, 2\pi\right) \tag{8}$$

Consequently, we can obtain the rotation angles α, β, θ based on Eqs. (6), (7) and (8).

4.3. Direction angle histogram based rotation estimation

If we directly use the direction angles to calculate the rotation angles by Eqs. (6), (7) and (8), we have to know the exact point correspondences for d_X and $\tilde{d}_X$, d_Y and $\tilde{d}_Y$, d_Z and $\tilde{d}_Z$ in two point clouds. However, it is very challenging to obtain these exact point correspondences between two point clouds in practice. Instead, we can obtain the statistical information of d_X, d_Y and d_Z for each point cloud, and then estimate the rotation using these statistics (e.g., histograms).

Let $\{a\}$ be a set of direction angles within the range of $[0, 2\pi)$, its histogram $Hist(a)$ can be denoted as:

$$Hist(a) = \{[H_k, R_k]\}, \quad k = 1, 2, \cdots, N_{bin} \tag{9}$$

where H_k is the number of points with their direction angle falling into the k-th bin, and mean value in the k-th bin is defined as R_k. Extensive experimental results show that the algorithm can obtain satisfactory computational efficiency when the number of bins N_{bin} is set to 3600.

Let $\{\tilde{a}\}$ be the direction angles for the points after a rotation with a rotation angle ε, then we have:

$$\tilde{a} = \text{mod}(a + \varepsilon, 2\pi) \tag{10}$$

We therefore have:

$$\begin{aligned} Hist(\tilde{a}) &= \left\{\left[\tilde{H}_k, \tilde{R}_k\right]\right\} \\ \tilde{R}_k &= \text{mod}\,(R_k + \varepsilon, 2\pi) \end{aligned} \tag{11}$$

Let $\{d_X\}$, $\{d_Y\}$, $\{d_Z\}$ be the sets of direction angles of the whole point cloud before rotation, $\left\{\tilde{d}_X\right\}$, $\left\{\tilde{d}_Y\right\}$, $\left\{\tilde{d}_Z\right\}$ be the direction angle sets after rotation, their histograms are defined as:

$$\begin{aligned} Hist\,(d_X) &= \{[H_{Xk}, R_{Xk}]\}\,, Hist\left(\tilde{d}_X\right) = \left\{\left[\tilde{H}_{Xk}, \tilde{R}_{Xk}\right]\right\} \\ Hist\,(d_Y) &= \{[H_{Yk}, R_{Yk}]\}\,, Hist\left(\tilde{d}_Y\right) = \left\{\left[\tilde{H}_{Yk}, \tilde{R}_{Yk}\right]\right\} \\ Hist\,(d_Z) &= \{[H_{Zk}, R_{Zk}]\}\,, Hist\left(\tilde{d}_Z\right) = \left\{\left[\tilde{H}_{Zk}, \tilde{R}_{Zk}\right]\right\} \end{aligned} \tag{12}$$

We can then obtain the rotation angle based on Eqs. (6), (7), (8) and (11), that is:

$$\begin{cases} \alpha = \text{mod}\left(\tilde{R}_{Xk} - R_{Xk}, 2\pi\right) \\ \beta = \text{mod}\left(\tilde{R}_{Yk} - R_{Yk}, 2\pi\right) \\ \theta = \text{mod}\left(\tilde{R}_{Zk} - R_{Zk}, 2\pi\right) \end{cases} \tag{13}$$

Consequently, we can easily calculate the rotation angles α, β, θ from the histograms of direction angle using Eq. (13). However, R_{Xk} and $\tilde{R}_{Xk}$, R_{Yk} and $\tilde{R}_{Yk}$, R_{Zk} and $\tilde{R}_{Zk}$ must correspond to the same points in the point cloud pairs. In fact, the peak bin (which is defined as the angle that occupies the largest number of points) in the direction angle histograms satisfies this condition, as shown in Fig. 4. Fig. 4 shows the three sole peaks in the direction angle histogram around the X, Y and Z axes (from left to right), which are marked by red

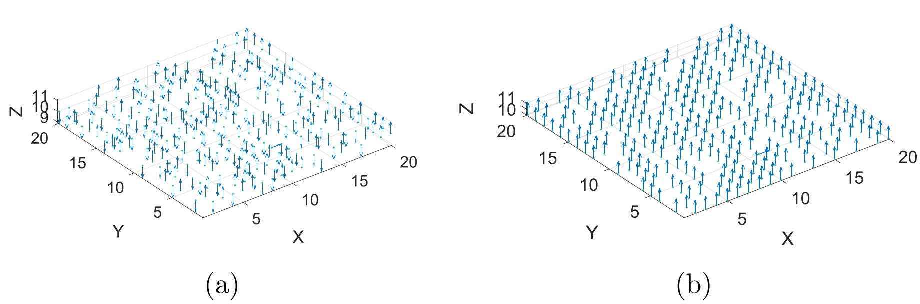


Fig. 3. An illustration of the normal normalization. (a) Normal vectors before normalization. (b) Normal vectors after normalization.

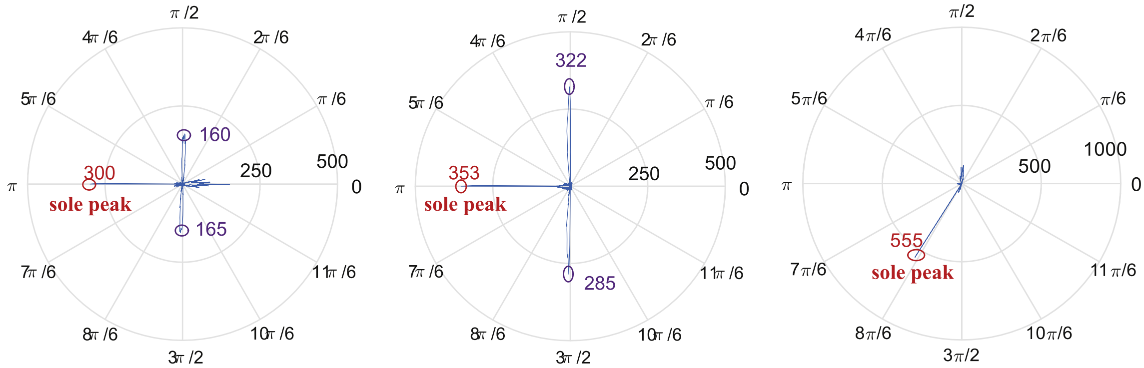
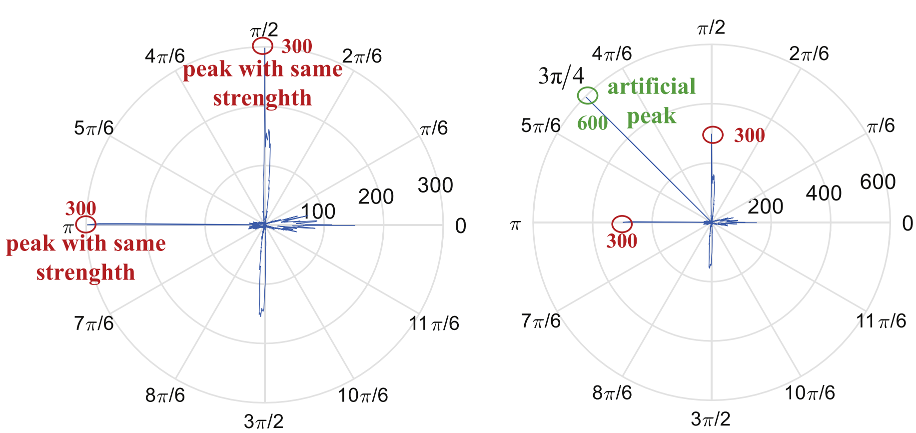


Fig. 4. The sole peak in each histogram (marked by red circle).

circles. If there is more than one peak, the average value of all the peaks is used as the final peak, as shown in Fig. 5. We define average value of all the peaks as to average their orientations in the histogram and sum their number of points. As shown in Fig. 5, if there are two exactly same peaks at angle $\pi 2$ and π with 300 points in the original histogram (as shown in Fig. 5 (a) marked by red circles), the artificial peak after the average turns to be a sole peak at $3\pi 4$ with 600 points (as shown in Fig. 5 (b) marked by green circles).

Fig. 6 shows an example of the direction angle histograms for an indoor point cloud. The direction angle histograms of the given point cloud are shown in Fig. 6 (a). We then rotate the point cloud by angles of $\pi 6$, $\pi 3$, $\pi 2$ round the X, Y, Z axes, respectively. The direction angle histograms of the rotated point clouds are shown in Fig. 6 (b). It can be seen that the differences between the peaks of the direction angle histograms of $\{d_X\}$, $\{d_Y\}$ and $\{d_Z\}$ are $\pi 6$, $\pi 3$, $\pi 2$, which is the same as the rotation angles between the original point cloud and the transformed point cloud. As shown in Fig. 7, the ground points are parallel to the XY plane. The points on the side wall are almost parallel to the YZ plane. The peaks in the direction angle histogram around the X axis are mainly generated by ground points, as shown in the right top of Fig. 7. The peaks in the direction angle histogram around the Y axis are mainly generated by ground points and wall points, as shown in the right middle of Fig. 7. The peaks in the direction angle histogram around the Z axis are mainly generated by wall points, as shown in the right down of Fig. 7. Since there is only one side wall, the difference between the peak values in the direction angle histogram around the X and Z axes are more obvious than the peak values in the direction angle histogram around the Y axis. When the rotation is only around the X axis, the peak values remain the same but the positions of the peaks are changed in the direction angle histogram around the X axis before and after rotation, as shown in the left figures in Fig. 6 (a) and (b). Thus, these positions of peaks in direction angle histogram around the X axis before and after rotation can be used to estimate the rotation angle around the X axis in our proposed algorithm. Similarly, the positions of peaks in the direction angle histograms around the Y and Z axes before and after rotation are used to estimate the rotation angle around the Y and Z axes, respectively. Therefore, the rotation around an axis can be determined by our histogram based algorithm without using any point correspondences.

Our algorithm can exactly extract the rotation angle when the point cloud is rotated around only one axis. However, a complete 3D rotation always consists of three rotation angles around the three coordinate axes. To obtain an accurate rotation estimation, an iteration is repeated to update the rotation matrix. During iteration, the rotation estimation process is iteratively performed around



(a) (b)

Fig. 5. Two peaks with exactly the same strength in the histogram and the artificial peak generated by averaging their orientations in the histogram and summing their number of points. (a) The two exactly same peaks in the histogram are shown in red circles. (b) The artificial peak generated from (a) is shown in green circle.

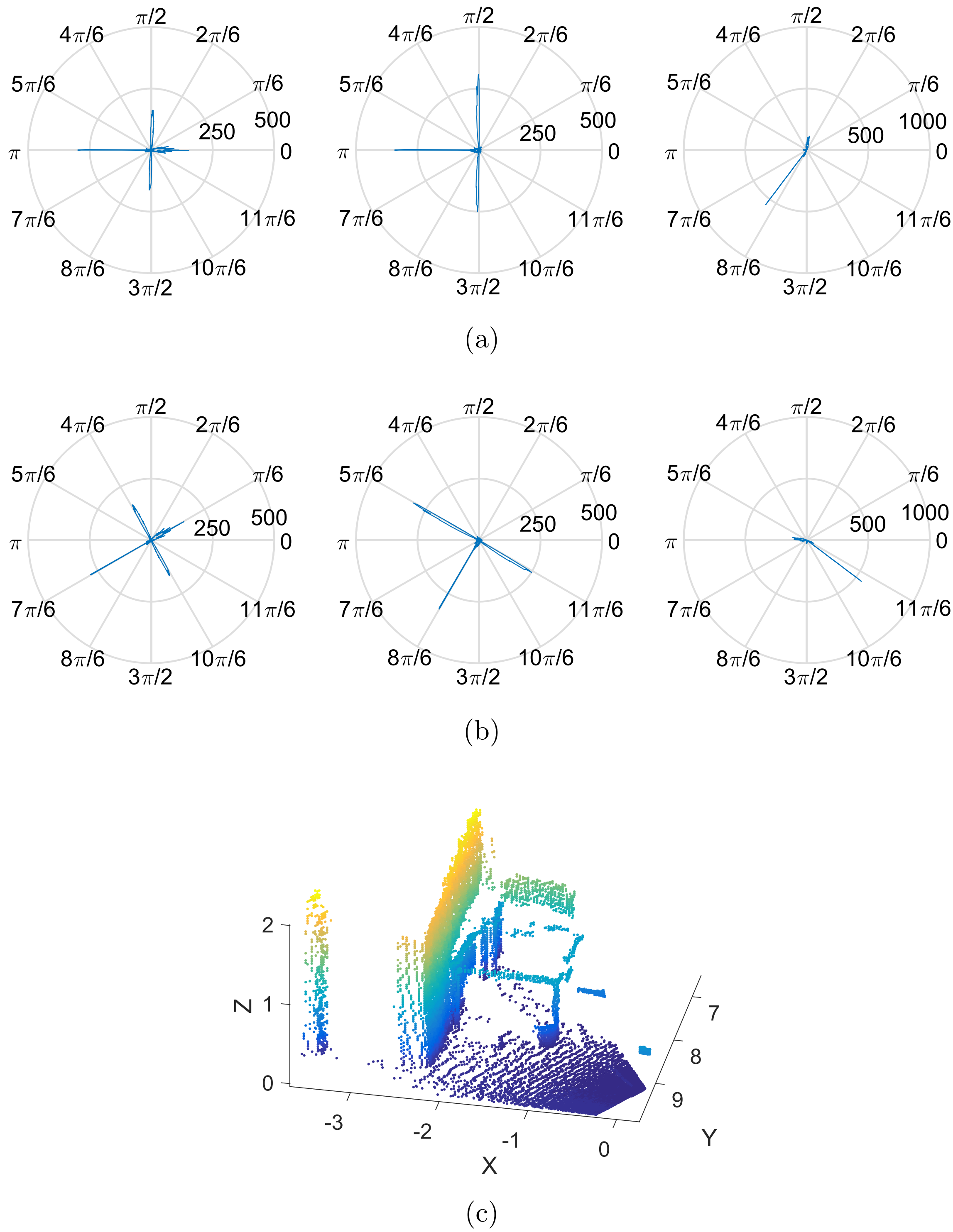


Fig. 6. An illustration of the direction angle histograms. (a–b) Three direction angle histograms of the original point cloud (as shown in (c)) and the rotated point cloud. From left to right, the figures show the direction angle histograms around the X, Y and Z axes, respectively. Due to the outliers and noise, the histograms are not smooth and even have more than 2 side peaks.

the Z, Y and X axes. Specifically, once the rotation around an axis is estimated, the rotation of the point cloud around this axis is corrected. Then, the rotation around next axis is estimated and corrected until the rotation around the three axes are estimated. The iteration is repeated until the maximum number of iterations is reached. During the process of rotation estimation and correction, the rotation of the point cloud around each axis is continuously reduced. Consequently, the rotations around three axes are accurately estimated. Experimental results show that the rotation estimation algorithm always achieves an acceptable result after about 3 iterations. The whole process for rotation estimation is summarized in Algorithm 1.

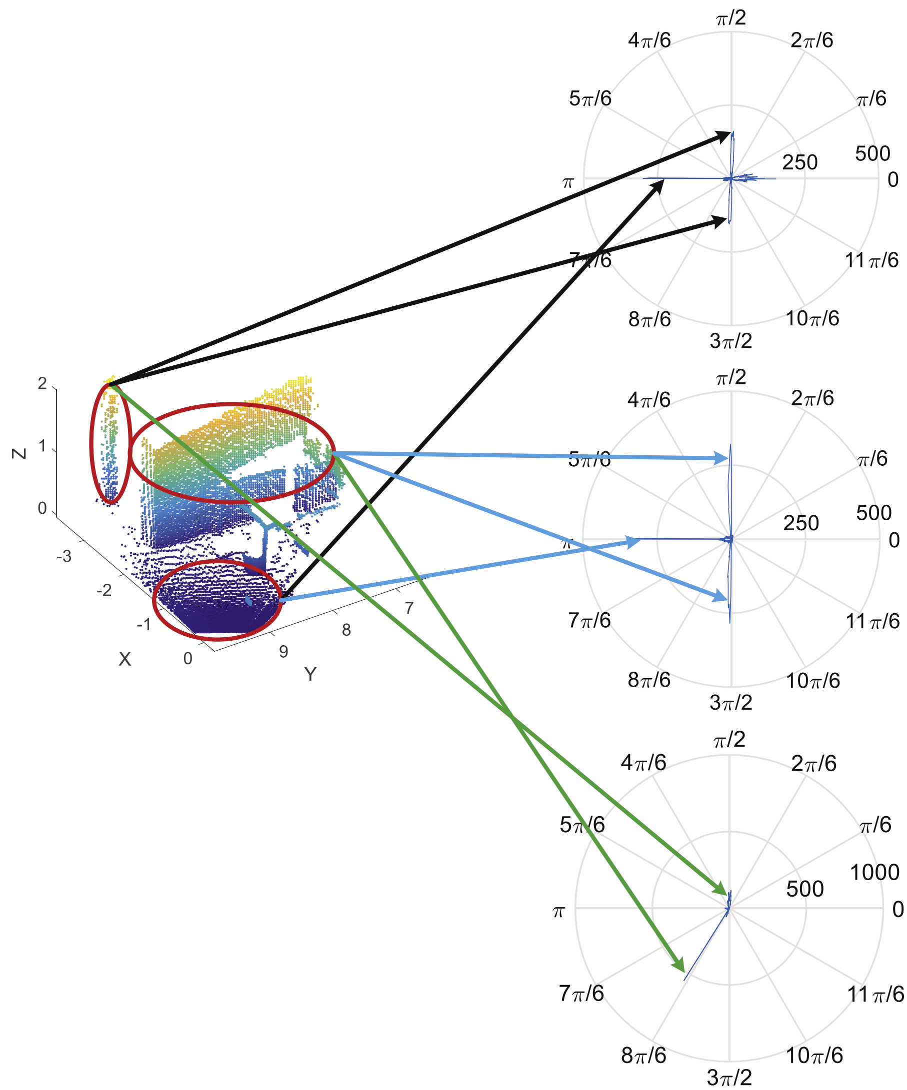


Fig. 7. The correspondence between the point cloud and the direction angle histogram.

Algorithm 1. Direction Angle based Rotation Estimation

Input: point clouds $\boldsymbol{P}$ and $\boldsymbol{Q}$, the max number of iterations N_{iter}

1. Calculate the histograms $Hist\,(d_X)$, $Hist\,(d_Y)$, $Hist\,(d_Z)$ of $\boldsymbol{P}$ using the normals of points in $\boldsymbol{P}$.
2. Set the rotation matrix $\mathbf{R}^*$ to be an identity matrix, set the iteration number $n_{iter} = 0$.
3. **while** $n_{iter} < N_{iter}$ **do**
4. Calculate the histogram $Hist\left(\widetilde{d}_Z\right)$ of $\boldsymbol{Q}$.
5. Compare the positions of the peaks in $Hist\,(d_Z)$ and $Hist\left(\widetilde{d}_Z\right)$ to obtain the rotation angle θ;
6. Calculate the rotation matrix $\mathbf{R}$ around the Z axis based on θ;
7. $\boldsymbol{Q} = \boldsymbol{Q}\mathbf{R}$, $\mathbf{R}^* = \mathbf{R}^*\mathbf{R}$;
8. Calculate the rotation angles α and β around the X and Y axes following the similar approach as for θ;
9. $n_{iter} = n_{iter} + 1$;
10. **end while**
11. **return** rotation matrix $\mathbf{R}^*$.

5. Experiments

To test the performance of our algorithm, a set of experiments were conducted. The proposed 3D rotation estimation algorithm was applied to both the 3D point cloud registration and global localization problems.

5.1. 3D point cloud registration

5.1.1. Experimental setup

To accomplish a full 3D point cloud registration, rotation estimation is achieved by our proposed algorithm, translation estimation is achieved by aligning the centers of corresponding points extracted from the point clouds after rotation normalization using a variant of ICP [15]. In these experiments, our algorithm was compared to five state-of-the-art approaches including Fully Automatic Registration of 3D point clouds [26] (denoted as FAR), CPD [18], ACPD [17], ICP using point-to-point distance [15] (denoted as ICP) and ICP using point-to-plane distance [30] (denoted as ICPP). Experiments were performed on the ETH Hauptgebaude dataset [46] (as shown in Fig. 8). The ETH Hauptgebaude dataset was acquired from the main building of ETH Zurich by moving a laser scanner Hokuyo UTM-30LX in a long corridor along a straight direction. Hokuyo UTM-30LX uses

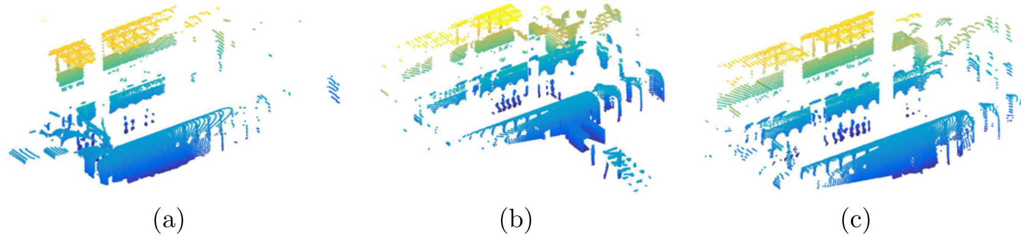


Fig. 8. Three point clouds in the ETH Hauptgebaude dataset.

laser source ($\lambda = 870$ nm) to scan a semicircular field with a viewing angle of 270° in azimuth. It measures the distance to objects and the coordinates of those points lying on objects using a step angle of 0.25°. Totally, there are 1080 measurement steps to cover the whole space of 270° in azimuth. Its maximum working distance is 30 m and its accuracy is 30 mm [46]. The point clouds in this dataset are already registered using the poses of the laser scanner acquired by an IMU sensor and a GPS system.

To speedup the process, we performed uniform downsampling on the given point cloud. All experiments were conducted in MATLAB 2016a on a PC with 4GHz Intel Core i7 CPU and 16G RAM.

To evaluate the performance of a registration algorithm, we measure the pairwise alignment error between the ground-truth pose and the estimated pose using Relative Rotational Error (RRE) e_R and Relative Translational Error (RTE) e_T, which is similar to [29]. RRE is calculated as:

$$e_R = \sum_{i=1}^{3} \boldsymbol{angle}(i)$$
$$\boldsymbol{angle} = f\left(\mathbf{R}_T^{-1}\mathbf{R}_E\right) \tag{14}$$

where $\mathbf{R}_T$ is the ground-truth rotation matrix, $\mathbf{R}_E$ is the estimated rotation matrix, and $f(\cdot)$ is a function to transform a rotation matrix to three Euler angles $\{\boldsymbol{angle}(i), i = 1, 2, 3\}$. RRE is the sum of the absolute differences in three Euler angles.

RTE is calculated as:

$$e_T = \|\boldsymbol{t}_T - \boldsymbol{t}_E\|_2 \tag{15}$$

where $\boldsymbol{t}_T$ is the ground-truth translation vector and $\boldsymbol{t}_E$ is the estimated translation vector.

5.1.2. Performance with different bin sizes

In this section, we evaluated the performance of our algorithm with respect to different bin sizes of the direction angle histogram. 100 pairs of point clouds with an average overlap of around 50% were selected for experiments. We applied a rotation with angles of 10°, 10° and 10° around the *X*, *Y*, and *Z* axes, respectively, and the translation was set to 0. The bin size in a histogram is increased from 1800 to 18,000. The bin size was determined by the bin width from 0.02° to 0.2° with a step of 0.02°. The results are shown in Fig. 9. It can be observed that with the increase of bin size, the performance of the proposed algorithms is gradually increased. Furthermore, when the bin size is larger than 4500, the performance is dramatically decreased. The best estimation result of about 0.6° is achieved when the bin size is between 3600 and 4500. Therefore, the bin size is set to 3600 to achieve a compromise between computational efficiency and estimation accuracy.

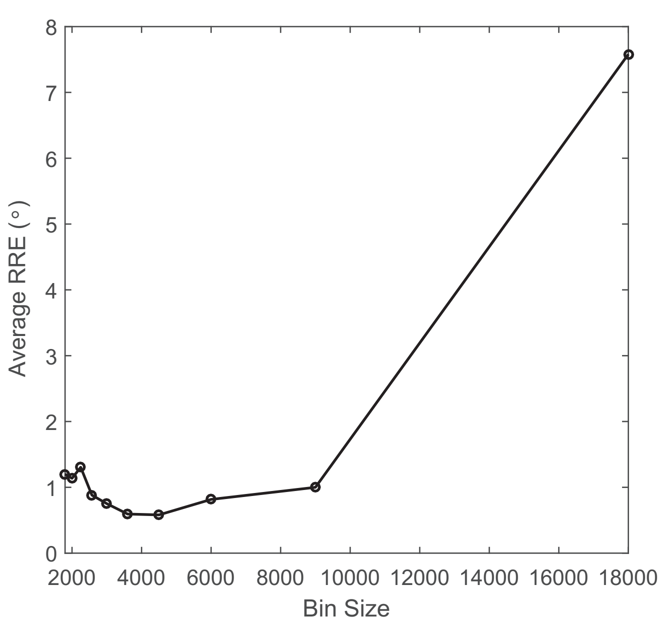


Fig. 9. The RRE result achieved on point clouds with different bin sizes of the direction angle histogram.

5.1.3. Robustness to different rotations

In this section, we evaluated the performance of our algorithm with respect to different rigid transformations (including rotation and translation) of point clouds. We set the maximum number of iterations as $N_{iter} = 3$ in our experiments, while the maximum number of iterations for ICP and ICPP were set to 100. The same parameters for FAR were used as in [26]. The parameters of each algorithm were fixed in different experiments. 100 pairs of point clouds with an average overlap of around 50% were selected for experiments. For each pair of point clouds, we only applied different rotations to one of the two point clouds. The translation between the point clouds was set to 0. The RRE results achieved by different algorithms are shown in Fig. 10.

From Fig. 10, it can be observed that the proposed algorithm achieves a very high accuracy on point clouds under large rotations. The RRE is as small as 4.08° when the rotation between point clouds is 80°. As the rotation angle increases, we observe a decrease of the performance of other algorithms. However, the performance of the proposed algorithm remains almost unchanged. Since there is an exhaustive search over the large space of corresponding peaks in FAR, FAR achieves the second best performance as compared to the proposed algorithm. ACPD and CPD achieve the third and fourth best performance. That is because the structured point cloud in ETH Hauptgebaude can be well modeled by GMM. Since the test point clouds contain a large number of planes with similar features and the overlap degree is low (about 50%), the point correspondence

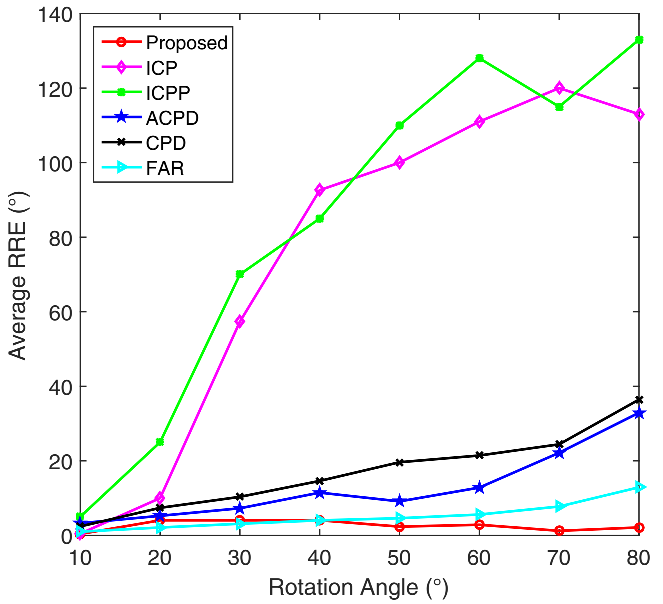


Fig. 10. The RRE result achieved on point clouds with different rotation angles around each axis.

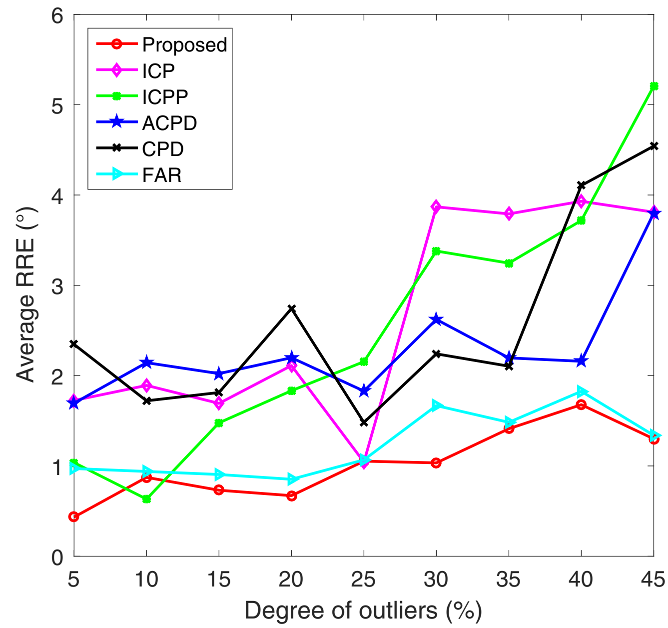


Fig. 11. The RRE results on point clouds with different degrees of outliers.

cannot be accurately estimated. Consequently, the ICP and ICPP algorithms obtain the worst performance, they even cannot be converged when the rotation angle is larger than 30°. In contrast, the proposed algorithm achieves the best performance. The large number of planes in the scene is highly beneficial to our direction angle histogram based rotation estimation algorithm.

5.1.4. *Robustness to outliers*

In this section, we evaluated the rotation estimation performance of our algorithm on point clouds with different degrees of outliers. We selected the same pairs of point clouds as in Section 5.1.2. We rotated a point cloud in each pair with the angles of 10°, 10° and 10°around the *X*, *Y*, and *Z* axes, respectively, and set the translation to (1 m, 1 m, 1 m).

To evaluate the performance of our algorithm under different levels of outliers, we randomly selected N_o points in one of the point cloud pair and added a displacement to each point along the normal direction. The mean of the displacement was set to 0 and the standard deviation of the displacement was set to 1 m. N_o was set to a selection ratio of the number of the point cloud, ranging from 5% to 45% with a step of 5%. Fig. 11 shows the RRE results achieved by different algorithms on the selected data under different levels of outliers.

From Fig. 11, it can be observed that with the increase of outliers, the performance of all algorithms is decreased. The proposed algorithm achieves the best performance under almost all the degrees of outliers. In that case, the structure information of the given point clouds remains unaffected when the degree of outliers is less than 45%. Therefore, our proposed algorithm is robust to outliers in real-world environments.

5.1.5. *Robustness to missing data*

To evaluate the performance of our algorithm under missing data, we randomly selected one point in a point cloud while eliminating N_d points closest to the selected point. N_d was set to a selection ratio of the number of the point cloud, ranging from 5% to 45% with a step of 5%. The results are shown in Fig. 12, it is clear that the performance of other algorithms is decreased as the degree of missing data is increased, except the proposed algorithm and FAR. However, our algorithm achieves a smaller RRE error. The results demonstrate that our proposed algorithm is robust to missing data.

5.1.6. *Robustness to the variation of point numbers*

In addition, we evaluated the performance of our algorithm on point clouds with the variation of point numbers. To obtain point clouds with different numbers, a uniform down-sampling operation was performed on the selected point clouds. The results are shown in Fig. 13, it can be observed that the proposed algorithm, CPD, ACPD and FAR achieve very stable performance on point clouds with the variation of point numbers. Our algorithm achieves the smallest RRE error on most point clouds with varying number of points. As the point number becomes larger, the number of noisy points is also increased. Consequently, the error of ICP and ICPP is increased. The

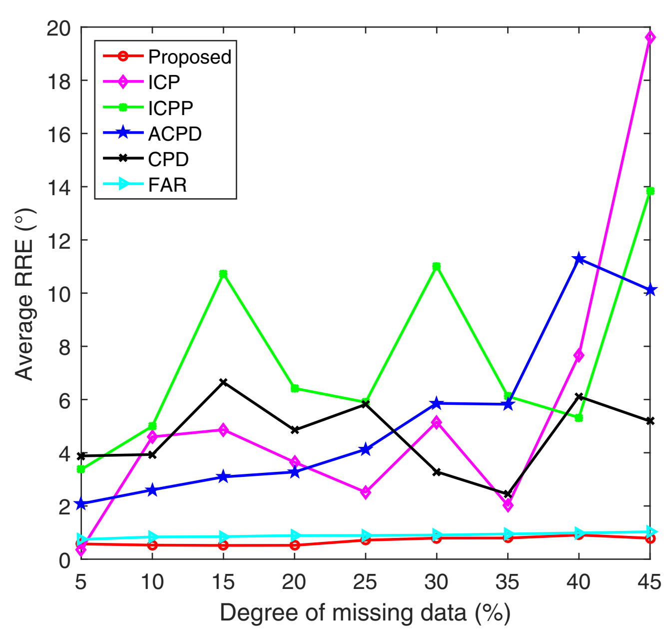


Fig. 12. The RRE results on point clouds with different degrees of missing data.

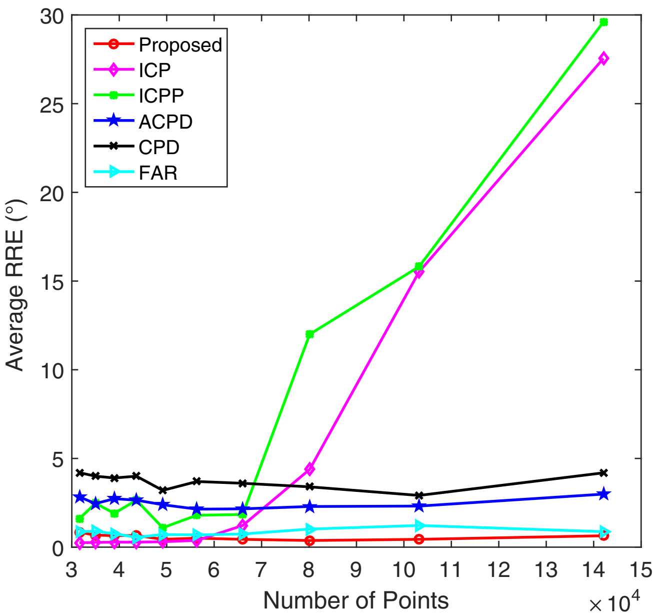


Fig. 13. The RRE results on point clouds with varying number of points.

experimental results demonstrate that the proposed algorithm is robust to the variation of point numbers.

5.1.7. Robustness to noise

To evaluate the performance of our algorithm under different levels of noise, we randomly added Gaussian noise to all of the points in one of the point cloud pair. For the given dataset, the standard deviation of measurement error is 30 mm. We evaluated the performance of registration algorithms under noise with standard deviations ranging from 2 mm to 12 mm with a step of 2 mm, while the mean value of noise was set to 0. The results are shown in Fig. 14, it can be seen that the performance of all algorithms is reduced as the degree of noise is increased. The proposed algorithm achieves the best performance under noise.

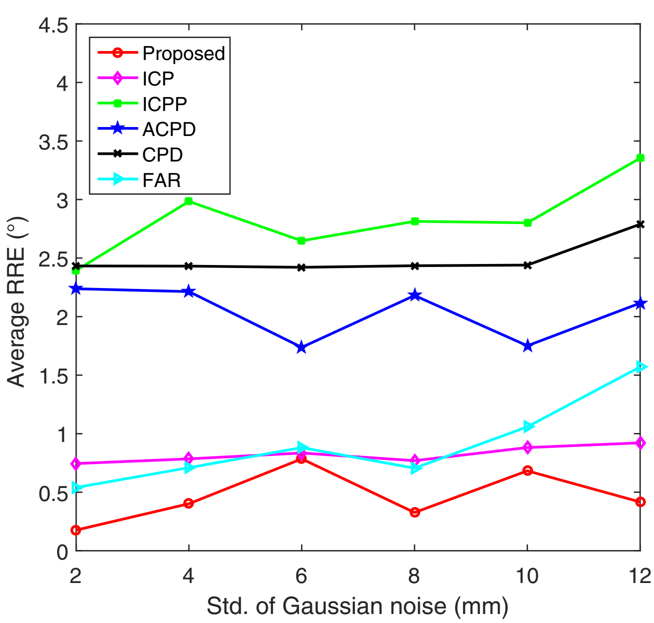


Fig. 14. The RRE results on point clouds with different degrees of noise.

5.1.8. Robustness to different overlaps

In this section, we evaluated the performance of our algorithm on point clouds with different degrees of overlaps. The ground-truth overlap degree between each two point clouds is given by the dataset, as shown in Fig. 15.

490 pairs of point clouds with different overlap degrees were used for our test. The rotation angles around the X, Y, and Z axes were set to 10°, 10° and 10°, respectively, and the translation was set to (1m, 1m, 1m). The final registration results are shown in Fig. 16. The RRE and RTE results are averaged on all point cloud pairs.

From Fig. 16, it can be observed that the registration performance of ICP, ICPP, ACPD, CPD and FAR deteriorates as the decrease of the overlap between point clouds. In contrast, the performance of the proposed algorithm remains stable. For rotation estimation, the proposed algorithm achieves the best performance even on point clouds with a small overlap less than 50%. The average RRE of our proposed algorithm on all point clouds is 0.71°, while the RRE achieved by FAR is 0.91°. Compared to FAR, we mainly used the direction information of the normal vectors. For points on the same plane, the direction information of their normal vectors is the same. Therefore, the direction information of a few points can represent the direction information of their coplanar points. Meanwhile, there are a large number of perpendicular or parallel planes in the structured scene, so rotation estimation can achieve an acceptable estimation results even if only a few overlapping points exist. It should be noted that these overlapping points should be distributed in different planes. The estimation is failed if these overlapping points are lying on the same plane.

Since the rotation estimation result has a significant influence on translation estimation, the proposed algorithm also achieves the best performance in translation estimation. The average RTE achieved by our proposed algorithm on all point clouds is 0.38 m. This clearly demonstrates the robustness of our algorithm with respect to small overlaps. For the ICP and ICPP algorithms, sufficient point correspondences are unavailable between two point clouds when the overlap is small. Similarly, for ACPD and CPD, a particular probabilistic model cannot be properly generated for two point clouds with small overlaps. In addition, a qualitative comparison of translation estimation between the proposed algorithm and FAR was presented. As reported in [26], when there is an average rotation angle of $2^\circ$and an average translation distance of 3.8 in., an average estimation error of 1.2 in. (computed as the minimum distance from each point in the

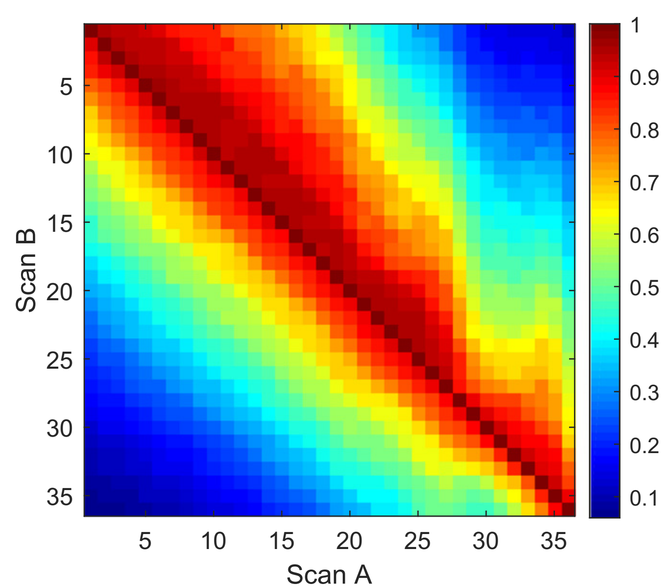


Fig. 15. The overlap degrees between different scans.

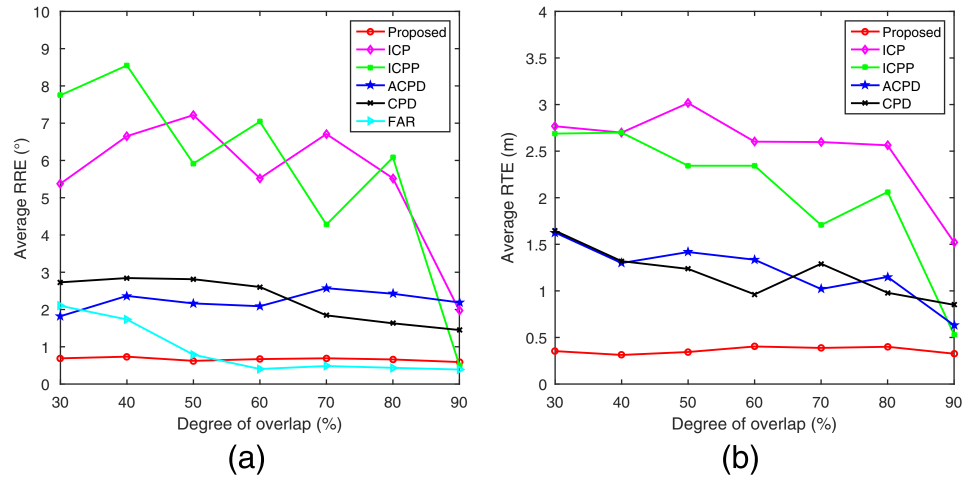


Fig. 16. The RRE (a) and RTE (b) results on point clouds with different degrees of overlaps.

aligned scan to the original scan) is obtained by FAR. We used a similar dataset and applied a similar rotation and translation with the same distribution of rotation angle and translation distance on the data. Our algorithm obtained an average translation estimation error of 0.65 inches and an average rotation estimation error of 0.05°. In contrast, the proposed algorithm fully employs the statistical information of surface normals. Consequently, it is not needed to extract a set of corresponding points from the overlapping areas. Therefore, the robustness of the algorithm is improved.

5.1.9. Computational time

In this section, we calculated the average computational time of each algorithm for registering each pair of point clouds in the ETH Hauptgebaude dataset, the results are shown in Table 1. In this experiments, the rotation angles around the X, Y, and Z axes were set to 10°, 10° and 10°, respectively, and the translation was set to (1 m, 1 m, 1 m). Each original point cloud includes about 200,000 points. To obtain an acceptable registration, we down-sampled the point clouds to about 55,000 points using a uniform down-sampling operation. It can be observed that the proposed algorithm achieves the fastest performance, followed by FAR and ICP. The proposed algorithm outperforms the other algorithms by an order of magnitude, it takes only 2.13 s for point cloud registration. Since there is an exhaustive search over the large space of corresponding peaks in FAR, FAR is more time-consuming than the proposed algorithm. In contrast, CPD and ACPD are the slowest algorithms, they take more than 38 s to register two point clouds. The computational burden of ACPD and CPD is mainly caused by the construction of GMM models. For ICP and ICPP, point corresponding process is time-consuming. In contrast, our algorithm uses the histograms of surface normals, which can be generated very efficiently.

5.2. Global localization

We further evaluated the performance of our algorithm for global localization. The map data is acquired by a Kinect sensor on an indoor robot. The Kinect device has two cameras: a RGB camera with a resolution of 640 × 480 pixels working at 30 Hz, and an IR camera with a resolution of 320 × 240 pixels working at 30 Hz. Its working distance is within the range of 0.7 m to 6 m, with a horizontal viewing angle of 57° and a vertical viewing angle of 43° [46]. The map data is acquired by a Kinect sensor mounted on an indoor robot, which is generated by the Real-Time Appearance-Based MAPping method (RTAB-MAP) [47]. Therefore, the acquired point clouds are vertically aligned. Given a global and a local 3D map acquired by the Kinect (as shown in Figs. 17 (a) and 18(a)), the direction angle is first extracted using the rotation estimation algorithm. Once the accurate direction is obtained, we then have to estimate the position of the local map in the global map. The task of position estimation is to calculate the translation between the local and global point clouds.

Table 1
The registration time for different algorithms.

Algorithm	ACPD	CPD	ICP	ICPP	FAR	Proposed
Time (s)	38.14	42.74	12.67	13.62	9.78	2.13

In practice, there is only translation on the ground plane for the data acquired by an indoor robot. Template matching can achieve an acceptable localization accuracy with less computational complexity than ICP for the global localization task. Therefore, we use template matching to estimate the translation in this paper. Given a large map and a small map (namely, template map), template matching is used to search in the large map for the part same as the template map. Template matching is used to address the problem when the template map is a subset of the large map. The main steps are described as follows. First, the two 3D point clouds are projected onto the XY plane (as indoor robots usually moves on the XY plane) to generate two 2D projection maps. Second, correlation operation is used to obtain the planar translation between the two projection maps.

5.2.1. Point cloud projection

First, the center of the point cloud is translated to the origin of the coordinate system, and the points whose Z values are within a certain range (e.g., below the height of the 3D sensor) are projected onto the XY plane. Second, the XY plane is partitioned into several bins and the number of projection points in each bin is counted, resulting in a projection map (see Figs. 17 (b) and 18(b)).

5.2.2. Translation estimation

Given a large map S with size of $W \times H$, and a template map $Temp$ with the size of $W_T \times H_T$. The search area in S is named as S_{ij}, where i, j indicate the global coordinates of the top-left pixel of the search area in S, $1 \le i \le W - W_T$, $1 \le j \le H - H_T$.

Then, the similarity $\rho(i,j)$ between $Temp$ and S_{ij} is calculated as:

$$\rho(i,j) = \frac{\sum_{x,y} Temp(x,y) \times S_{ij}(x,y)}{\sqrt{\sum_{x,y} \left\| S_{ij}(x,y) \right\|^2} \sqrt{\sum_{x,y} \left\| Temp(x,y) \right\|^2}} \quad (16)$$

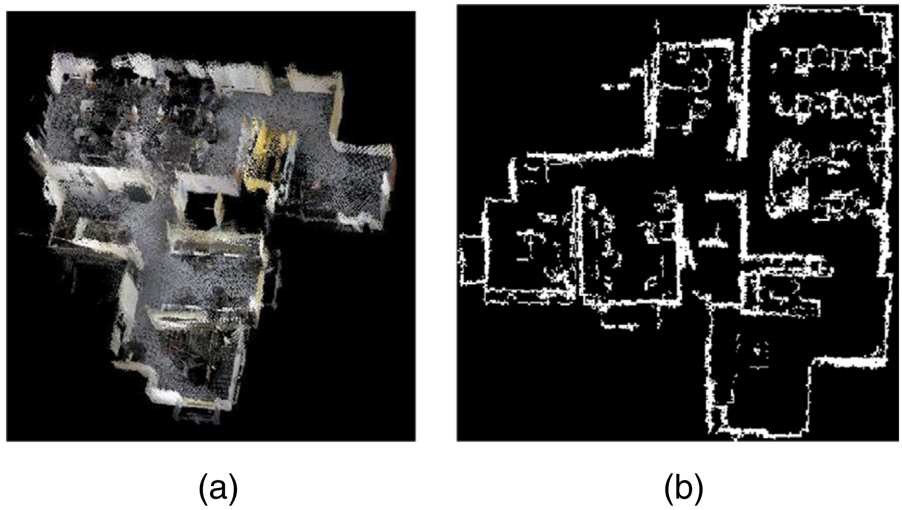


Fig. 17. Global map represented by a 3D point cloud (a) and its 2D projection map (b).

where x, y denote the pixel coordinate, $1 \leq x \leq W_T$, $1 \leq y \leq H_T$. The coordinate (c_i, c_j) with the largest value of $\rho(i,j)$ gives the required location, that is:

$$(c_i, c_j) = \arg\max_{i,j} \ \rho(i,j) \tag{17}$$

The proposed algorithm was tested on an indoor dataset acquired by a Kinect sensor mounted on a Turtlebot robot. The dataset contains a 3D global map, 10 manually labeled ground-truth positions and 100 local maps (i.e., 10 3D local maps at each position). Fig. 19 shows the labeled ground-truth positions (as shown in blue stars) and the localization results (as shown in red stars).

Since the global localization problem significantly depends on hardware, data sources and other factors, we therefore performed qualitative comparison between our algorithm and several existing algorithms. The average localization error achieved by our algorithm is 0.15 m, which is smaller than the radius of our robot (0.18 m). Fang et al. [41] achieved a localization error within 0.1 m with the support of sonar. Biswas and Veloso [40] achieved a failure rate of 2% at the end of 4 km trials using the Kinect sensor only. In [40], a trial was considered to be successful if the localization error was smaller than 1 m. Compared to these localization algorithms, experimental results clearly show that our algorithm achieves a promising global localization performance. The success of our localization algorithm mainly caused by the accurate direction estimation.

6. Conclusions

This paper has presented a novel rotation estimation algorithm for 3D registration. The rotation between two point clouds is estimated using histograms of direction angles. The proposed algorithm has been tested on two datasets for both 3D registration and global localization. Experiments have been conducted on real and synthetic datasets. Experimental results show that the proposed algorithm achieves both high accuracy and computational efficiency, it is also very robust to small overlaps, noise, missing data and large percent of outliers. It outperforms several existing algorithms including ICP,

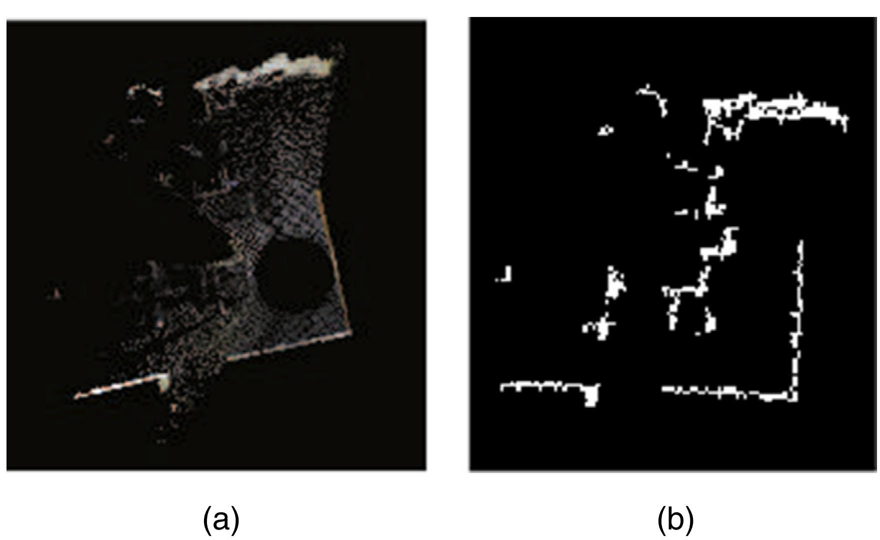


Fig. 18. Local map represented by a 3D point cloud (a) and its 2D projection map (b).

Fig. 19. The ground-truth positions and the localization results achieved at ten positions.

CPD, ACPD and the algorithm proposed in [26]. Our algorithm is further applied to indoor robot global localization, with a promising localization accuracy being achieved.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61602499, 61601488, 61471371, 61403265), the National Postdoctoral Program for Innovative Talents (No. BX201600172), and the Science and Technology Plan of Sichuan Province (No. 2015SZ0226).

References

[1] D. Holz, S. Behnke, Mapping with micro aerial vehicles by registration of sparse 3D laser scans, International Conference on Intelligent Autonomous Systems, 2016. pp. 1583–1599.
[2] F. Yu, J. Xiao, T. Funkhouser, Semantic alignment of LiDAR data at city scale, IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 1722–1731.
[3] C. Kerl, J. Stuckler, D. Cremers, Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras, IEEE International Conference on Computer Vision, 2015. pp. 2264–2272.
[4] R. Or-El, G. Rosman, A. Wetzler, R. Kimmel, A.M. Bruckstein, RGBD-Fusion: real–time high precision depth recovery, IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 5407–5416.
[5] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, KinectFusion: real-time dense surface mapping and tracking, IEEE International Symposium on Mixed and Augmented Reality, 2011. pp. 127–136.
[6] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J.M. Rendón-Mancha, Visual simultaneous localization and mapping: a survey, Artif. Intell. Rev. 43 (1) (2015) 55–81.
[7] H. Strasdat, J.M.M. Montiel, A.J. Davison, Visual SLAM: why filter? Image Vis. Comput. 30 (2) (2012) 65–77.
[8] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: large-scale direct monocular SLAM, European Conference on Computer Vision, 2014. pp. 834–849.
[9] M. Hernandez, J. Choi, G. Medioni, Near laser-scan quality 3-D face reconstruction from a low-quality depth stream, Image Vis. Comput. 36 (2015) 61–69.
[10] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, An integrated framework for 3D modeling, object detection and pose estimation from point-clouds, IEEE Trans. Instrum. Meas. 64 (3) (2015) 683–693.
[11] K. Chen, Y. Lai, Y.-X. Wu, R.R. Martin, S.-M. Hu, Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information, ACM Trans. Graph. 33 (6) (2014) 208.
[12] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, M. Lu, An accurate and robust range image registration algorithm for 3D object modeling, IEEE Trans. Multimedia 16 (5) (2014) 1377–1390.
[13] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, Rotational projection statistics for 3D local surface description and object recognition, Int. J. Comput. Vis. 105 (1) (2013) 63–86.
[14] R.O. Castle, G. Klein, D.W. Murray, Combining monoSLAM with object recognition for scene augmentation using a wearable camera, Image Vis. Comput. 28 (11) (2010) 1548–1556.
[15] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.
[16] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets, Auton. Robot. 34 (3) (2013) 133–148.
[17] M. Lu, J. Zhao, Y. Guo, Y. Ma, Accelerated coherent point drift for automatic 3D point cloud registration, IEEE Geosci. Remote Sens. Lett. 13 (2) (2016) 162–166.
[18] A. Myronenko, X. Song, Point set registration: coherent point drift, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2262–2275.
[19] B. Jian, B.C. Vemuri, Robust point set registration using Gaussian mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1633–1645.
[20] G.D. Evangelidis, D. Kounades-Bastian, R. Horaud, E.Z. Psarakis, A generative model for the joint registration of multiple point sets, European Conference on Computer Vision, 2014. pp. 109–122.
[21] H. Andreasson, A.J. Lilienthal, 6D scan registration using depth-interpolated local image features, Robot. Auton. Syst. 58 (2) (2010) 157–165.
[22] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N.M. Kwok, A comprehensive performance evaluation of 3D local feature descriptors, Int. J. Comput. Vis. 116 (1) (2016) 66–89.
[23] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3D object recognition in cluttered scenes with local surface features: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (11) (2014) 2270–2287.
[24] E.R. Nascimento, G.L. Oliveira, M.F. Campos, A.W. Vieira, W.R. Schwartz, BRAND: a robust appearance and depth descriptor for RGB-D images, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012. pp. 1720–1726.
[25] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, International Conference on 3-D Digital Imaging and Modeling, 2001. pp. 145–152.
[26] A. Makadia, A. Patterson, K. Daniilidis, Fully automatic registration of 3D point clouds, IEEE Conference on Computer Vision and Pattern Recognition, 2006. pp. 1297–1304.
[27] A. Segal, D. Haehnel, S. Thrun, Generalized-ICP, Robotics: Science and Systems, 2009.
[28] J. Yang, H. Li, Y. Jia, GO-ICP: solving 3d registration efficiently and globally optimally, IEEE International Conference on Computer Vision, 2013. pp. 1457–1464.
[29] J. Serafin, G. Grisetti, NICP: dense normal based point cloud registration, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015. pp. 742–749.
[30] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image Vis. Comput. 10 (3) (1992) 145–155.
[31] S. Granger, X. Pennec, Multi-scale EM-ICP: a fast and robust approach for surface registration, European Conference on Computer Vision, Springer. 2002, pp. 69–73.
[32] T. Tamaki, M. Abe, B. Raytchev, K. Kaneda, Softassign and EM-ICP on GPU, 2010 First International Conference on Networking and Computing, IEEE, 2010. pp. 179–183.
[33] Y. Liu, Automatic registration of overlapping 3D point clouds using closest points, Image Vis. Comput. 24 (7) (2006) 762–781.
[34] A. Bustos, T.J. Chin, D. Suter, Fast rotation search with stereographic projections for 3D registration, IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 3930–3937.
[35] J.C. Bazin, Y. Seo, M. Pollefeys, Globally optimal consensus set maximization through rotation search, Asian Conference on Computer Vision, 2012. pp. 539–551.
[36] R.I. Hartley, F. Kahl, Global optimization through rotation space search, Int. J. Comput. Vis. 82 (1) (2009) 64–79.
[37] J.C. Bazin, C. Demonceaux, P. Vasseur, I. Kweon, Motion estimation by decoupling rotation and translation in catadioptric vision, Comput. Vis. Image Underst. 114 (2) (2010) 254–273.
[38] B. Schiele, J.L. Crowley, A comparison of position estimation techniques using occupancy grids, International Conference on Robotics and Automation, 1994. pp. 1628–1634.
[39] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, J. Artif. Intell. Res. 11 (1999) 391–427.
[40] J. Biswas, M. Veloso, Depth camera based indoor mobile robot localization and navigation, International Conference on Robotics and Automation, 2012. pp. 1697–1702.
[41] F. Fang, X. Ma, X. Dai, Mobile robot localization based on improved model matching in hough space, International Conference on Intelligent Robots and Systems, 2006. pp. 1541–1546.

[42] Y. Ma, Y. Guo, J. Zhao, M. Lu, J. Zhang, J. Wan, Fast and accurate registration of structured point clouds with small overlaps, IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016. pp. 1–9.
[43] Y. Ma, Y. Guo, M. Lu, J. Zhao, J. Zhang, Global localization in 3D maps for structured environment, IEEE International Geoscience and Remote Sensing Symposium, 2016. pp. 6680–6683.
[44] H. Hoppe, T. DeRose, T. Duchampt, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, Comput. Graph. 26 (1992) 2.
[45] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D point cloud based object maps for household environments, Robotics and Autonomous Systems Journal 56 (11) (2008) 927–941.
[46] F. Pomerleau, M. Liu, F. Colas, R. Siegwart, Challenging data sets for point cloud registration algorithms, Int. J. Robot. Res. 31 (14) (2012) 1705–1711.
[47] M. Labbe, F. Michaud, Online global loop closure detection for large-scale multi-session graph-based SLAM, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014. pp. 2661–2666.