Fast and Accurate Registration of Structured Point Clouds with Small Overlaps

Yanxin Ma, Yulan Guo, Jian Zhao, Min Lu, Jun Zhang, Jianwei Wan
ATR National Key Laboratory, College of Electronic Science and Engineering
National University of Defense Technology, Changsha, Hunan, China
{mayanxin, yulan.guo, zhaojian, lumin, zhangjun, wanjianwei}nudt.edu.cn

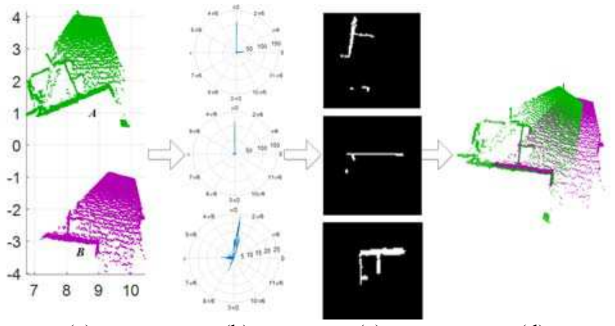
Abstract

To perform registration of structured point clouds with large rotation and small overlaps, this paper presents an algorithm based on the direction angles and the projection information of dense points. This algorithm fully employs the geometric information of structured environment. It consists of two parts: rotation estimation and translation estimation. 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. For translation estimation, the point clouds are projected onto three orthogonal planes and then a correlation operation is performed on the projection images to calculate the translation vector. Experiments have been conducted on several datasets. Experimental results demonstrate that the proposed algorithm outperforms the state-of-the-art approaches in terms of both accuracy and efficiency.

1. Introduction

Point cloud registration has been extensively investigated in the past three decades [1-4]. The task of point cloud registration is to calculate the rotation matrix and translation vector to minimize the alignment error between two point clouds. Point cloud registration plays an important role in a number of applications including Simultaneous Localization And Mapping (SLAM) [5-7], 3D reconstruction [8, 9], and object detection/recognition [10, 11]. Although remarkable progress has been achieved, 3D point cloud registration remains a challenging problem, especially for point clouds with small overlaps.

The Iterative Closest Point (ICP) algorithm [12] and its variants [13-16] have been frequently used to perform 3D point cloud registration. ICP uses a least square method to achieve optimal matching between two point clouds. Yang *et al.* [15] proposed a Global Optimal ICP (Go-ICP) method for point cloud registration. Serafin *et al.* [13] used normal features to obtain more accurate point correspondence, resulting in an improved registration accuracy for large-scale point clouds. Pomerleau *et al.* [17] developed a fast ICP algorithm for real-time SLAM. The



(a) (b) (c) (d)

Figure 1. The workflow of the proposed algorithm. (a) Point clouds for registration. (b) Direction angle histograms of point cloud *B* for rotation estimation (Sec. 2.1). (c) Projection images of point cloud *B* for translation estimation (Sec. 2.2). (d) Registration results. (Figure best seen in color.)

key step of these methods is to determine the point correspondence, which requires a large overlap between two point clouds [14, 17]. However, in real scenes, it is difficult to accurately search 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 [14, 17].

Another way to achieve point cloud registration is the probabilistic methods. It is shown that probabilistic methods [1, 18-21] outperform the ICP algorithm in the presence of noise and outliers. Jian *et al.* [19] used a Gaussian Mixture Model (GMM) to address the point cloud registration problem. Evangelidis *et al.* [21] 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. Meanwhile, Myronenko *et al.* [18] also proposed a Fast Gauss Transform based CPD algorithm (FGT-CPD). Lu *et al.* [20] proposed an Accelerated Coherent Point Drift (ACPD) algorithm to achieve fast registration. Experimental results show that the computational complexity of these probabilistic methods is high, especially for large-scale point clouds [20].

Most of the aforementioned algorithms require an initial alignment, and can only work on point clouds with a large overlap. Recently, a number of local feature based algorithms have been proposed [10, 22] to address the local optimization problem faced by the ICP and probabilistic based methods. However, feature extraction and matching is computationally expensive. Developing a fast and accurate registration algorithm for point clouds with small overlaps and large rotation is highly demanded. Therefore, this paper proposes a novel algorithm to handle this problem using the geometric information of structured environment (see Fig. 1). From Fig. 1, it can be observed that the rotation between two point clouds (as shown in Fig. 1(a)) can be inferred from their direction angles (as shown in Fig. 1(b)). Meanwhile, the translation between two point clouds can be inferred from their projection images (as shown in Fig. 1(c)). Based on this observation, our registration algorithm is decomposed into two parts: rotation estimation and translation estimation. For rotation estimation, a direction angle is first defined and the statistics information of the direction angles is used to obtain the rotation matrix. For translation estimation, the point cloud is projected onto three coordinate planes and the the translation vector is then extracted by calculating the correlation between the corresponding projection images. The proposed algorithm does not require any initial alignment, and can be implemented efficiently as it does not have to calculate the point correspondences. A set of experiments have been conducted on indoor point clouds. Experimental results show that the proposed algorithm outperforms ICP and FGT- CPD on point clouds with small overlaps.

The rest of this paper is organized as follows. Section 2 introduces our direction angle distribution based registration algorithm. Section 3 presents the comparative experimental results and analyses. Section 4 concludes this paper.

2. Point cloud registration

Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ to be two point clouds for registration, $\boldsymbol{p}_i = (x_i^p, y_i^p, z_i^p) \in \boldsymbol{P}, i = 1, 2, \cdots, n$, $\boldsymbol{q}_j = (x_j^q, y_j^q, z_j^q) \in \boldsymbol{Q}$, $j = 1, 2, \cdots, m$ are the points in $\boldsymbol{P}$ and $\boldsymbol{Q}$, $\{\boldsymbol{n}_i^p\}$ and $\{\boldsymbol{n}_j^q\}$ are the normals of the points. 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 :

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

Point $q_{j*} \in Q$ denotes the corresponding point of p_i .

A registration algorithm is usually designed to minimize the error defined by Eq. 1. To obtain the optimal results, the point correspondence relationship between two point clouds has to be determined. However, this process is difficult and also time-consuming.

It is observed that point clouds of indoor scenes usually have several planes perpendicular to each other, e.g., the floor and walls (as shown in Fig. 1). The angle between the corresponding planes before and after a certain rotation in the scene contains the information of the global rotation transformation. Similarly, the position of corresponding planes contains the information of the global translation. As a result, the angular and positional relationship between the corresponding planes can be used to estimate the transformation 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 gives 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 the projection can be used to calculate the translation (see Fig. 1(c)). Since planes cannot be completely occluded in a scene, the proposed algorithm is expected to be robust to small overlaps and large occlusion.

2.1. Direction angle distribution based 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. Actually a rotation problem in the Cartesian coordinate system can be converted to a translation 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.

2.1.1 2D direction angle

Given a 2D direction vector $\boldsymbol{Pt} = (x, y) \in \mathbb{R}^2$ 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** of $\boldsymbol{Pt}$ is defined as:

$$D_a(y, x) = \begin{cases} \arctan\left(\dfrac{y}{x}\right), & \text{if } y > 0 \,\&\, x \geq 0 \\ \arctan\left(\dfrac{y}{x}\right) + \pi, & \text{if } y < 0 \\ \arctan\left(\dfrac{y}{x}\right) + 2\pi, & \text{if } y < 0 \,\&\, x \geq 0 \end{cases} \tag{2}$$

where $\arctan(\cdot)$ denotes the arctangent function.

Figure 2 shows the D_a values of several selected points located on a unit circle. It is clear that the 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. Figure 3 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 seen from Fig. 3 that the angle function has a monotonous nature in a certain region.

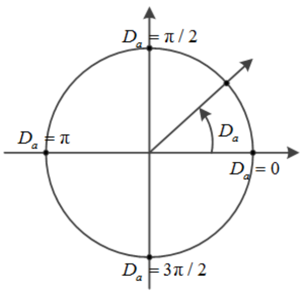


Figure 2. The 2D direction angle of several selected points located on a unit circle.

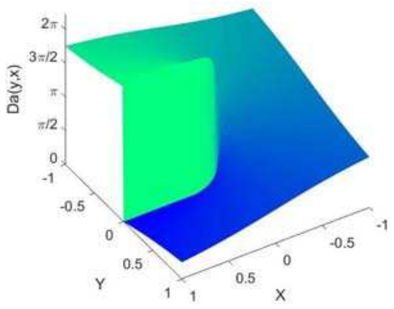


Figure 3. The 2D direction angle of points located on the XY plane. (Figure best seen in color.)

2.1.2 3D direction angle

Let $\boldsymbol{n} = (n_x, n_y, n_z)$ be the normal vector of 3D point $\boldsymbol{Qt} = (x, y, z) \in \mathbb{R}^3$, $\boldsymbol{n}$ uniquely determines the tangent plane of $\boldsymbol{Qt}$. 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 R_Z of normal $\boldsymbol{n}$ on the XY plane as:

$$R_Z = D_a(n_y, n_x), \quad \text{if } {n_x}^2 + {n_y}^2 \neq 0 \tag{3}$$

Similarly, the direction angle of normal $\boldsymbol{n}$ on the YZ plane is defined as R_X:

$$R_X = D_a(n_z, n_y), \quad \text{if } n_z^2 + n_y^2 \neq 0 \tag{4}$$

The direction angle of normal $\boldsymbol{n}$ on the XZ plane is defined as R_Y:

$$R_Y = D_a(n_z, n_x), \quad \text{if } n_z^2 + n_x^2 \neq 0 \tag{5}$$

The three angles R_X, R_Y and R_Z can be used to determine the rotation matrix.

Based on Eqs. (3-5) and the properties of the arctangent function, it can be derived that:

$$\begin{cases} \tan(R_Z) = \tan\left(D_a(n_y, n_x)\right) = \dfrac{n_y}{n_x}, & \text{if } {n_x}^2 + {n_y}^2 \neq 0 \\ \tan(R_X) = \tan\left(D_a(n_z, n_y)\right) = \dfrac{n_z}{n_y}, & \text{if } {n_z}^2 + {n_y}^2 \neq 0 \\ \tan(R_Y) = \tan\left(D_a(n_z, n_x)\right) = \dfrac{n_z}{n_x}, & \text{if } {n_x}^2 + {n_z}^2 \neq 0 \end{cases} \tag{6}$$

If the point $\boldsymbol{Qt}$ is rotated around the Z axis by an angle θ, then the new normal of point $\boldsymbol{Qt}$ is:

$$\boldsymbol{n}' = (n_x \cos\theta - n_y \sin\theta, n_x \sin\theta + n_y \cos\theta, z) \tag{7}$$

Based on Eq. (3), the new direction angle on the XY plane is calculated as:

$$R'_Z = D_a(n_x \sin\theta + n_y \cos\theta, n_x \cos\theta - n_y \sin\theta) \tag{8}$$

Combining Eqs. (6) and (8), it can be derived that:

$$\begin{aligned} \tan R'_Z &= \frac{n_x \sin\theta + n_y \cos\theta}{n_x \cos\theta - n_y \sin\theta} \\ &= \tan(R_Z + \theta) \end{aligned} \tag{9}$$

Similarly, if the point $\boldsymbol{Qt}$ is rotated around the X axis by angle α, the new direction angle on the YZ plane is R'_X that is:

$$\tan R'_X = \tan(R_X + \alpha) \tag{10}$$

If the point $\boldsymbol{Qt}$ is rotated around the Y axis by angle β, the new direction angle on the XZ plane is R'_Y that is:

$$\tan R'_Y = \tan(R_Y + \beta) \tag{11}$$

Considering the periodicity property of direction angles, we have

$$\begin{cases} R'_X = \operatorname{mod}(R_X + \alpha, 2\pi) \\ R'_Y = \operatorname{mod}(R_Y + \beta, 2\pi) \\ R'_Z = \operatorname{mod}(R_Z + \theta, 2\pi) \end{cases} \tag{12}$$

where $\operatorname{mod}(\cdot)$ denote the modulo operation.

We define the difference between angles $b \in [0, 2\pi)$ and $a \in [0, 2\pi)$ as $D(a, b)$:

$$D(a,b) = \begin{cases} a - b, & \text{if } 0 \le b \le a < 2\pi \\ 2\pi - (b - a), & \text{if } 0 \le a < b < 2\pi \end{cases} \tag{13}$$

Combining Eqs. (12) with (13), we have:

$$\begin{cases} \alpha = D\left(R'_X, R_X\right) \\ \beta = D\left(R'_Y, R_Y\right) \\ \theta = D\left(R'_Z, R_Z\right) \end{cases} \tag{14}$$

Consequently, we can obtain the rotation angles α , β , θ based on Eq. (14).

2.1.3 Direction angle histogram based rotation estimation

If we directly use the direction angles to calculate the rotation angles by Eq. (14), we have to know the exact point correspondences for R_X and R'_X , R_Y and R'_Y , R_Z and R'_Z in two point clouds. However, it is not easy to obtain these exact point correspondences in practice. Instead, we can generate the statistical information of R_X , R_Y , and R_Z for each point cloud, and then estimate the rotation using these statistics (e.g., histograms).

Let $\{R_a\}$ be a set of direction angles within the range of $[0, 2\pi)$, the histogram $H(R_a)$ of $\{R_a\}$ is defines as:

$$H(R_a) = \left[H_k^n, R_k^{bin}\right], \quad k = 1, 2, \cdots, n_{bin} \tag{15}$$

where H_k^n is the number of points with direction angles falling into the bin determined by R_k^{bin} . In this paper, the number of bins n_{bin} is set to 3600.

Let $\{R'_a\}$ be the direction angles for the points after a rotation (with a rotation angle of ε), then we have:

$$R'_a = \mathrm{mod}\left(R_a + \varepsilon, 2\pi\right) \tag{16}$$

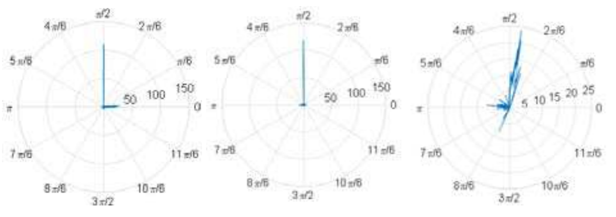
We therefore have:

$$\begin{gathered} H(R'_a) = \left[H'^n_k, R'^{bin}_k\right] \\ R'^{bin}_k = \mathrm{mod}\left(R_k^{bin} + \varepsilon, 2\pi\right) \end{gathered} \tag{17}$$

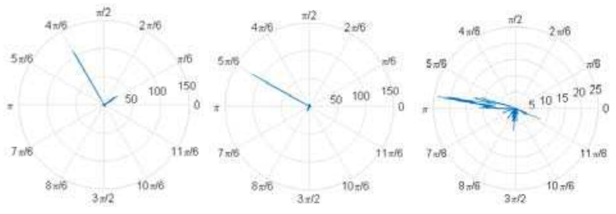
Let $\{R_X\}$, $\{R_Y\}$, $\{R_Z\}$ to be the sets of direction angles of the whole point cloud, then we can obtain the rotation angle based on Eqs. (14) and (17):

$$\begin{cases} \alpha = D\left(R'^{bin}_{Xk}, R^{bin}_{Xk}\right) \\ \beta = D\left(R'^{bin}_{Yk}, R^{bin}_{Yk}\right) \\ \theta = D\left(R'^{bin}_{Zk}, R^{bin}_{Zk}\right) \end{cases} \tag{18}$$

Consequently, we can easily calculate the rotation angles α , β , θ from the direction angle histograms using Eq. (18). Figure 4 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. 4(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. 4(b). It can be seen that the differences between the peaks of the direction angle histograms of R_X , R_Y and R_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. Therefore, the rotation around an axis can be determined by our histogram based algorithm without using any point correspondences.



(a) Three direction angle histograms of the original point cloud.



(b) Three direction angle histograms of the rotated point cloud.
Figure 4. An illustration of the direction angle histograms. (Figure best seen in color.)

Since a 3D rotation consists of three rotations around the X, Y and Z axes, the three rotation angles can therefore be estimated through an iterative approach. The whole process for rotation estimation is summarized in Algorithm 1.

2.2. Correlation based translation estimation

Once the accurate rotation is obtained, we then have to estimate the translation between two point clouds. A simple approach is to translate the centers of the two point clouds to the same position. However, this method only works well for highly overlapped point clouds. In practice, the point clouds acquired from real scenes usually have a small overlap, which makes the centers of the two point clouds significantly different. Figure 5 shows two point clouds with translation only (see Fig. 5(a)) and their projections on three planes (see Figs. 5(b-d)). Figure 5(b) shows the projection of the points with Z values ranging from 0.3m to 1m. Figure 5(c) shows the projection of the points with Y values ranging from -1m to -2m. Figure 5(d) shows the projection of the points with X values ranging from 3m to 9m. From Fig. 5, it can be seen that the translation of the wall and the floor planes can clearly be represented by their projection on the three coordinate planes. Consequently, we can estimate the translation vector using projection.

```
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 $\{H_X^P\}$, $\{H_Y^P\}$, $\{H_Z^P\}$ 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 $\{H_Z^Q\}$ of $\boldsymbol{Q}$.
5:     Compare the positions of the peaks in $\{H_Z^P\}$ and
       $\{H_Z^Q\}$ to obtain the rotation angle $\varphi_z$ and the
       rotation matrix $\mathbf{R}_Z$ around the Z axis;
6:     $\boldsymbol{Q} = \boldsymbol{Q} \cdot \mathbf{R}_Z$, $\mathbf{R}^* = \mathbf{R}^* \cdot \mathbf{R}_Z$;
7:     Calculate the rotation angles $\varphi_x$, $\varphi_y$ around the X,
       Y axes following the similar approach as for $\varphi_z$;
9:     $n_{iter} = n_{iter} + 1$;
10: end while
11: return rotation matrix $\mathbf{R}^*$.
```

In this paper, we use a correlation operation to calculate the translation vector. The main steps are as follows. **First**, the two point clouds are projected onto the three coordinate planes to generate three projection images. **Second**, the correlation operation is used to obtain the plane translation between the two projection images. **Third**, the translations in all planes are averaged to obtain the final 3D translation.

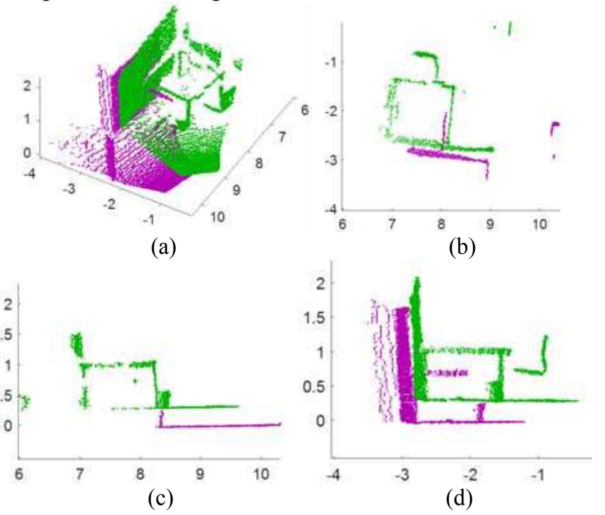


Figure 5. Two point clouds with a translation of (0.1m, 0.2m, 0.3m) and their projections on three coordinate planes. (a) The point clouds. (b) Projection on the XY plane. (c) Projection on the XZ plane. (d) Projection on the YZ plane. (Figure best seen in color.)

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. from 0.3m to 1m) are projected onto the XY plane. The points within that range obviously represent the structure of the scene, as shown in Fig. 5(b). **Second**, the XY plane is partitioned into several bins and the number of projection points in each bin is counted, resulting in a projection image (see Fig. 6 (a)). Similarly, the point cloud are then projected on the YZ and XZ planes to obtain another two projection images, respectively (see Figs. 6 (b-c)).

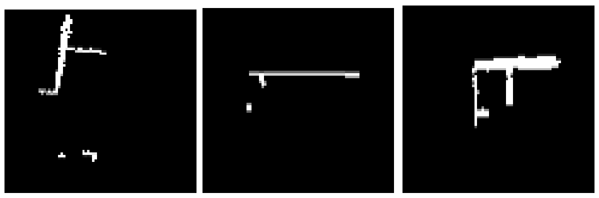


Figure 6. An illustration of point cloud projection. (a) Projection on the XY plane. (b) Projection on the YZ plane. (c) Projection on the XZ plane.

Translation Estimation. The correlation operation is performed on the projection image on the XY plane and the translation in the XY space is determined as $T_{xy} = (C_x, C_y, 0)$, where (C_x, C_y) is the position with the maximum correlation being achieved. Following a similar approach, we can extract the translation T_{yz} in the YZ space and the translation T_{xz} in the XZ space. The overall translation is finally obtained by averaging the three translations:

$$T = \frac{1}{2}(T_{xy} + T_{yz} + T_{zx}) \tag{19}$$

3. Experiments

To test the performance of our algorithm, a set of experiments were conducted. Our algorithm was also compared with several state-of-the-art approaches including FGT-CPD [18] and two ICP variants, i.e., ICP using "point-to-point" metric [12] (denoted by ICP) and ICP using "point-to-plane" metric [14] (denoted by ICPL). The experiments were performed on several indoor datasets. All experiments were conducted using MATLAB 2015b on a PC with 2.5GHz Intel Core i7 CPU and 8G RAM.

3.1. Experiment setup

To test 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 [20]. RRE is calculated calculated as:

$$\begin{gathered} E_R = \sum_{i=1}^{3} \left| angle(i) \right| \\ angle = F\left(R_T^{-1} R_E \right) \end{gathered} \tag{20}$$

where R_T is the ground-truth rotation matrix, R_E is the estimated rotation matrix, and $F(\cdot)$ transforms a rotation matrix to three Euler angles. RRE is actually the sum of the absolute differences in three Euler angles.

RTE is calculated as:

$$E_T = \left\| T_T - T_E \right\|_2 \tag{21}$$

where T_T is the ground-truth translation vector and T_E is the estimated translation vector.

3.2. Robustness to rotation and translation

In this section, we tested the performance of our algorithm with respect to different rigid transformations (including rotation and translation) of point clouds. We selected 100 point clouds with an average overlap of about 60% from the *office* dataset [23] for experiments. The dataset consists of a sequence of RGB and depth images acquired with a handheld Microsoft Kinect sensor. The point clouds generated from the depth images have about 280000 points in average. This dataset also contains the ground-truth camera poses obtained by a KinectFusion system [24]. Figure 7 shows a model of the *office* scene reconstructed by KinectFusion [24].

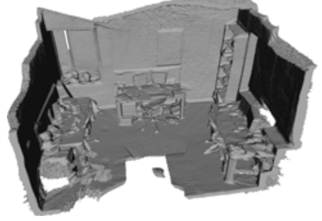


Figure 7. A model of the *office* scene.

For each pair of test point clouds, we apply different rotations and translations to one of the two point clouds. The registration results are shown in Tables 1-2, with the best results shown in bold face.

First, we only apply different levels of rotations to the point clouds. The rotation angles around each axis is set to 30°, 60°, 90°, and 120°, respectively. The translation between the point clouds is set to 0. We set the max number of iterations as $N_{iter} = 3$ in our experiments. The RRE results achieved by different algorithms are shown in Table 1. From Table 1, we can see that the proposed algorithm achieves a very high accuracy on point clouds in the presence of rotation, especially when the rotation is large. The RRE is as small as 1.91° when the rotation between point clouds is 90°. The ICP and ICPL algorithm obtains the similar performance. That is because the scene for test contains a large number of planes with similar features, and the point correspondence cannot be accurately estimated. Meanwhile, when the rotation is large than 60°, the error achieved by FGT-CPD increases dramatically. That is because the structured point cloud cannot be well modeled by GMM. In contrast, the proposed algorithm achieved the best performance. The large number of planes in the scene highly support our rotation estimation based on the direction angle histograms.

Second, we only apply different levels of translations to the point clouds. The translations along each axis is set to 0.1m, 0.2m, 0.3m, and 0.4m, respectively. No rotation is applied to the two point clouds. In our algorithm, we set the bin size to be 0.02m for generating projection images. The RTE results achieved by different algorithms are shown in Table 2. From Table 2, it can be seen that our algorithm achieves almost the same RTE results under different levels of translations and it outperforms other methods on all tests. That is because, the other methods address the translation estimation problem by assuming that the centers of two point clouds are located at the same position. Note that, the ICP used here is the original one without rejecting point pairs with large distance. Therefore the RTE of ICP and FGT-CPD is almost same as the distance between their centers. However, this assumption does not hold for most practical scenarios, where only a small overlapped part can be found between the two point clouds. In our algorithm, the translation is estimated from the projections of planes (i.e. floor and walls), and the translation can be accurately estimated using these projections on the coordinate planes.

	Relative Rotational Error (°)			
Rotation	30°	60°	90°	120°
FGT-CPD [18]	6.97	6.97	45.83	43.73
ICP [12]	8.04	5.53	12.57	17.95
ICPL [14]	8.70	12.12	11.40	10.48
Proposed	**2.81**	**2.82**	**1.91**	**4.94**

Table 1. Registration results on point clouds with rotations only. (The best results are shown in bold.)

	Relative Translation Error (m)			
Translation	0.1m	0.2m	0.3m	0.4m
FGT-CPD [18]	0.471	0.459	0.449	0.440
ICP [12]	0.481	0.465	0.457	0.452
ICPL [14]	0.603	0.593	0.573	0.554
Proposed	**0.043**	**0.042**	**0.043**	**0.043**

Table 2. Registration results of point clouds with translations only. (The best results are shown in bold.)

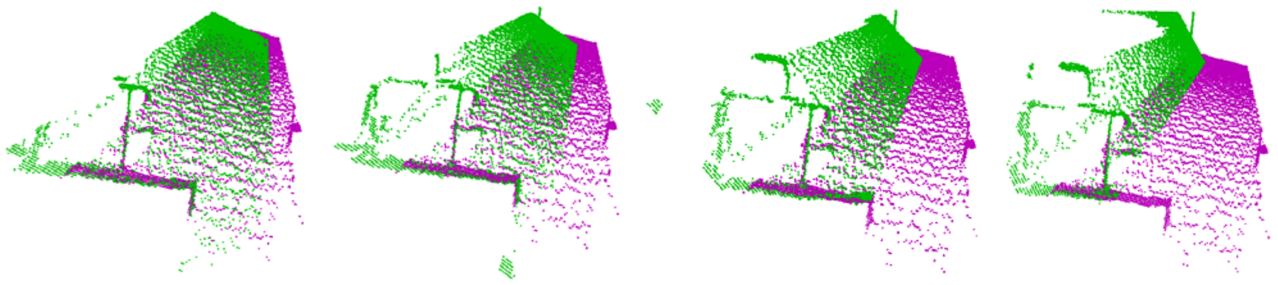
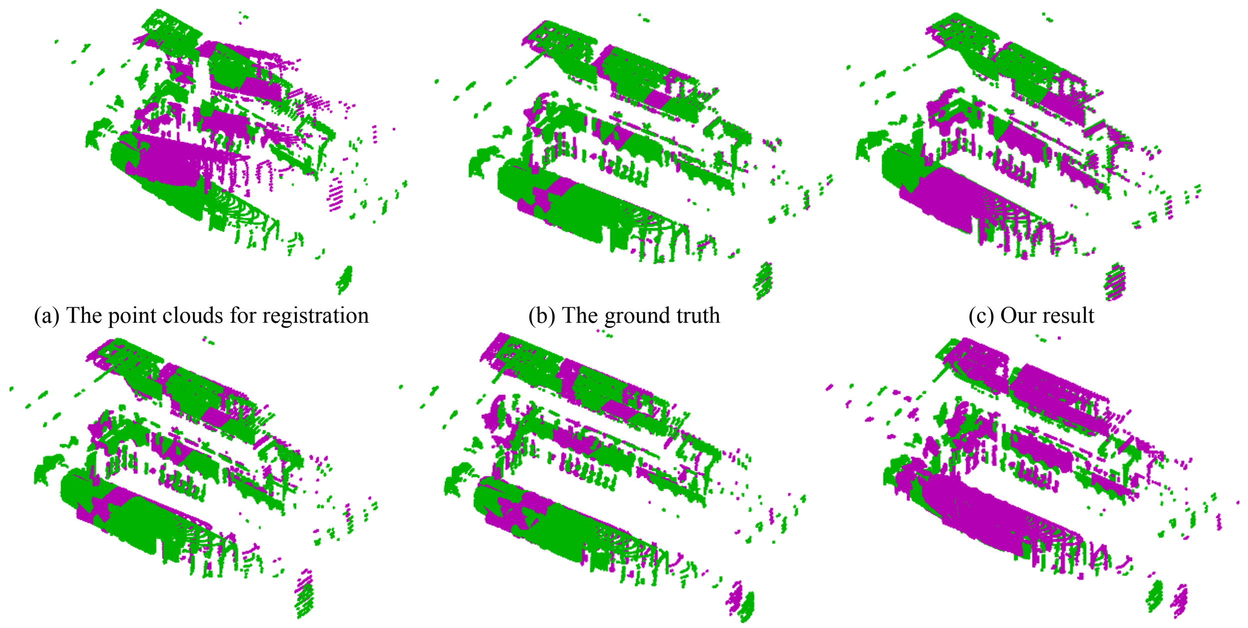

(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4

Figure 8. The point clouds with different degrees of overlaps. (Figure best seen in color.)

	Relative Rotational Error (°)				Relative Translation Error (m)			
Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 1	Scene 2	Scene 3	Scene 4
FGT-CPD [18]	4.76	1.70	5.85	7.32	0.186	0.194	0.219	0.431
ICP [12]	64.76	78.24	88.06	91.19	1.334	1.453	1.509	1.487
ICPL [14]	**1.40**	100.81	137.89	5.89	**0.029**	1.218	1.694	0.321
Proposed	1.64	**1.01**	**4.39**	**1.16**	0.030	**0.0131**	**0.083**	**0.080**

Table 3. Registration results on point clouds with different degrees of overlaps.


(a) The point clouds for registration (b) The ground truth (c) Our result

(e) ICP result (f) ICPL result (g) FGT-CPD result

Figure 9. Registration results on two large-scale point clouds. (Figure best seen in color.)

3.3. Robustness to different overlaps

In this section, we test the registration performance of our algorithm on point clouds with different degrees of overlaps.

Since the point clouds in the *office* dataset have almost the same degree of overlap, we acquired a new dataset in an office using Kinect. The ground-truth camera poses were obtained using the Real-Time Appearance-Based Mapping (RTAB-MAP) method [25]. Several pairs of point clouds with different degrees of overlaps were used for our test, an illustration of the point clouds is shown in Fig. 8. The rotation angles around the X, Y, and Z axes were set to 30°, 30°, 30°, respectively, and the translation was set to (0.1m, 0.1m, 0.1m). The final registration results are shown in Table 3, with the best results shown in bold face.

From Table 3, we can see that the registration performance of all these algorithms decreases as the overlap between point clouds is reduced. The proposed algorithm obtains the best performance on all scenes except Scene 1, it achieves acceptable performance even on point clouds with a very small overlap (i.e. Scene 4). For the most challenging case given by Scene 4, our algorithm achieves a small RRE of 1.16°, which is even better than the performance achieved by ICPL on the easiest case (Scene 1). This clearly demonstrates the robustness of our algorithm with respect to small overlaps.

For the ICP algorithms, insufficient point correspondences can be found between two point clouds when the overlap is small. Similarly, for FGT-CPD, a particular probabilistic model cannot be generated for two point clouds with small overlaps. In contrast, the proposed algorithm fully employs the statistical information of surface normals, it is not necessary to extract a set of corresponding points from the overlapping areas, and the robustness of the algorithm is therefore improved.

3.4. Results on LiDAR point clouds

These methods were further tested on two large-scale LiDAR point clouds given by [26]. These point clouds were acquired from an apartment, with a rotation of (10°, 20°, 30°) and a translation of (0.1m, 0.1m, 0.1m). The registration results are shown in Fig. 9. It can be shown that our algorithm also achieves good registration results on large-scale structured point clouds. It can be seen from Fig. 9 that our algorithm achieves better performance compared to the other methods.

3.5. Computational time

In this section, we calculated the computational time of each algorithm for registering two point clouds from the *office* dataset, the results are shown in Table 4. It can be observed that the proposed algorithm achieved the fastest performance when the number of points is about 280000, followed by FGT-CPD. The proposed algorithm outperforms the other methods by an order of magnitude, it takes only 25.6s for point cloud registration. In contrast, ICP is the slowest algorithm, it takes more than 300s to register two point clouds. The computational burden of FGT-CPD is mainly due to the construction of GMM models. For ICP, the process for determining the corresponding points is very time-consuming. In contrast, our algorithm uses the histograms of surface normals, which can be generated very efficiently.

FGT-CPD[18]	ICP[12]	ICPL [14]	Proposed
184.5 s	313.3 s	391.4 s	**25.6 s**

Table 4. The registration time for different algorithms.

4. Conclusion

This paper has presented a novel algorithm for point cloud registration. The rotation between two point clouds is estimated by iteratively generating histograms of direction angles. The translation between two point clouds is estimated by projecting the rotated point clouds onto three coordinate planes. The proposed algorithm has been tested on three different datasets. Experimental results show that our registration algorithm achieves both high accuracy and efficiency, it is also very robust to small overlaps. It outperforms several existing methods including ICP and FGT-CPD.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under contract No. 61471371.

References

[1] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, *et al.*. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE TVCG,* 19(7): 1199-1217, 2013.

[2] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. *ICCV,* pp. 2039-2047, 2015.

[3] W. Zeng, L. Lui, and X. Gu. Surface registration by optimization in constrained diffeomorphism space. *CVPR*, pp. 4169-4176, 2014.

[4] F. Bogo, J. Romero, M. Loper, and M. Black. FAUST: Dataset and evaluation for 3D mesh registration. *CVPR*, pp. 3794-3801, 2014.

[5] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. *ECCV*, pp. 834-849, 2014.

[6] F. Steinbrücker, J. Sturm, and D. Cremers. Real-time visual odometry from dense RGB-D images. *ICCV Workshops*, pp. 719-722, 2011.

[7] A. I. Comport, M. Meilland, and P. Rives. An asymmetric real-time dense visual localisation and mapping system. *ICCV Workshops*, pp. 700-703, 2011.

[8] F. Steinbrucker, C. Kerl, and D. Cremers. Large-scale multi-resolution surface reconstruction from RGB-D sequences. *ICCV*, pp. 3264-3271, 2013.

[9] C. Kerl, M. Souiai, J. r. Sturm, and D. Cremers. Towards illumination-invariant 3D reconstruction using ToF RGB-D cameras. *3DV*, pp. 39-46, 2014.

[10] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3D local surface description and object recognition. *IJCV,* 105(1): 63-86, 2013.

[11] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. An integrated framework for 3D modeling, object detection and pose estimation from point-clouds. *IEEE TIM,* 64(3): 683-693, 2015.

[12] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE TPAMI,* 14(2): 239-256, 1992.

[13] J. Serafin and G. Grisetti. NICP: Dense normal based point cloud registration. *IROS*, pp. 742-749, 2015.

[14] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *3-D DIM*, pp. 145-152, 2001.

[15] J. Yang, H. Li, and Y. Jia. Go-ICP: Solving 3d registration efficiently and globally optimally. *ICCV*, pp. 1457-1464, 2013.

[16] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing,* 21(13): 1145-1153, 2003.

[17] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart. Tracking a depth camera: Parameter exploration for fast ICP. *IROS*, pp. 3824-3829, 2011.

[18] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE TPAMI,* 32(12): 2262-2275, 2010.

[19] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE TPAMI,* 33(8): 1633-1645, 2011.

[20] M. Lu, J. Zhao, Y. Guo, and Y. Ma. Accelerated coherent point drift for automatic 3D point cloud registration. *IEEE GRSL,* 13(2):162-166, 2016.

[21] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A generative model for the joint registration of multiple point sets. *ECCV*, pp. 109-122, 2014.

[22] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3D local feature descriptors. *IJCV,* 116(1): 66-89, 2016.

[23] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi. Real-time RGB-D camera relocalization. *ISMAR*, pp. 173-179, 2013.

[24] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison*, et al.*. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, pp. 127-13, 20116.

[25] M. Labbe and F. Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. *IROS*, pp. 2661-2666, 2014.

[26] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research,* 31(14): 1705-1711, 2012.