

# Probability Driven Approach for Point Cloud Registration of Indoor Scene

Kun Dong<sup>1</sup> · Shanshan Gao<sup>2</sup> · Shiqing Xin<sup>3</sup> · Yuanfeng Zhou<sup>1</sup>

Received: 202x-xx-xx / Accepted: 202x-xx-xx

**Abstract** Point cloud registration is a crucial step of localization and mapping with mobile robots or in object modeling pipelines. In this paper, we present a novel probability driven algorithm for point cloud registration of the indoor scene based on RGB-D images. Firstly, we extract the key points in RGB-D images and map the key points to 3D space as preprocessing. Then, we build distance matrix and difference matrix for each point cloud, respectively in scalarization and vectorization, to encode the structural proximity. And establish the corresponding point set by computing the matching probabilities. At last, we solve the transform matrix that aligns the source point cloud to the target point cloud. The entire registration framework consists of two phases: coarse registration based on the distance matrix (in scalarization) and fine registration based on the difference matrix (in vectorization). The two-phase registration strategy is able to greatly reduce the influence of inherent noise. Experiments demonstrate that our method outperforms in registration accuracy than the state-of-the-art methods. Furthermore, our method is more efficient than existing methods in computing speed because we utilize the location relationship between key points instead of point features.

**Keywords** Point cloud registration · Probabilistic method · Indoor scene · Distance matrix · Difference matrix

## 1 Introduction

Point cloud registration is the process of consistently aligning two point sets. For this purpose, one has to invent some strategies to infer the transformation matrix that maps the source point cloud to the target point cloud. In Fig. 1, the left shows the situation that two point clouds are not well aligned while the right shows



**Fig. 1** The explanation of point cloud registration. (a) The point clouds before registration. (b) The point clouds after registration

✉ Yuanfeng Zhou  
yfzhou@sdu.edu.cn

Kun Dong  
dongkunnukgnod@163.com

<sup>1</sup> School of Software,  
Shandong University,  
Jinan, China

<sup>2</sup> School of Computer Science and Technology,  
Shandong University of Finance and Economics,  
Jinan, China

<sup>3</sup> School of Computer Science and Technology,  
Shandong University,  
Qingdao, China

the result after registration, which is ready for data fusion. And the right is the result after registration.

In recent years, this research has played a significant role in computer vision [1], augmented reality [9], and robotics [3]. More specifically, the problem has been experienced by many researchers who study indoor scene reconstruction [4], SLAM [5], camera relocalization [6], fusing data from different sensors [7] and other problems [8, 9].

There are numerous methods to solve the point cloud registration problem. Researchers usually adopt the ICP algorithm [10] or ICP variants [11, 12, 14, 15], which are

the most widely used methods for this task because of their high feasibility and clear concepts. They estimate the new transformation according to the closest point correspondences that built by the current transformation, then iterate the steps until the convergence. In addition, 4PCS [16] and super4PCS [17] can also perform well on point clouds with small overlap by using an efficient but practical data structure.

Different from general point cloud models, the point clouds of an indoor scene are usually obtained by color and depth cameras of Kinect with low-resolution. As a consequence, the point clouds are not accurate with a lot of noise and missing parts, which increases the difficulty of point cloud registration. Although the methods mentioned above are known for providing good estimation and are applied in many tasks, they are likely to be influenced by some noise and outliers due to the need of a good initialization or the right subsets. So they usually have poor performance on point cloud registration of the indoor scene.

In this paper, we present a novel algorithm for point cloud registration of the indoor scene based on RGB-D images. We pay attention to building corresponding point set, and our registration approach consists of coarse registration based on the distance matrix (in scalarization) and fine registration based on the difference matrix (in vectorization). Experiments show that our method is efficient and fast for point cloud registration of the indoor scene, by enhancing the accuracy and decreasing the running time greatly.

Our main contributions include:

- We propose a probability driven approach to calculate the matching probabilities of the key points from two specified separate point clouds. Our method is efficient and able to produce accurate registration results.
- We use the distance matrix (in scalarization) and the difference matrix (in vectorization) to respectively guide coarse registration and fine registration.

## 2 Related Work

As a classic and traditional task in computer vision, the problem of point cloud registration has been studied for many years. Researchers have raised many representative algorithms [11, 17, 19, 20] and some advanced methods [21, 22, 30, 31] in recent years. Usually, the typical processes of these algorithms consist of two stages: building corresponding points according to their features; computing and optimizing the transformation between point clouds.

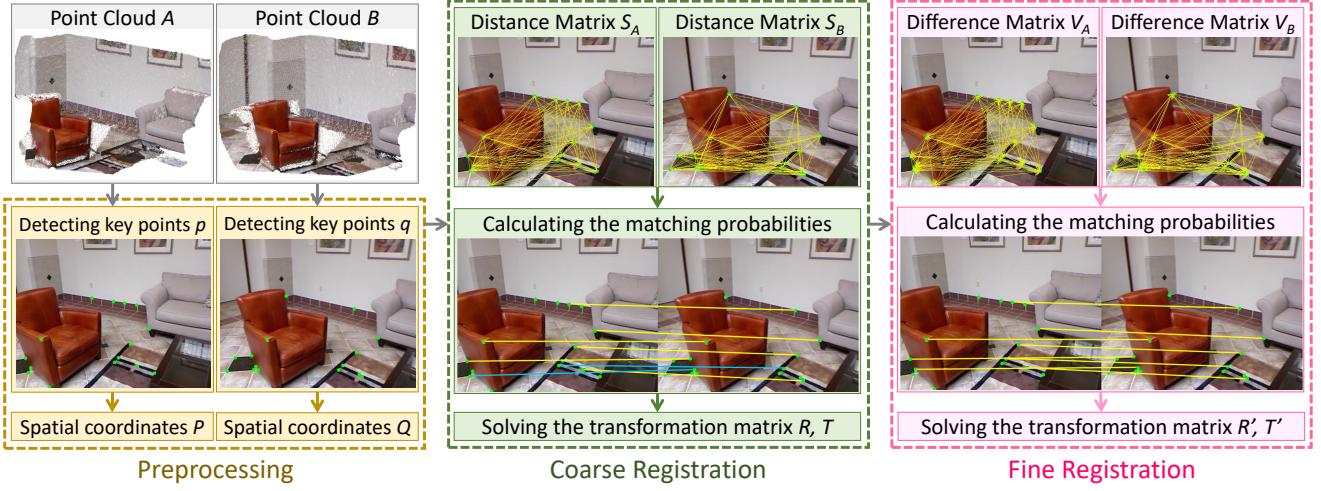
**ICP and its variants.** Based on local iterative optimization, ICP [10] is susceptible to local minima and relies on the quality of the initialization. To overcome the point cloud registration problem with outliers and missing data, some ideas based on the ICP algorithm have been come up with. GO-ICP [11] integrates local ICP into the BnB scheme, which speeds up the new method. Sparse ICP [13] formulates registration optimization by sparsity inducing norms, and achieves superior registration results. But when solving the problems without a good initialization or handling some smooth surfaces [18], these algorithms may easily fail to get optimal estimation.

**4PCS and its improvements.** 4PCS [16] is another classic method state-of-art sampling approach to solve this problem. It works on sampled tuples of four points in a plane, rather than randomly chooses minimal subsets of three points. Later, Mellado [17] improves this algorithm by decreasing the complexity of extraction to linear. However, these approaches may have a slim chance to sort out the right subsets under the situation that high outliers exist.

**Probabilistic registration methods.** Some algorithms, which are known as the correlation based methods [25–27], treat both observation points and model points as probabilistic distributions. Some algorithms [23–26] are also used to create compact Gaussian Mixture Model (GMM) representations. However, the probabilistic point sets registration methods are restricted by their slow processing speed. FilterReg [22] is a state-of-art probabilistic registration method with faster performance, because it transforms this problem to a maximum likelihood estimation and solves using the Expectation Maximization (EM) [28] algorithm.

**RANSAC-related algorithms.** For some point cloud registration applications, randomization strategy is still the main method, which is represented by the famous RANSAC [29]. When the data is noisy and the surfaces only partially overlap, existing pipelines often require many iterations to sample a good correspondence set and find a reasonable alignment. Recently, Huu M. comes up with SDRSAC [30] algorithm built on RANSAC, which make it possible to sample of large subsets but also combine an oracle with their sampling scheme.

Our algorithm is also inspired by the class of methods that aim at image registration problems [32, 33] and two-dimensional matching problems [34, 35], and work well on images. These enable us to find a new approach to build correspondence and apply this kind of ideas on point cloud registration.



**Fig. 2** The framework of the proposed algorithm. Our approach consists of three main parts: Preprocessing, Coarse Registration based on the Distance Matrix, and Fine Registration based on the Difference Matrix

### 3 The Proposed Method

#### 3.1 Overview

In general, the most important step in point cloud registration is to find corresponding points, and the accuracy of the corresponding point set determines the accuracy of the final point cloud registration result to a large extent. So our method focuses on the process of establishing the corresponding point set. Aiming at the common and inaccurate problem of point cloud registration of indoor scene, we combine the coarse registration based on the distance matrix (in scalarization) and the fine registration based on the difference matrix (in vectorization) with our probability driven approach.

The framework of our algorithm is shown in Fig. 2, including the following steps. In the first step, we use the Shi-Tomas method [36] to extract the key points, and map the key points to the three-dimensional space as preprocessing. In the second step, we build distance matrices (in scalarization) for key points, and then according to the distance matrices calculate the probabilities that any two key points are corresponding points, so as to obtain the coarse registration result. In the third step, we further establish difference matrices (in vectorization) according to the coarse registration result, and recalculate the corresponding probabilities between key points to obtain the fine registration result.

#### 3.2 Preprocessing

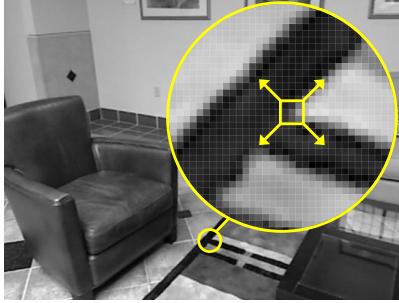
The preprocessing is to extract key points and map them to three-dimensional space. Many methods calculate the features and find corresponding points for all

points in the point cloud, which take a lot of time, and are easy to find the wrong corresponding point. We first extract some key points in each image, then build corresponding point set based on these key points, instead of searching all points. So our method can greatly reduce computational complexity, and improve the accuracy of finding corresponding points.

For point clouds of an indoor scene, they are usually obtained from depth images and color images. We obtain the depth information by depth cameras, and get the color information by color cameras. We use the Shi-Tomas method [36] to extract corner points of color image as key points. For the same part of the scene area, even if the images are obtained at different angles, the same key points still can be obtained. In other words, for a set of key points in the overlapping area, their corresponding points will be in another set of key points, so we don't need to search for other points in the point cloud during the process of finding corresponding points.

##### (1) Detecting key points.

We first gray the color images and detect corners on these grayscale images by the Shi-Tomas algorithm [36]. This method makes a fixed-size window (the window size is the neighborhood of a pixel and is  $3 \times 3$  by default) do a slight sliding in any direction in the image, and calculates whether the gray value of the window area before and after the move changes significantly. Take Fig. 3 as an example, for the neighborhood range drawn in the figure, if the window is slided in any direction, the gray value in the window changes greatly, we consider that there's a corner in the area where the window is located.



**Fig. 3** Detecting key points. Slide the yellow square window in any direction, and the gray value in the window will change greatly



**Fig. 4** Extracting key points. The green dots represent the position of the key points in two-dimensional

Through calculation, we extract  $n$  corner points in a grayscale image as key points, as shown in Fig. 4 (key points are marked in green on the color picture).

Suppose we register the two point clouds  $A$  and  $B$ , and we gray their color images to obtain the grayscale images  $I_A$  and  $I_B$ . The key points set extracted on  $I_A$  is  $\mathbf{p} = \{p_1, p_2, \dots, p_n\}$ , where  $p_i(i = 1, 2, \dots, n)$  represents the two-dimensional coordinates of the  $i_{th}$  key point on  $I_A$ . And the key points set extracted on  $I_B$  is  $\mathbf{q} = \{q_1, q_2, \dots, q_n\}$ , where  $q_j(j = 1, 2, \dots, n)$  means the two-dimensional coordinates of the  $j_{th}$  key point on  $I_B$ .

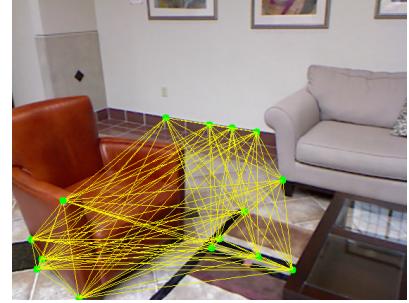
## (2) Calculating spatial coordinates of key points.

According to the two-dimensional coordinates of the key points, we need to calculate their coordinates in three-dimensional space. The conversion from the depth value  $d$  of one pixel in two-dimension to the coordinate in three-dimensional space can be calculated by the following formula [21] :

$$P(u, v, d) = \left( \frac{(u - c_x) \cdot d}{f_x}, \frac{(v - c_y) \cdot d}{f_y}, d \right)^\top, \quad (1)$$

where  $(u, v)$  is the coordinate of the pixel in two-dimension,  $f_x$  and  $f_y$  are the known focal lengths,  $(c_x, c_y)$  is the known principal point of the camera.

After calculation, we obtain the spatial coordinates of the key points, which are  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  and



**Fig. 5** Visual representation of distance matrix. The yellow lines connect any two key points

$\mathbf{Q} = \{Q_1, Q_2, \dots, Q_n\}$ , where  $P_i(i = 1, 2, \dots, n)$  represents the spatial coordinates of the  $i_{th}$  key point on  $I_A$ , and  $Q_j(j = 1, 2, \dots, n)$  represents the spatial coordinates of the  $j_{th}$  key point on  $I_B$ .

## 3.3 Coarse Registration based on Distance Matrix

In this section, we calculate the matching probability between any two key points according to the spatial distances between the key points. In order to represent the spatial distance relationship between key points clearly, we first construct distance matrix (in scalarization) for each set of key points.

### (1) Constructing distance matrix.

For each set of key points, we calculate the Euclidean distances from any key point to other key points to represent their spatial relationship. So the distance Matrix  $\mathbf{S}_A$  calculated from  $\mathbf{P}$  can be specified as:

$$\mathbf{S}_A = \begin{bmatrix} 0 & |P_1 - P_2| & \cdots & |P_1 - P_n| \\ |P_2 - P_1| & 0 & \cdots & |P_2 - P_n| \\ \vdots & \vdots & \ddots & \vdots \\ |P_n - P_1| & |P_n - P_2| & \cdots & 0 \end{bmatrix}, \quad (2)$$

where  $|P_i - P_j|$  is the spatial Euclidean distance between the  $i_{th}$  key point and the  $j_{th}$  key point.

Also, the distance matrix  $\mathbf{S}_B$  calculated from  $\mathbf{Q}$  can be specified as:

$$\mathbf{S}_B = \begin{bmatrix} 0 & |Q_1 - Q_2| & \cdots & |Q_1 - Q_n| \\ |Q_2 - Q_1| & 0 & \cdots & |Q_2 - Q_n| \\ \vdots & \vdots & \ddots & \vdots \\ |Q_n - Q_1| & |Q_n - Q_2| & \cdots & 0 \end{bmatrix}. \quad (3)$$

In other words, we can consider  $n$  key points as  $n$  vertices, and connect any two vertices to form an undirected complete graph (as shown in Fig. 5) to express the distance matrix, where the weight of each edge is the spatial Euclidean distance of the two connected key points.

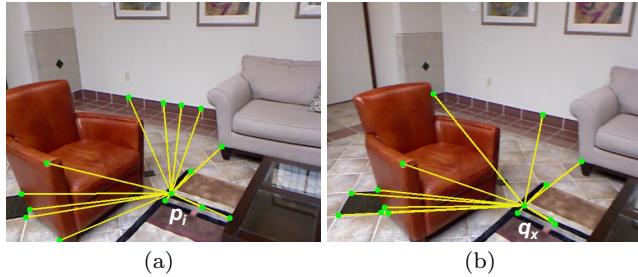
(2) Calculating matching probability based on distance matrix.

Taking one key point  $p_i$  in  $A$  and one key point  $q_x$  ( $x = 1, 2, \dots, n$ ) in  $B$ , we calculate the probability  $Pr\{i\_x\}$  that they are corresponding points, where  $i$  and  $x$  are the serial numbers of key points in  $\mathbf{p}$  and  $\mathbf{q}$  respectively. The higher the probability, the more likely they are corresponding points.

Based on the constructed distance matrices  $\mathbf{S}_A$  and  $\mathbf{S}_B$ , we compare the  $i_{th}$  row of  $\mathbf{S}_A$  and the  $x_{th}$  row of  $\mathbf{S}_B$ , and calculate the number of equal non-zero distance values as  $n_e$  (within the error range  $\epsilon$ ). Then we define the probability that  $p_i$  and  $q_x$  are corresponding points as:

$$Pr\{i\_x\} = \frac{n_e}{n}. \quad (4)$$

In other words, we can connect  $p_i$  to other points in  $p$  to form  $n - 1$  edges, and connect  $q_x$  to other points in  $q$  to form  $n - 1$  edges. As shown in Fig. 6, the lines in (a) represent the edges connected to  $p_i$ , and the lines in (b) represent the edges connected to  $q_x$ . Then we calculate the number of edges with equal weight (if the error of the weights of the two edges is less than  $\epsilon$ , they are considered equal).



**Fig. 6** The edges connected to one key point. The center point in (a) is  $p_i$ , and the center point in (b) is  $q_x$

After calculating all  $Pr\{i\_x\}$  ( $i = 1, 2, \dots, n$ ), we define the maximum corresponding probability of  $p_i$  to  $q_x$  as:

$$Pr\{i\_j\} = \max Pr\{i\_x\} \quad (x = 1, \dots, n), \quad (5)$$

where  $j$  means the key point  $q_j$  in  $q$ .

If  $Pr\{i\_j\} > \frac{1}{2}$ , we consider that the corresponding point of  $p_i$  is  $q_j$ .

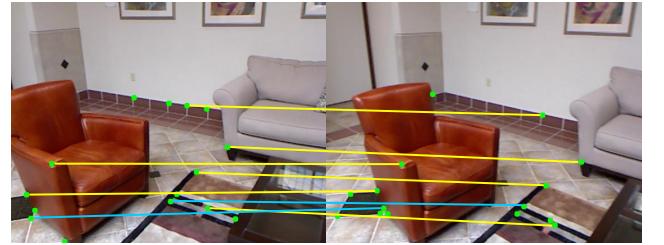
There are two things to explain: on one hand, we allow such small error in the corresponding probability calculation, because point clouds are generally captured by Kinect, whose low resolution and noise make the point clouds have some errors. On the other hand, if points  $p_i$  and  $q_j$  are corresponding points, then most of the edges connected to them are consistent, and the calculated probability will be larger.

After calculation, we find the corresponding points for most points in  $\mathbf{p}$  (the corresponding relationship is shown in Fig. 7), although there are still a few points that have no corresponding points and some wrong correspondences. And we establish new point sets  $\mathbf{p}'$  and  $\mathbf{q}'$  to record the corresponding relationship. We define:

$$\mathbf{p}' = \{p'_1, p'_2, \dots, p'_{n_1}\}, \quad (6)$$

$$\mathbf{q}' = \{q'_1, q'_2, \dots, q'_{n_1}\}, \quad (7)$$

where  $n_1$  is the number of key points in  $\mathbf{p}$  that has corresponding points,  $p'_i$  ( $i = 1, 2, \dots, n_1$ ) represents the  $i_{th}$  key point,  $q'_i$  ( $i = 1, 2, \dots, n_1$ ) represents the corresponding point of  $p'_i$ .



**Fig. 7** Correspondences after calculating matching probability by distance matrix. The yellow lines indicate the correct correspondences, while the blue lines indicate the wrong correspondences

### (3) Solving the transformation matrix.

Based on the key point sets  $\mathbf{p}'$  and  $\mathbf{q}'$ , we can establish their corresponding spatial coordinate sets  $\mathbf{P}'$  and  $\mathbf{Q}'$ . We use the Singular Value Decomposition method [38] to calculate the rotation matrix  $\mathbf{R}$  and the translation matrix  $\mathbf{T}$ , so as to convert  $\mathbf{P}'$  to  $\mathbf{Q}'$  and align them. Suppose the relationship between  $P'_i$  and  $Q'_i$  ( $i = 1, 2, \dots, n_1$ ) is:

$$Q_i = \mathbf{R}P_i + \mathbf{T}. \quad (8)$$

First, we normalize  $\mathbf{P}'$  and  $\mathbf{Q}'$ , take the centroid of each set of points as the new coordinate origin, and recalculate the coordinates of points:

$$P'_i = P'_i - \overline{P'}, \quad (i = 1, 2, \dots, n_1), \quad (9)$$

$$Q'_j = Q'_j - \overline{Q'}, \quad (j = 1, 2, \dots, n_1), \quad (10)$$

where  $\overline{P'}$  and  $\overline{Q'}$  are the centers of mass of  $\mathbf{P}'$  and  $\mathbf{Q}'$  respectively.

To facilitate singular value decomposition, we need to construct the standard orthogonal transformation matrix:

$$\mathbf{H} = P'^\top Q'. \quad (11)$$

Then, we decompose the matrix  $\mathbf{H}$  into three matrices according to Singular Value Decomposition [38]:

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^*, \quad (12)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal,  $\Sigma$  is nonnegative and diagonal.

Finally, the transformation matrix can be obtained by the following formula:

$$\mathbf{R} = \mathbf{V}\mathbf{U}^\top, \quad (13)$$

$$\mathbf{T} = \overline{\mathbf{Q}'} - \mathbf{R}\overline{\mathbf{P}'}. \quad (14)$$

The registration method based on the distance matrix is summarized in Algorithm 1. Based on the rotation matrix  $\mathbf{R}$  and translation matrix  $\mathbf{T}$ , we transform the point cloud  $A$  to  $A'$ , and align it with  $B$  to achieve the coarse registration based on the distance matrix.

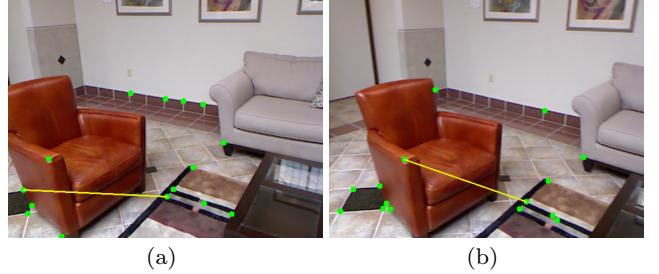
#### Algorithm 1 Registration Based on Distance Matrix

```

Require: Key points  $\mathbf{p}$  and their spatial coordinates  $\mathbf{P}$ ;
          Key points  $\mathbf{q}$  and their spatial coordinates  $\mathbf{Q}$ .
Ensure: Transformation  $\mathbf{R}, \mathbf{T}$  that aligns  $A$  to  $B$ .
1: function REGISTRATION BASED ON DISTANCE MATRIX( $\mathbf{p}$ ,
    $\mathbf{q}, \mathbf{P}, \mathbf{Q}$ )
2:    $S_A \leftarrow$  the distance matrix of  $\mathbf{p}$ ;
3:    $S_B \leftarrow$  the distance matrix of  $\mathbf{q}$ ;
4:   for each  $p_i \in \mathbf{p}$  do
5:     for each  $q_j \in \mathbf{q}$  do
6:        $Pr\{i,j\} \leftarrow$  the matching probability of  $p_i, q_j$ ;
7:       if  $Pr\{i,j\} > \frac{1}{2}$  then
8:          $q_j$  is the corresponding point of  $p_i$ ;
9:          $p' += p_i$ ;
10:         $q' += q_j$ ;
11:      end if
12:    end for
13:  end for
14:   $\mathbf{P}' \leftarrow$  the spatial coordinates of  $\mathbf{p}'$ ;
15:   $\mathbf{Q}' \leftarrow$  the spatial coordinates of  $\mathbf{q}'$ ;
16:   $\mathbf{R}, \mathbf{T} \leftarrow SVD(\mathbf{P}', \mathbf{Q}');$ 
17: end function
```

#### 3.4 Fine Registration based on Difference Matrix

However, our registration method based on the distance matrix (in scalarization) has a problem that causes inaccurate results and cannot be avoided: when only using spatial Euclidean distance as the measurement condition for key points matching, the distances of two pairs of unrelated key points may be equal. As shown in Fig. 8, there is no correspondence between two key points in (a) and two key points in (b), but the connection in (a) have the same distance with the connection in (b). For a few key points, this phenomenon can cause wrong corresponding judgment, which makes the registration result inaccurate.



**Fig. 8** Two pair of unrelated key points but with same distance. (a) connects two key points in  $\mathbf{p}$ , (b) connects two key points in  $\mathbf{q}$

Despite the errors, the two point clouds can be roughly aligned by coarse registration based on the distance matrix, and a good initial registration result is obtained. On this basis, we construct vectors between two sets of key points respectively. Because vectors can represent directions, the vectors of edges in Fig. 8 are different, which can solve this problem. Therefore, we propose a registration method based on the difference matrix (in vectorization).

The reason for not using the coarse registration based on the difference matrix directly is that before the registration, two point clouds are in different coordinate systems, so even the vectors of the same edge are different, and registration based on the difference matrix cannot be performed.

#### (1) Constructing difference matrix.

By coarse registration method based on the distance matrix, we can work out the new spatial coordinates of the key points  $\mathbf{P}' = \{P'_1, P'_2, \dots, P'_n\}$  and  $\mathbf{Q}' = \{Q'_1, Q'_2, \dots, Q'_n\}$ , where  $\mathbf{P}'$  is the coordinates of  $\mathbf{p}$  in the three-dimensional space of point cloud  $A'$ , and  $\mathbf{Q}'$  is the coordinates of  $\mathbf{q}$  in the three-dimensional space of point cloud  $B$ .

Imitating the construction method of distance matrix, we can calculate the spatial coordinate difference (in vectorization) of any key point to other key points to represent their spatial relationship. Therefore, the difference matrix  $\mathbf{V}_A$  calculated from  $\mathbf{P}'$  can be specified as a difference matrix:

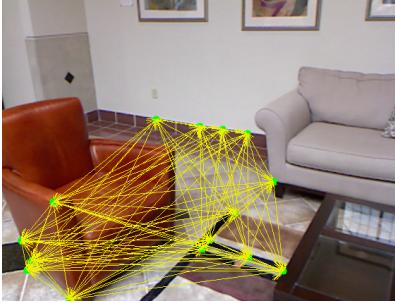
$$\mathbf{V}_A = \begin{bmatrix} \mathbf{0} & \overrightarrow{P'_1 - P'_2} & \cdots & \overrightarrow{P'_1 - P'_n} \\ \overrightarrow{P'_2 - P'_1} & \mathbf{0} & \cdots & \overrightarrow{P'_2 - P'_n} \\ \vdots & \vdots & \ddots & \vdots \\ \overrightarrow{P'_n - P'_1} & \overrightarrow{P'_n - P'_2} & \cdots & \mathbf{0} \end{bmatrix}, \quad (15)$$

where  $\overrightarrow{P'_i - P'_j}$  represents a vector from the  $i_{th}$  key point to the  $j_{th}$  key point.

The difference matrix  $\mathbf{V}_B$  calculated from  $\mathbf{Q}'$  can also be specified as a difference matrix:

$$\mathbf{V}_B = \begin{bmatrix} \mathbf{0} & \overrightarrow{Q'_1 - Q'_2} & \cdots & \overrightarrow{Q'_1 - Q'_n} \\ \overrightarrow{Q'_2 - Q'_1} & \mathbf{0} & \cdots & \overrightarrow{Q'_2 - Q'_n} \\ \vdots & \vdots & \ddots & \vdots \\ \overrightarrow{Q'_n - Q'_1} & \overrightarrow{Q'_n - Q'_2} & \cdots & \mathbf{0} \end{bmatrix}. \quad (16)$$

In other words, we can regard  $n$  key points as  $n$  vertices. Then start with any vertex and end with other  $n - 1$  vertices to make a directed edge. Thus form a directed complete graph (as shown in Fig. 9), in which the weight of each edge is a vector from the starting point to the endpoint.



**Fig. 9** Visual representation of difference matrix. The yellow arrows are the vectors among the key points

### (2) Calculating matching probability based on difference matrix.

Analogous to the registration based on the distance matrix, we calculate probability  $Pr'\{i\_x\}$  that the key points  $p_i$  and  $q_x$  ( $x = 1, 2, \dots, n$ ) are corresponding.

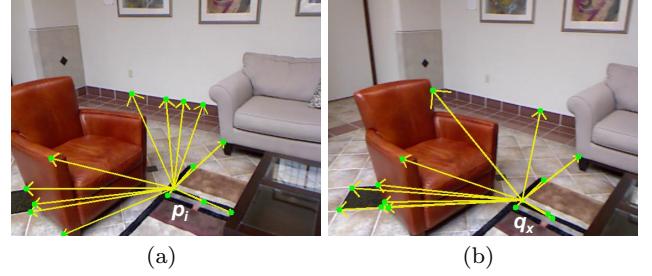
Based on the constructed difference matrices  $\mathbf{V}_A$  and  $\mathbf{V}_B$ , we compare the  $i_{th}$  row of  $\mathbf{V}_A$  and the  $x_{th}$  row of  $\mathbf{V}_B$ , and calculate the number of equal non-zero vectors as  $n'_e$  (within the error range  $\epsilon$ ). What should be noted is that vectors differ from distances because they are three-dimensional. We require two vectors to be equal in three dimensions, then they are called equal. So the probability that  $p_i$  and  $q_x$  are corresponding points is:

$$Pr'\{i\_x\} = \frac{n'_e}{n}. \quad (17)$$

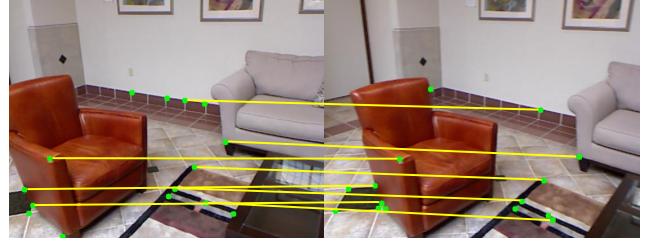
As shown in Fig. 10, the arrows in (a) represent the vectors starting from  $p_i$ , and the arrows in (b) represent the vectors starting from  $q_x$ . Then we calculate the number of equal vectors (two vectors are considered equal if their error is less than  $\epsilon$  in all three dimensions).

Similarly, after calculating all matching probabilities  $Pr'\{i\_x\}$  ( $i = 1, 2, \dots, n$ ), we define:

$$Pr'\{i\_j\} = \max Pr'\{i\_x\} \quad (x = 1, \dots, n). \quad (18)$$



**Fig. 10** The vectors starting from one key point. The center point in (a) is  $p_i$ , the center point in (b) is  $q_x$



**Fig. 11** Correspondences after calculating matching probability by difference matrices

If  $Pr'\{i\_j\} > \frac{1}{2}$ , we consider that the corresponding point of  $p_i$  is  $q_j$ .

After calculation, we find the corresponding points for most points in  $\mathbf{p}$  (the corresponding relationship is shown in Fig. 11). Then we establish a new point set  $\mathbf{p}''$  and  $\mathbf{q}''$  to record the corresponding points, and define:

$$\mathbf{p}'' = \{p''_1, p''_2, \dots, p''_{n_2}\}, \quad (19)$$

$$\mathbf{q}'' = \{q''_1, q''_2, \dots, q''_{n_2}\}, \quad (20)$$

where  $n_2$  is the number of key points in  $\mathbf{p}'$  that has corresponding points,  $p''_i$  ( $i = 1, 2, \dots, n_2$ ) represents the  $i_{th}$  key point,  $q''_i$  ( $i = 1, 2, \dots, n_2$ ) represents the corresponding point of  $p''_i$ .

### (3) Solving the transformation matrix.

Similar to the registration based on distance matrix, on the basis of the key point sets  $\mathbf{p}''$  and  $\mathbf{q}''$ , we establish their corresponding spatial coordinate sets  $\mathbf{P}''$  and  $\mathbf{Q}''$  in  $A$  and  $B$ . Then we also use the singular value decomposition method to calculate the rotation matrix  $\mathbf{R}'$  and translation matrix  $\mathbf{T}'$ . The registration method based on the difference matrix is summarized in Algorithm 2. Then the point cloud  $A$  can be transformed and further aligned with  $B$ , thereby achieving our fine registration based on difference matrix.

## 4 Experimental Results

### 4.1 Experimental Details

#### (1) Datasets.

**Algorithm 2** Registration Based on Difference Matrix

---

**Require:** Key points  $\mathbf{p}$  and their new spatial coordinates  $\mathbf{P}'$ ;  
     Key points  $\mathbf{q}$  and their spatial coordinates  $\mathbf{Q}'$ .  
**Ensure:** Transformation  $\mathbf{R}', \mathbf{T}'$  that aligns  $A$  to  $B$ .

```

1: function REGISTRATION BASED ON DIFFERENCE MATRIX( $\mathbf{p}, \mathbf{q}, \mathbf{P}', \mathbf{Q}'$ )
2:    $\mathbf{V}_A \leftarrow$  the difference matrix of  $\mathbf{p}$ ;
3:    $\mathbf{V}_B \leftarrow$  the difference matrix of  $\mathbf{q}$ ;
4:   for each  $p_i \in \mathbf{p}$  do
5:     for each  $q_j \in \mathbf{q}$  do
6:        $Pr'\{i,j\} \leftarrow$  the matching probability of  $p_i, q_j$ ;
7:       if  $Pr'\{i,j\} > \frac{1}{2}$  then
8:          $q_j$  is the corresponding point of  $p_i$ ;
9:          $\mathbf{p}'' \leftarrow \mathbf{p}'' + p_i$ ;
10:         $\mathbf{q}'' \leftarrow \mathbf{q}'' + q_j$ ;
11:      end if
12:    end for
13:  end for
14:   $\mathbf{P}'' \leftarrow$  the spatial coordinates of  $\mathbf{p}''$ ;
15:   $\mathbf{Q}'' \leftarrow$  the spatial coordinates of  $\mathbf{q}''$ ;
16:   $\mathbf{R}', \mathbf{T}' \leftarrow SVD(\mathbf{P}'', \mathbf{Q}'')$ ;
17: end function
```

---

To verify our experimental results, we conduct experiments on NYU-Depth V2 dataset [40], SUN3D dataset [41], and ICL-NUIM datasets [39]. The NYU dataset has labeled pairs of RGB and depth images from a variety of indoor scenes recorded by the Microsoft Kinect. The SUN3D dataset is a large-scale RGB-D video database for reconstruction captured by ASUS Xtion PRO LIVE sensor. The ICL-NUIM dataset aims at benchmarking RGB-D algorithms, providing two different scenes (the living room and the office room scene). Both NYU and sun3D datasets have some noise, which increases the difficulty of point cloud registration.

## (2) Evaluation criteria.

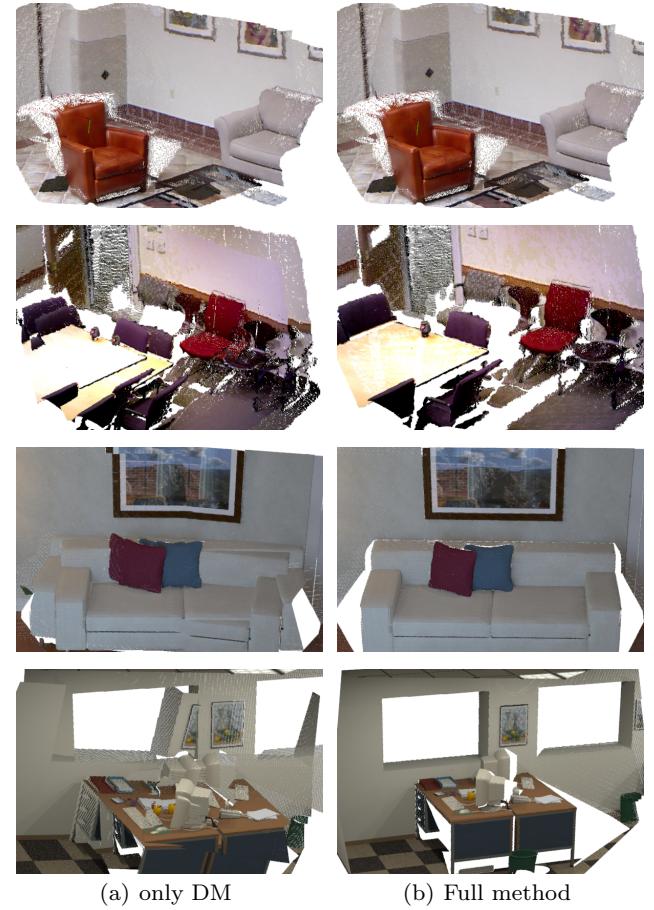
During the evaluation step of point cloud registration, the Largest Common Pointset (LCP) method is commonly used to calculate the overlap ratio of two point clouds. Suppose we transform point cloud  $A$  to  $A'$  and align it to  $B$ . Then for each point  $B_x$  in  $B$ , we find its closest point  $A'_x$  in  $A'$ . If the distance between  $B_x$  and  $A'_x$  is less than a tolerance range, we consider them to be a pair of overlap points.

Then we calculate the overlap ratio  $OR$  of the two point clouds:

$$OR = \frac{2N_c}{N_A + N_B}, \quad (21)$$

where  $N_c$  is the number of overlap point pairs,  $N_A$  and  $N_B$  are the number of points of  $A$  and  $B$  respectively.

In addition, we calculate the Root Mean Square Error (RMSE) to represent the error between two point clouds:



**Fig. 12** Visual performance of our registration results. (a) is registration results based on Distance Matrix (DM) only, (b) is our entire method

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_c} d_i^2}{N_c}}, \quad (22)$$

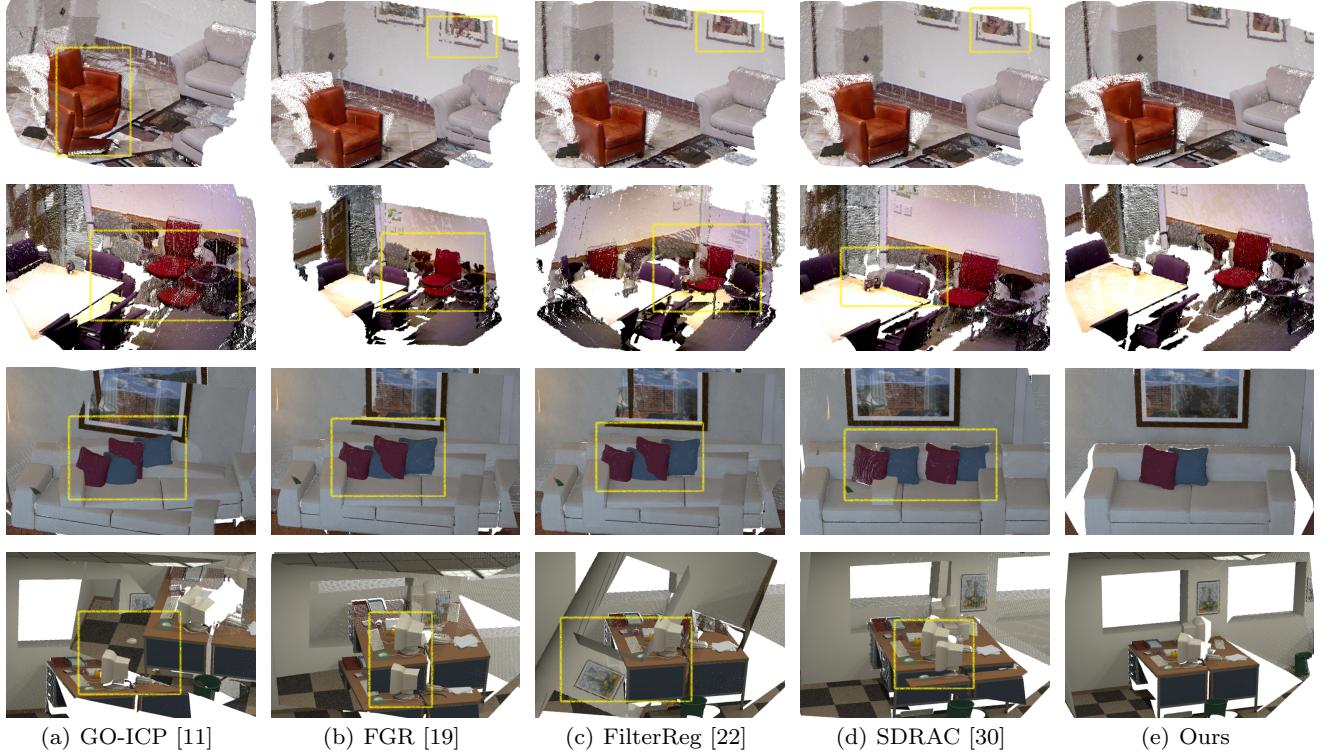
where  $d_i$  represents the distance between the  $i_{th}$  pair of overlap points.

## 4.2 Validity Test

To test the effectiveness of the two registration methods (coarse registration based on the distance matrix and fine registration based on the difference matrix), we test the evaluation criteria after removing each registration and compare the results.

It should be noted that when we only use the registration based on difference matrix, we cannot find any corresponding points that meet the conditions, so we cannot compute the transformation matrix.

As Fig. 12, (a) shows the results on different datasets of registration based on Distance Matrix (DM) only, and (b) shows our results without removing any step. And the first row is the results of the NYU dataset,



**Fig. 13** Comparison with other algorithms visually. The first row is results on NYU, the second row is results on SUN3D, the third row is results on ICL-NUIM (Living Room), and the fourth row is results on ICL-NUIM (Office Room). (a) is the results of GO-ICP, (b) is the results of FGR, (c) is the results of FilterReg, (d) is the results of SDRAC, (e) is the results of our method

Dataset	Evaluation Criteria	only DM	Full method
NYU	<i>OR</i>	75.59%	77.16%
	<i>RMSE</i>	0.0213	0.0214
SUN3D	<i>OR</i>	14.53%	87.32%
	<i>RMSE</i>	16.8029	10.1137
ICL-NUIM (Living Room)	<i>OR</i>	20.49%	86.06%
	<i>RMSE</i>	16.4409	6.3063
ICL-NUIM (Office Room)	<i>OR</i>	6.75%	69.21%
	<i>RMSE</i>	14.7743	10.7876

**Table 1** Numerical performance of our registration results. *OR* is their overlap ratios, and *RMSE* is their errors

the second row is the results of SUN3D, the third and fourth row are the results of the living room and office room of ICL-NUIM respectively. Table 1 displays the overlap ratios of point clouds and average errors in the above situations.

By observing and analyzing the results, we can find that: first of all, the best results cannot be obtained by registration based on the distance matrix or registration based on the difference matrix only, even registration based on the difference matrix cannot work directly. Only by combining the two registration meth-

ods can we get the best alignment results. In addition, we can get good results on a small number of datasets by registration based on the distance matrix only, but it is not always effective alone.

### 4.3 Experimental Comparison

To evaluate the performance of our proposed algorithm, we conduct experiments on different datasets and compare our approach against several state-of-art algorithms that can be used to calculate point cloud registration effectively, including GO-ICP [11], FGR [19], SDRAC [30] and FilterReg [22].

All experiments are executed on a standard Windows system with 16GB of RAM. SDRSAC is carried out using MATLAB, FilterReg, GO-ICP, FGR, and our algorithms are accomplished with the released C++ code.

The experimental results are shown in Fig. 13, comparing with several algorithms on different datasets. We can find that for those continuous and complete datasets (such as NYU in the first row), most methods can get good results. But when we focus on the sofa or the picture frames on the wall, we can realize that their registration results are not perfect. The edges

Dataset	Evaluation Criteria	GO-ICP [11]	FGR [19]	FilterReg [22]	SDRAC [30]	ours
NYU	<i>OR</i>	11.94%	72.31%	80.76%	<b>82.97%</b>	77.16%
	<i>RMSE</i>	0.0305	0.0245	0.0223	<b>0.0193</b>	0.0214
	Preprocessing Time	18.290s	36.271s	4.922s	none	<b>0.171s</b>
	Registration Time	0.094s	29.907s	2.631s	848.463s	<b>0.001s</b>
	Total Time	18.384s	76.178s	7.553s	848.463s	<b>0.172s</b>
SUN3D	<i>OR</i>	5.67%	31.67%	2.17%	42.02%	<b>87.32%</b>
	<i>RMSE</i>	15.4152	14.7856	15.3656	13.0682	<b>10.1137</b>
	Preprocessing Time	18.648s	39.205s	5.392s	none	<b>0.184s</b>
	Registration Time	0.074s	111.307s	2.768s	575.860s	<b>0.001s</b>
	Total Time	18.722s	150.512s	8.160s	575.860s	<b>0.185s</b>
ICL-NUIM (Living Room)	<i>OR</i>	9.90%	10.17%	11.04%	2.47%	<b>86.06%</b>
	<i>RMSE</i>	14.5945	14.7015	14.5660	15.2368	<b>6.3063</b>
	Preprocessing Time	18.911s	32.509s	6.192s	none	<b>0.195s</b>
	Registration Time	0.171s	198.868s	3.210s	487.763s	<b>0.001s</b>
	Total Time	19.082s	231.377s	9.402s	487.763s	<b>0.196s</b>
ICL-NUIM (Office Room)	<i>OR</i>	1.86%	3.22%	1.02%	2.68%	<b>69.21%</b>
	<i>RMSE</i>	14.8338	15.2293	15.1035	15.1535	<b>10.7876</b>
	Preprocessing Time	18.846s	31.478s	5.201s	none	<b>0.180s</b>
	Registration Time	0.078s	117.947s	2.924s	835.364s	<b>0.001s</b>
	Total Time	18.924s	149.425s	8.129s	835.364s	<b>0.181s</b>

**Table 2** Comparison with other algorithms numerically. The first row is the names of the datasets. The second row is the adopted evaluation criteria, including *OR*, *RMSE*, preprocessing time, registration time, and total time. The other rows are different methods. And the numbers in bold are the best results

of the frame are not fully aligned in their results, but can be nicely matched in our results. For those datasets with many noisy points and some missing parts (such as SUN3D in the second row), their algorithms perform poorly that can be seen from the table and chair. For those datasets with very few features (such as ICL-NUIM in last two row), especially datasets with low continuity (such as Office Room of ICL-NUIM in the fourth row), their results are in a mess when we pay attention to the pillows on the sofa and table alignment, while our results are acceptable.

The numerical comparison is displayed in Table 2. For time comparison, our algorithm takes the least time for all datasets. In preprocessing, we are at least tens of times faster than the fastest method, also in registration time and total time. For overlap-ratio, the overlap of our method is the most in SUN3D and ICL-NUIM. For *RMSE*, the error of our method is minimal in SUN3D and ICL-NUIM. Although the overlap-ratio is about 4% less than SDRSAC [30] and about 0.002 more than SDRSAC [30] in NYU, we have the best visual effect, which is more practical.

## 5 Conclusion

In conclusion, we propose a novel and efficient probability driven approach for point cloud registration of the indoor scene that achieves fast speed and high accuracy. The entire registration framework consists of two phases: coarse registration based on the distance matrix and fine registration based on the difference matrix. On one hand, the probability driven point cloud registration approach utilize the location relationship between key points instead of point features. On the other hand, to improve registration efficiency and accuracy, distance matrix (in scalarization) and difference matrix (in vectorization) are used to respectively guide coarse registration and fine registration. Experiments demonstrate that our method outperforms in registration accuracy than the state-of-the-art methods. Furthermore, our method is more efficient than existing methods in computing speed.

## Compliance with ethical standards

**Conflict of interest** All authors declare that they have no conflict of interest.

## References

1. Cédric Bobenrieth, Hyewon Seo, Arash Habibi, Frédéric Cordier.: Indoor scene reconstruction from a sparse set of 3D shots. Proceedings of the Computer Graphics International Conference. 27:1–27:5, Yokohama, Japan (2017)
2. Shichao Yang, Daniel Maturana, Sebastian A. Scherer.: Real-time 3D scene layout from a single image using Convolutional Neural Networks. IEEE International Conference on Robotics and Automation. 2183–2189, Stockholm, Sweden (2016)
3. Zhuo Li, Xingtong Liu, Xiang Xie, Guolin Li, Songping Mai, Zhihua Wang.: An optical tracker based registration method using feedback for robot-assisted insertion surgeries. IEEE International Symposium on Circuits and Systems. 1–4, Baltimore, MD, USA (2017)
4. Ziquan Lan, Zi Jian Yew, Gim Hee Lee.: Robust Point Cloud Based Reconstruction of Large-Scale Outdoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition. 9690–9698, Long Beach, CA, USA (2019)
5. Qing Hong Gao, Tao Ruan Wan, Wen Tang, Long Chen, Kai Bing Zhang.: An Improved Augmented Reality Registration Method Based on Visual SLAM. E-Learning and Games. 11–19, Bournemouth, UK (2017)
6. Ahmed Nassar, Karim Amer, Reda ElHakim, Mohamed ElHelw.: A Deep CNN-Based Framework for Enhanced Aerial Imagery Registration With Applications to UAV Geolocation. IEEE Conference on Computer Vision and Pattern Recognition Workshops. 1513–1523, Salt Lake City, UT, USA (2018)
7. Fernando I. Ireta Muñoz, Andrew I. Comport.: A proof that fusing measurements using Point-to-hyperplane registration is invariant to relative scale. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. 517–522, Baden-Baden, Germany (2016)
8. Guanyu Yang, Chenchen Sun, Yang Chen, Lijun Tang, Huazhong Shu, Jean-Louis Dillenseger.: Automatic Whole Heart Segmentation in CT Images Based on Multi-atlas Image Registration. Statistical Atlases and Computational Models of the Heart. 250–257, Quebec City, Canada (2017)
9. Manash Pratim Das, Zhen Dong, Sebastian A. Scherer.: Joint Point Cloud and Image Based Localization for Efficient Inspection in Mixed Reality. IEEE/RSJ International Conference on Intelligent Robots and Systems. 6357–6363, Madrid, Spain (2018)
10. Paul J. Besl, Neil D. McKay.: A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14, 239–256 (1992)
11. Jiaolong Yang, Hongdong Li, Dylan Campbell, Yunde Jia.: Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. IEEE Trans. Pattern Anal. Mach. Intell. 38, 2241–2254 (2016)
12. James Servos, Steven Lake Waslander.: Multi-Channel Generalized-ICP: A robust framework for multi-channel scan registration. Robotics and Autonomous Systems. 87, 247–257 (2017)
13. Sofien Bouaziz, Andrea Tagliasacchi, Mark Pauly.: Sparse Iterative Closest Point. Comput. Graph. Forum. 32, 113–123 (2013)
14. Bo Ren, Jia-Cheng Wu, Ya-Lei Lv, Ming-Ming Cheng, Shao-Ping Lu.: Geometry-Aware ICP for Scene Reconstruction from RGB-D Camera. J. Comput. Sci. Technol. 34, 581–593 (2019)
15. Mohamed Lamine Tazir, Tawsif Gokhool, Paul Checchin, Laurent Malaterre, Laurent Trassoudaine.: CICP: Cluster Iterative Closest Point for sparse-dense point cloud registration. Robotics and Autonomous Systems. 108, 66–86 (2018)
16. Dror Aiger, Niloy J. Mitra, Daniel Cohen-Or.: 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graph. 27, 85 (2018)
17. Nicolas Mellado, Dror Aiger, Niloy J. Mitra.: Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. Comput. Graph. Forum. 33, 205–215 (2014)
18. Qian-Yi Zhou, Vladlen Koltun.: Depth camera tracking with contour cues. IEEE Conference on Computer Vision and Pattern Recognition. 632–638, Boston, MA, USA (2015)
19. Qian-Yi Zhou, Jaesik Park, Vladlen Koltun.: Fast Global Registration. Computer Vision. 766–782, Amsterdam, The Netherlands, (2016)
20. Radu Bogdan Rusu, Nico Blodow, Michael Beetz.: Fast Point Feature Histograms (FPFH) for 3D registration. IEEE International Conference on Robotics and Automation. 3212–3217, Kobe, Japan (2009)
21. Jaesik Park, Qian-Yi Zhou, Vladlen Koltun.: Colored Point Cloud Registration Revisited. IEEE International Conference on Computer Vision. 143–152, Venice, Italy (2017)
22. Wei Gao, Russ Tedrake.: FilterReg: Robust and Efficient Probabilistic Point-Set Registration Using Gaussian Filter and Twist Parameterization. IEEE Conference on Computer Vision and Pattern Recognition. 11095–11104, Long Beach, CA, USA (2019)
23. Bing Jian, Baba C. Vemuri.: Robust Point Set Registration Using Gaussian Mixture Models. IEEE Trans. Pattern Anal. Mach. Intell. 33, 1633–1645 (2011)
24. Benjamin Eckart, Kihwan Kim, Jan Kautz.: Fast and Accurate Point Cloud Registration using Trees of Gaussian Mixtures. CoRR. abs/1807.02587 (2018)
25. Dylan Campbell, Lars Petersson.: An Adaptive Data Representation for Robust Point-Set Registration and Merging. IEEE International Conference on Computer Vision. 4292–4300, Santiago, Chile (2015)
26. Todor Stoyanov, Martin Magnusson, Achim J. Lilienthal.: Point set registration through minimization of the L2 distance between 3D-NDT models. IEEE International Conference on Robotics and Automation. 5196–5201, Paul, Minnesota (2012)
27. Bing Jian, Baba C. Vemuri.: A Robust Algorithm for Point Set Registration Using Mixture of Gaussians. IEEE International Conference on Computer Vision. 1246–1251, Beijing, China (2005)
28. Radu Horaud, Florence Forbes, Manuel Yguel, Guillaume Dewaele, Jian Zhang. Rigid and Articulated Point Registration with Expectation Conditional Maximization. IEEE Trans. Pattern Anal. Mach. Intell. 33, 587–602 (2011)
29. Martin A. Fischler, Robert C. Bolles.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commun. ACM. 24, 381–395 (1981)

30. Huu M. Le, Thanh-Toan Do, Tuan Hoang, Ngai-Man Cheung.: SDRSAC: Semidefinite-Based Randomized Approach for Robust Point Cloud Registration Without Correspondences. IEEE Conference on Computer Vision and Pattern Recognition. 124–133, Long Beach, CA, USA (2019)
31. Ziquan Lan, Zi Jian Yew, Gim Hee Lee.: Robust Point Cloud Based Reconstruction of Large-Scale Outdoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition. 9690–9698. Long Beach, CA, USA (2019)
32. Rahat Yasir, Mark G. Eramian, Ian Stavness, Steve Shirtliffe, Hema Sudhakar Duddu.: Data-Driven Multispectral Image Registration. Computer and Robot Vision. 230–237, Toronto, ON, Canada (2018)
33. Yeqing Li, Chen Chen, Fei Yang, Junzhou Huang.: Deep sparse representation for robust image registration. IEEE Conference on Computer Vision and Pattern Recognition. 4894–4901, Boston, MA, USA (2015)
34. D. Khuê Lê-Huu, Nikos Paragios.: Alternating Direction Graph Matching. IEEE Conference on Computer Vision and Pattern Recognition. 4914–4922, Honolulu, HI, USA (2017)
35. Xu Yang, Zhi-Yong Liu.: Adaptive Graph Matching. IEEE Trans. Cybernetics. 48, 1432–1445 (2018)
36. Jianbo Shi, Carlo Tomasi.: Good Features to Track. Conference on Computer Vision and Pattern Recognition. 593–600, Seattle, WA, USA (1994)
37. Christopher G. Harris, Mike Stephens.: A Combined Corner and Edge Detector. Proceedings of the Alvey Vision Conference. 1–6, Manchester, UK (1988)
38. Nathan Halko, Per-Gunnar Martinsson, Joel A. Tropp.: Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Review. 53, 217–288 (2011)
39. Ankur Handa, Thomas Whelan, John McDonald, Andrew J. Davison.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. IEEE International Conference on Robotics and Automation. 1524–1531, Hong Kong, China (2014)
40. Nathan Silberman, Derek Hoiem, Pushmeet Kohli, Rob Fergus.: Indoor Segmentation and Support Inference from RGBD Images. European Conference on Computer Vision. 746–760, Florence, Italy (2012)
41. Jianxiong Xiao, Andrew Owens, Antonio Torralba.: SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. IEEE International Conference on Computer Vision. 1625–1632, Sydney, Australia (2013)

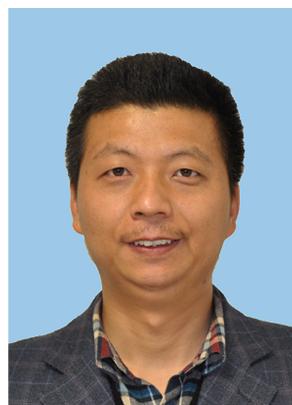
**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Kun Dong** received her B.S. degree in computer science and technology from Hangzhou Dianzi University, Hangzhou, in 2018. Currently, she is currently pursuing the Master degree at the School of Software, Shandong University, Jinan, China. Her interested research fields include computer vision, computer graphics, and artificial intelligence.



**Shanshan Gao** received the B.S. degree from the Shandong University of Technology, Zibo, China, in 2001, and the M.S. and Ph.D. degrees from Shandong University, Jinan, China, in 2005 and 2011, respectively. She is currently a Professor with the School of Computer Science and Technology, Shandong University of Finance and Economics. Her current research interests include computer graphics, image saliency detection, and image segmentation.



**Shiqing Xin** received the Ph.D. degree in applied mathematics from Zhejiang University, in 2009. He is an Associate Professor with the Faculty of School of Computer Science and Technology, Shandong University. His research interests include computer graphics, computational geometry, and 3D printing.



**Yuanfeng Zhou** received the masters and Ph.D. degrees from the School of Computer Science and Technology, Shandong University, Jinan, China, in 2005 and 2009, respectively. He held a post-doctoral position with the Graphics Group, Department of Computer Science, The University of Hong Kong, Hong Kong, from 2009 to 2011. He is currently a Professor with the School of Software, Shandong University, where he is also a member of the GDIV Laboratory. His current research interests include geometric modeling, information visualization, and image processing.