A ICP-Improved Point Cloud Maps Fusion Algorithm with Multi-UAV Collaboration

Hao Li[1], Xiaohan Qi[1], and Zhihua Yang[1,2(✉)]

[1] Communication Engineering Research Center, Harbin Institute of Technology, Shenzhen, Guangdong, China
lihao_hitsz@163.com, qixiaohan@stu.hit.edu.cn, yangzhihua@hit.edu.cn
[2] Pengcheng Laboratory, Shenzhen, Guangdong, China

Abstract. Using depth sensor devices to obtain 3D reconstruction maps is widely used in robotics and UAVs technology. For instance, large-scale environments reconstruction usually requires multiple or multiple angles to construct local point cloud maps, and then use 3D point cloud fusion technology to obtain global maps. In this paper, we present a complete point cloud fusion system for 3D map reconstruction of indoor environment based on traditional method, including initial fusion and precise fusion. Furthermore, we adopt the method of kd-tree search to match the points in the cloud of two point clouds, and eliminate the wrong matching or the matching point pairs with large error to improve the fusion efficiency. Our experiments show that, the convergence speed of the iterative process is improved, and the time complexity of the whole fusion algorithm is reduced while the final fusion effect achieves the required accuracy.

Keywords: 3D reconstruction · Point cloud map · Map fusion · ICP

1 Introduction

In recent years, with the continuous advancement of image acquisition equipment, it is convenient to obtain massive high-resolution image data using equipment, such as mobile phones, street-view vehicles, and drones. For unmanned vehicles, mobile robots, AR and other agents in a 3D environment, in addition to the understanding of 2D images, the interaction of the 3D environment is also required. In view of the above reasons, the classic problem of constructing a 3D world (image-based 3D modeling) through image data has increasingly become a hot spot for computer vision researchers.

Nowadays, it is of great significance to use computer graphics and computer vision to reconstruct the city building and object model. However, in most cases, we cannot obtain all the data needed for the object or environment to be reconstructed at one time, and it requires multi-angle and multi-time measurement

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 550–560, 2019.
https://doi.org/10.1007/978-3-030-19153-5_56

to build local point cloud maps. Therefore, point cloud fusion becomes an indispensable step in 3D reconstruction [1]. The fusion of 3D point cloud is a very important part in the reconstruction of 3D objects. The point cloud registration technology refers to the integration of point clouds in different coordinate systems into the same coordinate system by certain algorithms with the overlapping information of point clouds, to obtain a complete 3D point cloud model of objects or environments. At present, point cloud fusion is mainly divided into three categories: manual fusion, instrument-based fusion, and algorithm-based automatic fusion. In this paper, we study the algorithm-based point cloud fusion technology.

Furthermore, the process can be divided into an *initial fusion* and *precise fusion* [2], which initial fusion is to perform a rough registration transformation of two pieces of point cloud at any position, so that the overlapping regions are roughly in the same position, providing a suitable initial value for precise fusion. Common methods [3] for initial fusion include labeling method, turntable method, principal component analysis method, and curvature analysis and so on. At the same time, precise fusion is to accurately match the point cloud data after initial fusion, optimize the translation matrix and the rotation matrix.

At present, the point cloud registration algorithm is widely used as the *iterative nearest point* (ICP) algorithm proposed by Besl *et al.* [4] in 1992. Then, Magnusson *et al.* [5] used *normal distributions transform* (NDT) to register 3D point cloud. However, the ICP algorithm can fulfill the requirements of our registration task in many cases, there are still some problems. On the one hand, the biggest impact on the ICP algorithm is that the algorithm relies on a better initial registration result. If the initial registration position is poor, the algorithm is simple to fall into local optimization or infinite iteration [6], which greatly affects the convergence speed of the algorithm and increases the time complexity. On the other hand, the traditional ICP algorithm searches for the nearest neighbors at a time cost, and the iterative convergence speed is slow, which is difficult to use in systems that require real-time performance [7]. Because of these limitations, this paper conducts an in-depth study on the point cloud fusion algorithm. We have improved the initial fusion and precise fusion with the matching speed and the time complexity.

The contributions are specified as follow:

(a) we propose the initial fusion of the point clouds with the ISS-based SAC-IA algorithm, which can satisfy the correct matching of the initial matching punctually feature point pair.
(b) we develop the improved ICP with kd-tree to search the nearest neighbors, and propose a method to filter matching point pairs by the normal vector angle to minimize the error iteration.
(c) our experimental results show the feasibility of the algorithm and we analyze the convergence speed, iteration time and fusion error of point cloud fusion.

2 Problem Formulation

2.1 Reference Scenario

Recently, with the need for agents to more accurately locate and construct high-precision 3D maps, a single UAV 3D SLAM algorithm can no longer satisfy the requirements, especially for large-scale 3D reconstruction. It takes a lot of time and generates a large accumulation error, which leads to a significant drop in the accuracy of positioning and mapping. Therefore, we proposed the multi-UAV 3D SLAM algorithm framework that becomes a good solution.

In this paper, the purpose of point cloud fusion is to complete 3D map reconstruction of large-scale environment more efficiently. It is proposed that under the framework of multi-UAV collaboration, multiple UAVs equipped with the same RGB-D sensor are used to reconstruct the local 3D point cloud map using the SLAM algorithm, and we finally obtain the global point cloud map of the environment through the fusion of the map.

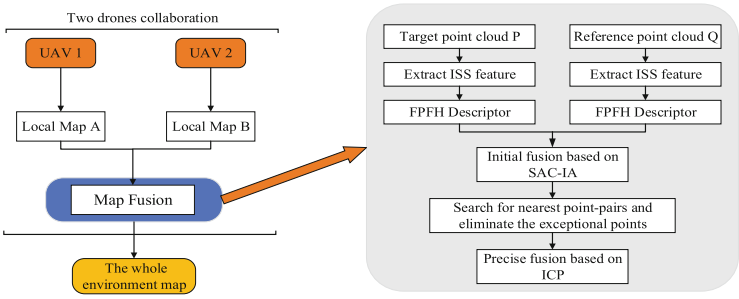


Fig. 1. Collaboration framework and point cloud fusion flow diagram

In Fig. 1, the flow chart of the two-UAVs collaboration is given. Each drone builds its own local point cloud map, and then integrates the map at the back to get the global map. In this paper, we mainly focus on the point cloud fusion part at the back end. The fusion flow diagram is given on the right side of the figure, and the detailed algorithm process is as follows.

2.2 Problem Definition

Suppose we have obtained two point cloud maps with overlapping areas, which are denoted as point cloud $\mathbf{P}$, expressed as $\mathbf{P} = \{p_i | p_i \in \mathbf{R^3}, i = 1, 2, \cdots, N\}$, the number of point clouds is N, and point cloud $\mathbf{Q}$, similarly, expressed as $\mathbf{Q} = \{q_i | q_i \in \mathbf{R^3}, i = 1, 2, \cdots, M\}$, the number of point clouds is M. They are three-dimensional point coordinates in different coordinate systems, where $\mathbf{P}$ is the target point cloud and $\mathbf{Q}$ is the reference point cloud. Our purpose is to unify point cloud $\mathbf{Q}$ through coordinate transformation to the same coordinate system as point cloud $\mathbf{P}$ and merge it into a point cloud map.

The rigid-body transformation can be realized through the following equation when the coordinate transformation of points $\mathbf{P}$ and $\mathbf{Q}$ in two different coordinate systems is carried out

$$q_i^{'}(x,y,z) = \boldsymbol{R}q_i(x,y,z) + \boldsymbol{t} \tag{1}$$

In Eq. (1), we fixed the coordinate system of $\mathbf{P}$, and obtained the $q_i^{'}(x,y,z)$ by the rotation and translation transformation of the point $q_i(x,y,z)$ in $\mathbf{Q}$. At this time, we think that $q_i^{'}(x,y,z)$ has been transformed into the coordinate system of $\mathbf{P}$,where $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{t}$ is the translation matrix. They can be expressed as

$$\boldsymbol{R_{3\times 3}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \begin{bmatrix} cos\beta & 0 & -sin\beta \\ 0 & 1 & 0 \\ sin\beta & 0 & cos\beta \end{bmatrix} \begin{bmatrix} cos\gamma & sin\gamma & 0 \\ -sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

and

$$\boldsymbol{t_{3\times 1}} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \tag{3}$$

where α, β and γ respectively represent the rotation angles of the point along the x, y and z axes, t_x, t_y and t_z respectively represents the translation of the point along the x, y and z axes.

The rigid-body transformation involves six unknowns $\alpha, \beta, \gamma, t_x, t_y$ and t_z, so at least six linear equations need to be determined. Therefore, three pairs of corresponding point pairs need to be found in the overlapping region of the point cloud to be matched, and cannot be collinear to complete the parameter estimation of the transformation matrix.

Let the rotation transformation matrix be $\boldsymbol{R}$ and the translation transformation vector be $\boldsymbol{t}$, and use $f(\boldsymbol{R}, \boldsymbol{t})$ to represent the error between the reference point set $\mathbf{Q}$ and the target point set $\mathbf{P}$ under the transformation matrix $(\boldsymbol{R}, \boldsymbol{t})$

$$f(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{M} \|p_i - (\mathbf{R}q_i + \boldsymbol{t})\| \tag{4}$$

$$\begin{aligned} s.t. \triangle f &< f_0 \\ \triangle \boldsymbol{R} &< \delta_1 \\ \triangle \boldsymbol{t} &< \delta_2 \end{aligned}$$

where f_0 is the distance error of point cloud registration, δ_1 and δ_2 are defined allowable error quantities, then the problem of solving the optimal transformation matrix can be transformed into solving the optimal solution $(\boldsymbol{R}, \boldsymbol{t})$ that satisfies $\mathbf{min}(f(\boldsymbol{R}, \boldsymbol{t}))$. Towards this goal, we can get the global fusion map which is formed by the fusion of multiple local point cloud maps.

3 Point Cloud Map Fusion Algorithm

Through the above description of the problem, we want to fuse the point cloud maps in different coordinate systems into a unified coordinate system, which is

the process of solving the transformation parameters $\boldsymbol{R}$ and $\boldsymbol{t}$. First, we need to filter the point cloud data to dry, then find the key points and describe the key points. After doing the above work, we use the initial fusion to obtain the transformation parameters $\boldsymbol{R}$ and $\boldsymbol{t}$, and finally optimize them by precise fusion.

3.1 ISS Feature Extraction

The method of finding the nearest point of the neighborhood usually uses the ε nearest neighbor or the k nearest neighbor. The former is to search for the points in the sphere with a center point P_i and a radius of ε, and the latter is to search for k points with a minimum geometric distance from the center point P_i. In order to improve the speed of feature point extraction, we improved the ISS (Intrinsic Shape Signatures) algorithm, using k nearest neighbor domain point $N^k(p_i)$.

In the specific search, use the kd-tree search method to find the k nearest neighbor point. For the case where the point cloud of the point cloud has different degrees of density, we set a neighborhood search radius r, and only select k_i points whose distance is less than r. Make it possible to select relatively few points in a sparse area and more points in a dense place. In this way, the point set formed by the nearest point of the k_i neighborhood of each point can obtain the center point c_i

$$c_i = \frac{1}{k_i} \sum_{p_{ij} \in N^k(p_i)} p_{ij} \qquad \|p_i - p_{ij}\| < r \tag{5}$$

where the 3D point cloud is written by $\mathbf{P} = \{p_i\}_{i=1}^{N}$ and $N^{k_i}(p_i) = \{p_{ij}\}, 1 \leq j \leq k_i$ representatives the k_i nearest neighbor point. Further, solve the correlation matrix $\boldsymbol{C_i}$

$$\mathbf{C}_i = \sum_{p_{ij} \in N^{k_i}(p_i)} (p_{ij} - c_i)(p_{ij} - c_i)^T \tag{6}$$

Calculate the eigenvalue of the matrix as $\{\lambda_0, \lambda_1, \lambda_2\}, \lambda_0 \leq \lambda_1 \leq \lambda_0$. If p_i satisfies the following formula, it is regarded as a feature point.

$$\frac{\lambda_2}{\lambda_1} < \varepsilon_1, \frac{\lambda_3}{\lambda_2} < \varepsilon_2 \tag{7}$$

where ε_1 and ε_2 are the preset threshold.

3.2 Initial Fusion

Sample consensus initial alignment (SAC-IA) can obtain the corresponding relationship between the feature points after extracting the feature from the point cloud. Then, calculate an initial transformation matrix, so that the two point cloud sets obtain a relatively good initial position.

This algorithm relies on point feature histogram (PFH) to obtain a description of the feature, but in order to reduce the time cost, as far as possible to

calculate the descriptor for a small number of point clouds. We use an accelerated description algorithm FPFH that is improved for PFH. Therefore, before the implementation of this algorithm, you should calculate the feature point clouds fast point feature histogram (FPFH) [10].

The steps are described as follows:

Algorithm 1. Initial Fusion of FPFH Feature Description

Input: Target point cloud **P** and reference point cloud **Q**

Begin
Set maximum distance threshold d_0 *and* e_0
Calculate the FPFH of point sets of $\boldsymbol{P}$ *and* $\boldsymbol{Q}$
// Search for match point pairs
while ($d > d_0$ and $e > e_0$) **do**
  Find points in $\boldsymbol{P}$ *that have similar FPFH characteristics to points in* $\boldsymbol{Q}$
  Randomly select n point pairs from these similar points
  $d = \frac{1}{n} \sum \|p_i - q_i\|$
  Calculate $p_i = \boldsymbol{R}q_i + \boldsymbol{t}$
  $e_i = p_i - (\boldsymbol{R}q_i + \boldsymbol{t})$
end while
End

Output: Initial transformation matrix $\boldsymbol{R}$ and $\boldsymbol{t}$

The ultimate goal of the above registration is to find a set of optimal transforms in all transforms to minimize the value of the error function, and the transformation at this time is the final registration transformation matrix. Then, the registration result can be further obtained with precise fusion.

3.3 Improved ICP Algorithm Based on kd-tree

We complete the precise fusion process by ICP algorithm. However, the classical ICP algorithm that it demands a high relative position between the point clouds and has larger amount of computation. Besides that, the iterative process may not converge to the global optimal solution. In order to improve the matching speed and precision of ICP algorithm, this paper improves the classical ICP algorithm based on kd-tree.

Kd-tree is a data structure used in multi-dimensional data space segmentation. It divides point cloud data in k-dimensional space according to a certain segmentation criterion, and applies the most extensive segmentation criterion to maximum variance segmentation. The point clouds used in this paper are scattered point clouds, so this article uses kd-tree search to implement nearest neighbor queries.

The time complexity of the traditional ICP algorithm is $\mathcal{O}(MN)$, where M and N are the number of two point clouds respectively. However, when the point cloud data is large, it takes a lot of time to search for the nearest point. An improved ICP algorithm based on kd-tree search is used to improve the fusion

registration speed of the algorithm. The time complexity of the algorithm can be reduced to $\mathcal{O}(logN)$. Therefore, the kd-tree nearest neighbor search method is used in this paper to accelerate the search of corresponding point pairs. Furthermore, the point pairs are screened out to eliminate the mismatched point pairs so as to improve the convergence speed while ensuring the fusion effect.

Improved algorithm steps are as follows:

Algorithm 2. Improved ICP Algorithm

Input: Point cloud after initial fusion

Begin
Set $K = 0$*, give a series of thresholds* $r_0, d_0, f_0, \delta_1, \delta_2$
// Search for match point pairs
for $i = 1, 2, \cdots, k$ **do**
  Use kd-tree search p_i *and* $q_i (r < r_0)$
  Remove the mismatched point pairs
  if $d(p_i, q_i) < d_0$ **then**
    Solve the coordinate transformation,get $\boldsymbol{R}$ *and* $\boldsymbol{t}$
  end if
end for
// Optimized transformation matrix
while $(\triangle f > f_0$ or $\triangle \boldsymbol{R} > \delta_1$ or $\triangle \boldsymbol{t} > \delta_2)$ **do**
  $q_i^{'} = \boldsymbol{R} q_i + \boldsymbol{t}$
  $f(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{M} \| p_i - (\boldsymbol{R} q_i + \boldsymbol{t}) \|$
  $K = K + 1$
end while
End

Output: Transformation matrix $\boldsymbol{R}$ and $\boldsymbol{t}$

4 Experimental Results

In this experiment, we apply the improved ICP algorithm based on kd-tree to register point cloud maps of indoor environment which are constructed by SLAM algorithm. Experimental device: ubuntu 16.04 LTS, 4 GB, intel Core i5-7500 CPU @3.40 GHz.

According to the algorithm steps described in our paper, we first extract the feature points from the point cloud, and filter the extracted feature points to obtain the feature descriptors. As shown in Fig. 2, on the left is the source point cloud data with the RGB color and on the right is obtained by extracting feature points from the origin cloud map, where the green colors in the picture indicate the extracted feature points.

The experimental results of initial fusion and related data are shown in Table 1. We know the registration score of SAC-IA based on kd-tree is better (where the score represents the Euclidean distance between the corresponding

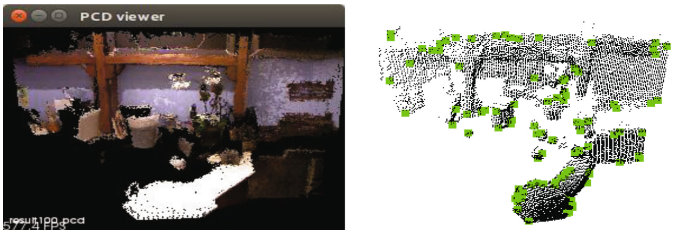
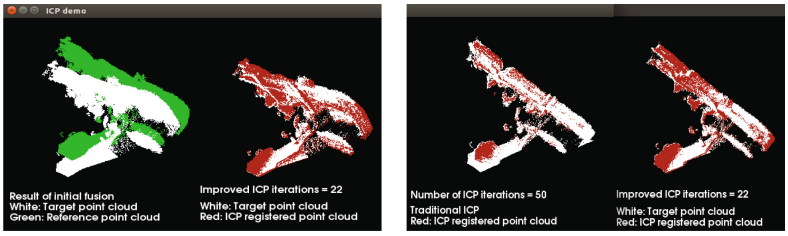


Fig. 2. Feature detection and description

Table 1. Experimental results of initial fusion

Algorithm	SAC-IA	SAC-IA based on kd-tree
Score	0.236	0.162
Rotation matrix	$\begin{bmatrix} 0.968 & 0.252 & 0.006 \\ -0.252 & 0.869 & -0.007 \\ 0.004 & 0.008 & 1.021 \end{bmatrix}$	$\begin{bmatrix} 1.108 & 0.312 & 0.010 \\ -0.298 & 0.969 & -0.009 \\ 0.006 & 0.019 & 0.954 \end{bmatrix}$
Translation vector	$\begin{bmatrix} 0.230 & 0.252 & 0.423 \end{bmatrix}$	$\begin{bmatrix} 0.243 & 0.265 & 0.419 \end{bmatrix}$

point pairs, and the smaller the better), and transformation matrix is similar. The transformation matrix obtained by SAC-IA is inaccurate, so it can only be used for rough registration. With a large number of points, the feature extraction and calculation of feature descriptors are very time-consuming, making the SAC-IA algorithm inefficient. From the experimental results, the introduction of kd-tree accelerated search can reduce the time cost while ensuring the accuracy of the transformation matrix.



(a) Improved ICP (b) Compared with Traditional ICP

Fig. 3. ICP iterations results

After initial fusion of two point clouds with overlapping areas, we get the registration result on the left side of Fig. 3. It can be seen that the fusion effect is not very good but acceptable. On this basis, we use the improved ICP algorithm

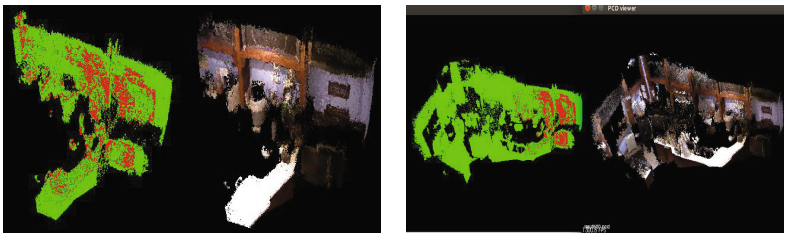
for accurate fusion registration, and the simulation results show the results on the right side of Fig. 3.

Figure 3 shows two images (a) and (b), in which (a), the content shown in the figure on the left is the result of initial fusion, while showing on the right side as a result of precise registration. It can be seen that a better fusion effect can be obtained by using the improved ICP precision registration. In additional, picture (b) shows the comparison between the final fusion map and the real point cloud map. It can be seen that the error of fusion effect is relatively small.

Table 2. Experimental results of precise fusion

Algorithm	ICP	Improved ICP
Time (s)	52.842	20.463
Iterations	50	22

Table 2 describes the time and iterations of the algorithm. The improved ICP algorithm can achieve the required fusion effect with fewer iterations, thus greatly reducing the time cost of the algorithm.



(a) 156575 and 311829 points (b) 156575 and 870104 points

Fig. 4. Compare the fusion effect of different cloud data

In Fig. 4, the number of point clouds is 156575 and 311829 in (a). In order to study the impact of the huge number of point clouds on the ICP algorithm, we performed two fusions experiments with 156515 and 870104 point clouds shown in (b). When the amount of cloud data is up to hundreds of thousands or even more, the improved algorithm can significantly improve the computational efficiency while maintaining the approximate registration accuracy compared with the traditional algorithm.

In Table 3, we set up 4 sets of experimental data A, B, C and D for comparison of simulation experiments, where P represents target point cloud and Q represents reference point cloud. The evaluation results are shown in Fig. 5, in which A+Improved represents the result of the improved ICP algorithm of the A group data, and A+Traditional represents the result of the traditional ICP

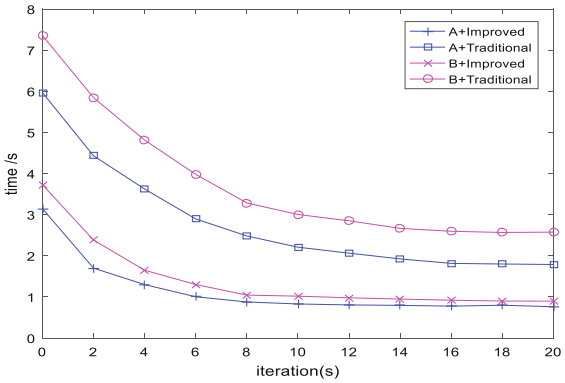


Fig. 5. Time of each iteration

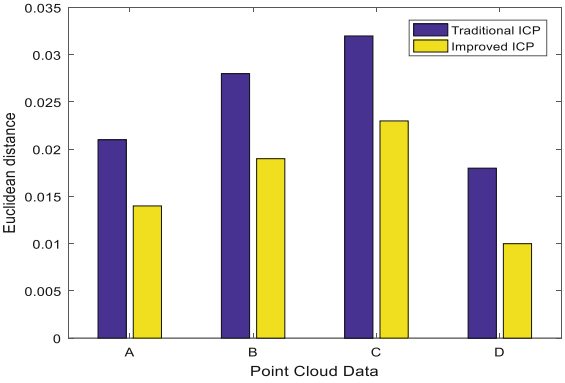


Fig. 6. Error of fusion evalution

Table 3. Point cloud data

Group	A	B	C	D
P	311829	639235	870104	156575
Q	156575	311829	639235	156575

algorithm of the A group data, and the rest are similar. It can be seen from the simulation results, compared with the traditional ICP algorithm, the improved algorithm can reduce the iteration time, shorten the convergence speed, and can achieve higher precision map fusion with fewer iterations. When the point cloud data increases, the time cost of the improved algorithm increases less, which is a big advantage compared to the traditional algorithm. The Fig. 6 shows the fusion result error of four groups of data. We use Euclidean distance to represent the magnitude of the error, it can be seen that the improved algorithm can reduce the fusion error and get better global map.

5 Conclusion

From the above experiments, the proposed collaboration framework is of great help for the 3D reconstruction of large-scale indoor environment, and can be obtained by using local sub-map fusion. Aiming at the shortcomings of traditional initial fusion and slow convergence of ICP algorithm, this paper proposes an improved algorithm based on kd-tree nearest point search. According to the density of the point cloud, feature points can be extracted and the nearest neighbors can be searched effectively. In the same way, the angle threshold of the normal vector is set to eliminate the wrong matching point pair, which can avoid the algorithm to increase the time cost in the wrong direction.

Finally, in this paper, a 3D point cloud maps of indoor environment are constructed by using the intelligent body equipped with the SLAM algorithm for fusion experiments. The results show that for large-scale point cloud data maps, the improved algorithm can improve the convergence speed of ICP algorithm with higher precision, which proves that the study in this paper has certain application value.

Acknowledgments. The authors would like to express their high appreciations to the supports from the National Natural Science Foundation of China (61871426) and Basic Research Project of Shenzhen (JCYJ20170413110004682).

References

1. He, H., Wang, H., Sun, L.: Research on 3D point-cloud registration technology based on Kinect V2 sensor. In: 2018 Chinese Control and Decision Conference (CCDC). IEEE (2018)
2. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. Image Vis. Comput. **25**(5), 578–596 (2007)
3. Qiu, S., Luo, Y.: Point cloud registration based on improved ICP algorithm. Henan Science and Technology (2017)
4. Besl, P.J., Mckay, N.D.: Method for registration of 3-D shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, pp. 239–256 (1992)
5. Jun, L., Wei, L., Donglai, D., Qiang, S.: Point cloud registration algorithm based on NDT with variable size voxel. In: Control Conference, pp. 3707–3712 (2015)
6. Sharp, G.C., Lee, S.W., Wehe, D.K.: ICP registration using invariant features. IEEE Trans. PAMI **24**(1), 90–102 (2002)
7. Schmuck, P., Chli, M.: Multi-UAV collaborative monocular slam. In: IEEE International Conference on Robotics and Automation, pp. 3863–3870 (2017)
8. Zhong, Y.: Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: IEEE International Conference on Computer Vision Workshops, pp. 689–696 (2010)
9. Mur-Artal, R., Tards, J.D.: ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
10. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: IEEE International Conference on Robotics and Automation, pp. 3212–3217 (2009)