SAC-IA Algorithm Based on Parallel KD-Tree Search and Improved Feature Point Selection

Wei Wei
University of Science and Technology of China
Hefei, China
ustcwei@mail.ustc.edu.cn

2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE) | 979-8-3503-3157-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCECE58074.2023.10135352

***Abstract*—In recent years, with the rapid development of autonomous driving and intelligent robots, point cloud stitching technology has attracted people's attention. At present, the sampling consistency initial registration algorithm is often used for coarse registration of point clouds, but this algorithm has problems such as large randomness in feature point selection and low search efficiency. this paper proposes a feature point selection method considering the relationship between coordinate distance and spatial similarity, which prevents the occurrence of local optimization by setting the distance threshold, and uses the FPFH to calculate the spatial feature similarity relationship between the source point cloud and the target point cloud point pair. The randomly selected feature pairs with similar spatial characteristics are saved to ensure that the selected point pairs are overlapping as much as possible, which is conducive to the solution of rotation matrix and translation vector. By comparing the registration results of the traditional algorithm, it is concluded that the error of the proposed algorithm is reduced by 15.70%. In addition, the KD-Tree parallel technology based on OpenMP is also used to greatly improve the search efficiency of the corresponding point.**

Keywords-component; SAC-IA; OpenMP; spatial features; Point cloud stitching; KD-Tree

I. Introduction

In recent years, with the development of lidar camera technology, concepts such as three-dimensional reconstruction and reverse engineering have gradually emerged. 3D Reconstruction[1]-[2].In a broad sense, it refers to the recovery and reconstruction of a three-dimensional object or three-dimensional scene, which is convenient for computer representation and processing according to the reconstructed model. The point cloud data including x-axis, y-axis, and z-axis coordinates acquired through lidar scanning is used to reconstruct the measured object.

In practical application scenarios, it is often necessary to deploy multiple lidars, so there will be a problem of unifying the coordinate system of multi-point cloud data. The Iterative Closest Point (ICP) algorithm proposed by BESL and MCKAY[3] is currently the most commonly used algorithm for point cloud registration, but the ICP algorithm has low registration accuracy and is easy to fall into the local optimal solution[4], so coarse registration is required to adjust the initial distance and pose between the two feature clouds to provide a good initial position for the ICP algorithm. In order to solve such problems, Rusu et al.[5] proposed a Sample Consensus Initial Initial Alignment (SAC-IA) algorithm based on Fast Point Feature Histogram (FPFH). This method achieves the effect of coarse splicing by finding point pairs with similar geometric relationships, but the selection of feature points of the SAC-IA method is random, and the time complexity of searching for point pairs with similar geometric relationships is high.

In actual scenarios, the point cloud data generated by multiple lidar scans often have multiple non-overlapping parts, which makes some points in the source point cloud do not have corresponding points on the target point cloud. If Most of the point pairs selected by the SAC-IA algorithm come from non-overlapping parts, which may cause unimaginable errors to the registration results. By consulting relevant literature[6]-[10], FPFH is a multidimensional vector, which describes the spatial geometric relationship between points and domains based on point coordinates, normal vectors, and curvature. Similar pairs of FPFH points will have similar spatial geometric relationships. If the feature point pairs of similar FPFH are selected, the occurrence of the above situation can be greatly reduced.

But the use of FPFH will also greatly increase the execution time. K-Dimensional Tree (KD-Tree) and Open Multi-Processing (OpenMP)[11]-[12] are often used for fast searching of point clouds. By creating multiple threads to execute KD-Tree search tasks in parallel, the operating efficiency can be greatly improved. KD-Tree is often used in the field of point cloud search. In the multi-dimensional space, data of different dimensions are segmented, and the segmented data is used to search and search for adjacent points in the point cloud. Since KD -Tree is a kind of balanced binary tree, so the time complexity of searching is $O(\log_2 N)$. Its search efficiency is high.

Therefore, a SAC-IA algorithm based on parallel KD-Tree search and improved feature point selection is proposed. As shown in Figure 1, the disadvantage of large randomness of feature point selection is optimized, and the parallel model of OpenMP is used to greatly improve the search efficiency. Finally, the error comparison and search efficiency comparison with the traditional algorithm are verified.

The main contributions of this paper are as follows:

1) Optimize the selection of SAC-IA features to improve the accuracy of the final registration
2) Use the parallel computing model to improve the search efficiency of similar points

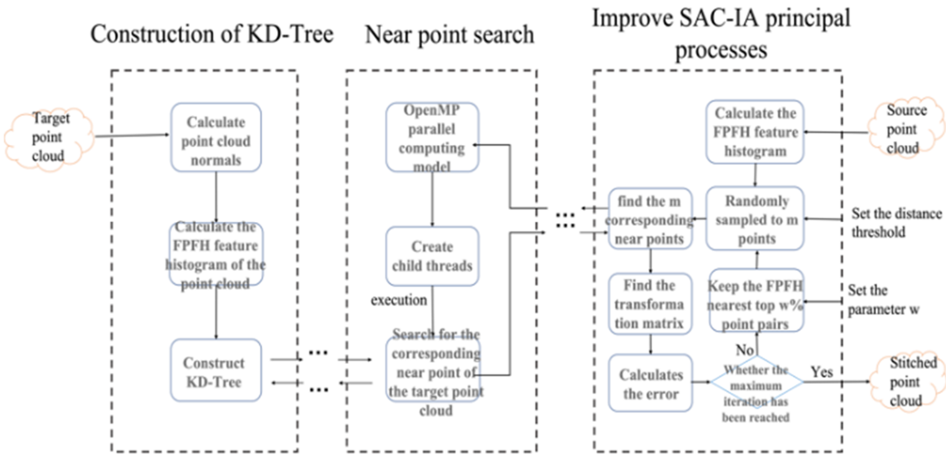

Figure 1 General structure

II. RELATED WORK

In the process of 3D reconstruction, point cloud stitching is one of the key technical links, due to the limited accuracy of 3D feature matching technology. Therefore, scholars at home and abroad have carried out a lot of research on it.

Faugeras[13] used the quaternion solution matrix method for the first time on point cloud splicing, and proposed a quaternion-based splicing algorithm based on the quaternion method. The ICP algorithm proposed by Besl[3] is the most widely used point cloud splicing algorithm, but the ICP algorithm has certain requirements for the initial value of iteration, and the large error of the initial value will lead to the error of local convergence, and the calculation amount is huge and consumes a lot of time. Many subsequent point cloud splicing is based on ICP to solve these two defects.

Ding et al.[14] proposed an object detection and transformation matrix estimation method based on sensor 3D point cloud information, in which the sensor samples and extracts the FPFH features, and uses the FPFH features of the point cloud to obtain the pose information of the target in the scene using SAC-IA and ICP algorithms. Liu et al.[15] In this paper, an improved SAC-IA algorithm based on voxel nearest neighbor search is proposed to improve the efficiency and accuracy of the algorithm, and the original SAC-IA algorithm is optimized with low efficiency and low accuracy, which cannot meet the problems of real-time application.

Although the above work has achieved good results in the field of point cloud stitching, it does not consider the feature point selection problem of the SAC-IA algorithm. The feature point selection is random and the distance between points is too close. Moreover, in many scenarios, the point cloud data scanned by multiple lidars will also have non-overlapping parts. If the feature points of the source point cloud just select this part, it is difficult to search for the corresponding near point according to the FPFH value. In addition, searching the target point cloud also consumes a lot of time

III. METHODOLOGY

This article has made the following improvements. First of all, in order to prevent the aggregation of multiple feature points from falling into the local optimal solution, it is necessary to ensure that the coordinate distance between randomly selected feature points is greater than the set threshold. Second, save the point pairs with similar FPFH feature histograms to the next iteration, because FPFH describes the spatial geometric relationship between feature points and their neighbor points, saving similar point pairs in each iteration can ensure that the point cloud of the next iteration and the points of the target point cloud overlap as much as possible in the spatial relationship. Finally, the KD-Tree search model based on OpenMP is used to improve the speed of the algorithm

For the above feature point selection and similarity point search, this paper describes it in more detail as follows:

A. Feature point selection considering coordinate distance and spatial similarity

In previous studies of feature point selection, much work[16]-[18] has been done to optimize selection by setting geometric constraints. However, in actual scenarios, the two point cloud data are often scanned by different lidar devices, so there may be a situation where the source point cloud feature point cannot find the corresponding point in the target point cloud.

In the work of this paper, the nearest ω% point pairs of the FPFH feature histogram are kept for each iteration to the next iteration, and the next iteration will continue to randomly select feature points until the feature points are equal to m. Because the FPFH feature histogram can represent the 3D spatial relationship between points and domains, point pairs with similar FPFH can be considered to have similar 3D spatial features. Retaining similar point pairs in each iteration can avoid the situation that the source point cloud feature points cannot find corresponding points in the target point cloud. At the same time, set a distance threshold for each feature point to prevent local optimization due to too close distance between feature points

B. Parallel search of KD-Tree based on OpenMP

In order to search for the corresponding point of the target point cloud more efficiently, this paper uses the KD-Tree search based on OpenMP[19]. The OpenMP framework, maintained by the OpenMP Architecture Review Board, a non-profit technology consortium, is an implementation of multithreading, a parallelization method. The master thread derives the specified number of slave threads, the system assigns tasks to slaves, then the threads run concurrently, and the runtime environment assigns threads to different processors.

Using the FPFH of the target point cloud to create a KD-Tree, the constructed KD-Tree can quickly find the nearest point q_i of the reference point cloud point p_i in the target point cloud. As shown in the Figure 2, the construction process of KD-Tree is as follows:

Step 1: Sorting the first dimension feature data of the target point cloud histogram, the left subtree and the right subtree respectively store the point cloud collection whose first dimension data is less than or equal to the median and store the point cloud whose first dimension data is greater than the median gather.

Step 2: Then sort the point cloud sets of the left and right subtrees constructed above according to the second dimension data, and then construct the left and right subtrees, left subtree and right subtree respectively on the basis of step 1

Step 3: Store point cloud collections whose second-dimensional data is less than or equal to the median and store point cloud collections whose second-dimensional data is greater than the median

Step 4: In step 2, construct the left and right subtrees according to the subsequent multidimensional data.

Step 5: Perform steps 1, 2, and 3 until the set is empty.

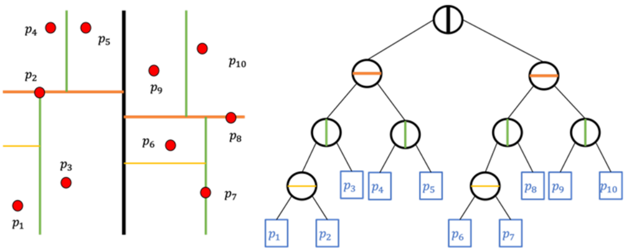

Figure 2 KD-Tree structure of point cloud

For the feature points from the source point cloud, the OpenMP main thread is used to derive multiple slave threads to perform the KD-Tree search task, and the multidimensional data of the FPFH feature histogram are compared in turn during the search process, and finally the reference point cloud point p_i the closest point q_i FPFH in the target point cloud can be found. The idea of parallel search is used to greatly improve the overall search efficiency of the algorithm.

IV. Experiments

The point cloud data used in this experiment was collected by Livox Mid-70 LiDAR. Livox Mid-70 has the advantages of large field of view and high cost performance, and has a wide range of applications in the field of autonomous driving and mobile robots, which is more suitable for practical application needs[20]-[23]. The radar parameters are shown in Table 1

Table 1 LiDAR parameters

Laser wavelength	Point cloud output	Angular error	Blind spots	Ambient temperature
905nm	100,000 points/s	< 0.1°	0.05 m	-20°C~65°C

This is shown in Figure 3. The green point cloud has 279552 point cloud data for the source point cloud, and the red point cloud has 179712 point cloud data for the target point cloud, and the point cloud data is saved in PCD format through LiDAR. In this paper, two scanned point cloud data are used to be registered using the improved SAC-IA algorithm and compared with the traditional algorithm

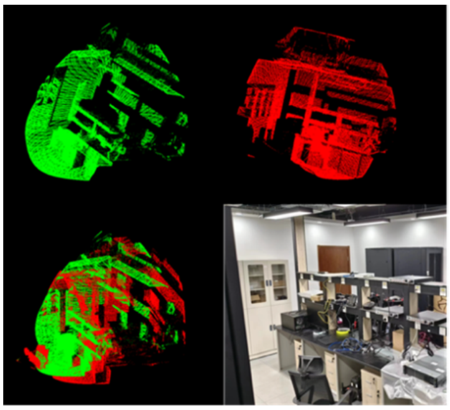
Figure 3 Scan the acquired point cloud data

A. Error Analysis

According to the above cloud data, the improved SAC-IA algorithm is compared with the traditional SAC-IA algorithm. The mean squared error of the conventional algorithm is 0.0493. The experimental results of the improved SAC-IA algorithm are shown in Table 2. When the parameters ω of the similar point pairs of the FPFH feature histogram are taken as 50 and 85, the minimum mean squared error is 0.04156 and 0.04174, respectively, which is much smaller than the error of the traditional algorithm; however, when the ω value is 30, the mean squared error is the largest, which is 0.05162, which is greater than the error result of the traditional algorithm

Table 2 Improved algorithm stitching of result data

ω value	Improved mean squared error of SAC-IA
10	0.04482
15	0.04494
20	0.04586
25	0.04794
30	0.05162
35	0.04882
40	0.05027
45	0.04983
50	0.04156
55	0.05048
60	0.05019
65	0.04819
70	0.04751
75	0.04748
80	0.04592
85	0.04174
90	0.04414

The results are compared, and when the improved algorithm parameter ω is taken as 10, 15, 20, 25, 35, 45, 50, 65, 70, 75, 80, 85, 90, the error is lower than the traditional algorithm, and when ω is taken as 50, the error is reduced by 15.70% on the basis of the original algorithm.

As shown in Figure 4, the right side is the stitching effect of the improved algorithm, and the left side is the stitching effect of the traditional algorithm. The splicing effect is compared, and the parameter w of the improved algorithm here is taken as 50, and it can be seen from the figure that the improved algorithm registration effect is better, which further affirms the improvement work in this paper

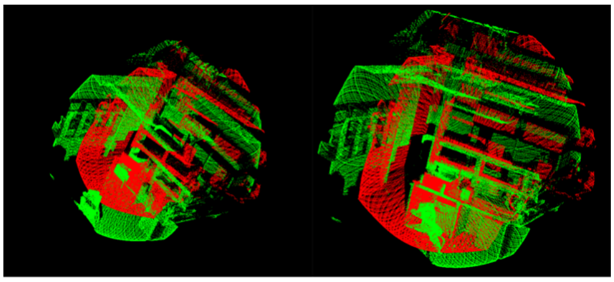

Figure 4 Contrast of stitching effects

B. *Search efficiency analysis*

The running environment of this experiment is shown in Table 3. Without using the parallel OpenMP model and KD-Tree search technology, it takes 56736s to find the corresponding point. If the KD-Tree search technology is used, it takes 149s, which greatly improves the search efficiency on the original basis; in a dual-core environment, using the OpenMP-based parallel computer framework to create two threads to execute the KD-Tree search in parallel takes 130s, which is the same as Compared with the parallel model without OpenMP, the search efficiency is increased by 12.75%.

TABLE 3 RUNNING ENVIRONMENT

Processor	Memory	Disk	Graphics card
2 GHz dual-core Intel Core i5	16 GB 1867 MHz LPDDR3	1TB	Intel Iris Graphics 540 1536 MB

V. CONCLUSIONS

In this paper, a SAC-IA algorithm based on parallel KD-Tree search and improved feature point selection is proposed. Aiming at the problems of large randomness and low search efficiency of feature point selection of existing methods, this paper based on the parallel computing model of OpenMP and combined with the efficient search technology of KD-Tree, by further introducing the feature point selection technology considering the relationship between coordinate distance and spatial similarity, it can effectively solve the problem that the source point cloud feature points cannot find the corresponding point in the target point cloud, and optimize the local optimization solution caused by feature point aggregation, thereby greatly reducing the error of point cloud registration results and improving the accuracy.

By sampling the actual scene and comparing with the registration results of the traditional algorithm, it can be concluded that the algorithm in this paper can reduce the error by 15.70% at most. Using parallel OpenMP technology improves the search efficiency by 12.75% compared with single-threaded.

ACKNOWLEDGMENT

Thanks to Professor Yang for his guidance

REFERENCES

[1] Kang Z, Yang J, Yang Z, et al. A review of techniques for 3d reconstruction of indoor environments[J]. ISPRS International Journal of Geo-Information, 2020, 9(5): 330.

[2] Munoz-Silva E M, González-Murillo G, Antonio-Cruz M, et al. A Survey on Point Cloud Generation for 3D Scene Reconstruction[C]//2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE). IEEE, 2021: 82-87.

[3] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. Spie, 1992, 1611: 586-606.

[4] Sonka M, Hlavac V, Boyle R. Image processing, analysis, and machine vision[M]. Cengage Learning, 2014.

[5] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.

[6] Zhao T, Li H, Cai Q, et al. Point cloud segmentation based on FPFH features[C]//Chinese Intelligent Systems Conference. Springer, Singapore, 2016: 427-436.

[7] Da-he M A, Guo-zhu L I U. Improved method of point cloud registration based on FPFH feature[J]. Computer and Modernization, 2017 (11): 46.

[8] Lin S, Peng Y, Cao G. Low Overlapping Plant Point Cloud Registration and Splicing Method Based on FPFH[C]//Chinese Conference on Image and Graphics Technologies. Springer, Singapore, 2022: 103-117.

[9] AV S M, Kanna B R. Parallel FPFH SLAM for Aerial Vehicles[C]//2021 IEEE Conference on Norbert Wiener in the 21st Century (21CW). IEEE, 2021: 1-4.

[10] Hai-peng L I, Dan X U, Yu-ting F U, et al. A scattered point cloud simplification algorithm based on FPFH feature extraction[J]. Journal of Graphics, 2022, 43(4): 599.

[11] Kwedlo W, Łubowicz M. An OpenMP parallelization of the K-means algorithm accelerated using KD-trees[C]//International Conference on Parallel Processing and Applied Mathematics. Springer, Cham, 2020: 457-466.

[12] Wu G, Cao L, Tian H, et al. HY-DBSCAN: A hybrid parallel DBSCAN clustering algorithm scalable on distributed-memory computers[J]. Journal of Parallel and Distributed Computing, 2022.

[13] Faugeras O D, Hebert M. A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces[C]//Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 2. 1983: 996-1002.

[14] Ding Q, He J. The detection of non-cooperative targets in space by using 3D point cloud[C]//2019 5th international Conference on Control, Automation and Robotics (ICCAR). IEEE, 2019: 545-549.

[15] Liu B, Liu L, Tian F. An Improved SAC-IA Algorithm Based on Voxel nearest Neighbor Search[J]. Critical Reviews™ in Biomedical Engineering.

[16] Shi X, Peng J, Li J, et al. The iterative closest point registration algorithm based on the normal distribution transformation[J]. Procedia Computer Science, 2019, 147: 181-190.

[17] Kim H, Hilton A. Influence of colour and feature geometry on multi-modal 3d point clouds data registration[C]//2014 2nd international conference on 3D vision. IEEE, 2014, 1: 202-209.

[18] Wang B, Xie J, Wang X, et al. A new method for measuring the attitude and straightness of hydraulic support groups based on point clouds[J]. Arabian Journal for Science and Engineering, 2021, 46(12): 11739-11757.

[19] Chapman B, Jost G, van der Pas R. Using OpenMP in the Real World[J]. 2007.

[20] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 3126-3131.

[21] Van Nam D, Gon-Woo K. Solid-state LiDAR based-SLAM: A concise review and application[C]//2021 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2021: 302-305.

[22] Kelly C, Wilkinson B, Abd-Elrahman A, et al. Accuracy Assessment of Low-Cost Lidar Scanners: An Analysis of the Velodyne HDL–32E and Livox Mid–40's Temporal Stability[J]. Remote Sensing, 2022, 14(17): 4220.

[23] Ortiz Arteaga A, Scott D, Boehm J. Initial investigation of a low-cost automotive lidar system[C]//ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Copernicus GmbH, 2019, 42: 233-240.