Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A sliding window based feature extraction method with its application to 3D grid map fusion*

Changcheng Qiu, Rongchuan Sun, Shumei Yu**, Liang Chen
School Of Mechanical And Electrical Engineering
Soochow University
Suzhou, 215006, China
yushumei@suda.edu.cn

***Abstract* - As the scale of the environment increases, the single robot simultaneous localization and mapping (SLAM) is not the best choice for building maps. In recent years, more and more researchers have focused on multi-robot SLAM. Multi-robot SLAM has the advantages of high efficiency, high precision of mapping and strong fault tolerance. However, multi-robot SLAM needs to solve the problem of map fusion. To solve this problem, this paper proposes an algorithm based on the maximum common subgraph for 3D grid map fusion, which does not require the pose relationship between robots. According to the characteristics of the indoor environment, the points where the three faces intersect are defined as corner points. Firstly, this paper proposes a sliding window-based feature point extraction method for 3D grid map. Moving the window to determine whether the point is a flat point, an edge point or a corner point according to the change in the number of occupied cubes in the window. Then we fit a plane to reject points that are not intersecting by three faces. Secondly, the backtracking method is used to find the maximum common subgraph. Finally, it calculates the transformation matrix according to the matched feature points to realize the fusion of 3D grid maps. The experimental results show that the proposed method can accurately fuse the 3D grid map with rotations.**

Keywords: Feature points extraction, maximum common subgraph, transformation matrix, map fusion

I. INTRODUCTION

Robots have been applied in many aspects of our lives, which has greatly improved people's life. Mobile robot is an important branch of robot, it senses the surrounding environment through sensors carried by itself and locate itself according to the established map, which is called SLAM.

At present, the SLAM algorithm for mobile robots is mainly divided into filter based method[1] and graph based optimization method[2]. Since the filter-based SLAM can only predict and update the state of the current pose, if an error occurred in the pose of the robot at a certain moment, then it will accompany the whole process. Therefore, the filter-based SLAM algorithm is not suitable for large-scale environmental mapping. The SLAM algorithm based on graph optimization is divided into front end and back end. The front end completes the graph building, which includes scan matching and closed loop detection, and the back end optimizes the pose of the robot. In this paper, the SLAM algorithm based on graph optimization is selected as the method to build a map for mobile robot.

When a mobile robot performs a task, it needs a precise map to navigate. Environmental maps can be divided into two-dimensional maps[3] and three-dimensional maps[4]. Real environments can be described well by three-dimensional maps, such as point cloud map and 3D grid map. However, point cloud map only models the surface of objects in the environment, which is not divided into occupied areas, free areas and unknown areas. Therefore, the mobile robot can't use point cloud map for navigation.

As the scale of the environment increases and becomes more complex, it is difficult for a single robot to build accurate maps. In recent years, multi-robot SLAM[5] has been proposed, which has the advantages of high accuracy, high efficiency and strong fault tolerance. At the same time, an important problem that multi-robot SLAM needs to solve is to fuse the maps established by multiple robots to form a global map.

Extracting features from the map to be fused and then performing feature matching is an effective way to reduce the computational complexity of the fusion algorithm. The concept of corner point was first proposed by Moravec[6]. He used the points that are clearly different from the surrounding pixels as corner points. After thirty years of research, many effective corner extraction algorithms have been proposed. Ivan Sipiran [7] proposed an adaptive neighbourhood technique to calculate the Harris response value of a vertex to complete the feature point extraction of 3D objects. The adaptive neighbourhood technique is used to determine the neighbourhood and the neighbourhood points are used for surface fitting. Then the Harris response value of the point is calculated, and the point with higher Harris response value is taken as the feature points. Helin Dutagaci[8] proposed three evaluation measures, namely False Positive, False Negative Errors and Weighted Miss Error to compare feature points detection algorithms. This article analyses and compares six feature points extraction algorithms including Mesh saliency, Salient points, 3D-Harris[7], 3D-SIFT[9], Scale-dependent corners, HKS-based interest point detection, among which Mesh saliency, 3D-Harris were found to have good robustness. Zhong yu[10] proposed the ISS feature point detection algorithm. First, the weight of each point in the neighbourhood of any point was calculated. Second, calculating the covariance matrix of the point and the eigenvalue of the matrix. If the ratio between the eigenvalues meets the threshold, it is a feature point. Rusu R.B et al.[11] proposed a registration algorithm based on FPFH features.

*This research is supported by the National Natural Science Foundation of China (61673288，61773273).
**Corresponding author: (e-mail: yushumei@suda.edu.cn).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

First, find the K neighborhood of a point. Second, calculate the Euclidean distance and the normal angle difference between any two points in all points in the k neighborhood. Then get a full description of the geometry of the point. This method has a higher computational complexity.

The purpose of extracting feature points is to complete the map fusion by matching the feature points. At present, the fusion of three-dimensional maps is mainly concentrated on point cloud maps. There are few studies on the fusion of 3D grid maps. This paper proposes a sliding window-based feature point extraction method. By moving the window, it is judged whether the point is a plane point or an edge point or a corner point according to the change in the number of occupied cubes in the window. The maximum common subgraph algorithm based on backtracking method is used to find the correspondence between feature points in different maps, and the transformation matrix is calculated to complete the fusion of the map.

The distribution of this paper is as follows, and the second section introduces the definition of the fusion of 3D grid maps. The third section describes the sliding window-based feature extraction method. The fourth section shows the detailed process of 3D grid map fusion. The fifth section is the experiment, and the sixth section gives the conclusion and future work.

II. 3D GRID MAP ASSOCIATION PROBLEM DEFINITION

When multiple local maps are created by multiple robots or the same robot at different times, these maps are in the corresponding robot coordinate system. In order to achieve map fusion, firstly, we need to find the overlapping parts of the two local maps, and find out the significant features of the overlapping parts. Then, we solve the transformation matrix according to the correspondence of the features of the overlapping parts, and finally transform one of the local maps to the coordinate system of another local map to achieve the global map. The transformation matrix consists of a 3×3 rotation matrix R and a 3×1 translation vector Tran as below:

$$T = \left[R_{(\alpha,\beta,\gamma)}\ \text{Tran}\right] \tag{1}$$

Therefore, one cube of the 3D grid map $P=[x, y, z]^T$ can be transferred to another 3D grid map coordinate through an transformation matrix T:

$$f(T,P)=T\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

Suppose we have got two local maps G_A and G_B of the global map, which are created by different mobile robots and have overlapping regions. By now, the problem of the local 3D grid map association is to find an optimal transformation matrix to transform the local map G_B into the coordinates of map G_A, then we get the global map. The process of map fusion as shown in Fig. 1.

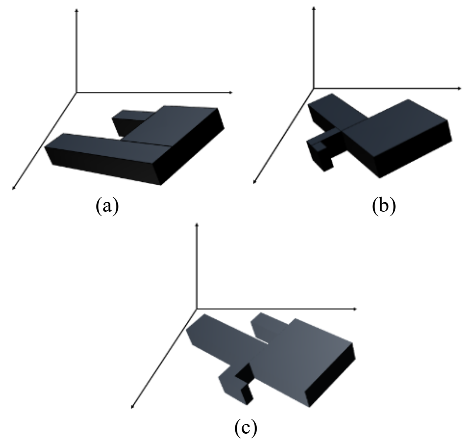


Fig. 1. The process of map fusion. (a) and (b) represent the local maps; (c) represents the global map.

III. SLIDING WINDOW-BASED FEATURE EXTRACTION

Combining the characteristics of the indoor environment and 3D grid maps, this paper proposes a sliding window-based feature point extraction method. The flow chart of this method is shown in Fig. 2.

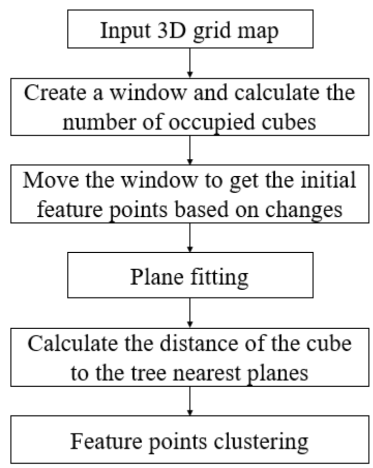


Fig. 2. Feature points extraction flow

In this paper, feature points are defined as points where three faces intersect, as shown in Fig. 3.

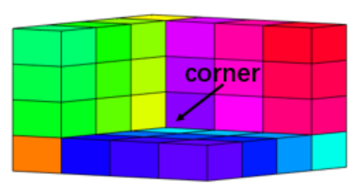
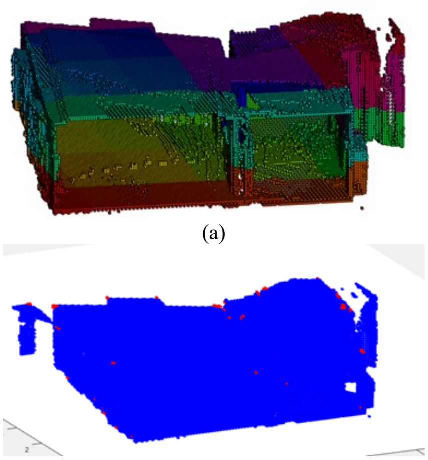


Fig. 3. Corner points defined in this paper

The region-based segmentation algorithm[12] uses neighbourhood information to classify nearby points with similar properties to obtain segmentation regions and to distinguish differences between different regions. First, calculate the curvature of all points and take the minimize curvature as the initial seed. Second, search its neighbourhood and calculate the angle between the normal of each neighbourhood point and the current seed normal. If it is less than the smoothing threshold, it belongs to the same plane. Finally, the curvature of each neighbourhood point is calculated, which is less than the curvature threshold as the next seed. The advantage of this algorithm is that it can segment the indoor environment of mobile robot well (such as corridors and rooms, etc.), while the disadvantage is that it is easy to generate excessive segmentation and insufficient segmentation, especially at the edges. Therefore, the feature points can't be directly extracted by the plane segmentation method based on region growth.

The feature point extraction method proposed in this paper is the sliding window-based method. First, create a 7*7*7 window centred on each cube in the 3D grid map and calculate the number of occupied cube in the window. Then, if the number of occupied cubes in the window changes significantly when moving the window in any direction, this cube is the initial feature point. However, there are many wrong corners in the initial feature points. As shown in Fig. 4, we choose a part of the local map for analysis. The initial feature points contain the correct feature points, but there are also many wrong corner points.



(a)

(b)

Fig. 4. (a): the part of the local map; (b) the initial feature points extracted based on sliding window method.

Then, in order to eliminate the wrong feature points, the region growth-based algorithm is used to segment the plane of 3D grid map and we perform plane fitting on the segmented points. After we calculate the plane equation, we can calculate the distance from the point to the plane. If the distance between the initial feature point and its nearest three planes is less than the distance threshold, then this initial feature point is the correct feature point. The distance threshold conditions are as follows:

$$d=\frac{|A*x_0+B*y_0+C*z_0+D|}{\sqrt{A^2+B^2+C^2}}<\varepsilon \tag{3}$$

Where A, B, C and D are the parameters of the plane equation and ε is the distance threshold.

In order to reduce the time of feature point matching, we cluster the feature points extracted by the sliding window-based feature point extraction method proposed in this paper. That is, there are only a few feature points at the same corner.

IV. 3D GRID MAP FUSION

In Session III, we extracted the feature points of the local map. In this section, we examine how to use corner points to search the maximum common subgraph for map fusion.

A. Introduction to backtracking method

Backtracking is a search method that uses depth-first strategies to select the optimal result[13]. Starting from the parent node, access all nodes in the solution space. When accessing a child node, determine whether the child node satisfies the requirements of the problem solution, and if so, continue to search downward; otherwise, trace back to the parent node.

Fig. 5 is the schematic diagram of a backtracking method. If there are m sets, there are n balls in each set. The color of the ball has a random one of red, yellow and blue. Now use the backtracking method to take a ball from each set to form a new set, and each time you get the ball must be different from the previous two, that is, any three balls in the order must be different. The second level node represents all the source nodes. As shown in Fig. 5, When the current level can find the ball with the different colors from the previous two levels, continue to explore the branch. If the current level can't find the ball with the different colors of the upper two levels, then the branch will not be branched. At this time, the layer is backtracked and continues to explore the remaining branches. In the end, we can find the branch with the largest number of balls in the set.

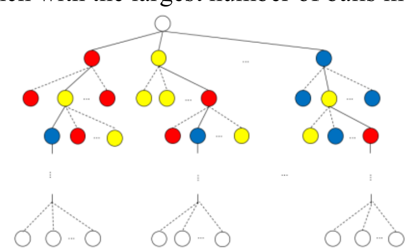
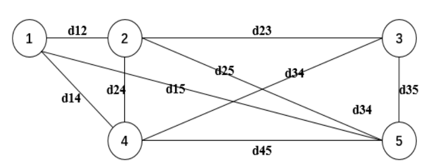


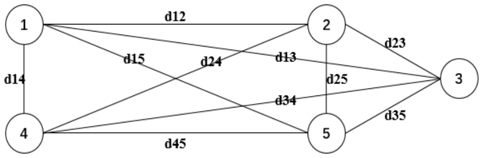
Fig. 5. The schematic diagram of a backtracking method. The solid line represents the optimal search path, and the dashed line represents the search path that does not meet the requirement.

B. Maximum common subgraph

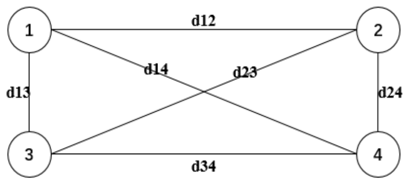
The maximum common subgraph is the area of overlap between the two maps. The problem of maximum common subgraph is an important one in structure theory. As shown in Fig. 6, d_{ij} represents the distance between node i and node j in the graph structure. If the difference between d_{ij} in structure A and d_{mn} in structure B is less than the threshold, and then there is a possibility that the node i and the node j in graph structure A have a match with the node m and the node n in structure B. The maximum common subgraph is composed of all nodes whose distances between nodes are less than the threshold, and the distance between any two nodes must be less than the threshold.



(a) Graph structure F1



(b) Graph structure F2



(c) The maximum common graph

Fig. 6. (a) and (b) represent two graph structure; (c) represents the maximum common graph of (a) and (b).

C. *Maximum common subgraph al based on backtracking*

Suppose there are two maps A and B. Feature points are extracted from the map using the methods described in this article. Since there is no initial match, there is a possibility that the feature points in map A match any of the feature points in map B. The backtracking method can be used to find nodes in two graphs that satisfy certain constraints. The constraint is:

$$\forall\{(c_A^i,c_B^i),(c_A^j,c_B^j)\}\in \mathrm{MCS}\ \Big\| \|c_A^i-c_A^j\|-\|c_B^i-c_B^j\| \Big\| < D \quad (4)$$

Where, MCS is the maximum common subgraph scheme, (c_A^i,c_B^i) and (c_A^j,c_B^j) are the matching feature points of the two groups in this scheme, $\|c_A^i-c_A^j\|$ is the distance between feature point i and feature point j in map A, $\|c_B^i-c_B^j\|$ is the distance between feature point i and feature point j in map B, D is the error threshold. The detailed steps are shown in Algorithm 1.

Algorithm 1 Maximum Common Subgraph Algorithm based on Backtracking

Input: F_A, F_B
Output: M

1. i=1;
2. $M=\emptyset$;
3. Check if the i-th line has any unaccessed feature points,
4. **if** success **then**
5. Check compatibility, **if** success **then**
6. join M;
7. i=i+1;
8. **else if** i=i+1
9. **return** to step 3;
10. Check if you are accessing the last line, if success then
11. i=i-1;
12. **return** to step 3;
13. **when** i=0, **end**

In order to speed up the search speed, when there are five sets of matches in the matrix M, an initial transform matrix can be calculated based on the five sets of matches and their neighbors in the local map. The feature points in the map B are transformed into the coordinate system of the map A, and the feature points with close distances after the transformation are added to the matrix M and the matrix M is taken as the scheme of the current MCS.

According to the above steps, we get all the MCS schemes, and the most matching number of feature points in the MCS is taken as the optimal MCS scheme. If there is more than one MCS scheme with the most number of matching, calculate the transformation matrix and select the one with the most number of fusion cubes as the optimal MCS scheme.

D. *Map Fusion*

According to the matched feature points in the optimal maximum common subgraph scheme, the transformation matrix T_A^B can be calculated, and according to the transformation matrix, the fusion of the 3D grid maps can be realized.

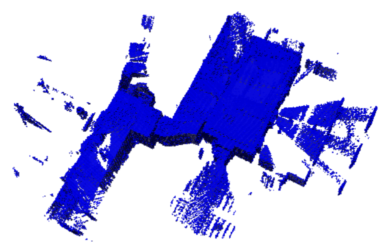
The fusion principle of the 3D grid map is: in the overlapping area of map A and map B, if the cube is occupied in map A or map B, the cube is still occupied after fusion. If the cube is free in both map A and map B, the cube is free after fusion, otherwise the merged cube is unknown.

V. EXPERIMENT RESULT

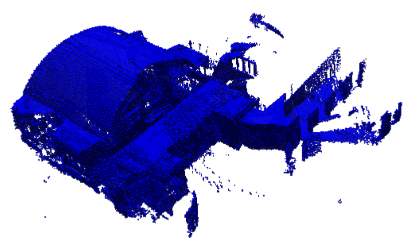
This section uses the methods proposed in this article to fuse local maps build by Google's public dataset.

A. *Creating local maps*

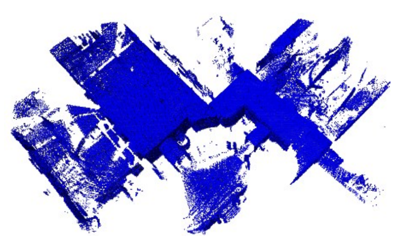
This experiment uses two Google public datasets, which are b3-2016-01-19-13-50-11.bag and b3-2016-02-09-13-17-39.bag[14]. For the two data bags, we use Google open source SLAM algorithm based on graph optimization, cartographer, to build local maps which shows in Fig. 7[15].



(a)



(b)



(c)

(d)

(e)

Fig. 7. Local maps: (a) and (b) represent two local maps of the data b3-2016-01-19-13-50-11.bag; (c), (d) and (e) represent three local maps of the data b3-2016-02-09-13-17-39.bag.

B. *Map Fusion*

After creating a series of local maps, we fuse these local maps to get the global maps. Taking the local maps (a) and (b) for example. First, we use the sliding window-based feature point extraction method proposed in this paper to extract corner points for local maps, as shown in Fig. 8. Second, we cluster the extracted feature points and only have a small number of feature points at the same corner, which greatly reduces the computational complexity of the algorithm. Fig. 9 shows the effect of feature point clustering. Third, wo use the Algorithm 1 to find the maximum common graph.

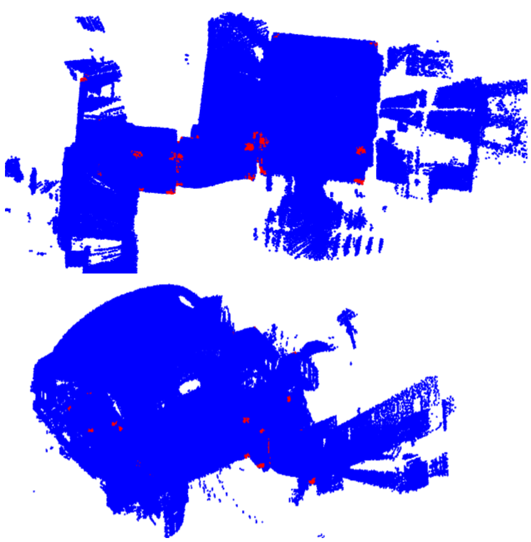


Fig. 8. Feature points of 3D grid map

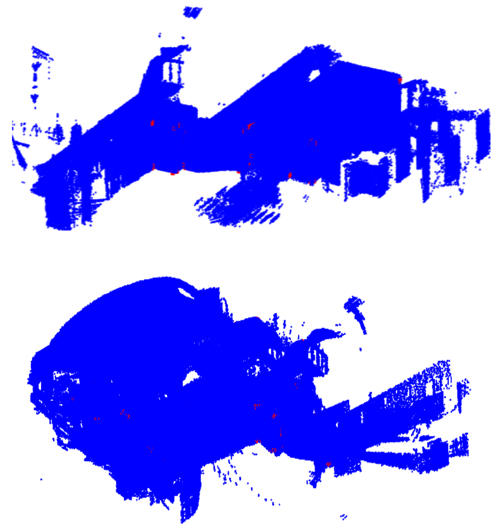


Fig. 9. Feature points clustering

Table 1 Analysis of the method proposed in this paper

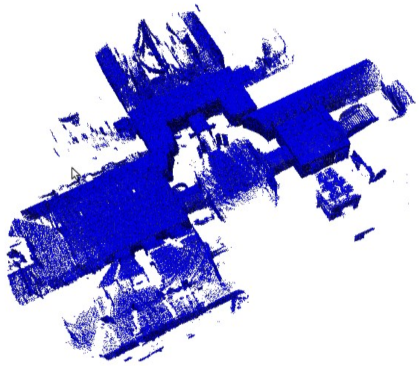
Local maps / Analysis	map1	map2
Number of corners in the map	22	24
Our method	23	22
Correct / Error	20/3	20/2
Number of feature points in this method	3054	4533
Number of feature points after clustering	82	88

It can be known from Table 1 that the method proposed in this paper is able to extract most of the feature points in the map, and only three feature points are not extracted. The number of feature points after clustering is greatly reduced, and there are only a few feature points in the same corner, which greatly reduces the computational complexity of the map fusion algorithm.

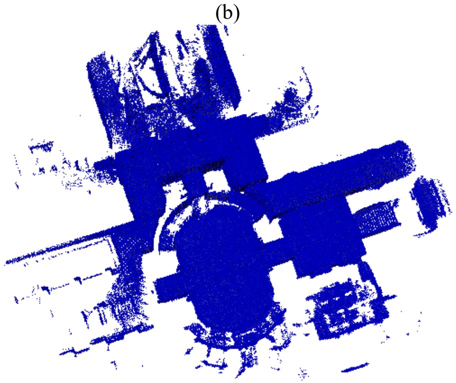
According to the maximum common subgraph, we get the correspondence between feature points from different maps. In order to calculate the exact transformation matrix, we extract the neighboring points of the feature points, add them to the corresponding relationship matrix, and calculate the exact transformation matrix according to the new matrix.

Finally, we calculate the transformation matrix from the correspondence of the feature points in the maximum common subgraph. Then, we do a map fusion to get a global map. As can be seen from the Fig 10, the method proposed in this paper can accurately realize the fusion of 3D grid maps.

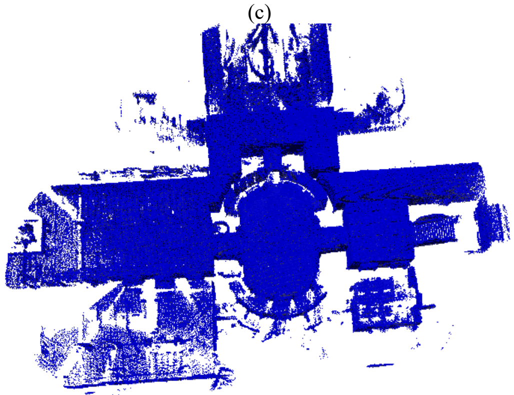
(a)



(b)



(c)



(d)

Fig. 10. Map fusion results: (a) represents the map fusion result of (a) and (b) in Fig. 7; (b) represents the map fusion result (c) and (d) in Fig. 7; (c) represents the map fusion result of the fusion of (d) and (e) in Fig. 7; (d) represents the result of the fusion of (c), (d) and (e) in Fig. 7.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a sliding window-based feature point extraction method for 3D grid maps and realize the fusion of 3D grid maps based on the maximum common subgraph. We created multiple local maps with overlap using the SLAM algorithm based on graph optimization. First, we create a window and move the window to determine whether the point is a flat point, an edge point or a corner point based on the change in the number of occupied cubes in the window. In order to eliminate the wrong feature points, we also performed a plane fitting. Second, we use the backtracking method to find the maximum common subgraph. In order to reduce the complexity of the algorithm, we clustered the feature points. Finally, according to the correspondence between feature points in different local maps in the maximum common subgraph, the transformation matrix is calculated to realize the fusion of 3D grid maps.

Since there is no initial match, the complexity of the algorithm increases greatly as the feature points grow. In future work, we need to reduce the number of feature points while ensuring that the feature points are accurate and the method of feature point matching for different local maps will be optimized to improve the matching speed.

REFERENCES

[1] G. Vallicrosa, P. Ridao, "H-SLAM: Rao-Blackwellized Particle Filter SLAM Using Hilbert Maps". *Sensor*,2018,18(5):1386.

[2] L. Carlone, R. Aragues, J. A.Castellanos and B. Bona, "A fast and accurate approximation for planar pose graph optimization". *The International Journal of Robotics Research,* 2014,33:965-987.

[3] A. Filatov, et al, "2D SLAM Quality Evaluation Methods". *The 21st Conference of Open Innovations Association*, 2017,120-126.

[4] L XuanPeng, Belaroussi and Rachid, "Semi-Dense 3D Semantic Mapping from Monocular SLAM", *Computer Vision and Pattern Recognition*,2016.

[5] D. Fox, J. Ko, K. Konolige and B. Limketkai, "Distributed Multirobot Exploration and Mapping". *Proceedings of the IEEE*,2006,94(7):1325-1339.

[6] Z. CongPeng, Wei. XueGuang, "Rectangle detection based on Harris corner". *Optics & Precision Engineering,* 2014,152:3479-3491.

[7] I. Sipiran and B. Bustos, "A robust 3D interest points detector based on Harris operator". *Eurographics Workshop on 3D Object Retrieval*. 2010.

[8] H. Dutagaci, C. PanCheung and A. Godil, "Evaluation of 3D interest point detection techniques via human-generated ground truth". *The visual Computer*,2012,28(9),901-917.

[9] A. Godil and A. Imdad Wagan, "Salient local 3D features for 3D shape retrieval". *Three-Dimensional Imaging, Interaction, and Measurement, 2011,vol.7864.*

[10] Y. Zhong, "Intrinsic shape signatures: a shape descriptor for 3D object recognition". *IEEE International Conference on Computer Vision Workshops*, 2009:689-696.

[11] R. Bogdan Rusu, et al, "Fast point feature histograms (FPFH) for 3D registration". *IEEE International Conference on Robotics and Automation*, 2009: 3212-3217.

[12] L. Truong-Hong, et al, "Octree-based region growing for point cloud segmentation". *ISPRS Journal of Photogrammetry and Remote Sensing,* 2015,104, 88-100.

[13] J. J Mcgregor, "Backtrack search algorithms and the maximal common subgraph problem". *Software: Practice and Experience*,12(1),23-34.

[14] https://google-cartographer-ros.readthedocs.io/en/latest/data.html#d-cartographer-backpack-deutsches-museum.

[15] W. Hess, D. Kohler, H. Rapp and D. Andor. "Real-Time Loop Closure in 2D LIDAR SLAM". *2016 IEEE International Conference on Robotic and Automation(ICRA).*2016,1271-1278.