An Improved ORB-SLAM Algorithm for Mobile Robots

Xinhua Liu[1,2], Linjun Chen[1,2], Xiaodan Wang[1,2], Hailan Kuang[1,2], Xiaolin Ma[1,2]

1.School of Information Engineering, Wuhan University of Technology, 430070, Wuhan , China

2.Key Laboratory of Fiber Optic Sensing Technology and Information Processing, Wuhan University of Technology, Ministry of Education, Wuhan, 430070, China

E-mail: liuxinhua@whut.edu.cn; chenlinjun@whut.edu.cn; wxxd@whut.edu.cn; kuanghailan@whut.edu.cn; maxiaolin0615@whut.edu.cn

Abstract—In the ORB-SLAM system for mobile robots, there are many problems such as large matching error, slow operation speed, low positioning accuracy and small map application scope. To solve these problems, this paper first adds the depth information based on saliency detection and preprocessing the scene images to improve the efficiency of the system. Then, the ORB feature extraction is carried out in the scale space with large isolation. The improved multi-probe LSH and PROSAC algorithm were used to optimize the matching strategy and it will improve the matching accuracy and efficiency. Aiming at the large number of error closed-loop in the closed-loop detection algorithm, an improved closed-loop detection algorithm based on the region of interest of scene image and the idea of hierarchical weighted matching is proposed to improve the accuracy and recall of closed loop detection. Finally, the final motion trajectory and 3D environmental map are obtained through the pose image optimization and the octree building. The experimental results show that the method can effectively improve the positioning accuracy and computation efficiency. At the same time, the octree map can not only greatly save a lot of storage space, but also meet the following requirements such as navigation, obstacle avoidance and interaction.

Keywords—Mobile Robots, SLAM, ORB, Loop closure detection

1 INTRODUCTION

With the increase of social demand and the development of related technologies, the research of mobile robots has become the main research direction of artificial intelligence, from the industrial application to the daily life of various service robots. In an unknown and complex environment, it is a very basic and key problem for mobile robots to build an incremental map and self-positioning based on its own sensing sensor. The problem is called Simultaneous Localization and Mapping (SLAM), and was first proposed by Smith, Self and Cheeseman in 1988 [1]. It is one of the most popular research directions in the field of robotics. With the development of computer vision technology, the vision of visual as an external sensor has become a hot spot in SLAM research field [2].

The ORB (Oriented FAST and Rotated BRIEF) features because of its extraction fast speed, easy to match advantages brought to the attention of the researchers [3]. In 2015, Mur-Artal R et al. [4] put forward that introducing ORB into SLAM algorithm is one of the relatively easy and perfect systems in modern SLAM system. However, there are still some shortcomings in the ORB-SLAM. First, the entire system needs to extract the ORB features for each frame captured, which will consume a lot of computing time. Secondly, it is easy to drop frames when the poses changes greatly especially only a rotation transformation. Finally the environment map constructed by the system is very sparse and cannot be used in other practical applications.

Based on the above analysis, this study adopted Kinect2.0 as ambient sensors, added scene image preprocessing stage to improve system operation efficiency. Then, the ORB feature extraction is carried out in the scale space with large isolation. The improved multi-probe LSH [5] and PROSAC [6] algorithm were used to optimize the matching strategy and improve the matching accuracy and efficiency. Aiming at the large number of error closed-loop in the closed-loop detection algorithm, an improved closed-loop detection algorithm based on the region of interest of scene image and the idea of hierarchical weighted matching is proposed. Finally, the final motion trajectory and environmental map are obtained through the pose image optimization and the octree.

2 SCENE IMAGE PREPROCESSING

2.1 Detection of region of interest

When the mobile robot collects images in the unknown environment and extracts and matches in the whole image, the image contains a large number of interfering objects, which not only takes a lot of time but also easily produces mismatch. If the image feature extraction and matching are limited in the area of interest, it can eliminate a lot of

This work is supported by National Nature Science Foundation under Grant NOs.61502361 and 61772088.

978-1-7281-0106-4/19/$31.00 ©2019 IEEE

interference information and improve the efficiency and precision of the algorithm. In this paper, a saliency detection algorithm is used to extract the interest area of images, and optimized it by the depth information, to eliminate the large amount of background interference and invalid information in the image. The algorithm flow chart is as figure 1.

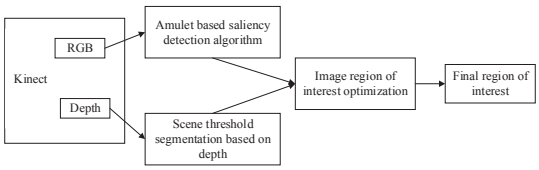


Fig. 1. Flow chart of scene image preprocessing.

In this paper, we use the generic aggregating multi-level convolutional feature framework, namely Amulet(Aggregating Multi-level Convolutional Features), which effectively utilizes multi-level features of FCNs(Fully convolutional neural networks) for salient object detection [7]. However, the interest area obtained by the Amulet also contains some interfering objects. Therefore, the paper proposes an image scene segmentation method that combines depth information to filter out the interference area in the scene. Image scene segmentation is a process of classifying pixels into different categories by setting certain threshold values [8]. The scene is segmented by the maximum depth threshold and minimum depth threshold. The gray value of the depth image pixel is $I(x,y)$, and the depth image is transformed into the segmented binary depth image by using formula (1).

$$I(x,y)=\begin{cases} 255 & d_{\min} \le d(x,y) \le d_{\max} \\ 0 & otherwise \end{cases} \quad (1)$$

After the scene image preprocessing, the visual odometer module based on the feature point method is studied. It is mainly discussed how to estimate the precise trajectory of the robot to provide better initial value for the back end based on the information of adjacent images.

3 VISUAL ODOMETER BASED ON FETUR

3.1 Improved ORB algorithm

ORB feature extraction is divided into two steps: oFAST [9] detection and rBRIEF [10] description, which have fast operation, rotation invariance, but lack of scale invariance. Therefore, the feature detection is carried out in the scale space, so that the acquired characteristics have the scale invariance.

First, we need to construct a scale space consisting of n octave layer li and n intra-octave layer bi, in which the octave layer l0 represents the original image. The first layer of intra-octave layer b0 is obtained by 1.5 times the lower sampling of the original image, and then l0 and b0 are sampled by layer by layer to get the octave layer li and intra-octave bi of each layer next. Because the working environment of the robot is not too large in the indoor environment, the scale of objects in the scene will not change too much when positioning and composing. For too far objects, it is considered as background filtering in the image domain preprocessing, so in order to reduce the time consumed in the construction scale Pyramid stage, the n=2 is selected, and the scale space consists of four image layers.

Then the image is detected by FAST corner detection in the scale space to obtain the rough image feature point location and scale, and the least square method is used to further precision. Finally, the real scale and coordinates of feature points are obtained by interpolation. Then we use the original algorithm's feature point orientation calculation method to calculate the direction of the feature points and generate the corresponding descriptors to match the features.

3.2 Optimized feature matching strategy

Because the ORB algorithm generates descriptors that are a series of binary strings, the matching of features is usually achieved by violence comparing Hamming distance. This method not only has a large amount of computation, but also produces a large number of mismatch pairs when the feature vector dimension is high. In this paper, the Multi-Probe LSH algorithm is used to optimize the matching strategy. The specific matching process is as follows:

- First allocate certain memory to store the hash value of the feature, and build the hash table as shown in Figure 2 according to the determined hash function (formula (2)).where hi is the value of the generated 32-bit new string digit. According to the classification of the hash value, the hash table formed by the hash function has 528 buckets.

$$H(x)=h_{31}\times 32+h_{30}\times 31+...+h_0\times 1 \quad (2)$$

- Then all the ORB features of the first frame are randomly sampled, 32 bits are formed into new strings, and to the corresponding bucket in the table by the hash function hash. To facilitate the subsequent Hamming distance calculation, we place the feature descriptors in the same bucket on a linked list.
- For each ORB feature point of a new frame, repeat step (2), and then use the set of disturbance sequence vectors defined to obtain a series of "adjacent" barrels to the hash bucket, which is directed by the current feature. If the feature descriptors have already existed in these barrels, calculated the hamming distance between these descriptors. Finding the minimum hamming distance in these distances, when the minimum hamming distance is less than threshold, the two feature points corresponding to the minimum hamming distance are matched successfully, and the ratio of the minimum distance and the next nearest neighbor is less than the threshold value (0.5), and it is considered to be the correct match. Otherwise, the new feature descriptor is added to the bucket.
- Repeat steps (3) until the robot is finished.

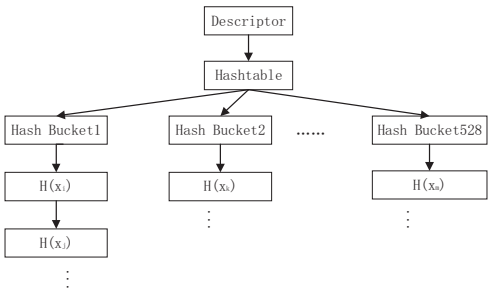


Fig. 2. Hashtable.

Then the PROSAC algorithm is used to eliminate mismatches and get more accurate matching results. The ICP algorithm is used to estimate motion.

4 loop closure detection

Using the visual odometer designed in the previous chapter, we get the motion trajectory and initial pose map. There will be a lot of cumulative error due to the work mode of the mileage, which leads to the failure to build the global consistent trajectory and map. So this chapter carries out closed loop detection and back-end optimization for the data obtained in the previous chapter.

4.1 Key frame extraction

When mobile robot is walking, the system will collect environmental information continuously and calculate the movement transformation between successive frames. However, for the two adjacent frames, it is likely that the same scene, so that most of the information between the two frames is repeated. If each frame is used for closed loop detection or pose graph construction, a large number of redundant phenomena will be caused, which not only wastes a lot of memory consumption of the system, but also increases the computational complexity of the system. Therefore, before the closed loop detection, we need to select the key frames for scene image frames.

First, each key frame needs to ensure that the extracted feature points are evenly distributed and sufficient. Then, in order to perform accurate motion estimation, it is necessary to guarantee that at least 5% matching points can be obtained between the current frame and the previous key frame. At the same time, in order to ensure the completeness and redundancy of the image information, the matching between the current frame and the previous key frame cannot be more than 90%. In the end, when the new key frame is to be inserted, the size of the motion between the previous key frames is suitable, neither too large nor too small. The following formula is used as a measure of the motion size.

$$\min\_norm \le \left\|\Delta t\right\| + \min\left(2\pi - \left\|r\right\|, \left\|r\right\|\right) \le \max\_norm \quad (3)$$

where Δt represents displacement vectors between adjacent frames, R indicates rotation angle. Therefore, formula (3) uses the norm of displacement and rotation to describe the size of robot motion.

4.2 Loop closure detection

Closed loop detection is essentially a data association process. By calculating whether the similarity between the current scene and the previous scene is less than a certain threshold, we can detect whether there is a closed loop, and then use the closed loop to correct the deviation of the trajectory. In this paper, the closed loop detection based on Bag-of-Visual-Words (BoVW) [11] is selected, but the problem of closed loop error will occur because of the visual confusion and the ambiguity of the words. Therefore, an improved closed loop detection algorithm based on the scene image is proposed to improve the ability to distinguish the visual confusion, and the hierarchical weighted matching idea is combined to enhance the similarity calculation to improve the recall rate.

First, the ORB features in the region of interest are expressed as visual words. Describe the region of interest through the words in the visual dictionary with a k branch and a d depth, and describe the scene by the region of interest. In order to search for similar images quickly, a reverse index from word to scene is established to achieve fast retrieval of pre match.

Then the candidate's closed loop is determined by calculating the region of interest matching score in each pre-matching scene. The visual words of all tree nodes are treated equally in the original closed loop detection algorithm. It ignores the fact that different levels of words have different representational abilities and the fact that images have different similarities at different levels. Therefore, this paper combines the hierarchical weighted matching idea to improve the original TF-IDF score matching criteria for calculation. The specific calculation process is as follows:

First, calculate the TF-IDF entropy of the current frame region of interest X at the i(i∈{1,2,⋯,kl})node of the l(l ∈{0,1,⋯,d}) level.

$$w_i^l(X) = \frac{n_i}{n}\log\left(\frac{N}{N_i}\right) \quad (4)$$

Where ni represents the number of times a word appears in an image. n represents the number of times all words appear in the image. Ni indicating the number of times the word appears in the dictionary and N represents the number of all feature words in the dictionary.

According to [12], using the inverse proportion function, we get the similarity scores of X and Y at a single node O_i^l.

$$S_i^l(X,Y) = \frac{1}{\left|w_i^l(X) - w_i^l(Y)\right| + 1} \quad (5)$$

Then the similarity score in the l layer is as followings.

$$S^l(X,Y) = \sum_{i=1}^{k^l} S_i^l(X,Y) = \sum_{i=1}^{k^l} \frac{1}{\left|w_i^l(X) - w_i^l(Y)\right| + 1} \quad (6)$$

Since the visual dictionary is hierarchical clustering, it means that the upper layer contains the similarity of the next layer of space. In order to avoid repetition of

computational similarity, the final similarity matching score is made by using the similarity increment between each layer as the weight.

$$S(X,Y)=\sum_{l=1}^{L}\beta_l\Delta S^l(X,Y)=\sum_{l=1}^{L}\frac{1}{k^{L-l}}(S^l(X,Y)-S^{l+1}(X,Y)) \quad (7)$$

where β_l represents the matching intensity coefficient of the L level of the visual dictionary tree. If the score is less than the given global similarity threshold, the candidate region of interest in the current frame and the matching scene of the region of interest are selected as the candidate closed loop. However, in the candidate's closed loop, there is often a false positive closed loop because of the ambiguity of the word, so finally the final closed loop confirmation of the time continuity is carried out.

After completing the closed loop detection, we need further optimize the pose map. The optimization of pose map is essentially the process of minimizing the nonlinear error function. In this paper, the G2O [13] framework is used to solve the optimization problem in order to obtain the global optimal trajectory of the robot position and the global consistent motion. Finally, the map is constructed through this trajectory. In order to save storage space and get more practical value map, this paper uses octree [14] to construct the map.

5 EXPERIMENTS

The experimental platform is Ubuntu 14.04 operating system, computer CPU is 1.4 GHz Intel i5, memory 4 GB, and then ROS indigo system is built. The experimental data set uses four RGB-D packets of fr1_desk, fr1_room, fr2_desk and fr2_large_no_loop in the TUM RGB-D Benchmark datum [15], which was published by Technical University of Munich in 2012.

5.1 Scene image preprocessing

In this paper, based on the results of Amulet saliency detection, as shown in Fig. 3(a), the binarized depth image obtained by threshold segmentation of the scene image using depth values is shown in Fig. 4(b). The image region of interest is shown in Figure 4(c).

（a）The results of Amulet

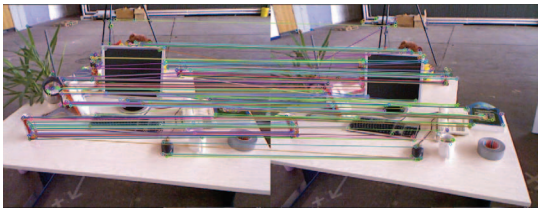
（b）Depth threshold segmentation

（c）Final image region of interest

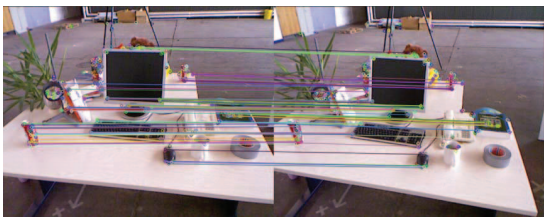
Fig. 3. Preprocessing result.

5.2 Visual odometer

In this paper, we compare the feature matching between the violence matching method and the optimized matching strategy. The experimental results are shown in Figure 4.



（a）The violence matching



（b）The optimized matching strategy

Fig. 4. Contrast result of different matching methods.

It can be clearly seen from Figure 5 (a) that there are obviously incorrect matching points after the violent match method. The matching accuracy is only 31.8%, and the time consumed is 73.807ms. Although it is possible to eliminate incorrect matches by changing the threshold size, only when the threshold is small enough can we completely remove the mismatch. However, when the threshold is too small, the number of effective feature pairs is too few to be applied to subsequent operations. As can be seen from Figure 5 (b), the overall matching line is roughly parallel after the optimized matching strategy and the PROSAC algorithm eliminate the external points. The matching accuracy is 93.7%, and the time consumed is 40.474ms.

Then, in order to further improve the efficiency of feature detection and matching, we first use the preprocessing of scene images mentioned in the previous chapter to detect ROI. Then we use the improved ORB algorithm and the optimized matching strategy, and PROSAC to mismatch the feature extraction and matching in the region of interest. The result of matching is shown in Figure 5, and 79 pairs of matching points are obtained. The time taken is 29.647ms. Compared with the feature extraction and matching of the whole image, the accuracy of the matching is improved while the efficiency is increased by about 37%.

Fig. 5. Matching result based on region of interest.

Using the above matching results, we use ICP algorithm for motion estimation and get Table 1. As shown in Table 1, the convergence rate of the ICP algorithm is accelerated due to the elimination of the initial value iteration process. The shorter iteration time and less iteration times have achieved more accurate calculation results, which shows that the improved ICP algorithm has better real-time performance and can meet the real-time requirements of the SLAM system.

Table 1. The Results on ICP

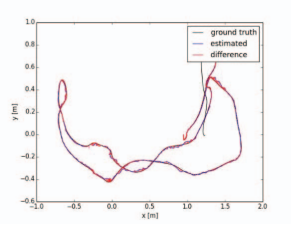
Group	Number of iterations		Time of iterations	
	Original ICP	*Improved ICP*	*Original ICP*	*Improved ICP*
1	23	9	33.876	6.077
2	30	12	40.014	9.919
3	42	17	53.634	15.918
4	26	11	35.302	7.926
5	35	12	42.055	11.116

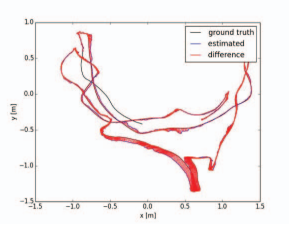
5.3 Back-end experiment

The improved algorithm is used to estimate the motion trajectories of the Kinect camera from the key frames obtained from the four sets of data sets. Compared with the ground true provided by the dataset, the schematic diagram is shown in Figure 6. It can be roughly seen from the diagram that the estimated motion trajectory is basically the same as the whole motion trend of the real trajectory, and its relative deviation is not too large and is basically close.

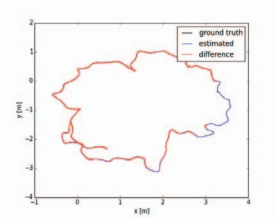
Following two aspects of the accuracy and real-time performance of the algorithm, the improved algorithm's RMSE and algorithm operation time are compared with the original ORB-SLAM and the literature [16] and the literature [17]. The comparison results are shown in Table 2 and table 3.

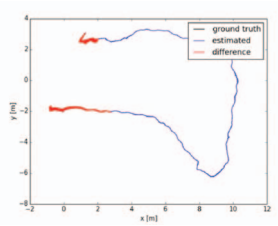
As shown in Table 2, in terms of accuracy, the average RMSE of this algorithm is reduced by 50% compared with the original ORB-SLAM algorithm, reduced by 12% compared with the literature [16], reduced by 78% compared with the literature [17]. This is mainly due to the optimization and improvement of the two parts of visual odometer and closed loop detection. As shown in Table 3, in terms of real-time performance, we can see that the algorithm runs faster under lower processor performance. Compared with the original ORB-SLAM algorithm, it is increased by 58%, which is 14% higher than that in the literature [16], which is 52% higher than that in the literature [17]. This is mainly due to the preprocessing stage, key frame extraction and octree map addition.





（a）fr1_desk （b）fr1_room





（c）fr2_desk （a）fr2_large_no_loop

Fig. 6. Matching result based on region of interest.

Table 2. The Results on RMSE（m）

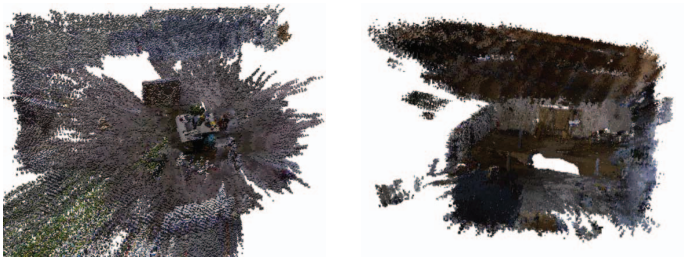
Dataset	*Ours*	*[16]*	*[17]*	*ORB-SLAM*
fr1_desk	0.015	0.015	0.026	0.019
fr1_room	0.047	0.065	0.087	0.054
fr2_desk	0.008	0.008	0.057	0.011
fr2_largr_no_loop	0.158	0.17	0.86	0.351

Table 3. The Results on Time（s）

Dataset	*Ours*	*[16]*	*[17]*	*ORB-SLAM*
fr1_desk	40.6	42.1	35.9	74.3
fr1_room	89.2	93.9	94.3	207.2
fr2_desk	147.7	200	390.3	373.3
fr2_largr_no_loop	205.9	224.1	478.6	471.6

Then, the octree map is constructed by using the results of the trajectory obtained above, as shown in Figure 7. From Figure 7, we can see that objects in space have been reconstructed very well, and the outline of objects in the environment can be clearly identified. Moreover, the octree map constructed in the absence of color information memory is only KB, saving a lot of memory space. Even if we add color information to it, the map is much smaller than the point cloud map. By adjusting the resolution of the map, it can be used to adapt to different applications. It can also easily query the occupying probability of any point, and design and implement the navigation, obstacle avoidance and path planning methods in the map, so that the map is more practical.

（a）fr1_desk （b）fr1_room



（c）fr2_desk （a）fr2_large_no_loop

Fig. 7. Octomap.

6 CONCLUSION

Aiming at the large amount of operation, poor matching precision and limited application range of ORB-SLAM system, this paper added the scene image preprocessing stage to improve the system efficiency. Then, feature detection is carried out in the scale space with large isolation, and the ORB algorithm is improved to have scale invariance. And improved Multi-Probe LSH algorithm combined with PROSAC algorithm to optimize matching strategy to achieve more accurate matching. In view of the visual confusion and the ambiguity of the closed loop detection, a closed loop detection based on the region of interest in the scene image is proposed to improve the accuracy and recall of the closed loop, and the octree is used to build the graph. The experimental results show that the method can effectively improve the positioning accuracy and efficiency, and the generated OctoMap maps can not only save a lot of storage space, but also meet the other application requirements.

REFERENCES

[1] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in robotics[C]// IEEE International Conference on Robotics and Automation. Proceedings. IEEE, 2003:435-461.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Pi S, He B, Zhang S, et al. Stereo visual SLAM system in underwater environment[C]// Oceans. IEEE, 2014:1-5.

[3] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.

[4] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.

[5] Ying H, Ting H, Shui Z, et al. Improved locality-sensitive hashing method for the approximate nearest neighbor problem[J]. Chinese Physics B, 2014, 23(8): 217-225.

[6] Chum O, Matas J. Matching with PROSAC - progressive sample consensus[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2005:220-226.

[7] Zhang P, Wang D, Lu H, et al. Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2017:202-211.

[8] Zhu S, Xia X, Zhang Q, et al. An Image Segmentation Algorithm in Image Processing Based on Threshold Segmentation[C]// International IEEE Conference on Signal-Image Technologies and Internet-Based System. IEEE, 2008:673-678.

[9] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]// European Conference on Computer Vision. Springer-Verlag, 2006:430-443.

[10] Calonder M, Lepetit V, Strecha C, et al. BRIEF: Binary Robust Independent Elementary Features[C]// European Conference on Computer Vision. 2010:778-792.

[11] Kato H, Harada T. Image Reconstruction from Bag-of-Visual-Words[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:955-962.

[12] Yongfeng L I, Zhang G, Feng W, et al. An Improved Loop Closure Detection Algorithm Based on Historical Model Set[J]. Robot, 2015(May).

[13] Kümmerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization[C]// IEEE International Conference on Robotics and Automation. IEEE, 2011:3607-3613.

[14] Hornung A, Kai M W, Bennewitz M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots, 2013, 34(3):189-206.

[15] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2012:573-580.

[16] Lv Q, Lin H, Wang G, et al. ORB-SLAM-based tracing and 3D reconstruction for robot using Kinect 2.0[C]// Control and Decision Conference. IEEE, 2017:3319-3324.

[17] Endres F, Hess J, Sturm J, et al. 3-D Mapping With an RGB-D Camera[J]. IEEE Transactions on Robotics, 2017, 30(1):177-187.