MATEC Web of Conferences **160**, 07001 (2018) https://doi.org/10.1051/matecconf/201816007001
EECR 2018

A Micro SLAM System Based on ORB for RGB-D Cameras

Fei Wang[1, 2] , Xiaogang Ruan[1, 2], Pengfei Dong[1, 2] and OUATTARA SIE[1, 2]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*

Abstract. In this paper, a micro SLAM system based on ORB features for RGB-D cameras has been proposed. With only a RGB-D sensor, this method can be applied in small environment for localization and mapping. Furthermore, the task of 3D reconstruction can also be accomplished by using the approach. The pose graph based on Bundle Adjustment is adopted for reducing the estimation error. In order to further speed up computing to meet the requirement of real-time, we have proposed the piecewise optimization strategy. The approach is evaluated on public benchmark datasets. Compared with several state-of-the-art scheme, this method has proven to work well in these environments.

1 Introduction

Simultaneous Localization and Mapping(SLAM) is the key technology for autonomous navigation of robot. It acquires information about itself and environment by sensors such as laser sensors, cameras and inertial measurement units. Visual SLAM has become a hot research in the field of computer vision and robotics in recent years. Compared with other sensors, RGB-D camera can obtain more abundant environmental information, and help to achieve accurate and robust positioning, and has been widely used recently.

Visual SLAM mainly estimates the motion of the camera through the matching of image features, which is commonly called visual odometry [1]. The commonly used visual features are SIFT[2], SURF[3], ORB[4], etc. The commonly used method is to obtain the feature matching between two adjacent frames, then compute the estimated camera motion by PNP, ICP[5][6] and some other algorithms. The estimation error of camera motion will accumulate over time, leading to more and more inaccurate estimation of motion [7]. So it is difficult to build a globally consistent map. Therefore, after the estimation, it is necessary to optimize the pose and environment map.

In this paper, we build a SLAM system based on the ORB feature for small environments. Figure 1 shows the different environment map using tum data. ORB has the advantages of rotation and scale invariance, while the extraction time is short, which meets the real-time requirements. We use a simple matching method to select the key frame. We use the method of matching the frame to the environment model to create the map and use the bundle adjustment to optimize the camera pose and feature point position. It usually takes a lot of time to optimize the entire map because of the large number of feature points. We proposed a piecewise optimization strategy. For the relocalization problem, we select a number of previous key frames to match the current frame.

The evaluation was conducted on a publicly RGB-D data set. The result shows that our method can achieve higher accuracy and robustness in the estimation of motion and map points positions.

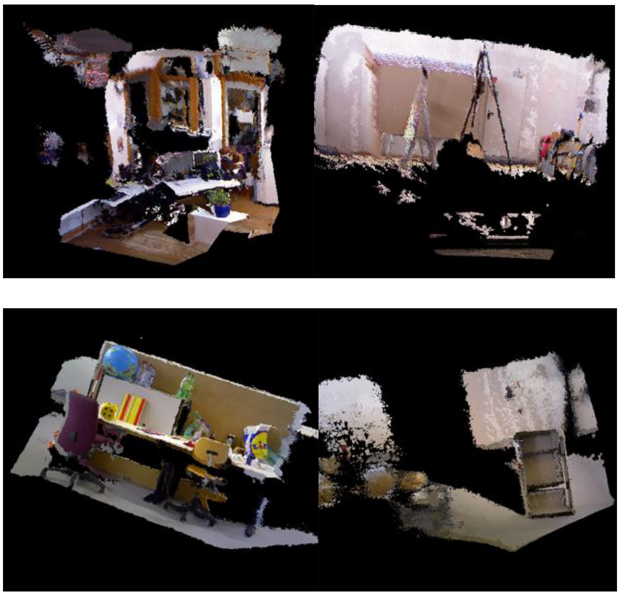


Figure 1. The building map of different environment.

2 Related work

The image feature is a set of information related to the computational task, and the main features used in visual slam are SIFT, SURF, ORB [2,3,4], etc.. SIFT features arediscriminative, because their descriptors are represented by high dimensional vectors, and have rotation invariance, scale invariance, and robustness to noise and illumination variations[8]. The SIFT feature is used in visual SLAM, but the time complexity of SIFT feature is too high because of its high vector dimension. SURF features have scale invariance, rotation invariance, and the speed of the algorithm relative to the SIFT feature is increased by 3 to 7 times. In [9,10,11], SURF was used for the slam system, compared with the SIFT characteristics, the time complexity is reduced. The Euclidean distance between two feature vectors is usually calculated when matching the SIFT and SURF features of two images, which is used as the similarity measure of feature points.

ORB feature is the combination of FAST [12] feature detection operator and BRIEF [13] descriptor, and some improvements are made on the basis of it. The extraction efficiency of ORB is 100 times the SIFT feature, 10 times the SURF feature, which is the biggist advantage of ORB. In [6,14,15,16,17], ORB was used in SLAM system which greatly accelerate the speed of the algorithm. ORB feature matching is based on the Hamming distance of BRIEF binary descriptor as similarity measurement.

Frame to frame alignment will cause large cumulative floating, so there is always error in the pose estimation process. In order to reduce the error caused by frame alignment, the SLAM [6,18,19,20] method based on key frames is proposed. HENRY P, et al. [20] propose that a new key frame is created when the common feature points of the two images are below a certain threshold. KERL C, et al [6] propose a method to select key frames based on entropy, computing similarity entropy ratio for each frame, if the value is less than a predefined threshold, the previous frame is selected as the new key frame, and insert it to the map, which greatly reduces the drift.

In monocular slam, the motion of the camera can be obtained by calculating the fundamental matrix, the essential matrix or the homography matrix, so as to realize initializing. 3D positions of feature points will be obtained by triangulation, and then the feature matching can be done by PNP. In RGB-D images, we can obtain the depth information of the feature points directly through the depth image, and then estimate the camera motion using iterative closest point (ICP) [21].

Error accumulation is inevitable in the process of 3D reconstruction, and closed loop is an important means to eliminate it. The key point of loop closing is to detect whether the camera passed the same location. Williams B, et al. compared the loop closing methods [22] and draw the conclusion that the matching performance of image to image is better than that of map to map and image to map. Eda E D and Drummond T W propose a unified approach to relocation and closed-loop detection, which continually searches for visited locations using dictionary methods based on 16-dimensional SIFT features [23]. Document [24, 25] uses the dictionary method based on the SURF descriptor for loop-closing detection, and SURF feature extraction takes about 400 ms. In the literature [14], the position recognition is carried out by the dictionary method based on the ORB feature. The method can identify the position from different views because of the invariance of rotation and scale.

It is very important for a robot working in complex and dynamic environment to quickly generate a 3-D map, which will play a key role in localization and path planning. Our SLAM system overview is shown in Figure 2. Loop-closing constraint will be added to the map for correction after successful loop closing detection. The optimization of map can be solved by bundle adjustment [26], which is to optimize the all the camera pose and feature points for the same time. However it is difficult to achieve real-time optimization because of the high calculation complexity. An executable method is pose graph optimization. The camera pose is represented as the vertex in the graph, and the error between two frame transformation is the edge. Errors are distributed along the graph, namely evenly distributed over all pose on the graph. The optimization technique of pose graph is used to realize the effective correction of rotation, translation and scale floating [27]. Essential graph is constructed after successful loop detection, and then using pose graph method to executing optimizing[6]. All the key frame are included in the essential graph, while there are less contraint between edges compare to covisibility graph[28]. The overview of our SLAM system is shown in Figure 2.

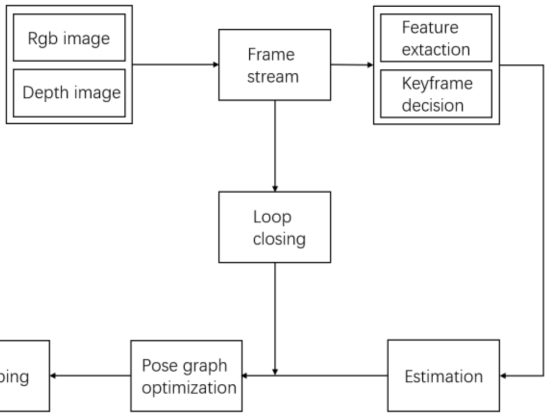


Figure 2. The SLAM system overview in the paper. The pose graph optimized by bundle adjustment using g2o.

3 Approach

3.1 Motion model

The movement of the camera in the three-dimensional space belongs to the rigid body motion and can be decomposed into the rotation R and the translation t. Where R satisfies the following conditions:

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathrm{R}^{3\times 3} \,|\, \mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathrm{T}, \det(\mathbf{R}) = 1\} \qquad (1)$$

The rotation matrix R and translation t will not satisfy the linear relationship when they are using to express continuous rigid motion. Thus, homogeneous coordinates and transformation matrices are introduced:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2}$$

$$\mathrm{SE}(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathrm{R}^{4\times4} \mid \mathbf{R} \in \mathrm{SO}(3), \mathrm{t} \in \mathrm{R}^3 \right\} \tag{3}$$

The corresponding homogenous coordinate of point in 3-dimension space is defined as p=(X, Y, Z, 1). The coordinates of point p after transformation T are represented as p'=Tp.

Since the rigid motion in three-dimensional space has only six degrees of freedom, while there are 12 values in T matrices. So T is a redundant expression. Therefore, we use Lie algebra se(3) ξ corresponding to Lie group SE(3) to express the motion of camera. ξ is a six dimensional vector, and the transformation matrix T can be obtained by the exponential mapping of Lie algebra. The backend optimization problem can be transformed into an unconstrained one using Lie algebra to express the transformation.

3.2 Camera model

The internal matrix of the camera is expressed as follows:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where f_x, f_y are the focal length and c_x, c_y are the primary coordinate of the imaging plane.

For the spatial point p=(X, Y, Z, 1), the corresponding pixel coordinates are denoted as p_{uv}=(u, v) , and their corresponding relations are as follows:

$$\begin{cases} u = f_x \dfrac{X}{Z} + c_x \\ v = f_y \dfrac{Y}{Z} + c_y \end{cases} \tag{5}$$

3.3 Feature extraction and matching

We hope that the slam system can track and construct the map in real time, so we select the ORB feature points with high efficiency and invariance of rotation and scale. Feature points generally include key points and descriptors. It is necessary to compute the derivative of pixels when using Harris corners, which will increase the complexity of detection algorithm. The improved fast corner is adopted in ORB. We take two steps to extract the key points. First of all, we extract key points as usual. And then the thresholds are then reduced appropriately in areas with less key points, where there are often lacking in texture features or weak contrast. Thus ensuring that the key points can be evenly distributed over the image. After that, the description of the corresponding key point is computed, using an improved BRIEF descriptor, usually represented by a 128-bit or 256-bit binaries. The detected feature points are shown in Figure 4.

Feature matching is a key step in visual slam, which solves the problem of data association in slam, that is, to determine the relationship between the current landmarks and the front markers. The accurate matching of the descriptors between image to image or image to map can reduce the heavy burden for the subsequent pose estimation and optimization.

However, due to the local characteristics of image features, the situation of mis-matching is widespread and has not been effectively solved for a long time. We can find some mismatched points in Figure 4. The main methods used to remove the wrong match include the feature point distance check and geometric relationship check. The former sets a distance threshold. And the matching point whose distance is longer than the threshold will be regarded as mismatching point. This is because the ORB feature points have various forms of invariance, so the distance of correctly matched feature points is naturally relatively close. The latter randomly selects n pairs, calculates the transformation matrix between these chosen images, and then checks with the remaining pairs. Repeat the process above until finding the best result. The results of removing mismatches are shown in Figure 5.

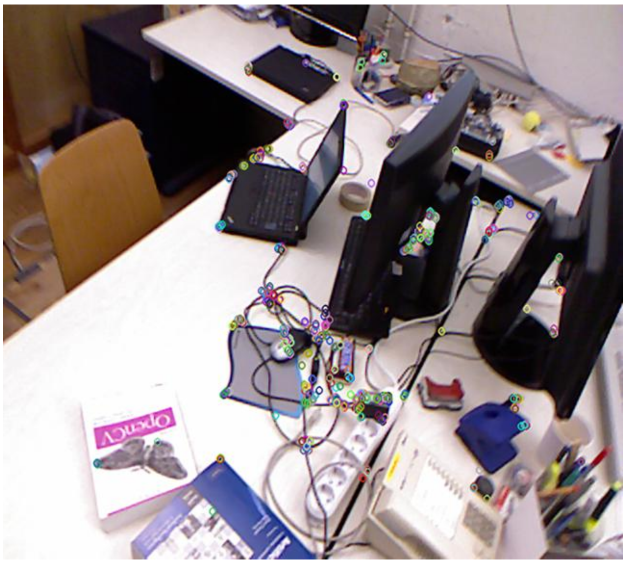


Figure 3. Feature points detected.

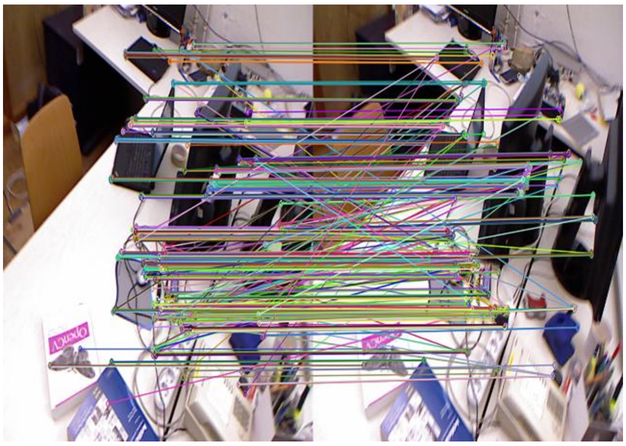


Figure 4. Feature matching results.

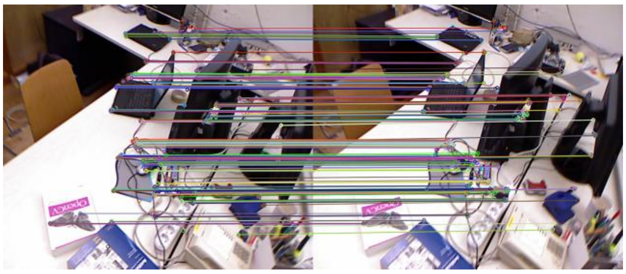


Figure 5. Feature matching results after removing the mismatched key points.

3.4 Pose estimation

We can get the 3D coordinates of the feature points directly, after extracting the feature points. Camera motion can be estimated using ICP algorithm. ICP is called iterative closest point algorithm [17], which is a common method for aligning point clouds of two frames. The algorithm has two key aspects: one is to find the corresponding point pair between the two-frame point cloud, and the other is to compute the transformation matrix that minimizes the distance of the two-frame point cloud according to these corresponding points. The algorithm can accurately calculate the transformation matrix of point clouds of two frame accurately. However, the algorithm is sensitive to initial values. When the initial transformation is not selected properly, the algorithm may fall into local minima. Moreover, when the point cloud is a dense point cloud, because of the large amount of data and the long running time of the algorithm, it can not meet the requirements of real-time. In this paper, the transformation matrix T_t between the current frame F_t and the environmental model M_t is calculated using ICP algorithm, because both of them are sparse feature points, the data volume is small, and therefore with the fast speed. And we assume that the camera moves at constant speed. Between t-2 and t-1, the amount of change in camera movement is $\Delta T = T_{t-1} / T_{t-2}$. In order to calculate the T_t , the initial value of the ICP algorithm is set as $T_{init} = T_{t-1}\Delta T$. The transformation matrix of the camera relative to the global coordinate system is obtained by continuous iteration.

3.5 Bundle adjustment

The transformation matrix obtained by the ICP algorithm is only the initial estimate of the pose. The error of the transformation matrix obtained by ICP algorithm will be larger, when the number of feature points match the current frame and model is small. In addition, there is also possible for ICP algorithm being trapped into local minima. At the same time, with the gradual accumulation of the camera motion error, the posture estimation is becoming more and more inaccurate. Therefore,T is not the optimal pose estimation.

We need to optimize the pose to solve the estimation deviation and accumulation error. In this paper, the bundle adjustment method is used to optimize the pose and map. Ideally, the projection point which is reprojected from the spatial by the transformation matrix obtained by ICP should be coincident with the obsevered point. However, due to the existence of errors in the transformation matrix, it is impossible to completely coincide. Therefore, we optimize the transformation pose by constructing a least squares problem to minimize the re-projection error. Assuming that the 3-D coordinates of the observation point is *P,* the observed pixel coordinates is P_{uv} and the position posture of the camera is expressed using the Lie algebra ξ . The least squares problem can be constructed as follows:

$$\xi^* = \arg\min_{\xi} \sum \rho(\left\| P_{uv} - \frac{1}{s} K \exp(\xi) P \right\|^2) \qquad (6)$$

where ρ is a kernel function. In the paper, the Hubel kernel is used.

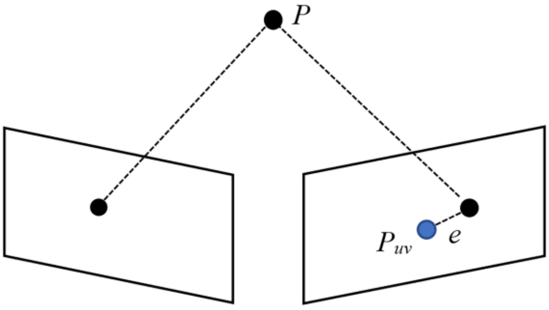


Figure 6. Minimzing the error of reprojection.

The sketch map of reprojection error is shown in Figure 6. We use piecewise optimized strategies in our experiments. Each time *K* frames will be optimized, that is, the first time from the 1-k frame are optimized and the second time the k+1 -2k frame are optimized, which can further accelerate the optimization speed. And finally performing global optimization.

3.6 Loop closing

Loop closing detection is an important method to eliminate accumulation error. Bag of wors method has been widely used in loop detection because of its effectiveness. The bag of words refers to the technique of converting the contents of an image into a digital vector using a visual dictionary tree. Firstly, the feature of the training image set is extracted, and then the feature descriptor space is clustered by K-means.

We use a *k* tree to express the dictionary. Similar to hierarchical clustering, it is a direct extension of k-means. Suppose we have *N* feature points, and we will build a tree with the depth of *d* and divided into k branch each time. The steps are as follows :

1. In the root node, all samples are clustered into *k* class by k-means, obtaining the first layer.
2. For each node of the previous layer, the samples belonging to the node are clustered into *k* class, getting the next layer.
3. And so on, finally get the leaf layer. The leaf layer is called Words.

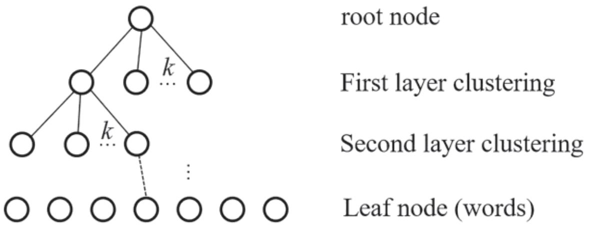


Figure 7. Sketch map of *k* branch tree dictionary.

We will get a *k*-branch, *d*-depth tree that can accommodate k^d words. The k branch tree are shown in Figure 7. When searching for a word corresponding to the given words, we can get the needed one by comparing it with the clustering center of each intermediate node, ensuing the logarithmic level of search efficiency, speeding up the detection.

4 Evaluation

In order to verify the effectiveness of the algorithm and to ensure consistent and comparable experiments in this paper, we use the publicly available Tum data set provided by Technical University of Munich. It contains indoor benchmark data set acquired by RGB-D vision sensor. The motion trajectory of the camera is captured by a high precision motion capture system. The experiment runs on a laptop with Intel Core i7-6700hq CPU.

The constraint between frames can be added through loop detection, further improving the robustness of the system, and the tracking and mapping quality of the whole SLAM is also improved. In this algorithm, loop closing detection is performed after the key frame decision, and the results of the ICP solution are added to the graph optimization as shown in Figure 8.

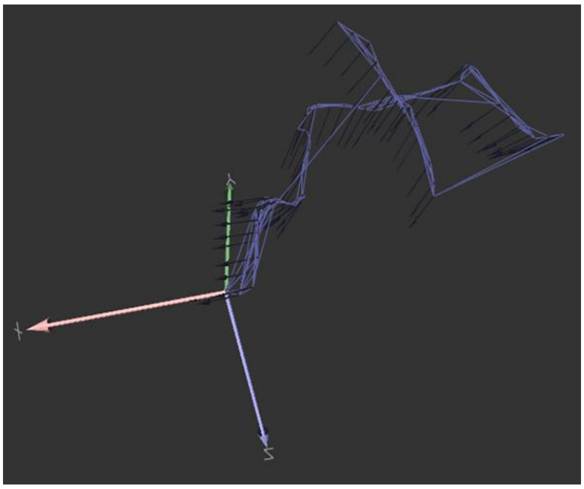
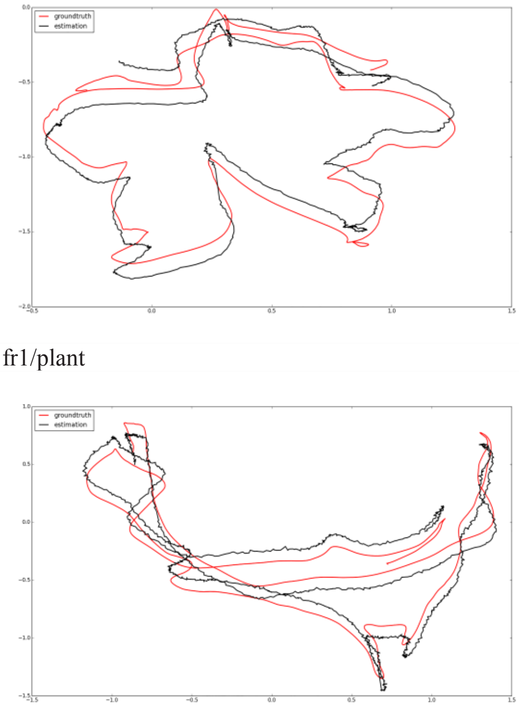


Figure 8. Constraint produced by loop closing.

In order to obtain the change of robot pose, we use the method of ICP to estimate the adjacent pose of the robot by using the 3D-3D feature point constraint, and optimize the position and posture with the help of graph optimization. The results of the experiment are shown in Figure 9 and Figure 10. The experimental trajectory of the robot and the reference trajectory of the data set under different data sets are given, for making the trajectory more intuitive. Where the red line represents the tracjectory computed by the ground truth file, the black line represents the estimated one of the robot by our method. The error of the robot posture can be effectively evaluated by comparing the root mean square error(RMSE). It can be seen that this method can realize the trajectory tracking of robot well. The root mean square error between the real pose and the estimation of the robot's trajectory in the 3-dimensional space is recorded in Table 1.



(a) fr1/plant

(b) fr1/room

Figure 9. The comparison between groundtruth and estimation.

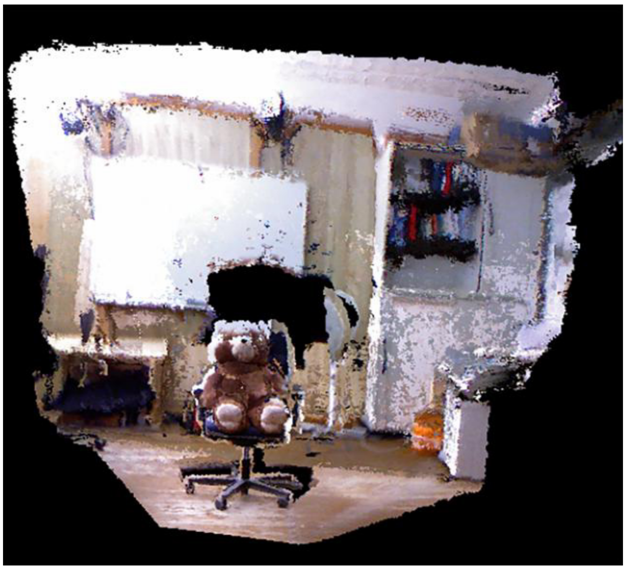


Figure 10. The constructed map.

Table 1. Comparison of RMSE (*m*)

Data set	RGB-D SLAM	DVO SLAM	ORB SLAM2	Our method
fr1/desk	0.026	0.021	0.016	0.023
fr1/desk2	0.048	0.046	0.022	0.042
fr1/room	0.087	0.043	0.047	0.06

5 Conclusion

In this paper, we proposed a method of localization and map building for RGB-D cameras. ORB feature points are adopted for image matching between adjacent frame. And then we use ICP algorithm to estimate the camera pose and 3-D coordinate of feature points. Because there is always drift in pose estimation and map building. It is necessry to reduce the drift at the same time. We take two steps to optimize the location and map. First，we use Bag of words to detect loop closure. Then, we build the pose graph to optimize the pose estimation obtained by ICP.We have proposed a piecewise optimization strategy, which can speed up the optimizing process. And finally, a global optimization is executed. The trajectory of camera and environment map can be obtained at last.

For the next step, we plan to use the method of machine learning to detect loop closing such as convolutional neural networks(CNN), which has been proven to be very effective in the field of image recognition. Further more, we are planning to bulild the environment map with higher quality to realize robot navigation.

References

1. D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004
2. D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, vol. 60, no 2, pp.91-110, 2004.
3. H. Bay, T. Tuytelaars, and L. Van Gool, “ Surf : Speeded up robust features”, in Computer Vision-ECCV 2006, pp.404-417, Springer, 2006.
4. E. Rublee, V. Rabaud, K. Konolige, and G. Braski, “ Orb : an efficient alternative to sift or surf”, in *2011 IEEE International Conference on Computer Vision(ICCV)*, pp. 2564-2571, IEEE, 2011.
5. R. A. Newcombe, S Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P.Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion : Real-time dense surface mapping and tracking”, in *IEEE Intl. Symp. On Mixed and Augmented Reality(ISMAR)*, 2011.
6. Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.
7. C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for RGB-D cameras”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
8. Ali, A.M., Nordin, M.J., SIFT based monocular SLAM with multi-clouds features for indoor navigation, *TENCON 2010 - 2010 IEEE Region 10 Conference*, 2010, pp. 2326–2331
9. Z. Zhang, Y. Huang, C. Li, Y. Kang, Monocular vision simultaneous localization and mapping using SURF, in *World Congress on Intelligent Control and Automation*, 2008, pp. 1651–1656.
10. Y. Ye, The research of SLAM monocular vision based on the improved surf feature, *International Conference on Computational Intelligence and Communication Net-works*, Hong Kong, China, 2014:344–348.
11. Y. T. Wang, Y. C. Feng, Data association and map management for robot SLAM using local invariant features. *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2013.
12. E. Rosten and T. Drummond. Machine learning for highspeed corner detection. In *European Conference on Computer Vision*, volume I, 2006.
13. M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, 2010.
14. R. Mur-Artal and J. D. Tardos, Fast relocalisation and loop closing in keyframe-based SLAM, in IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 2564-2571.
15. X. Fen and W. Zhen, “An embedded visual SLAM algorithm based on Kinect and ORB features,” in *34th Chinese Control Conference*, July 2015, pp. 6026–6031.
16. G. Xin, X T. Zhang, X.Wang, and J. Song, “A RGBD SLAM algorithm combining ORB with PROSAC for indoor mobile robot”, International Conference Computer Science and Network Technology, Harbin, pp. 71-74, 2015.
17. L. Jun, T. Pan, K. Tseng, J. Pan, Design of a monocular simultaneous localisation and mapping system with ORB feature. In *Proceedings of 2013 IEEE International Conference Multimedia and Expo*, San Jose, CA, USA, 15–19 July 2013; pp. 1–4.
18. G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’07)*, Nara, Japan, November 2007. 1
19. G. Klein and D. Murray. Improving the agility of keyframebasedSLAM. In *European Conference on Computer Vision*, 2008.
20. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments. In *the 12th International Symposium on Experimental Robotics (ISER)*, December 2010.
21. P. J. Best and N. D. Mckay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 14, no. 2,1992.
22. B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. D. Tardos, A comparison of loop closing techniques in monocular SLAM, *Robotics and*

Autonomous Systems, vol. 57, no.12, pp. 1188-1197,2009.

23. E. D. Eade and T. W. Drummond. Unified loop closing and recovery for real time monocular SLAM. In *Proc. BMVC*, 2008.
24. D. Galvez-Lopez and J. D. Tardos. Real-time loop detection with bags of binary words. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 51-58, September 2011.
25. M. Cummins and P. Newman, “Appearance-only SLAM at large scale with FAB-MAP 2.0,” The International Journal of Robotics Research, vol. 30, no. 9, pp. 1100–1123, 2011.
26. B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle Adjustment - a Modern Synthesis, *LNCS*, Springer Verlag, 1883:298-375, 2000.
27. H. Strasdat, J. M. M. Montiel, and A. Davison, “Scale drift-aware largescale monocular SLAM,” in *Proc. of Robotics: Science and Systems(RSS)*, 2010.
28. H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, Double window optimisation for constant time visual SLAM, in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011, pp. 2352–2359.