Advances in Computer Science Research, volume 94
International Conference on Big Data, Electronics and Communication Engineering (BDECE 2019)

Monocular SLAM Feature Point Optimization Based on ORB

Run Tan, Minling Zhu* and Yuefan Xu
Beijing Information Science and Technology University, Beijing, China
*Corresponding author

***Abstract*—Simultaneous localization and mapping (SLAM) of mobile robots is a hotspot in the field of computer vision and intelligent robots. Through experiments on the monocular SLAM system indoors, the existence of problems was found that the map is sparse and feature point matches are mistaken in some environments. Therefore, it is proposed to optimize the feature point acquisition and match of the ORB algorithm, filter the feature points according to the actual hardware parameters, and perform image pre-processing on the mismatch problem generated after using the Hamming distance, and then obtain the improved RANSAC algorithm, more precise match points will be got. It makes the image matching in different environments possessing better robustness, and also improves the sparse problem of the construction of point cloud.**

Keywords—ORB-SLAM; monocular SLAM; hamming distance; homography Matrix; RANSAC

I. INTRODUCTION

Simultaneous localization and mapping of mobile robot is one of the hot issues in the field of computer vision and mobile robot. It has a wide range of applications, from the most mature sweeping robots to the gradual landing of driverless vehicles.

Thus, SLAM plays an important role. Given the differences in sensors acting on SLAM, there are two main areas of research. One is laser SLAM with lidar and the other is camera-mounted vision (VSLAM) discussed in this paper. Visual SLAM has the advantage of low cost and sensor less detection distance limit, so VSLAM has a good market prospect.

Mur-Artal proposed a relatively complete ORB-SLAM algorithm in 2015(Oriented FAST and Rotated BRIEF, referring to extracting a characteristic of rotational invariance) promoted the rapid development of the field of visual SLAM. The improved ORB-SLAM2 on top of this algorithm is a current mainstream system that can be further applied to modes such as binoculars and RGB-D. ORB-SLAM2 innovatively uses three parallel threads, including real-time tracking of feature points, local mapping, and loop-back detection. This paper briefly introduces the feature point screening and matching links of ORB algorithm in tracking. According to the actual operation effect, it is proposed to improve the problems those selecting feature points and eliminating mismatches, and make the system have better practical application results.

II. ORB ALGORITHM DESCRIPTION

A. Feature Point Filtration

ORB-based front-end has always been the mainstream method in visual mileage designs, with more efficient and fast advantages over SUFT algorithms and SIFT algorithms[1].The ORB algorithm is divided into two parts, FAST corner detection and BRIEF descriptors[2]. Among FAST corner detection main detected the local pixel rescale change softer difference, the basic idea is in a certain pixel domain, selecting a pixel p, assuming that its brightness is I_p, Setting a threshold t, 16 pixels on a circle with a radius of 3 with a pixel p as the center[3]. As shown in Figure 1, if there are continuous n-points larger or both smaller than $I_p \pm t$, selecting this point as corner (n normally take 12, that is FAST-12).

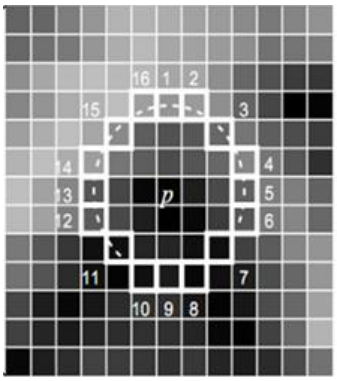


FIGURE I. FAST FEATURE POINTS

Because FAST corner points do not have directional information and scales, so the ORB algorithm adds a description of scale and rotation to the feature points[4]. In order to achieve feature scale invariance, the image pyramid is constructed, a fixed scale drop sample for the original frame (layer 0) in turn to accumulate 8 pictures[5], and each picture is detected corner points, as shown in Figure 2:

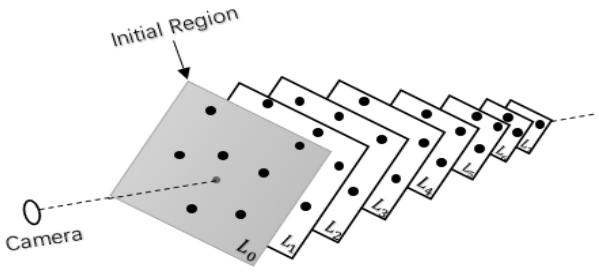


FIGURE II. IMAGE FEATURE EXTRACTION PYRAMID

In order to achieve the rotation of the feature, the Gray Centroid method is used. The specific steps are:

a) In a small image block V, define the moment of the image block as formula (1):

$$m_{pq} = \sum_{x,y \in V} x^p y^q I(x,y) \quad p,q = \{0,1\} \tag{1}$$

b) The centroid of the image block is calculated by Moment as a formula (2):

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2}$$

c) Connecting the geometric center O and centroid C of the image block to get its directional vector $\overrightarrow{OC}$, the direction of the feature point is defined as a formula (3):

$$\theta = \arctan(m_{01}/m_{10}) \tag{3}$$

Through the above steps, the FAST corner points have a description of scale and rotation[6].

After extracting the FAST corner points, the described subs are calculated using the improved BRIEF algorithm for each point. BRIEF algorithm calculates a binary string[7]. Each represents the size relationship of two pixels near the corner point. And if it is less than the relationship, 1 is placed and the other relationship is set at 0. In order to increase its noise resistance, Gaussian smoothing of the image needs to be pre-processed[8]. Michael Calonder's paper tested five treatment methods, with GII (both *p and q* Gaussian distribution in line with (0, $S^2/25$)) enjoying a small advantage over the others in most cases[9]. On this basis, the current rBRIEF (rotation-aware BRIEF) algorithm improves the correlation between rotation invariance and descriptor, so that ORB still performs well under the transformation of zoom, pan and rotation.

B. *Feature Point Matching*

After obtaining the corner point and the descriptor, it is necessary to match the same characteristic point between the two frames to achieve data association[10]. The main problem is to extract feature points $x_t^m, m = 1,2,\ldots,M$ in frame I_t, Extracting feature points $x_{t+1}^n, n = 1,2,\ldots N$ in the successor frame I_{t+1}, and how to find the correspondence between two collection elements. For the BRIEF descriptor, the current mainstream uses the Hamming distance[11] (the number of individuals in the two binary strings) as a measure; the procedure is as measured in Figure 3:

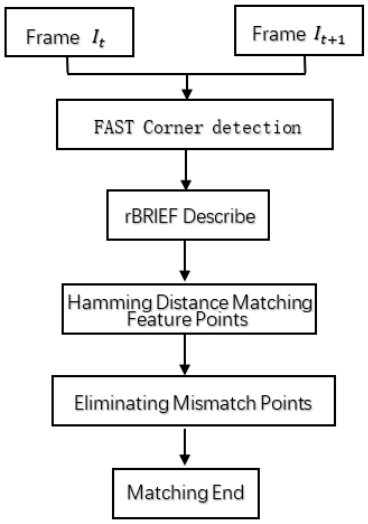


FIGURE III. FEATURE MATCHING PROCESS

When the number of feature points is large, the traditional Brute-Force Matcher method consumes more time and cannot meet the real-time requirements, and it could be better to use the fast approximation (FLANN) algorithm for matching[12]. The basis of the characteristic point matching filter is that the hamming distance is less than twice the minimum distance, which will produce many wrong match points, hence the elimination of the wrong match point is required, and more random sample consistency (RANSAC) method is used[13], so the improvement of the matching feature point algorithm is proposed[14].

III. ANALYSIS OF PROBLEMS IN PRACTICAL APPLICATIONS

A. *Problems with Sparse Drawings and Feature Point Matching Errors*

In fact, experiments with single-eye ORB-SLAM in indoor environments will reveal that when you go straight on a path with more textures[15], you encounter a large turn and then walk in a straight line which will cause a problem of path distortion, as shown in Figure 4:

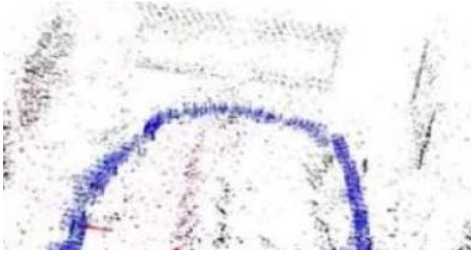


FIGURE IV. POST-TURN

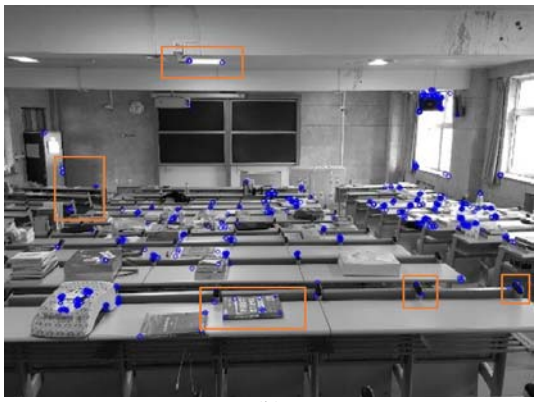
This is also a pure rotation problem that is common in single SLAM[16]. This is due to the limitation of the calculation formula solved by the monolith SLAM through similar transformation space Sim (3), we can only use the algorithm to minimize the error[17], and cannot completely solve. It can be found through multiple experiments. The angle orientation of the camera is still normal after a large turn, and the displacement distance is incorrect, which is characterized by a small.

At the same time, it will be found that after the turn of the feature points will be significantly less[18], at this time the number of mis-matched points will also increase, resulting in the final construction of the point cloud map is very sparse, cannot achieve a good practical use of the effect[19]. Therefore, the feature point extraction and matching part of the ORB algorithm are chosen to be improved, and the reject algorithm of mis-matching is optimized accordingly.

B. *Feature Point Improvement*

The camera resolution has certain limitations on the selection of FAST corner points, such as when the resolution is small. Due to the number of near points is less thus solid will be add-ed with more far points, the corner points are more concentrated in the middle[20], as shown in the box in Figure 5(a); It is also more dispersed, as shown in the box below 5 (b). At the same time, in order to improve the problem of sparse feature points after the drawing, the number of feature points can be increased appropriately[21], if the resolution of the frame from 512 *384 to 752 *480. You can select about 1000 feature points, if the re-solution is increased. Such as TIKIT data set, you can extract 1500 feature points. Second, you can set the value of n to a small point when FAST extracts the corner point. FAST-9, for example, can also play a role in dispersing feature points, but at the same time, it also caused the problem of the number of mismatched points increasing.

(a)



(b)

FIGURE V. (A) FEATURE EXTRACTION WHEN RESOLUTION IS LOW (B) FEATURE EXTRACTION AT HIGH RESOLUTION

C. *Mismatched Pre-processing*

You can eliminate absolute mismatch before you eliminate the wrong match point. Because Fast-12 is taken to increase feature point dispersion, it can be pre-processed in the following way: Suppose there is a matching point collection C in frame I_t:

$$C = \{ c_\alpha | c_\alpha \in C, \alpha = 1,2,\cdots \}$$

Suppose there is a matching point set D in the successor frame I_{t+1}:

$$D = \{ d_\alpha | d_\alpha \in D, \alpha = 1,2,\cdots \}$$

Where C and D should be an one-to-one corresponding relationship, the sub-label is α. The main reason for using Hamming code to generate mismatch is a similar orientation problem[22], which causes the wrong matching point differ in the relative position between two frames and the correct matching point between the two frames, as shown in Figure6:

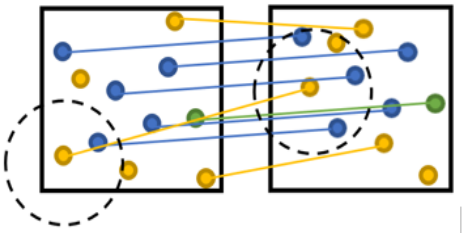


FIGURE VI. MISMATCH POINT RELATIVE POSITION

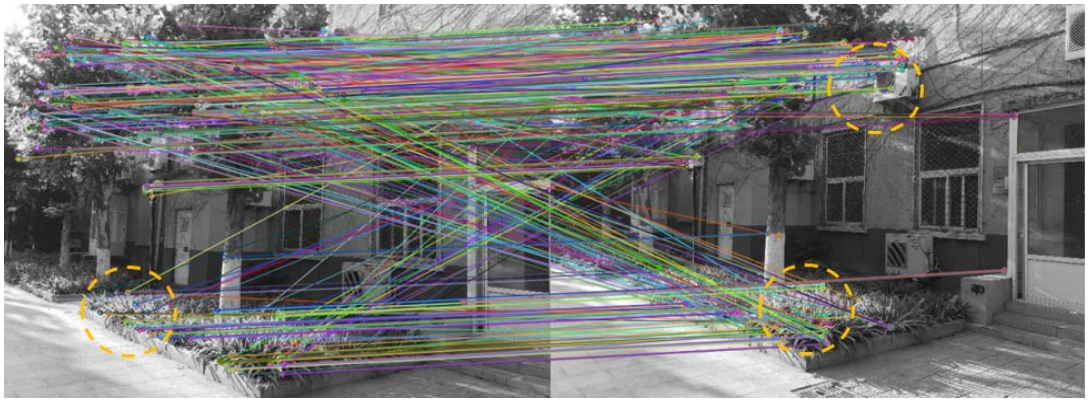
For a pair of matching points, $<c_\alpha, d_\alpha>$ the matching points within their range of r are the number of points C'_α and D'_α, such as formulas (4) and (5):

$$C'_\alpha = |X|, \quad X = \{x | |x - c_\alpha| \leq r, x \in C\} \qquad (4)$$

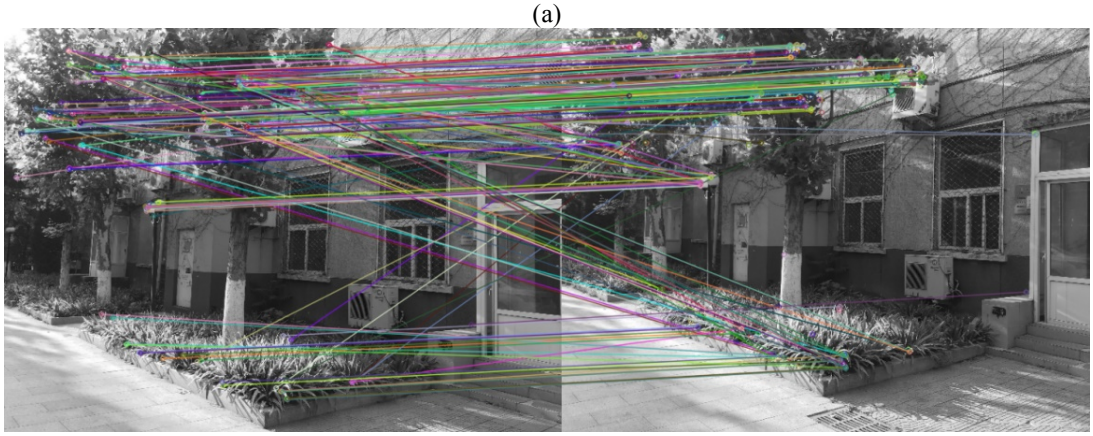
$$D_\alpha^t = |Y| \ , \ Y = \{y \| y - d_\alpha| \leq r, y \in D\} \quad (5)$$

For the $<c_\alpha, d_\alpha>$ match point, set acceptable thresholds. If it appears $|C_\alpha^t - D_\alpha^t| \leq t$, which illustrate that the match is correct, vice versa the wrong match point is discarded, and a new set of match points is formed. It can be considered that the match is correct, and vice versa is that the wrong match point is discard-ed[23], and a new set of match points S is formed.

Figure 7(a) and Figure 7(b) are two frames of pictures taken outdoors[24]. Figure 7(a) is a matching picture that has not been pre-processed, and Figure 7(b) is a result of using pre-processing[25]. It can be seen that pre-treatment for outdoor using can remove some scattered mis-matching points. Similarly, Figures 8(a) and 8(b) are the results of experiments conducted indoors, respectively, and it can be seen that the pre-treatment effect is reduced when there are more textures in the room[26].

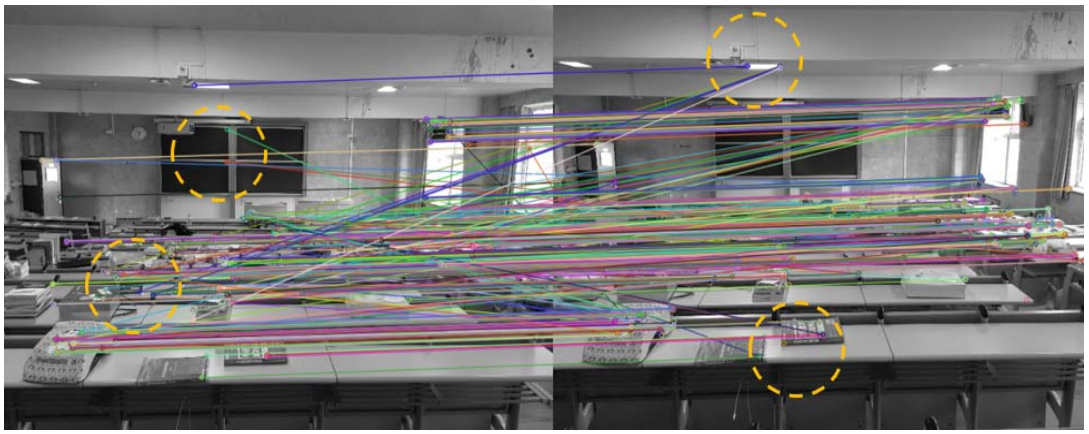


(a)



(b)

FIGURE VII. (A)OUTDOOR UNUSED PRE-PROCESSED MATCHES (B)MATCH AFTER USE OF PRE-PROCESSING OUTDOOR USE



(a)

(b)

FIGURE VIII. (A)UNUSED PRE-PROCESSED MATCHES IN THE ROOM (B) MATCH AFTER USE OF PRE-TREATMENT IN THE ROOM

D. RANSAC Improvement

Homography H describes the mapping relationship between two planes. And if the feature points between the two frames fall on a smoother plane, such as a wall or floor, you can choose to use H for motion estimation. At the same time, H is more suitable for pure rotation[27]. When pure rotation occurs, the freedom of the basic matrix F decreases. The noise interference will be particularly obvious. In order to avoid causing degradation, it is usually necessary to estimate the basic matrix F and single matrix H, choosing the lower projection error as the final motion estimation matrix. Homography H describes the mapping relationship between two planes, and if the feature points between the two frames fall on a smoother plane, such as a wall or floor, you can choose to use H for motion estimation[28]. At the same time, H is more suitable for pure rotation, when pure rotation occurs, the freedom of the basic matrix F decreases, the noise interference will be particularly obvious, in order to avoid causing degradation, it is usually necessary to estimate the basic matrix F and single matrix H at the same time, choose the lower projection error as the final motion estimation matrix.

The RANSAC algorithm is mainly to find an optimal homology matrix H, the scale of his 3*3. In the actual processing, it is usually multiplied by a non-zero factor to make h_9=1 (when taking non-zero), So H has 8 unknown parameters to be optimized, needing 8 linear equations to solve. A pair of matching points can list two equations, then the homography matrix with a degree of freedom of 8 can be calculated from four pairs of matching points. As formula (6):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{6}$$

Where (x, y) represents the point in the current frame, and (x', y') represents the point in the successor frame, and the transformation relationship is $I_{t+1} = HI_t$ the sorted formula (7) and the formula (8):

$$\alpha = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + 1} \tag{7}$$

$$\beta = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + 1} \tag{8}$$

The main steps of the RANSAC algorithm are as follow:

a) From the matching point set S, 4 matching pairs of non-common lines are randomly drawn out, and H', as model M, is calculated.

b) Calculate the projection error of all matching point pairs and model M in S, and if the error is less than the threshold, adding the inner point set A.

c) To determine whether the number of elements in the current inner point set A is greater than the optimal inner point set A', and if so, updating the number of iterations k.

d) Exit if the number of iterations is k or k', otherwise the number of iterations plus 1, repeating (1) - (3) to continue the iteration.

Where the k is constantly updated at no greater than the maximum number of iterations, not fixed values, then k's calculations such as formulas (9):

$$k' = \frac{\log(1-p)}{\log(1-w^m)} \tag{9}$$

(Where p is confidence, generally take 0.995. w is the proportion of the inner point. m is the minimum number of samples required to calculate the model and it is equal to 4)

Projection error is the cost function. If this model is the optimal model M, the minimum cost function is formula (10):

$$\sum_{i=1}^{n}((x'\alpha)^2 + (y'\beta)^2) \quad (10)$$

Because there are more feature points in the actual detection, iterative calculation time is long for the formula (10), so it needs to be optimized[29]. You can filter out in the matching point setting S to fit the matching feature point-to-set S', which consists of a matching pair with a steady projection error in a continuous n iteration, n can be determined by the specific environment or the proportion of the number of feature points selected. At the same time, in the next random selection of 4 pairs of match points can be avoided to avoid the selection of the matching point pairs in the collection of S', thus reducing the process of repeated lying iterations.

Because the matching points in S' have linear stability characteristics for the model M they make up, it can be said that their contributions to the model are certain. Next, if you choose to the wrong match point to calculate H', you will find that the cost function and consider the cost function built by the elements in the Collection of S' is a large gap, choose a threshold t, if the threshold t, you can judge that the wrong match point accounted for a larger, can be discarded. This reduces the number of iterative rounds required within the acceptable range in order to reduce the time-consuming algorithm.

IV. CONCLUSION

In this paper, the monocular ORB-SLAM is introduced, specifying the key steps of ORB feature extraction and matching, and by thinking about the feature point collection problems found in the actual operation, the ORB feature extraction and matching algorithm is improved accordingly. Through the different hardware selection, the number of extracted feature points is increased accordingly, and the steps of image pre-processing are added after the coarse match, and the optimization idea of a longer iteration time in practical application is put forward in the improved RANSAC algorithm. The corresponding improvements made in this paper in practical application solve some of the problems. According to actual situations, there is a room for perfection.

ACKNOWLEDGMENTS

This work was supported by 2018 Talent Development Quality Enhancement Project of BISTU (5101923400); The Level Project of Science Research in Colleges and University-Beijing Information Science and Technology University (5211910957); National Natural Science Foundation of China (51675055).

REFERENCES

[1] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2564-2571.

[2] Linyan Cui,Fei Wen. A monocular ORB-SLAM in dynamic environments[J]. Journal of Physics: Conference Series,2019,1168(5).

[3] Calonder M , Lepetit V , Strecha C , et al. BRIEF: Binary Robust Independent Elementary Features[C] Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV. 2010.

[4] Patrik Schmuck,Margarita Chli. CCM‐SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams[J]. Journal of Field Robotics,2019,36(4).

[5] RUBLEE E, RABAUD V, KONOLIGE K, et al.ORB:an efficient alternative to SIFT or SURF[C].2011 International Conference on Computer Vision, New York: IEEE, 2011.

[6] Yong-ge WEN. A Modified Image Registration Algorithm Based on Color Invariance and RANSAC[A]. Science and Engineering Research Center.Proceedings of 2017 2nd International Conference on Test, Measurement and Computational Method(TMCM 2017)[C].Science and Engineering Research Center:Science and Engineering Research Center,2017:6.

[7] Tran Ngoc-Trung,Le Tan Dang-Khoa,Doan Anh-Dzung,Do Thanh-Toan,Bui Tuan-Anh,Tan Mengxuan,Cheung Ngai-Man. On-device Scalable Image-based Localization via Prioritized Cascade Search and Fast One-Many RANSAC.[J]. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society,2018.

[8] S. D. Ma, A Self-Calibratin Technique for Active Vision System, IEEE Trans Robotics and Automation, Feb, 1996.

[9] Montemerlo M, Thrun S. Fast SLAM: a factored solution to the simultaneous localization and mapping problem. Proc of the Eighteenth National Conf on Artificial Intelligence, Alberta: AAAI, 2002: 593-598.

[10] David Silver, Deryck Morales, Loannis Pekleitis, et al. Arc carving: Obtaining accurate, low latency maps from ultrasonic range sensors. In Proceedings of the IEEE International Conference on Robotics and Auto-mation(ICRA), 2001, 2: 1554-1561.

[11] Lowe D G. Object Recognition from Local Scale-Invariant Features[C]. The Proceedings of the Seventh IEEE International Conference, 1999: 1150-1157.

[12] H. C. Longue-Hoggins. A computer algorithm for reconstructing a scene from two projections, Nature, 1981, 293: 133-135.

[13] Kaess M, Ranganathan A, Dellaert F. iSAM: Fast incremental smoothing and mapping with efficient data association. 2007 IEEE International Con-ference on Robotics and Automation, ICRA'07, Apr 10-14 2007. Rome, Italy: IEEE Press,Piscataway, NJ 08855-1331, United States, 2007. 1670-1677.

[14] Smith R, Cheeseman P. Estimating uncertain spatial relationships in robotics[C].Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence, 1986(4): 435-461.

[15] Engelhard N, Endres F, Hess J, et al. Real-time 3D visual SLAM with a hand-held RGB-D camera. In: Proc of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum. Vasteras, 2011.

[16] Wu D, Mendel J M. Enhanced karnik-mendel algorithms[J]. IEEE Transactions on Fuzzy Systems, 2009, 17(4):923-934.

[17] Castellanos J A, Montiel J, Neira J, et al. a probabilistic framework for simutaneous localization and map building. Journal of Transactions on Robotics and Automation, 1999, 15(5): 948-952

[18] Milella A, Siegwart R. Stereo-based ego-motion estimation using pixel tracking and iterative closest point[C]. Computer Vision Systems, IEEE International Conference on, 2006: 21-21.

[19] Tan W, Liu H, Dong Z. Robust monocular SLAM in dynamic environments }C}// IEEE International Symposium on Mixed and Augmented Reality. IEEE, 2013:209-218.

[20] Lim H, Lim J, Kim H J. Real-time 6-DOF monocular visual SLAM in a large-scale environment}C}// IEEE International Conference on Robotics and Automation. IEEE, 2014:1532-1539.

[21] NIST D. An Efficient Solution to the Five-Point Relative Pose Problem}J}. IEEE Transactions on Pattern Analysis&Machine Intelligence, 2004, 26(6): 756.

[22] CIVERA J, DAMSON A J, MONTIEL J M M. Inverse Depth Parametrization for Monocular SLAM}J}.IEEE Transactions on Robotics, 2008, 24(5): 932-945.

[23] Mur-Artal R, Tardos J D. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM}C}// Robotics: Science and Systems. 2015.

[24] GALUEZ-L PEZ D, TARDOS J D. Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28(5): 1188-1197.

[25] HORN B K P. Closed-form solution of absolute orientation using unit quaternions[J]. Journal of the Optical Society of America A, 1987, 4(4): 629-642.

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

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

[28] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces}C}// IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007:1-10.

[29] Song S, Chandraker M, Guest C C. Parallel, real-time monocular visual odometry}J}. 2013:4698-4705.