An improved feature matching ORB-SLAM algorithm

An improved feature matching ORB-SLAM algorithm

Qiang Li[a] Jia Kang*, Yangxi Wang, Xiaofang Cao

Lanzhou Jiaotong University School of Electronics and Information Engineering, Lanzhou, 730070

Corresponding author mailbox: 690516002@qq.com

[a]dshen@mail.lzjtu.cn

Abstract: To solve the problems of low accuracy and poor real-time performance in traditional mobile robot vision simultaneous positioning and map construction (SLAM), the original algorithm was improved. First, the ORB features of adjacent images are extracted, and the PROSAC algorithm is used to achieve feature point matching. At the same time, the PROSAC algorithm is improved and optimized, and the execution time of the optimized PROSAC algorithm is significantly reduced; finally, based on the graph optimization model, a global Bundle Adjustment algorithm based on the largest common-view weight frame is proposed to achieve dense and sparse map creation. The algorithm is verified by Tum data set, and the experiment shows that the root mean square error has dropped significantly. The results on the data set effectively prove the effectiveness of the algorithm in this paper.

1.INTRODUCTION

Simultaneous Location And Mapping(SLAM) is the use of sensors carried by the robot to achieve the positioning and map construction of the robot in a strange environment, and is an important technology for the robot to achieve complete autonomy. According to the different sensors carried by the robot, it is divided into laser slam and visual slam. Compared with the visual slam, the laser slam has an early start, the research is relatively mature, and the laser sensor is relatively expensive. At present, many researchers are devoted to the research of the visual slam, and the research of the visual slam has made good progress. In 2007, Davison et al [1] proposed the first monocular visual SLAM system based on Extended Kalman Filter (EKF). This system is the first real-time visual SLAM system, but the application scenario is very narrow. Sparse feature points are easy to lose; Zhang Yi and so on [2] proposed an iterative extended Kalman filter method on this basis, through multiple iterative updates to obtain a higher posterior probability and improve positioning accuracy. However, compared with the filter method, SLAM based on graph optimization has advantages in reducing the amount of calculation and the applicability of the actual environment, so the method based on graph optimization is now commonly used. In 2007, Klein et al. [3] proposed PTAM (Parallel Tracking And Mapping), which realized the parallelization of tracking and mapping, and was the first scheme to use graph optimization as a back-end, such as obvious flaws that easily lose track; Endres et al [4] proposed SLAM (Simultaneous Localization And Mapping for RGB-D camera, RGBD-SLAM) based on depth camera in 2014, which integrates image features, closed-loop detection, point cloud and other technologies to achieve three-dimensional mapping of indoor scenes , But because the scale-invariant feature transform (Scale Invariant Feature Transform, SIFT) feature extraction calculation is large, it is easy to lose the road sign; Long Chao et al [5] proposed a three-dimensional visual SLAM based on Kinect and a visual dictionary to improve positioning accuracy; in 2015, Mur-Artal et al [6] based on the PTAM (Parallel Tracking And Mapping) architecture, added map initialization and closed-loop detection. Function, optimized the selection of key frames, etc., and achieved good results in processing speed and map accuracy; Li Haifeng et al [7] proposed a visual feature method based on point, line, and surface fusion in 2017, which is superior to the point feature method in positioning accuracy. However, in traditional mobile robot vision simultaneous positioning and map construction (SLAM), there are still problems of low accuracy and poor real-time performance, which need to be further solved by researchers.

To solve the problems of low accuracy and poor real-time performance in traditional mobile robot vision simultaneous positioning and map construction (SLAM), the original algorithm was improved. First, extract the ORB features of adjacent images, and then use the fast approximate nearest neighbor algorithm to quickly match the feature points, and then use the PROSAC algorithm to replace the RANSAC algorithm to achieve feature point matching, excluding false matching points; Finally, based on the graph optimization model, a global Bundle Adjustment algorithm based on the largest common-view weight frame is proposed to achieve dense and sparse map creation. Finally, the TUM data set is used to verify the algorithm of this paper.

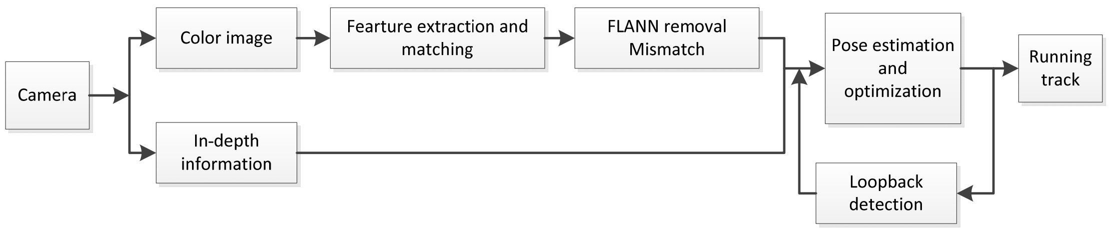


Figure 1. The overall framework of the algorithm in this paper

2.ORB algorithm analysis

2.1.ORB feature extraction and matching

In SLAM research, it is generally divided into a front end and a back end. The front end generally studies the registration between two adjacent frames of images, and estimates the position of the camera based on the registration result. There are direct methods and feature methods in front-end processing. The selection speed and accuracy of feature and descriptor selection have a great influence on the feature method. SIFT [8] and SURF [9] feature descriptors are very popular in traditional RGB-D SLAM, but both require a long time to calculate descriptors. A new feature descriptor calculation method was proposed by RuBlee et al in 2011. The algorithm uses an improved FAST corner detection method to obtain feature points, and uses the BRIEF feature descriptor to form a feature descriptor. This algorithm is called the ORB algorithm. Based on the fast calculation of FAST and BRIEF, compared with SIFT and SURF, the ORB algorithm has a huge advantage in speed. It does not need to be accelerated by the GPU, which reduces the requirements for using the device. The steps of the ORB algorithm are divided into oFAST feature point detection and rBRIEF descriptor extraction.

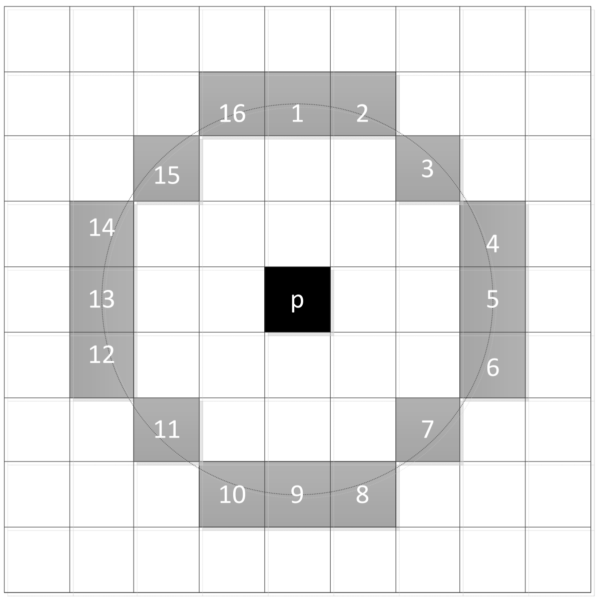


Figure 2. FAST feature points

Figure 2 shows the FAST feature points, and its neighborhood matrix can be defined as:

$$\mathrm{m}_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{1}$$

Where I (x, y) is the grayscale expression of the image, (x, y) is the position relative to the FAST feature point,m_{pq} is the (p+q)-th order matrix of feature points, p, q=0, 1, 2,...The centroid of the image block can be obtained from the feature matrix:

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

The direction of FAST feature points is expressed as:

$$\theta = arc\tan\left(m_{01}/m_{10}\right) \tag{3}$$

ORB uses RBRIEF characterization. RBRIEF is a binary descriptor that has rotation invariance. The descriptor segmentation function T of the corresponding image block is defined as

$$T(d;x,y) = \begin{cases} 1 & \left(d(x) < d(y)\right) \\ 0 & \left(d(x) < d(y)\right) \end{cases} \tag{4}$$

Where d(x) and d(y) are the gray functions at x and y in the neighborhood, The BRIEF descriptor contains n vectors in binary form:

$$f_n(d) = \sum_{1 \le i \le n} 2^{i-1} T(d; x_i, y_i) \tag{5}$$

Subsequently, the BRIEF descriptor is used to generate a description of the detected feature points. The brief descriptor has the characteristics of fast calculation speed, simple representation and small space occupation, but it does not have direction invariance. For a set of N points around feature points, define S as a 2 × N matrix:

$$S=\begin{pmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{pmatrix} \tag{6}$$

According to the rotation information θ of the image block and the corresponding rotation matrix R_θ, a matrix $S_\theta=R_\theta S$ with rotation information is calculated. The new feature descriptor can be expressed by the following formula:

$$g_n(p,\theta)=f_n(P)\big|(x_i,y_i)\in S_\theta \tag{7}$$

2.2 PROSAC algorithm principle
Feature matching is a crucial part in the realization of visual SLAM. Random Sample Consensus (RANSAC) algorithm is commonly used in the field of computer vision to optimize matching pairs. Randomness of RANSAC sampling points will lead to operational efficiency Low and degradation problems. Therefore, the PROSAC algorithm with better robustness and faster speed is used to eliminate false matches. The PROSAC algorithm sorts the sampling points according to the matching results from high to low, and only needs to draw samples from a high-quality data subset, and does not need to extract all the data. Assuming that the number of interior points of the sampling data set is more than the exterior points, the N data set points are represented as μ_N, and the two sampling points μ_i, μ_j in μ_N are arranged in descending order according to the correlation function q(u):

$$u_i,u_j\in\mu_N : i<j\Rightarrow q(u_i)\geq q(u_j) \tag{8}$$

Using μ_N to represent a set of N high-quality interior points requires designing a growth function of μ_N to maintain the balance between excessively optimistic reliance on quality prediction classification and excessively pessimistic RANSAC methods to treat all correspondences equally. If the probability of sampling the correct data point u_i can be known, then in principle the Bayesian method can be used: After each sampling and testing cycle, the posterior probabilities are recalculated for all matching relationships contained in the sample, and then sorted according to their posterior probabilities, and the sample with the highest probability is extracted. However, this method has two disadvantages that make it unusable: First of all, probability p (u_i) is not independent. Except for the simplest model, it cannot represent all the joint probabilities. Second, the estimation error of $p(u_i)$ is propagated through Bayesian formula and is emphasized, so if the initial estimate of $p(u_i)$ based on matching similarity is incorrect, then the posterior probability quickly becomes worthless. Therefore, a new method is used to obtain the growth function.

First make the following monotonous assumptions about the connection between $p(u_i)$ and the similarity function $q(u_i)$:

$$q(u_i)\geq q(u_j)\Rightarrow p(u_i)\geq p(u_j) \tag{9}$$

Then make the following inference:

$$i<j\Rightarrow p(u_i)\geq p(u_j) \tag{10}$$

Then use standard RANSAC method to extract T_N samples of size m from μ_N data set, expressed as $\{Mi\}_{i=1}^{TN}$, and arranged in descending order according to the sample quality, described by the formula as:

$$i<j\Rightarrow q\left(M_{(i)}\right)\geq q\left(M_{(j)}\right) \tag{11}$$

Sampling in the order of $M_{(i)}$, the sample set with high correlation will be sampled first, and the data points containing low-quality functions will be collected with delay. Let T_n denote the average number of samples in $\{Mi\}_{i=1}^{TN}$ containing high-quality interior points from μ$_n$, then

$$T_n = T_N \frac{\binom{n}{m}}{\binom{N}{m}} = T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i} \tag{12}$$

Then can get:

$$\frac{T_{n+1}}{T_n} = \frac{T_N}{T_N} \prod_{i=0}^{m-1} \frac{n+1-i}{N-i} \prod_{i=0}^{m-1} \frac{N-i}{n-i} = \frac{n+1}{n+1-m} \tag{13}$$

The final recursion relationship obtained from this is:

$$T_{n+1} = \frac{n+1}{n+1-m} T_n \tag{14}$$

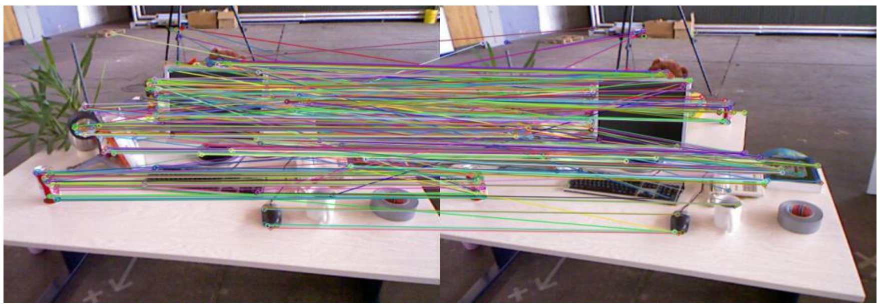
Because μ$_{n+1}$=μ$_n$U{μ$_{n+1}$}, then the sample of T$_{n+1}$-T$_n$ contains m-1 data points from u$_{n+1}$ and μ$_n$.Since the value of T$_n$ is generally not an integer, you can make T$'_m$=1 and T'$_{n+1}$=T^1_n+[T$_{n+1}$+T$_n$], then the growth function can be defined as:

$$g(t) = \min\left\{n : T_n^{'} \geq t\right\} \tag{15}$$

Where the t-th sample set M$_t$ is expressed as

$$M_t = \left\{\mu_{g(t)}\right\} \mathrm{U} M_t^{'} \tag{16}$$

Where M$'_t$ ⊂μ$_{g(t)-1}$is a set of |M$'_t$|=m-1 randomly selected from μ$_{g(t)-1}$, and the actual meaning of parameter T$_n$ is the minimum number of samplings after the PROSAC algorithm reaches the same effect as the RANSAC algorithm after multiple samplings.


a. Matching before unscreening

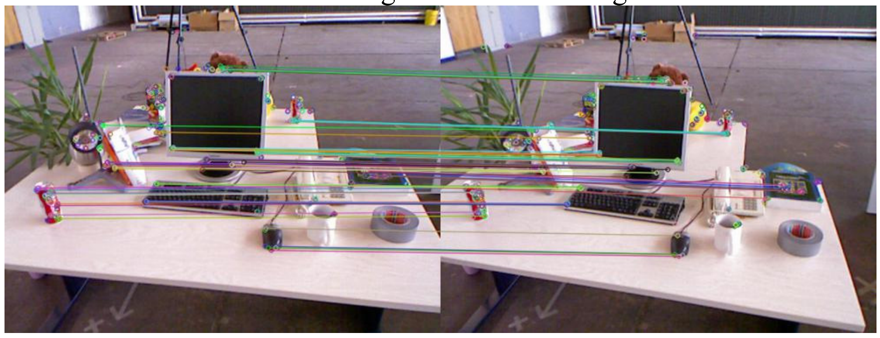

b. Feature matching after screening

Figure 3. Comparison of matching effect

It can be seen from b that there are many matching pairs based on the direct matching of ORB features, and there are many mismatches as a result. Compared with the PROSAC algorithm based on ORB features, the two algorithms obtain basically the same number of internal points, but the PROSAC algorithm is used to match Both the total time and the mismatch elimination time are greatly reduced.

2.3.Bundle Adjustment algorithm based on minimizing reprojection errors

After the pose tracking is completed, the initial pose will be obtained, but sometimes it is not very accurate. In order to improve the accuracy of the pose, the pose needs to be optimized. Each frame of observation of the camera will contribute an error, and the error of each frame adds up to constitute the total error. Due to the consideration of the amount of calculation, each frame is not optimized, but the selected key frames are optimized. In order to reduce the amount of calculation and ensure the robustness of the algorithm, a Bundle Adjustment method based on minimizing ghosting errors is proposed for optimization.

First, you need to determine the optimization objective function, and then obtain the camera pose by minimizing the objective function. Consider w three-dimensional space points **P** and its projection **p**, to get the camera pose **R**, **t** (the Lie algebra form of pose is expressed as ζ^).Suppose the coordinate of the spatial point is $\mathbf{P_a=[X_a, Y_a, Z_a]^T}$, and the internal parameter matrix of the camera is **K**, where the projected pixel coordinate is $\mathbf{u=(u_a, v_a)^T}$, and $\mathbf{S_a}$ is the zoom factor. The relationship between the pixel position and the spatial point position is shown in the formula.

$$S_a \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = K \exp\left(\zeta^\wedge\right) \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} \tag{17}$$

The pixel coordinates and the space point coordinates are projected according to the current estimated pose to obtain the position of the space point in the image after reprojection (due to the error and the real position do not coincide), and the optimization objective function is constructed as shown in the formula.

$$\varepsilon^* = \arg\min \frac{1}{2} \sum_{a=1}^{b} \left\| u_a - \frac{1}{s_a} K \exp\left(\zeta^\wedge\right) p_a \right\|_2^2 \tag{18}$$

In the formula: Ɛ* represents the reprojection error; b represents the total number of spatial points.

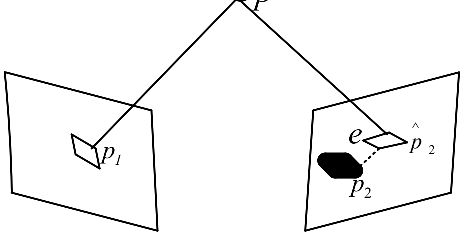


Figure 4. Schematic diagram of reprojection error

As shown in the figure, according to feature matching, p_1 and p_2 can be projected at the same spatial point **P**, but the camera pose is unknown. In the initial value, the projection p_2 of $\hat{p}_2$ has a certain distance from the actual p_2, so it needs to be adjusted The camera pose is that this distance becomes smaller.

In this paper, the graph optimization model is used to optimize the variables, the variables to be optimized are defined as vertices, and the errors are defined as edges. In order to obtain the updated amount of pose and map points, the optimization objective function derivates the pose and map points respectively to obtain the formula

$$\frac{\partial e}{\partial \delta\zeta} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z^2} & -\frac{f_x X'Y'}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y^2}{Z^2} & -\frac{f_y X'Y'}{Z^2} & \frac{f_y X'}{Z'} \end{bmatrix} \tag{19}$$

$$\frac{\partial e}{\partial p} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z^2} \end{bmatrix} R \tag{20}$$

3. Analysis of test results

In order to verify the feasibility and effectiveness of the proposed method, the experimental verification based on the data set was selected. The computer used in the experiment was an Acer notebook and the running platform was ubuntu18.04. The experiment uses the TUM data set for experimental demonstration. The data set is provided by Munich University of Technology and Freiburg University. The data set is to obtain indoor color information and depth information through the Microsoft Kinect sensor, to capture the movement of the camera through a high-precision motion capture system, to provide the true trajectory of the camera, and to give a test tool that can evaluate the results. The information of the data set used in this experiment is shown in Table 1:

Table 1 Test data set information table

Data set	Distance/m	Time/s
Fr1_desk	9.263	23.40
Fr1_desk2	10.161	24.86
Fr1_room	15.989	48.90
Fr1_floor	12.569	49.87

3.1 Comparison of running time and running effect

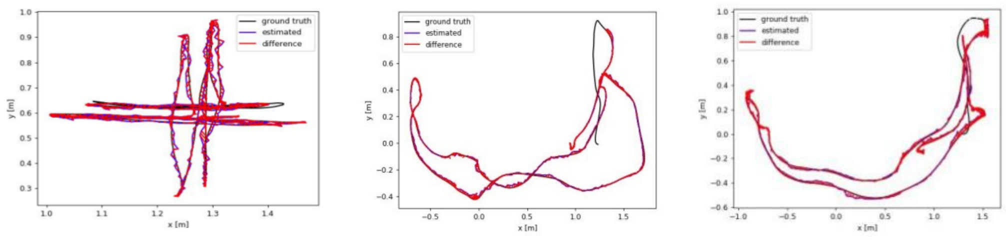
Running the above data set, as shown in Table 2 is the comparison of the running time after the improvement and the running time before the improvement. The improved running time is significantly reduced from the original running time, and the efficiency is significantly improved.

Table 2 Comparison of time before and after improvement

Data set	Original run time/s	Improved runtime/s
Fr1_desk	44.72	29.48
Fr1_desk2	49.24	30.32
Fr1_room	267.89	156.76
Fr1_floor	50.13	28.34

3.2 Analysis of trajectory error

Absolute trajectory error (ATE) and relative pose error (RPE) are used to compare the error between the estimated trajectory and the real trajectory. The absolute estimation error directly measures the difference between the point of the real trajectory and the estimated trajectory, suitable for testing SLAM. Absolute trajectory error (ATE) and relative pose error (RPE) are used to compare the estimated trajectory with the real trajectory Error. The absolute estimation error directly measures the difference between the point of the real trajectory and the estimated trajectory, which is suitable for testing the performance of the SLAM system. The deviation of the relative attitude error measurement estimated value from the true value is suitable for measuring the drift of the SLAM system. Calculate the ATE error and RPE error using the evaluation tools included in the TUM data set. Figure 4 shows the ATE error between the estimated trajectory and the real trajectory. Figure 4(a), (b) and (c) are the comparison charts of running data sets fr1_xyz, fr1_desk, fr1_desk2 and the estimated trajectory. The error does not change much and almost coincides with the real trajectory. It can be seen that the improved algorithm has errors on the trajectory Reduction has a certain effect.



(a) fr1_xyz (b)fr1_desk (c)fr1_desk2

Figure 5. Error between estimated trajectory and real trajectory

Sturm et al. used Kinect to record a series of image sequences, while using a motion capture system to record the camera's precision pose information. The image sequence includes a color map and a depth map. The resolution is 640*480, the frame rate is 30 Hz, and the real trajectory value is obtained by 8 high-speed tracking cameras (100 Hz). At the same time, they also provide an automatic evaluation tool, which can automatically calculate the root mean square error (RMSE) between the real trajectory and the algorithm estimated trajectory. In order to verify the effectiveness of the algorithm in this paper, this paper selects the data in the above test set for positioning and mapping experiments, and at the same time compares with the literature [14-15].

3.3 Analysis of positioning error

This article selects all frames of fr1/xyz, fr1/rpy, fr1/desk, fr1/desk2, fr1/floor, fr1/room data sets in the fr1 data package to experiment, and each set of data sets performs 6 repeated experiments. The camera pose of each key frame is estimated and the evo evaluation tool calculation algorithm RMSE is used. Table 3 shows the detailed test results of this algorithm.

Table 3 Detailed results of this algorithm

Data pack	Maximum/m	Minimum value/m	Average value/m	Standard deviation
fr1/xyz	0.020	0.017	0.018	0.00130
fr1/rpy	0.033	0.029	0.031	0.00205
fr1/desk	0.06	0.001	0.013	0.00918
fr1/desk2	0.061	0.002	0.020	0.01203
fr1/floor	0.038	0.002	0.008	0.00553
fr1/room	0.195	0.007	0.067	0.03795

At the same time, in order to reflect the superiority of the algorithm in this paper, this paper compared with other algorithms. Table 4 compares the root-mean-square error of the algorithm in this paper with the algorithms in [14] and [15].

Table 4 Comparison of the root mean square error between the algorithm in this paper and the algorithms in [14] and [15].

Data pack	Literature [14]	Literature [15]	Algorithm
fr1/xyz	0.021	0.017	0.017
fr1/rpy	0.042	0.032	0.029
fr1/desk	0.049	0.045	0.039

fr1/desk2	0.102	0.082	0.078
fr1/floor	0.219	0.250	0.209
fr1/room	0.055	0.050	0.059

4. Conclusion
This paper proposes an improved visual slam algorithm. This algorithm uses ORB features to reduce the performance requirements of computing devices. The fast nearest neighbor algorithm is used to match feature points and the image features are quickly matched. The PROSAC algorithm is used. To eliminate the mismatch, the global Bundle Adjustment algorithm based on the maximum common-view weight frame is used to achieve accurate estimation of the camera pose. Experimental simulations show that the algorithm proposed in this paper has a good effect.

References
[1] DAVISON A J, REIDID, MOLTON N D, et al. Mono SLAM: real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29 (6): 1052 - 1067.
[2] Yi Zhang, Longfeng Wang, Jiahang Yu. Construction of 3D Map of Mobile Robot Indoor Environment Based on Depth Information[J]. Computer application, 2014, 34(12): 3438 – 3445.
[3] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces [C] / / ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Washington, DC: IEEE Computer Society, 2007: 1 – 10.
[4] ENDRES F, HESS J, STURM J, et al. 3-D mapping with an RGB-D camera[J]. IEEE Transactions on Robotics, 2014, 30 (1): 177-187.
[5] Chao Long, Bo Han, Yu Zhang.3D SLAM based on kinect and visual dictionary[J]. Computer application, 2016, 36(3): 774 – 778.
[6] MUR-ARTAL R, MONTIEL J M M, TARDóS J D. ORB-SLAM: aversatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147 – 1163.
[7] Haifeng Li, Zunhe Li, Xinwei Chen. PLP-SLAM: Visual SLAM method based on point, line, and surface feature fusion[J]. Robot, 2017, 39 (2): 214 – 220.
[8] Lowe D G. Distincitive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
[9] Bay H, Tuytelaars T, Gool L V.SURF: Speeded up robust features[J]. Computer Vision & Image Understanding, 2006, 110 (3): 404-417.
[10] Wu Chang chang. Sift GPU: A GPU implementation of scale invariant feature transform (SIFT) [EB/OL].(2011-04-19).http: //www.cs.unc.edu/~ccwu/siftgpu/.
[11] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//International Conference on Computer Vision, 2011: 2564-2571.
[12] Rosin P L.Measuring corner properties[J].Computer Vision & Image Understanding, 1999, 73(2): 291-307.
[13] Wei Chang, Yun Liu. An improved fast panoramic image stitching algorithm[J]. Electronic measurement technology, 2017, 40(7): 90-94.
[14] Endres F, Hess J, Engelhard N, et al. An evaluation of the RGB- D SLAM system[C]//IEEE International Conference on Robotics and Automation, 2012: 1691-1696.
[15] Mengyin Fu, Xianwei Lü, Tong Liu,. Real-time SLAM algorithm based on RGB-D data[J].Robot, 2015, 37 (6): 683-692.