RGB-D Indoor Simultaneous Location and Mapping Based on Inliers Tracking Statistics

Xiaoning Niu¹, Hongzhe Liu¹* and Jiazheng Yuan²

¹Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China
²Beijing Open University, Beijing 100081, China

*Corresponding author e-mail: liuhongzhe@buu.edu.cn

Abstract. In this paper, an algorithm based on Inliers tracking statistics is proposed to solve the problem of motion-blur interference of RGB-D SLAM front-end pose estimation and back-end optimization. Firstly, the feature extraction and matching of the RGB image are carried out and the inliers is obtained by the RANSAC algorithm. Then, the blur image affected by the camera motion is removed by the tracking and statistics of the inliers quantity, and the camera position is solved by the nonlinear optimization method. Finally, the motion trajectory and the three-dimensional dense point cloud are plotted by and optimized global pose. Experimental results based on standard test set show that the algorithm based on Inliers tracking can improve the robustness and precision of SLAM system.

1. Introduction

With the research and development of intelligent robots, Simultaneous localization and mapping (SLAM) has become an increasingly popular research topic. SLAM is considered to be one of the key issues for robot autonomy. It has important research significance in the areas of navigation, control, and mission planning of robots, and been used in driverless vehicle, augmented reality and space exploration [1, 2, 3].

Visual SLAM uses the industrial monocular camera [4], binocular camera [5] or RGB-D camera [6] as the only sensor, which has enormous advantages in industrial applications as a result of high efficiency, low cost and excellent real-time system. The current RGB-D SLAM system can be classified as the feature method [6, 7] and the direct method [8-10]. In recent years, many studies on RGB-D SLAM have been proposed. Endres et al. [6] proposed a SLAM system that includes feature-based pose estimation and pose graph optimization by RGB-D camera. Mur-Artal et al. [7] proposed a novel SLAM system that uses the ORB (Oriented FAST and Rotated BRIEF) feature and uses monocular ordinary camera, binocular camera or RGB-D camera to realize the parallel processing of tracking and mapping. The above methods studied the feature selection and optimization problems respectively, but did not analyze the influence of motion blur on the SLAM system.

In this paper, aiming at the pose estimation of the SLAM frontend and the backend optimization being susceptible to motion blur interference, an algorithm based on Inliers tracking statistics is proposed and applied to the SLAM frontend pose estimation and backend key frame selection. The relationship between motion blur and feature matching inliers is analyzed. The inliers are obtained by the RANSAC algorithm; then the blurred images affected by the camera motion are eliminated and the
camera pose is solved by the ICP nonlinear optimization method by tracking and counting the number of inliers. This paper carries on the standard test set test and the actual environment test to the algorithm separately. The experimental result shows that the algorithm based on the Inliers tracking statistics can effectively improve the accuracy of the SLAM system construction.

2. Tracking and Statistics on Inliers to Eliminate Motion Blur Images

Feature matching is usually divided into inliers and outliers. In motion estimation, constraint equations are constructed from the correspondences of inliers to find the rotation matrix and the translation matrix of the camera motion. Therefore, the accuracy of inliers is crucial for pose estimation. By analyzing the tracking statistics of inliers, the following conclusions are drawn: due to the influence of motion blur on the feature points, the inliers suddenly decrease when the normal image and the blurred image match; when the blurred image and the blurred image match, the influence of ambiguity are similar by the motion, and the number of inliers will increase significantly. As shown in Fig. 1, the left image shows the results of feature matching between the normal image (a) and the normal image (b), where the number of inliers is 63; the middle picture shows the feature matching results of the normal image (b) and the blurred image (d), where the number of inliers is 8. The following figure shows the result of feature matching between the blurred image (b) and the blurred image (d), where the number of inliers is 41.

![Figure 1. Inliers matching results after matching each image](image)

In order to effectively eliminate motion blur images, this paper designs an algorithm for the relationship between inliers and motion blur. The idea is as shown in Algorithm 1. Among them, \( \text{curInliers} \) is the current number of inliers, and \( \text{lastInliers} \) is the number of historical inliers. When the inliers threshold satisfies equation (1), it is determined to be a normal image, and conversely, it is a motion-blurred image, which is excluded.

Algorithm 1 can effectively remove motion blur images similar to those in Fig. 1, where step 55 in equation 8 (1) is an empirical value, which indicates that the current matching inliers number A is greater than the historically matched inliers number 55%. When A is lower than 55% of B, it means that the current matching and historical matching variation is too large, and it is determined that the test frame is a motion-blurred image.

\[
\frac{\text{curInliers} \times 100}{\text{lastInliers}} > 55
\]

Algorithm 1 can effectively eliminate motion blur images similar to those in Fig. 1, where 55 in equation 8 (1) is an empirical value, which indicates that the current matching inliers number \( \text{curInliers} \) is greater than 55% of the historically matched inliers number \( \text{lastInliers} \). When \( \text{curInliers} \) is lower than 55% of \( \text{lastInliers} \), it means that the current matching and historical matching variation is too large, and it is determined that the test frame is a motion blur image.
Algorithm 1: Removing motion-blurred images based on inliers

Input: RGB images $1_{RI}$ and $2_{RI}$, depth images $1_{DI}$ and $2_{DI}$, camera is the internal parameter of the camera.
Output: Camera's rotation matrix $A$ and translation matrix $B$.

1: Input images $1_{RI}$, $2_{RI}$, $1_{DI}$ and $2_{DI}$;
2: Using $k$NN algorithm to match $1_{RI}$ and $2_{RI}$ for getting matches;
3: Get the depth information of features from $1_{DI}$ and $2_{DI}$, and eliminate feature points with a depth value of 0;
4: Using the RANSAC Algorithm to get inliers;
5: if $\frac{\text{curInliers} \times 100}{\text{lastInliers}} > 55$ & & $\frac{\text{lastInliers}}{\text{lastInliers}} < 120$ do
6: The 3D point pairs obtained by inliers are computed, and the rotation matrix $R$ and the translation matrix $t$ are calculated;
7: return $R$, $t$;
8: else if $\frac{\text{curInliers} \times 100}{\text{lastInliers}} < 55$ do
9: Determine the current frame as a fuzzy frame, discard the frame;
10: Considering that the similarity of two frames is decremented due to camera movement, a loss rate of 0.85 is set for historical $A$, that is: lastInliers* = 0.85;
11: continue;
12: else
13: Consider that the number of historical $A$ is too large, affecting the judgment of the next frame, So order $\frac{\sum_{i=1}^{20} \text{lastInliers}[i]}{10}$;
14: return $R$, $t$;

3. Principles and Algorithms of RGB-D SLAM

By selecting the ORB feature to match key frame, this paper constructs an RGB-D SLAM system. The overall architecture of the system is shown in Figure 2. The system consists of two parts: the frontend, including feature extraction and matching, blurring out blurred images, motion estimation and closed-loop detection; backend, including global pose optimization, point cloud and motion trajectory.

![Figure 2. RGB-D SLAM System Flow Chart](image-url)

3.1. ORB Feature Extraction and Matching

The ORB (Oriented FAST and Rotated BRIEF) features consist of key points and descriptors, which were proposed by Rublee et al. [13] in 2011.

After feature extraction, features between two adjacent frames are matched. In the real-time SLAM system, the Brute-Force Matcher algorithm is used to match the descriptors of feature in two adjacent frames. The matching result of the Brute-Force matcher is shown in Figure 3(a).

![Figure 3. Matching results. (a) Brute-Force algorithm, (b) Results after removing the false-matches](image-url)
From Figure 3(a) we can see that there are many outlines by the Brute-Force matcher. In order to remove these mismatches, k-Nearest Neighbor (kNN) matcher is used to remove the existing false-positive matches and false-negative matches. Then the RANSAC algorithm is used to estimate the camera’s homography matrix and get Inliers. The matching result after removing false matches is shown in Figure 3(c).

### 3.2. Pose Estimation and Optimization Algorithm

A set of 3D point pairs obtained by feature matching is a 3D-3D pose estimation problem. This paper uses the Iterative Closest Point (ICP) nonlinear optimization method to solve the problem. Let the matching points of the first frame image $I_1$ and the second frame image $I_{t+1}$ be: $P = \{p_1, ..., p_n\}$, $P' = \{p'_1, ..., p'_n\}$, respectively. When the Lie algebra is used to represent the pose, the pose transformation relationship from $P'$ to $P$ is:

$$\forall i, p_i = \exp(\xi \wedge) p'_i$$  \hspace{1cm} (2)

Where $\xi$ denotes the pose of the camera and symbol $\wedge$ denotes the conversion of a six-dimensional pose vector into a four-dimensional matrix. This paper defines the error term for the $i$th pair of points as:

$$e_i = p_i - \exp(\xi \wedge) p'_i$$  \hspace{1cm} (3)

Then, a least-squares problem is constructed by finding the sum of squared errors, and the optimal camera pose $\xi^*$ is found to minimize it:

$$\xi^* = \arg\min_{\xi} \sum_{i=1}^{n} \| p_i - \exp(\xi \wedge) p'_i \|_2$$  \hspace{1cm} (4)

This paper uses Equation Levenberg-Marquardt to solve Equation 4.

Through loop closure detection and global optimization, the system will obtain an estimated pose trajectory, and then project the pixel points of the key frame into the world coordinate system according to the edges and poses on the trajectory to form a 3D point cloud map.

### 4. Experimental Results and Analysis

In this paper, we use the data from the FR1 and FR2 groups in the standard dataset [12] to perform experiments to estimate camera pose trajectories and construct a three-dimensional dense map. The experimental results are shown in Table 1. The three-dimensional modeling of the dataset image and the comparison of the estimated trajectory with the Ground Truth are shown in Figs. 4 and 5, respectively. In order to verify the superiority of the algorithm, the algorithm of this paper is compared with the RGB-D SLAM proposed in the paper [6]. The off-line estimation pose data of RGB-D SLAM is from the paper [12].

| Dataset   | Length(m) | Per frame processing time (s) | RMSE(m)       | RGB-D SLAM | Our system |
|-----------|-----------|-------------------------------|---------------|------------|------------|
| fr1_desk  | 9.26      | 0.038                         | 0.026         | 0.026      | 0.006      |
| fr1_desk2 | 10.16     | 0.042                         | 0.042         | 0.026      | 0.006      |
| fr1_floor | 12.57     | 0.037                         | 0.035         | 0.014      | 0.015      |
| fr1_plant | 14.79     | 0.043                         | 0.061         | 0.028      | 0.025      |
| fr1_rpy   | 1.66      | 0.043                         | 0.043         | 0.028      | 0.009      |
| fr1_xyz   | 7.11      | 0.042                         | 0.013         | 0.013      | 0.009      |
As is illustrated in the Table 1, the average processing time of each frame of data is 38 ms in the test of the data set in this algorithm. Kinect's frame rate is 30Hz, so this system can basically accommodate the real-time requirement. In terms of accuracy, through comparison with the classical RGB-D SLAM trajectory error, we can see that the RMS (root mean square) error of the estimated trajectory of this algorithm is obviously less than that of RGB-SLAM, and the average error is centimeters.

5. Conclusion
In order to reduce the impact of motion blur on pose estimation and construction accuracy, this paper proposes a SLAM system based on Inliers tracking statistics to effectively eliminate motion blurred images. The experimental results show that the proposed method can accurately locate the position of the robot. The RMS error of the pose trajectory is on the order of centimeters. In an ideal environment, the error can be reduced to millimeter. At the same time, the overall processing speed of the system is 25 frames/s, which can meet the real-time requirements of indoor robot construction.

Acknowledgments
This work was financially supported by Supporting Plan for Cultivating High Level Teachers in Colleges and Universities in Beijing (IDHT20170511). Talents Cultivation and Cooperation Oriented to Intelligent Vehicle Industrialization (Grant number: UK-CIAPP/324), Newton Fund Project supported by Royal Academy of Engineering of UK. The National Key Technology R&D Program 2015BAH55F03)

References
[1] Zhenhe Chen, Jagath Samarabandu, Ranga Rodrigo. Recent advances in simultaneous localization and map-building using computer vision[J]. Advanced Robotics, 2007, 21(3-4):233-265.
[2] Durrant-Whyte H, Bailey T. Simultaneous Localization and Mapping: Part I[J]. IEEE Robotics &amp;amp; Automation Magazine, 2006, 13(2):99-110.
[3] Cheng Y, Maimone M W, Matthies L. Visual odometry on the Mars Exploration Rovers - A tool to ensure accurate driving and science imaging[J]. IEEE Robotics &amp; Automation Magazine, 2006, 13(2):54-62.
[4] Davison. Real-time simultaneous localisation and mapping with a single camera[C]// IEEE International Conference on Computer Vision, 2003. Proceedings. IEEE, 2008:1403-1410 vol.2.
[5] Paz L M, Pinies P, Tardos J D, et al. Large-Scale 6-DOF SLAM With Stereo-in-Hand[J]. IEEE Transactions on Robotics, 2008, 24(5):946-957.
[6] 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.
[7] 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.
[8] Kerl C, Sturm J, Cremers D. Robust odometry estimation for RGB-D cameras[C]// IEEE International Conference on Robotics and Automation. IEEE, 2013:3748-3754.

[9] Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]// IEEE InternWhelan T, Kaess M, Johannsson H, et al.

[10] Whelan T, Kaess M, Johannsson H, et al. Real-time large-scale dense RGB-D SLAM with volumetric fusion[J]. International Journal of Robotics Research, 2015, 34(4-5):598-626.

[11] 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.

[12] 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.