Feature-Based Monocular Real-Time Localization for UAVs in Indoor Environment

Yu Zhang, Zhihao Cai, Jiang Zhao, Zhenxing You and Yingxun Wang

1 Introduction

UAV applications in recent years have become more and more widely for its small size and excellent mobility. Most UAVs used to depend on GPS or inertial system for autonomous flight, but in the indoor or GPS denied environment it becomes a great challenge. Because of the small size and load capacity constraints, traditional sensors are not suitable for small UAV platforms, such as lasers [1, 2]. As a result, visual sensor provides a good solution as it contains a wealth of motion and environment information.

The traditional visual odometry which can be seen as pairwise and structure-less lacks robustness and easily accumulate drift. In this paper, we realize the real-time localization for UAVs in indoor environment based on the idea of the monocular visual odometry. Figure 1 shows main components of the system and we elaborates on in the paper. In addition, to improve the performance and overcome the shortcomings of traditional method, we use g2o and a map based method to optimize the estimated pose and reduce the drift of estimated trajectory in the case of prolonged motion.

Y. Zhang (✉) · Z. Cai · J. Zhao · Z. You · Y. Wang
School of Automation Science and Electrical Engineering,
Beihang University, Beijing 100191, China
e-mail: zhangyu5782@163.com

Z. Deng (ed.), *Proceedings of 2017 Chinese Intelligent Automation Conference*,
Lecture Notes in Electrical Engineering 458,
https://doi.org/10.1007/978-981-10-6445-6_39

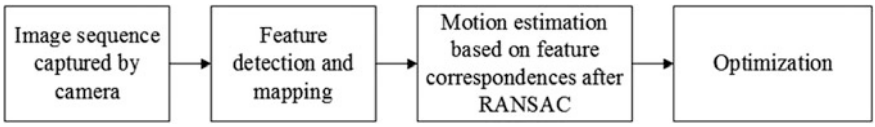

Fig. 1 Main components of the system

2 Feature Detection and Mapping

2.1 ORB Feature Detection and Description

ORB features consists of two parts: the oriented multi-scale FAST keypoints and the rotated BRIEF descriptor [3]. As a result, they are rotation invariant and it is extremely fast to compute and match ORB.

FAST corners are widely used because of its fast detection speed, but they do not have an orientation component and are not multi-scale. To overcome these weaknesses, ORB add scale and rotation description. Scale invariance is achieved by building a scale pyramids of the image and detecting corners at each level and the rotation description is added by intensity centroid. The moments of a patch are defined as:

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

And the centroid of it can be found by the moments defined above:

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

A vector OC is get by connecting the corner's center and the centroid. The orientation of the features is defined as:

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

The BRIEF descriptor is a binary descriptor which describes the pixels information around the detected keypoints. The "Steer BRIEF" feature after the rotation can be calculated by using the direction information, so that the descriptor of the ORB has good rotation invariance.

According to the statistics, the calculation speed of ORB is about 10 times faster than SURF and SIFT. To compute ORB in an image of size 752 × 480, it detects 500 features in 0.011679 s. Comparing to SURF and SIFT on the same image, the time cost are as follows: 0.091570 s, 0.116692 s.

2.2 Feature Matching

The feature matching step solves the problem of data association, which is critical because it can reduce the burden for subsequent pose estimation, optimization, and so on by accurate feature matching.

We adopt FLANN (Fast Library for Approximate Nearest Neighbors) in the feature matching step as it is fast and efficient. It significantly improves matching efficiency depending on the Randomized K-d Three Algorithm and The Priority search K-means Three Algorithm. For better performance, we remove error matches based on the condition that Hamming distance is less than twice the minimum distance. In the next step, we will use random sampling consensus strategies to remove more outliers for more accurate motion estimation.

3 Motion Estimation

3.1 Motion Estimation Based on the Epipolar Constraint

After feature matching, the geometric relations between two images can be computed from feature correspondences. Furthermore, we can compute the relative motion to get real-time position information. The main property of the epipolar constraint illustrated in the below, which is the basis of 2d-2d motion estimation [4].

P2 is the corresponding feature point of P1 in the other image, and they are both normalized coordinates. The epipolar constraint between them can be formulated as:

$$p_2^T E p_1 = 0 \tag{4}$$

E is the essential matrix which describes the geometric relations between two images. In this paper, we use the classic eight-point algorithm to compute E in the following form, where u and v is the normalized coordinates:

$$\left[u_1 u_2, u_1 v_2, u_1, v_1 u_2, v_1 v_2, u_2, v_2, 1\right] \cdot E = 0 \tag{5}$$

The rotation matrix R and the translation matrix t can directly be extracted from E by SVD. And the rigid body transformation can be composed as:

$$T_{i+1,i} = \begin{bmatrix} R_{i+1,i} & t_{i+1,i} \\ 0 & 1 \end{bmatrix} \tag{6}$$

In summary, the motion estimation based on feature correspondences are as follows. Step 1: Feature detection and matching in continuous frame. Step 2:

Compute essential matrix E based on corresponding features. Step 3: Extract the rotation and translation parts R, t from E. So the current pose can be computed by the transformation T if we get the pose at last timestamp.

3.2 RANSAC Algorithm

Matched features are usually contaminated by outliers that may be caused by blur, illumination and so on. Since the motion estimation is based on matched features, the wrong data association will have a bad effect on the accuracy on estimation.

For robust estimation, in the motion estimation step we take RANSAC (random sampling consensus) algorithm as a solution to outlier removal [5, 6]. The main idea behind the RANSAC algorithm is that based on data sets that randomly sampled the model hypotheses can be computed and the hypothetical model can be verified on other data. After iterations, the hypotheses which has the highest accordance is selected as a solution.

4 Optimization

4.1 Bundle Adjustment

Since the pose and trajectory of the camera are computed incrementally, the errors of each image will increase with time gradually. To keep the drift as small as possible, we use bundle adjustment to obtain more accurate estimation. 3D point P is computed based on correspondences P1, P2 by triangulation. The projection position of point P2′ is known according to the current pose. The error distance between P2 and P2′ is reprojection error.

In bundle adjustment, the goal is to minimize the reprojection error:

$$\arg\min_{C_k} \sum_{i,k} \left\| p_k^i - g(X_i, C_k) \right\|^2 \tag{7}$$

where C_k is the camera pose in kth frame. In this paper, we use g2o to realize the bundle adjustment. G2o is a library based on graph optimization [7]. In a pose graph, the camera poses and all 3D points are presented as nodes and the image projection of all 3D points are edges between nodes. Bundle adjustment tried to optimize the poses and 3D points to minimize the reprojection error.

4.2 Map Based Visual Odometry

The traditional visual odometry computes the relative transformations T from the adjacent images and then concatenate the transformation to recover the full trajectory [8]. It can be seen as pairwise and structure-less VO, because it only concentrates the motion between two frames. In addition, it ignores the features that once used so it saves the amount of computation but losses lots of information. Another drawback of the pairwise VO is that once there is a bad estimate in one frame, the error will always effect the following estimate.

To solve the problem mentioned above and take advantage of features extracted in old frames, we can use map based visual odometry instead. The map is a collection of 3D points by triangulating features in each frame. We match points in the map and the features extracted in the current frame to compute the pose of camera directly. The benefit of this method is that we are able to maintain a constantly updated map. As long as the map is correct, even if a frame goes wrong, it is possible to estimate the correct position of following frames. Besides, only feature points that close to the current position are kept in the map and those points out of view field will be dropped for computation efficiency.

The difference between traditional VO and map based VO is described in Fig. 2 in detail.

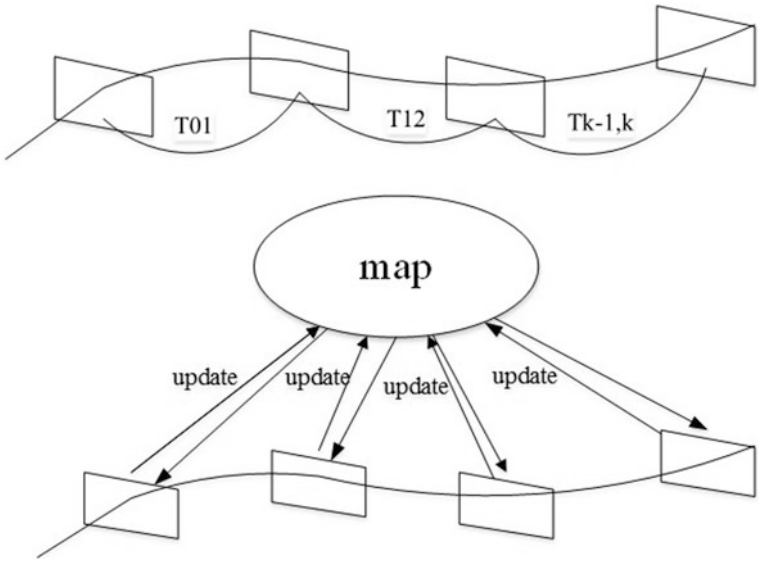

Fig. 2 The difference between the pairwise VO and map based VO

5 Results and Analysis

To test the effect of the bundle adjustment and evaluate the performance of the system, we have performed several experiments that are demonstrated in detail as follows.

The system runs with Intel Core-i7 and 16G RAM and in the platform of ROS (Robot Operating System), and the image is captured by the IDS-ueye camera at 30 fps. The resolution of the image is 752 × 480.

1. Accuracy evaluation in the TUM dataset

The TUM dataset is good for evaluating the accuracy as it provides image sequences with accurate ground_truth which is given by motion capture system. The following experiments are performed in freiburg1_xyz dataset.

In Fig. 3, the black line is the ground truth, and the blue is the estimated trajectory. The red line is the difference between them. Figure 3a shows the estimated trajectory before bundle adjustment, and Fig. 3b is the result after bundle

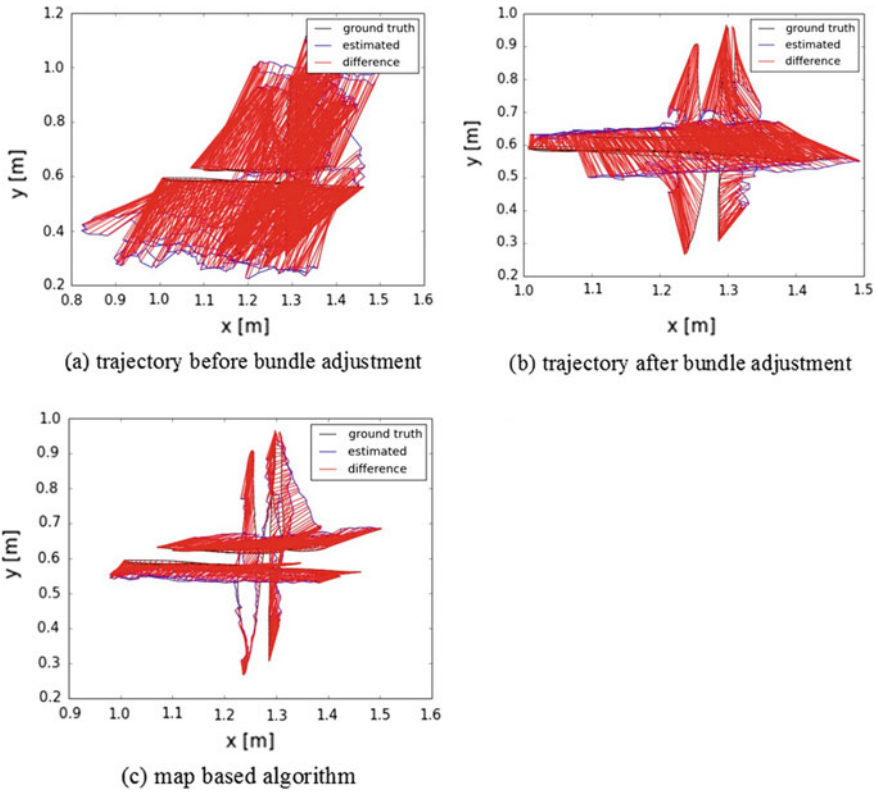

Fig. 3 Comparison of the estimated trajectory using different methods

adjustment with pairwise algorithm. Figure 3c shows the performance with map based algorithm.

- Experiment on bundle adjustment

Figure 4 shows that the reprojection error decreases a lot after bundle adjustment. In addition, from Fig. 3a, b and the first two lines in Table 1, it can be concluded that after bundle adjustment the localization error is reduced and the estimated trajectory is clearly more accurate.

- Pairwise algorithm Versus map based algorithm

Comparing Fig. 3b, c and the last two lines in Table 1, map based visual odometry obviously shows a better performance. And Fig. 5 plots the error respectively in x direction and y direction based on map based visual odometry.

2. Real-time localization experiments in indoor environment

Besides the experiments in TUM dataset, we have done the real-time localization experiments in the lab environment.

- Handheld experiment

We handheld the camera and walked around a table with a rectangular track in the lab environment. The environment and the real-time estimated trajectory is shown in Fig. 6. And from Table 2, it shows the computation speed of the system can satisfied the real-time need for UAVs.

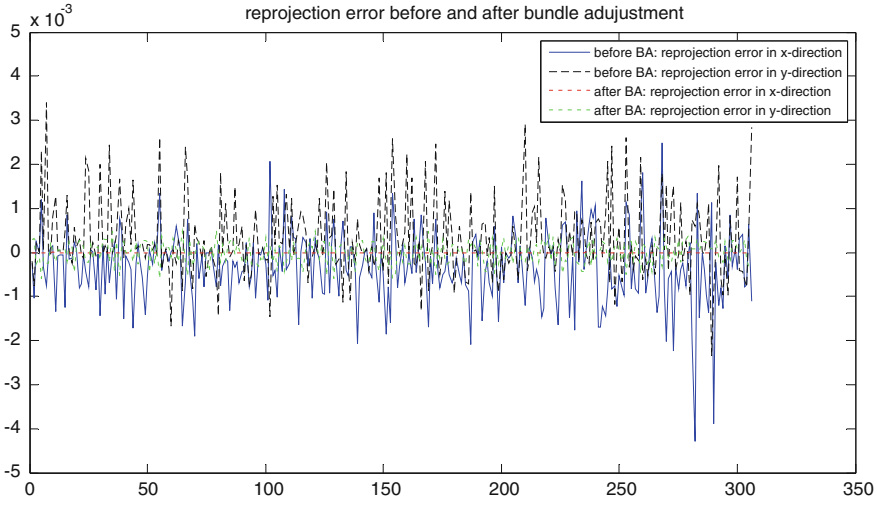

Fig. 4 Comparison of reprojection error before and after bundle adjustment

Table 1 Localization error comparison in different methods

Algorithms	Translational error min (m)	Translational error max (m)	Translational error mean (m)
Before BA	0.084973	0.413918	0.280460
After BA	0.015544	0.308868	0.109293
Map based algorithm	0.010236	0.137224	0.067179

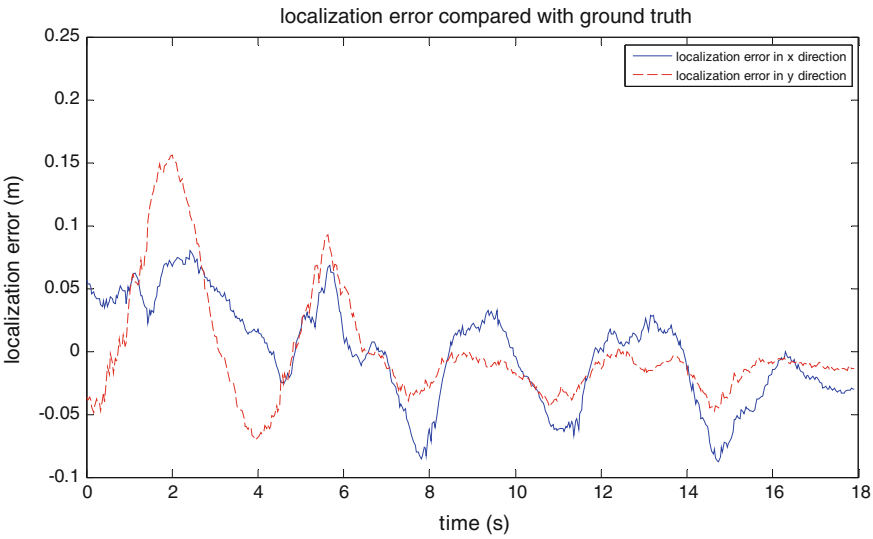

Fig. 5 Localization error

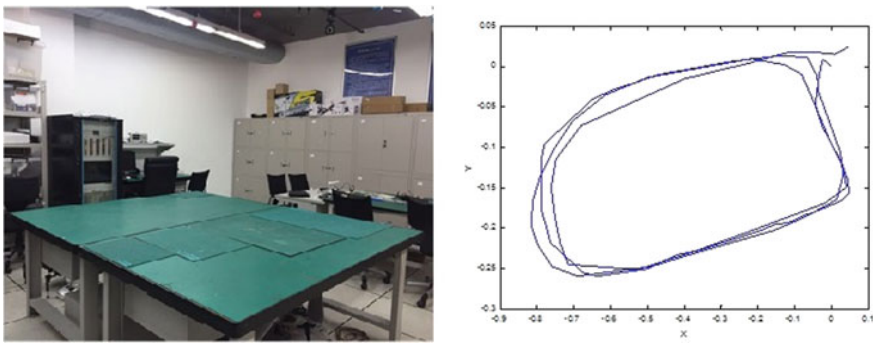

Fig. 6 Real-time rectangular trajectory around a table

Table 2 Computation time in one frame

	Feature detection	Feature matching	Motion estimation and optimization	Total
Time (s)	0.01665	0.01182	0.011342	0.039812

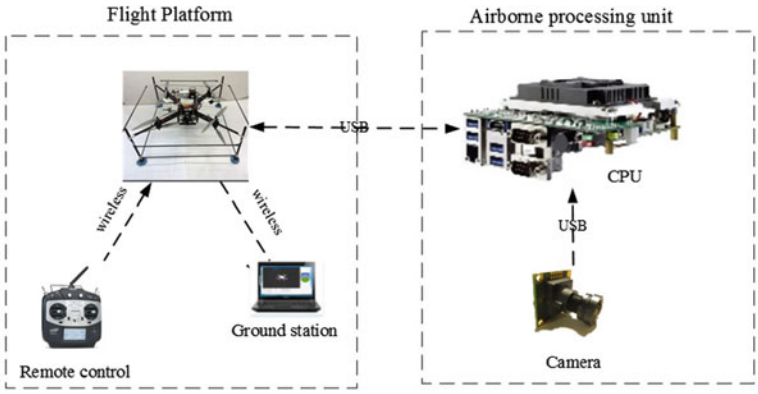

Fig. 7 The components of the system

Fig. 8 The images during the flight and the estimated trajectory

- Flight experiment

We use a small UAV with the wheelbase of 80 cm and 1 ~ 2 m/s flight speed to perform the experiments in the indoor environment. The components of the hardware is showed in Fig. 7. Figure 8 shows some images during the flight and the estimated flight trajectory.

6 Conclusion

A real-time localization method based on monocular is proposed to solve the navigation problem for UAVs in indoor environment in this paper. Several improvements have been made on traditional VO to satisfy the need of real-time and accuracy of UAVs. With the experiments carried out both in dataset and indoor environment, it turns out the method is feasible for UAV applications.

References

1. Zhang Y, Wang T, Cai Z et al (2017) The use of optical flow for UAV motion estimation in indoor environment. In: Guidance, navigation and control conference. IEEE, pp 785–790
2. Wang T, Zhang Y, Cai Z et al (2016) Visual attention based target detection and tracking for UAVs. In: IEEE chinese guidance, navigation and control conference. IEEE, pp 895–900
3. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: IEEE international conference on computer vision (ICCV), Barcelona, Spain, pp 2564–2571
4. Fraundorfer F, Scaramuzza D (2012) Visual odometry: part II: matching, robustness, optimization, and applications. IEEE Robot Autom Mag 19(2):78–90
5. Nistér D (2005) Preemptive RANSAC for live structure and motion estimation. Mach Vis Appl 16(5):321–329
6. Chum O, Matas J, Kittler J (2013) Locally optimized RANSAC. Lect Notes Comput Sci 2781:236–243
7. Kümmerle R, Grisetti G, Strasdat H et al (2011) G2o: a general framework for graph optimization. In: IEEE international conference on robotics and automation. IEEE, pp 3607–3613
8. Scaramuzza D, Fraundorfer F (2011) Visual odometry: part i: the first 30 years and fundamentals. IEEE Robot Autom Mag 18:80