Study on the ORB algorithm in the application of Monocular SLAM

Jiwu Wang
School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University
Beijing 100044, China
E-mail: jwwang@bjtu.edu.cn

Shunkai Zheng, Yao Du
School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University
Beijing 100044, China

Sugisaka Masanori
Alife Robotics Corporation Ltd, Japan and Open University, United Kingdom
E-mail: ms@alife-robotics.co.jp

Abstract

In order to reduce the accumulative errors in our monocular SLAM (simultaneous localization and mapping), the loop closing (detection + correction) method based on PTAM (Parallel Tracking and Mapping) is applied. Here natural environment features are necessary to be extracted efficiently, so the ORB (Oriented FAST and Rotated BRIEF (Binary Robust Independent Elementary Features)) algorithm is used for the feature extraction and matching. The experiment results show that there is strong feature recognition power in ORB so that it can realize environment recognition under the conditions of severe viewpoint change. Moreover, it is so fast to extract and match (without using multi-threading or GPU (Graphics Processing Unit) acceleration) that it can accurately track and map in real time. Its capability of fast and efficiency is verified with outdoor scene experiments.

Keywords: SLAM; ORB; Monocular Vision, Mobile Robot

1. Introduction

Monocular SLAM has been developed for a long time, from the path filter at initial stage to the most modern key frame-based SLAM systems now. One of the most representative key frame-based systems is Parallel Tracking and Mapping, PTAM.[1] PTAM has become a criterion in monocular vision, and it has been recently adapted for MAVs(micro aerial vehicles) navigation.[2,3]

As we all known, using Stereo and RGB-D cameras, the depth of each image can be calculated. But it is a more complex problem with just one camera. That is because the depth of observed features in the image is unknown. In some cases, monocular techniques are still necessary. So using multiple view geometry is necessary to solve the problem.

Published by Atlantis Press

In this paper we focus on two open problems about ORB and loop closing, which are essential for SLAM applications.

ORB features are oriented multi-scale FAST corners with a 256 bits descriptor associated.[4] They are extremely fast to compute and match. And they are highly invariant to viewpoint. That allows matching them from wide baselines, but the accuracy sometimes is not good.

Loop closing is the detecting task when a robot is revisiting a previously mapped place, in order to correct the error accumulated in the robot trajectory during exploration.[5] Tracking data association is able to close small loops. In the case of large loops, it sometimes is not successful. This is because the prediction of the camera location accumulates more errors, that data association fails to establish the relationship between the current image and the map. Then the system takes those images as a new place and thus duplicating previously mapped areas.

In this paper, based on the mobile robot in the lab, the problems of ORB and loop closing are studied.

2. System Overview

The system is as shown in Fig.1. In the system, it is composed of three main functions: tracking, local mapping and loop closer, which runs in parallel in a multi-core machine.

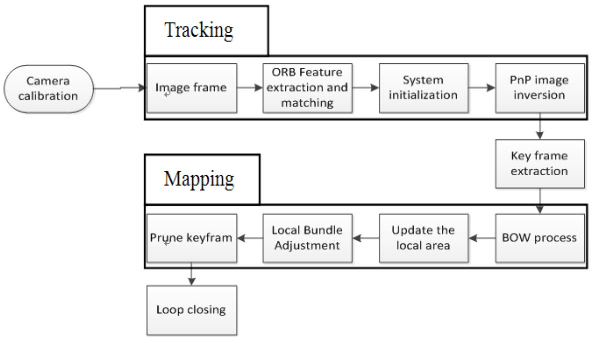

Fig.1.Illustration of system overview

2.1 ORB Extraction and Tracking

In the tracking process, the map initialization is performed to find a homography (planar scenes) or an essential matrix from two neighbour frames in an automatic process. After an initial map is established, the tracking process will estimate the camera pose with every incoming frame.

At first, it is to extract ORB features in the frame, in which a scale image pyramid is computed and FAST key points are detected at each level. Then a subset of these key points will be only described. If tracking was successful in the last frame, the tracking tries to get a first estimation of the camera pose from the last frame. Each ORB from the last frame, which has a map point associated, searches a match in the current frame in a small area around its previous position.

In the large scale environment, feature extraction and matching are influenced by Gaussian blur, rotation, scaling, illumination and processing speed. In order to verify the ORB features extraction performance, the following experiment is carried on.

Taking image sequence Fig.2 as original image that captured in the lab room.

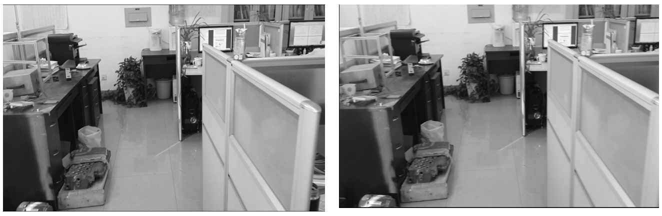

Fig.2. The original frame and its neighbor frame

Then using the affine transformation of the specified argument and mathematical model including Gaussian blur, rotation, scaling, illumination for the right side frame in Fig.2 to generate the sequences. It is similar to take pictures in the complex scene with cameras in the moving process, the same as the performance in the practical applications.

Use RANSAC (Random Sample Consensus) to find out the transformation matrix between the generated sequences and original frame. Then the sequences frame of the feature points is projected to the original frame. The size of the projection error is calculated. If the error is less than the set threshold value (in this experiment the threshold value is 2), the point is the correct match point, otherwise, it is a false matching point. The relation between the calculation is:

$$ratioMatch = \frac{correctMatch}{Match} \times 100\% \qquad (1)$$

In Fig.3, it shows that the feature matching performance of ORB is the best under the condition of fuzzy, because it can greatly improve the stability of the algorithm. The ORB is the most stable for rotation in Fig.4. The several feature extraction algorithms are similar to the change of the scale and brightness in a certain range in Fig.5 and in Fig.6.

In table1, the comparison of extracting all the feature points of the all images is given. The speed of ORB is the fastest and 17 times faster than SURF. For every feature point of the image, ORB is 3 times faster than SURF.

The above results show that the excellent performance of the ORB feature extraction algorithm. It demonstrates that ORB features have enough recognition power to enable place recognition from severe viewpoint change and it is so fast to extract and match (without using multi-threading or GPU acceleration). The ORB can satisfy the real time accurate tracking and mapping.

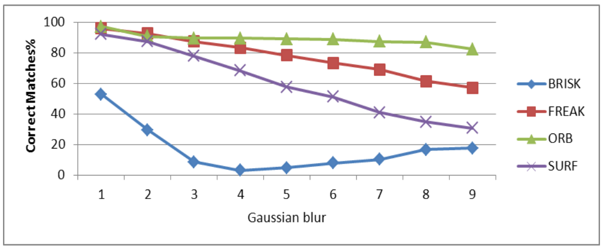

Fig.3. The verification with Gaussian blur

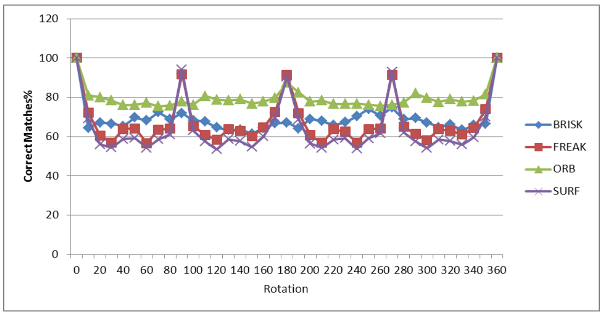

Fig.4. The verification with rotation angle (0-360°/per 20°)

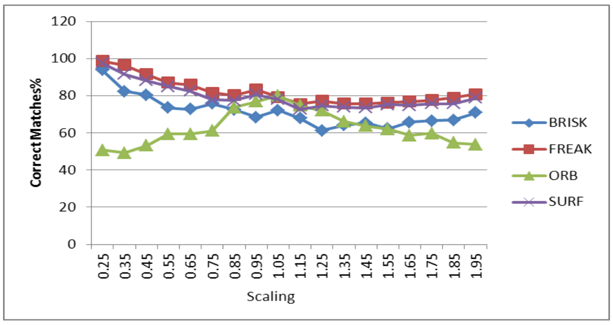

Fig.5. The verification with Scaling (0.25-1.95/per 0.1)

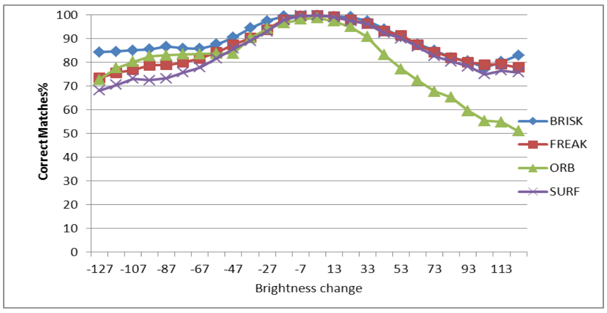

Fig.6. The verification with Additional brightness values

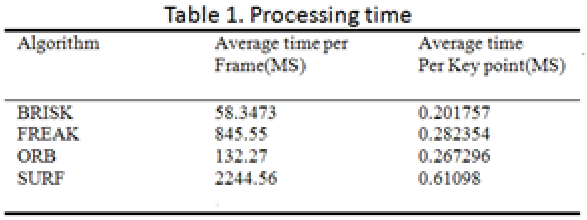

Table 1. Processing time

Algorithm	Average time per Frame(MS)	Average time Per Key point(MS)
BRISK	58.3473	0.201757
FREAK	845.55	0.282354
ORB	132.27	0.267296
SURF	2244.56	0.61098

2.2 Loop Closing

The loop closing thread takes the last key frame (K_i) processed by the local mapping, and tries to detect and close loops. At first we compute the similarity between the bag of words vector of K_i and all its neighbors in the covisibility graph ($\theta_{\min}$=30) and retain the lowest score S_{min}. Then we search in the recognition database and discard all those keyframes whose score is lower than S_{min}. It is a similar operation to gain robustness as the normalizing score in DBoW2 which is computed from the previous image, but here we use covisibility information. In addition all those key frames directly connected to K_i are discarded from the results. To accept a loop candidate we must detect consecutively three loop candidates that are consistent (key frames connected in the covisibility graph). There can be several loop candidates if there are several places with similar appearance to K_i.

To close a loop we need to compute a similarity transformation from the current key frame K_i to the loop key frame that informs us about the error accumulated in the loop. The computation of this similarity will serve also as geometrical validation of the loop. The correspondences are computed firstly between ORB associated to map points in the current key frame and the loop candidate key frames. Then the 3D to 3D correspondences for each loop candidate is obtained.

We alternatively perform RANSAC iterations with each candidate. If a similarity S_{ij} with enough inliers is found, we optimize it and perform a guided search of more correspondences. We optimize it again and the loop with K_i is accepted if S_{ij} is supported by enough inliers.

3. Experiments and Conclusions

3.1 Experiment with Public Data Sets

The odometry benchmark in the KITTI dataset[6] contains sequence images taken by a car driven around a residential area with accurate ground data from GPS and a Velodyne laser scanner. It is a very challenging dataset for monocular vision due to fast rotations, areas with lot of foliage which make more difficult for data association, and relatively high car speed (the sequences recorded at 10 fps).

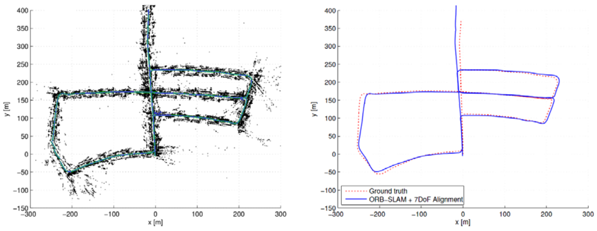

Fig.7. Left: points and key frame trajectory
Right: trajectory and ground truth

Table 2. Results of the KITTI dataset

Sequence	Dimension ($m\times m$)	KFs	RMSE (m)
KITTI 05	478x425	850	7.23

The result of the No.05 sequences of the KITTI dataset is shown as table 2: the RMSE of that experiment is 7.23m. The results demonstrate that our developed system is very accurate because of the trajectory error typically around the 1% of its dimensions.

3.1 Experiment with the Sequence Images around Laboratory Building

Experiments are carried on with the image sequences generated along the road around my laboratory. The actual map is as Fig.8.

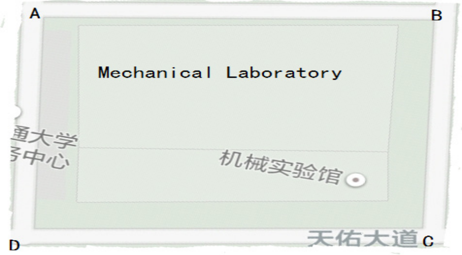

Fig.8. The map of the road A-B-C-D around the lab building

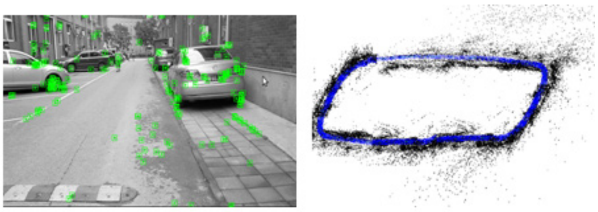

Fig.9. ORB feature extraction and generated map of ABCD

At last, in the ORB experiments, the characteristics of the corner can be lost, and the error of the distance is estimated. It verify the developed system is reliable..
In the future, we will focus on solving the robustness of the map initialization, the processing at the corner and the sparsification of the pose graph to be optimized by the loop closer.

References

1. G. Klein and D. Murray, Parallel tracking and mapping for small AR workspaces. International Symposium on Mixed and Augmented Reality (ISMAR), 2007.
2. D. Scaramuzza, M. Achtelik, et al., Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments, IEEE Robotics & Automation Magazine, Vol.21(3)(2013) 26–40,
3. J. Engel and D. Cremers, Scale-aware navigation of a low cost quadrocopter with a monocular camera, Robotics and Autonomous Systems, vol.62(11)(2014)1646-1656.
4. E. Rublee, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In IEEE International Conference on Computer Vision (ICCV), (2011)2564-2571.
5. B. Williams and J. D. Tard, A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems, vol.57(12)(2009)1188–1197.
6. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, Vision meets robotics: The KITTI dataset, The International Journal of Robotics Research, vol.32(11)(2013)1231–1237.