Collaborative Localization for Multiple Monocular Vision-Based MAVs

Tong Qin✉, William Wu, and Shaojie Shen

Aerial Robotics Group, The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
{tong.qin, wwuar}@connect.ust.hk, eeshaojie@ust.hk

Abstract. In this paper, we present a collaborative localization framework for a swarm of Micro Aerial Vehicles (MAVs) using monocular visual-inertial systems (VINS). Unlike traditional swarm applications which rely on external position equipment (GPS or Motion Capture System), our system achieves globally consistent localization based on internal sensors (onboard camera and IMU). Each MAV is equipped with one camera and one IMU. It estimates own pose onboard and sends visual information to a centralized ground station. The ground station collects collaborative information from all MAVs, and maintains a globally consistent coordinate by pose graph optimization. Then the global localization is feedbacked to each MAV for global control purpose. The ground station not only aligns all MAVs in a global coordinate, but also correct accumulated drifts for each MAV. The proposed system is validated by metric evaluation in experimental results. Real-world swarm application further demonstrates the robustness and practicability of our system.

Keywords: vision-based MAV; collaborative localization; monocular visual-inertial systems
Supplementary Material: Code&Video (https://github.com/qintonguav/Co-VINS)

1 Introduction

Micro aerial vehicles (MAVs) have played an important role in robotic community over the last decade. Due to their small size, light weight and six-degree-of-freedom flexibility, they can easily reach the place where human and ground vehicles can't get into. Meanwhile, they can collect information in a novel perspective, birds-eye view, which provides us with all-around observation. Therefore, MAVs have been widely used in inspection, search and rescue missions. A swarm of MAVs is able to cover larger areas and collect more information within a short time. Hence, research about swarm attracts more and more attention recently.

1.1 Problem Statement

Among the swarm of MAVs, the cooperative operation is of vital importance. The first issue is global localization. Every individual aerial vehicle needs to know both own pose and relative poses with respect to others. Therefore, a global consistent coordinate

which can locate all aerial vehicles is inevitable. Up to now, the localization system can be classified into two categories, external and internal localization. External localization system relies on external equipment, such as Global Positioning Systems (GPS) and Motion Capture System. Each of them has the limited range of application. GPS provides absolute longitude and latitude with respect to the earth, which is suitable only for the large outdoor environment. In the indoor environment, where GPS is unavailable, Motion Capture System locates objects through accurate tracking from multiple infrared cameras. Infrared cameras need to be installed all around. Cameras and site are carefully calibrated in advance. The objects can only move in the defined area, which limits its usage. Compared with external sensors, the internal sensors, such as Lidar and vision, have a larger range of application. They can be used in either indoor environment or outdoor environment. However, these internal localization methods pose a significant challenge in fusing data under a globally consistent coordinate. The data is always collected by different robots in local frames. In other words, each individual easily knows there own position, but it is hard to know its relative position with respect to others. To this end, we present a vision-based cooperative localization framework, which merges data from each aerial vehicle and locates them in a globally consistent coordinate in real time.

1.2 Related Work

Traditional external localization systems, e.g. GPS and Motion Capture, have conducted impressive cooperative missions on MAVs. In [1, 2], multiple quadrotors were precisely located by indoor Motion Capture systems. All quadrotors were located and controlled in a pre-defined global frame. For internal localization methods, the most commonly used sensors are cameras [3–8]. In [3], Achtelik *et al.* equipped two MAVs with monocular cameras, which formed a flexible stereo rig. The relative pose of two robots was recovered from this configuration, that is the simplest collaborative framework. Zou *et al.* merged multiple cameras simultaneously to build a global map in [4]. Schneider *et al.* presented a framework, which can merge multiple visual-inertial sequences offline. The vision-based collaborative SLAM applied on MAVs was first validated by Forster *et al.* in [6]. In this work, a centralized ground station collected data from individual MAV and detected overlaps between them. A global coordinate frame was maintained on the ground station by optimizing relative poses. Further on, more similar applications towards MAVs was shown up, such as [7, 8]. Karrer *et al.* validated visual-inertial collaborative SLAM towards multiple MAVs on benchmarking datasets in [8]. These internal vision-based algorithms have shown the potential ability of collaborative localization. However, none of them demonstrated real-time experiments towards a swarm of MAVs based on vision-based localization. Recently, Weinstein *et al.* demonstrated visual-inertial odometry (VIO) swarm on several MAVs in real time in [9]. It's a huge breakthrough for swarm under internal sensors. However, every MAV only estimated its only pose with respect to the individual frame instead of maintaining a global coordinate. Individual frames were just aligned by specified start points, which means they didn't fuse visual observation with each other. The swarm formation suffered from initial alignment error and accumulated VIO drifts.

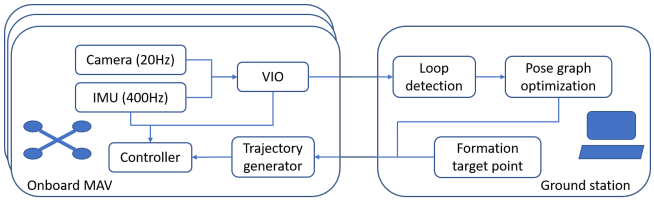


Fig. 1. A diagram of the overall system architecture.

1.3 Contributions

In this paper, we present a cooperative localization system, which provides a globally consistent coordinate towards multiple monocular visual-inertial MAVs. This work is a systematic and experimental extension of our previous monocular visual-inertial estimator and global pose graph optimization [10–12]. We demonstrate the practical capacity of our previous algorithm by conducting real-world experiments towards a swarm of MAVs. Each MAV runs monocular VIO onboard. The centralized ground station collects minimum data from every MAV. The ground station further detects visual overlap and optimizes all individual measurements in a globally consistent coordinate. Finally, the global localization results are sent back to every MAV for swarm behavior. We identify our contribution as twofold:

- We propose a globally cooperative localization framework, which can locate multiple monocular visual-inertial MAVs in a global coordinate and correct accumulated drift for every MAV.
- Real-time swarm application is performed to validate proposed global localization framework.

2 Technical Approach

The architecture of proposed system is shown in Fig. 1. Every MAV is equipped with one IMU and one camera, which supply 400Hz inertial measurements and 20Hz image respectively. The onboard VIO estimator fuses visual and inertial measurements in the tightly coupled optimization, which achieves an accurate estimation of robot's pose, velocity, IMU bias and environmental landmarks. The VIO estimator products 10Hz pose estimation. The 10Hz estimation is propagated with 400Hz IMU to achieve 400Hz low-latency pose feedback for control purpose. The keyframe is compressed into feature descriptors, which consumes low memory. The feature descriptors, along with robot's pose and landmarks' position, are sent to the ground station. The ground station receives visual data from every MAV. It detects internal overlap for one vehicle and external overlap among different vehicles by visual descriptor matching. Once loop connection is detected, global pose graph optimization is performed to eliminate the drift of each MAV and align multiple MAV together.

We define important notations here. Each MAV is numbered with a unique serial number from 1 to n. We use k to denote the serial number. Onboard VIO estimator works in local frame $\{L_k\}$. We use $\{\cdot\}^{L_k}$ to denote variables with respect to k MAV's local frame. The global frame maintained on the ground station is noted by $\{W\}$. The transformation from local frame to global frame is $\mathbf{T}_{L_k}^{W} \in SE(3)$, which is the unknown value needed to be estimated in our framework.

2.1 Monocular Visual-Inertial Odometry (VIO)

The monocular visual-inertial odometry used in this framework is a nonlinear optimization of visual and inertial measurements, which achieves an accurate estimation of robot's pose, velocity, IMU bias and environmental landmarks. It comes from our previous work, VINS-Mono [10]. In fact, any similar keyframe-based VIO algorithms can be used here. Several camera frames and IMU measurements are kept in a bundle. The bundle size usually is limited to bound computation complexity. A local bundle adjustment (BA) jointly optimizing camera and IMU states, as well as landmarks' locations. The VIO estimator products 10Hz pose estimation onboard MAV. The 10Hz estimation is propagated with 400Hz IMU to achieve 400Hz low-latency pose feedback for the controller. The keyframe information is sent to the ground station after efficient compression.

2.2 Keyframe Message

To save bandwidth, every MAV just sends necessary and compact messages to the ground station. Since each MAV tracks its own position using keyframe-based VIO algorithm, the message of keyframe is sent to the ground station. To be specific, keyframe message contains feature descriptors, 3D landmarks and pose of this keyframe. For one keyframe image, we detect 1000 Shi-Tomasi corners [13] and describe them by BRIEF descriptors [14]. These descriptors and their 2D coordinates on the image plane are packed into the message. These descriptors will be used for loop detection on the ground station. Also, VIO estimator constructs approximately one hundred of 3D landmarks. We packed these landmarks, as well as their descriptors into the message. These landmarks will be used to build 3D-2D connection in the following. Finally, we put the pose of this keyframe into the message. The pose is described in MAV's local frame. The main memory consumption is the descriptor. Each BRIEF descriptor takes 32 Byte, so the whole keyframe message takes less than 50 KB, which is much less than one raw image. The average keyframe selection frequency is around 1Hz. So the required bandwidth of every MAV is 50KB/s, which is acceptable for wire and wireless network in real time. In practice, this bandwidth can be lower since we don't need to send a message for every keyframe. In the experiment, we send one keyframe message among every five keyframes, where the required bandwidth is around 10KB/s.

2.3 Loop Detection

The ground station collects keyframe messages from all MAVs. DBoW2 [15], a bag-of-words place recognition approach is used for loop detection. Other similar place recog-

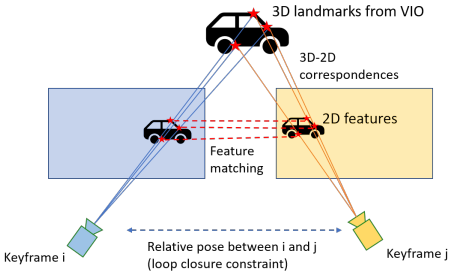


Fig. 2. An illustration of calculating relative pose between two similar keyframes. Considering two similar keyframes i and j, which are collected by k_i and k_j MAVs respectively. By solving the 3D-2D connections (3D landmarks in L_{k_i} frame and 2D features on j frame), we get the pose of j frame with respect in L_{k_i} frame. Hence, we can get the relative pose between i and j, $\hat{\mathbf{T}}_j^i = {\mathbf{T}_i^{L_{k_i}}}^{-1} \mathbf{T}_j^{L_{k_i}}$.

nition approaches can also be used in our framework. Descriptors of every keyframe are treated as the visual word and added into the visual database. DBoW2 returns similar keyframe by visual word matching. Once the similarity is beyond a certain threshold, we try to detect loop connection between two similar frames. We match the descriptors of 3D landmarks (from VIO) on current keyframe with 2D features on the similar keyframe. Directly descriptor matching may cause a lot of outliers. So we evaluate 3D-2D connections by solving PnP problem with RANSAC [16]. Once the number of inlier correspondences beyond a certain threshold, it is a valid loop detection.

Meanwhile, we can get the relative pose of these two similar keyframes by solving PnP problem if they are valid loop frames. An illustration of this process is shown in Fig. 2. Considering two similar keyframes i and j, which are collected by k_i and k_j MAVs respectively. By solving the 3D-2D connections (3D landmarks in L_{k_i} frame and 2D features on j frame), we get the pose of j frame with respect in L_{k_i} frame. Hence, we can get the relative pose between i and j, $\hat{\mathbf{T}}_j^i = {\mathbf{T}_i^{L_{k_i}}}^{-1} \mathbf{T}_j^{L_{k_i}}$. This relative pose will be used for pose graph optimization in the following section.

2.4 Pose Graph Optimization

The pose graph optimization is an extension of our previous work [12]. We specially take consideration of dealing with data collected by different robots simultaneously.

When the ground station receives keyframe messages, keyframes are added into different groups according to their serial numbers. Every keyframe serves as a vertex in the pose graph, and it connects with other vertexes by two types of edges.

– **Sequential Edge** This edge only exists inside one group. A keyframe establishes several sequential edges (here 4) to its previous keyframes in the same group. A sequential edge represents the relative transformation between two neighbor keyframes, which is taken directly from VIO. Considering keyframe i and one of

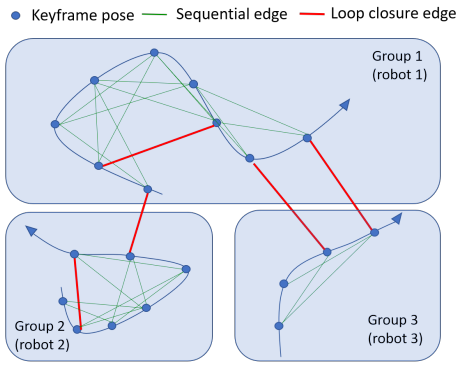


Fig. 3. An illustration figure of pose graph. Every keyframe pose serves as a vertex. It connects its neighbor vertexes by sequential edges in the same group. It connects loop closure frame by loop closure edge. The loop closure edge can be inside one group and between different groups.

its previous keyframes j in group k, the relative pose can be derived as $\hat{\mathbf{T}}_j^i = {\mathbf{T}_i^{L_k}}^{-1}\mathbf{T}_j^{L_k}$.

– **Loop Closure Edge** This edge exists both inside one group and between different groups. If the keyframe has a loop connection, it connects loop closure frame by a loop closure edge in the pose graph. Similarly, the loop closure edge represents the relative pose between two similar frames, which comes from Sect. 2.3.

The illustration figure of pose graph is depicted in Fig 3. Every keyframe pose serves as a vertex. It connects its neighbor vertexes by sequential edges in the same group. It connects loop closure frame by loop closure edge. The loop closure edge can be inside one group and between different groups.

The variables in the pose graph are transformations from every local frame to global frame $T_{L_k}^W$ and keyframes' poses in every local frame $T_*^{L_k}$. We adjust these variables, such that they match sequential edges and loop closure edges as much as possible. The cost function of pose graph optimization can be derived as:

$$\min_{\mathbf{T}_{L_k}^W, \mathbf{T}_*^{L_k}} \sum_{k=1}^{n} \sum_{(i,j)\in\mathcal{S}_k} \left\| \hat{\mathbf{T}}_j^i \ominus ({\mathbf{T}_i^{L_k}}^{-1}\mathbf{T}_j^{L_k}) \right\|^2 + \sum_{(i,j)\in\mathcal{L}} \rho\left(\left\| \hat{\mathbf{T}}_j^i \ominus (\mathbf{T}_{L_i}^W \mathbf{T}_i^{L_i})^{-1} \mathbf{T}_{L_j}^W \mathbf{T}_j^{L_j} \right\|^2 \right), \tag{1}$$

where $\mathcal{S}_k$ is the set of all sequential edges for k group, and $\mathcal{L}$ is the set of all loop closure edges inside and between groups. we add another Huber norm $\rho(\cdot)$ to further reduce the impact of any possible wrong loops. The notation $\ominus$ denotes the sum of position residual and angle residual. Intuitively, the loop closures edges inside one group are used to correct accumulated drift for one MAV. The loop closure edges between different groups are used to align different local frames into a global frame. Such that we can maintain a globally consistent coordinate. In practice, we treat the first local frame as global frame ($\mathbf{T}_{L_1}^W = \mathbf{I}$). In other words, we align the local frames of other robots with the first robot.

3 Experimental Results

In order to validate proposed system, one experimental evaluation and one swarm application is performed.

3.1 Sensors & Hardware Architecture

Sensors and hardware platform that we used is shown in the Fig 4. For each MAV, the sensor set contains a monocular camera (mvBlueFOX-MLC200w, 20Hz) and an IMU (400Hz) inside the DJI A3 controller[1]. The camera captures 752×480 image with a fisheye lens. The MAV structure is self-designed and 3D printing. DJI A3 controller is used for low-level attitude control. The onboard computer is NVIDIA TX2, which is equiped with quad-core CPU (ARM A57 at 2GHz). Only CPU is used on onboard computer. The VIO runs onboard computer at 10Hz. The ground station includes an Intel i7-3770 CPU at 3.40GHz. The pose graph optimization runs on ground station in real time.

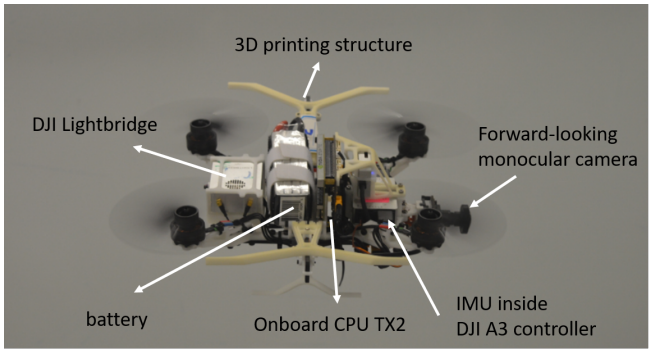


Fig. 4. A picture of sensors and hardware platform. One monocular camera (mvBlueFOX-MLC200w, 20Hz) and an IMU (400Hz) inside the DJI A3 controller are equipped on every MAV.

3.2 Experimental Verification

We first validate our algorithm's ability to perform global localization with multiple robots. In this experiment, we collect four sequences in the indoor environment, using above-mentioned sensor set. Each sequence is collected at different and unknown start points with unknown attitude. It means that each VIO initializes in different positions with different yaw angles. So their local frames are misaligned and can't be merged directly. The four sequences sent messages to the ground station at the same time. The

[1] `http://www.dji.com/a3`

ground station extracts relative transformations of each local frame from loop closure, and registers local frames into a global frame. We set the first sequence's local frame as the global frame. Other sequences are aligned with the first sequence in the pose graph optimization. For metric evaluation, we take OptiTrack[2] as ground truth, which is an external motion capture system. The OptiTrack provides accurate localization results in a globally consistent frame.

The VIO results on every local frame are shown in Fig. 5 respectively. The four sequences are collected in the same indoor environmental. The scene in four sequences is overlapped. Many parts of these four trajectories are repeated. However, these four trajectories cannot be merged directly, since their origin points and origin angles are different. Our system merged them together by proposed loop detection and pose graph optimization, as shown in Fig. 6(a). Compared with ground truth 6(b), the proposed pose graph optimization correctly align four trajectories into a global coordinate. The proposed system not only aligns multiple local coordinates, but also correct accumulated drift to achieve consistency. To be specific, we evaluate the numerical accuracy of pure VIO and pose graph results. The VIO and pose graph results are evaluated by absolute trajectory error (ATE) [17]. The RMSE is shown in Table. 1. It can be seen that the RMSE from pose graph is obviously smaller than VIO. In other words, the pose graph optimization corrects accumulated drift for every sequence efficiently. Hence, a globally consistent localization is achieved by our system.

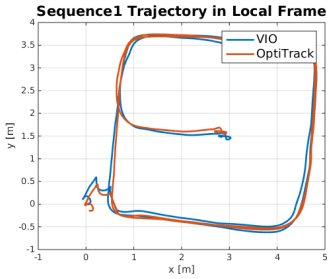
Table 1. RMSE of absolute trajectory error (ATE) in meters.

	Seq 1	Seq 2	Seq 3	Seq 4	Average
Local VIO	0.136	0.178	0.087	0.134	0.138
Global pose graph	0.106	0.082	0.084	0.120	0.100

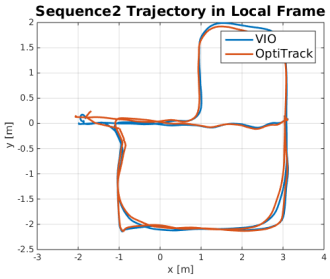
3.3 Swarm Application

The swarm application is performed to show the practicability and robustness of our system. Four MAVs were used due to the space limitation of flying area. The minimal inter-robot distance is set as 1.5m to avoid the collision. These MAVs were initialized at an arbitrary position. To achieve loop closure, they looked at the same direction. Their forward-looking cameras detected same natural features without any artificial landmarks. After a few seconds, their local frames were aligned to a global frame. Then we perform swarm formation flight. We set the target points for every MAV. Each MAV planed a smooth trajectory on board. A simple PID controller is used to force the MAV following the trajectory.

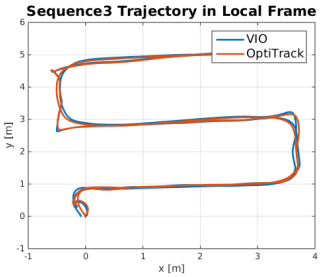
[2] https://optitrack.com/



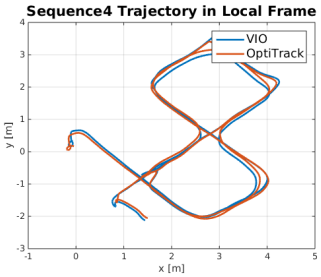
(a) VIO Trajectory of sequence 1



(b) VIO Trajectory of sequence 2

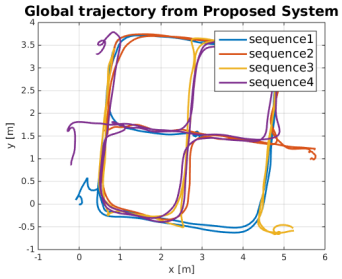


(c) VIO Trajectory of sequence 3

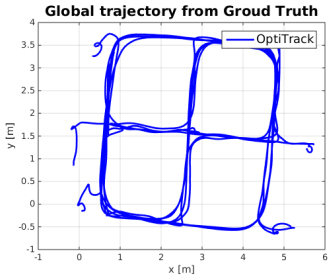


(d) VIO Trajectory of sequence 4

Fig. 5. The VIO results of every sequence on its local frame.

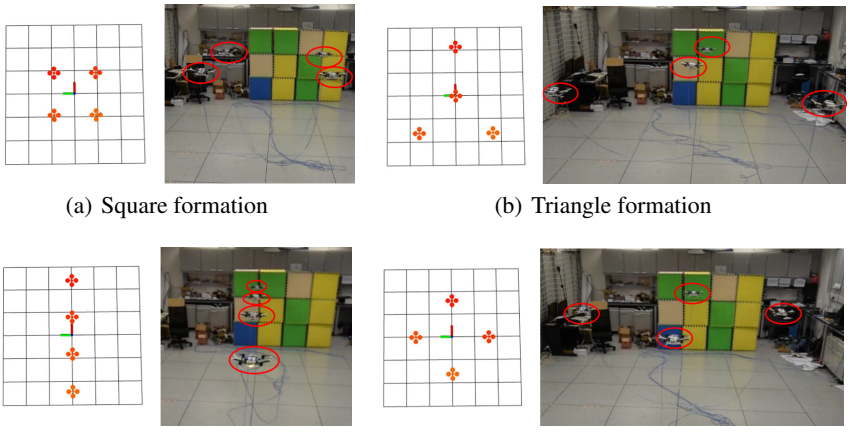


(a) Four trajectories are aligned in a global frame in proposed system.



(b) Four trajectories from OptiTrack

Fig. 6. Global trajectory from proposed system compares against ground truth.



(a) Square formation (b) Triangle formation

(c) Line formation (d) Cross formation

Fig. 7. Designed formations and snapshots of real-world swarm formations.

The formation flight includes several formations, as shown in Fig. 7. A designed formation and a snapshot of real swarm flight are listed. The four MAVs formed square, triangle, line, and cross formations. The details can be found in the supplementary video.

4 Conclusions and Future Work

In this work, we present a novel framework of collaborative localization, which can achieve global localization for multiple monocular visual-inertial MAVs. Unlike other swarm application that relies on the external positioning system, our algorithm depends only on internal sensors. Experiments verify the accuracy and effectiveness of our system. We further demonstrate the successful application of our algorithm on multiple MAVs to achieve swarm behaviors. Future works will focus on a decentralized collaborative framework which achieves the same globally consistent localization without the centralized ground station.

References

1. Kushleyev, A., Mellinger, D., Powers, C., Kumar, V.: Towards a swarm of agile micro quadrotors. Autonomous Robots **35**(4) (2013) 287–300
2. Ritz, R., Müller, M.W., Hehn, M., D'Andrea, R.: Cooperative quadrocopter ball throwing and catching. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2012) 4972–4978
3. Achtelik, M.W., Weiss, S., Chli, M., Dellaerty, F., Siegwart, R.: Collaborative stereo. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2011) 2242–2248
4. Zou, D., Tan, P.: Coslam: Collaborative visual slam in dynamic environments. IEEE transactions on pattern analysis and machine intelligence **35**(2) (2013) 354–366

5. Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., Siegwart, R.: maplab: An open framework for research in visual-inertial mapping and localization. IEEE Robotics and Automation Letters **3**(3) (2018) 1418–1425
6. Forster, C., Lynen, S., Kneip, L., Scaramuzza, D.: Collaborative monocular slam with multiple micro aerial vehicles. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2013) 3962–3970
7. Schmuck, P.: Multi-uav collaborative monocular slam. In: Proc. of the IEEE Int. Conf. on Robot. and Autom. (2017) 3863–3870
8. Karrer, M., Chli, M.: Towards globally consistent visual-inertial collaborative slam. In: Proc. of the IEEE Int. Conf. on Robot. and Autom. (2018)
9. Weinstein, A., Cho, A., Loianno, G., Kumar, V.: Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors. IEEE Robotics and Automation Letters **3**(3) (2018) 1801–1807
10. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. arXiv preprint arXiv:1708.03852 (2017)
11. Qin, T., Shen, S.: Robust initialization of monocular visual-inertial estimation on aerial robots. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2017)
12. Qin, T., Li, P., Shen, S.: Relocalization, global optimization and map merging for monocular visual-inertial slam. In: Proc. of the IEEE Int. Conf. on Robot. and Autom. (2018)
13. Shi, J., Tomasi, C.: Good features to track. In: Proc. of the IEEE Int. Conf. on Pattern Recognition, Seattle, WA (June 1994) 593–600
14. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010 (2010) 778–792
15. Gálvez-López, D., Tardós, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics **28**(5) (October 2012) 1188–1197
16. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. International journal of computer vision **81**(2) (2009) 155–166
17. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2012) 573–580