Towards Globally Consistent Visual-Inertial Collaborative SLAM

Towards Globally Consistent Visual-Inertial Collaborative SLAM

Marco Karrer and Margarita Chli
Vision for Robotics Lab, ETH Zurich, Switzerland

***Abstract*— Motivated by the need for globally consistent tracking and mapping before autonomous robot navigation becomes realistically feasible, this paper presents a novel back-end to monocular-inertial odometry. As some of the most challenging platforms for vision-based perception, we evaluate the performance of our system using Unmanned Aerial Vehicles (UAVs). Our experimental validation demonstrates that the proposed approach achieves drift correction and metric scale estimation from a single UAV on benchmarking datasets. Furthermore, the generality of our approach is demonstrated to achieve globally consistent maps built in a collaborative manner from two UAVs, each equipped with a monocular-inertial sensor suite, showing the possible gains opened by collaboration amongst robots to perform SLAM.**
Video – `https://youtu.be/wbX36HBu2Eg`

I. Introduction

One of the key pre-requisites in the quest of employing mobile robots with navigational autonomy is the development of their ability to perceive their workspace and estimate their ego-motion within it, which is commonly referred to as Simultaneous Localization And Mapping (SLAM). While initial attempts to address SLAM have been utilizing range sensors, it was the emergence of monocular and real-time capable SLAM systems, such as [6] and [12] that paved the way towards the use of SLAM onboard small Unmanned Aerial Vehicles (UAVs). The employment of Visual-Inertial (VI) sensing cues and the successful demonstration of vision-controlled flights using onboard sensing only [21], rendered this sensor suite as the standard choice for the control and navigation of small aircrafts.

With increasing maturity and robustness in this field, two state of the art methods for Visual-Inertial Odometry (VIO) open-sourced their implementations, namely OKVIS [15] and ROVIO [3]. Such systems permit reliable state estimation even during complicated UAV maneuvers. However, as these algorithms are only local, the current UAV pose that is being estimated is prone to drift over longer trajectories. Aiming to address drift during real-time monocular state estimation, ORB-SLAM [16] pushed the state of the art, tackling large-scale loop correction at an unprecedented robustness and accuracy in monocular systems. Incorporating additional inertial data to the monocular setup, the most recent VI-ORB-SLAM [17] was the first VI-SLAM system capable of correcting drift via loop-closure detection and optimization, while maintaining an estimate of metric scale with high accuracy. Despite constituting a milestone, VI-ORB-SLAM remains closed source and based on the authors' evaluation [17] as the only source of information, its accuracy is reportedly fluctuating across different datasets, highlighting the need for deeper analysis in VI-SLAM.

This research was supported by EC's Horizon 2020 Programme under grant agreement n. 644128 (AEROWORKS) and the Swiss National Science Foundation (SNSF, Agreement no. PP00P2_157585)

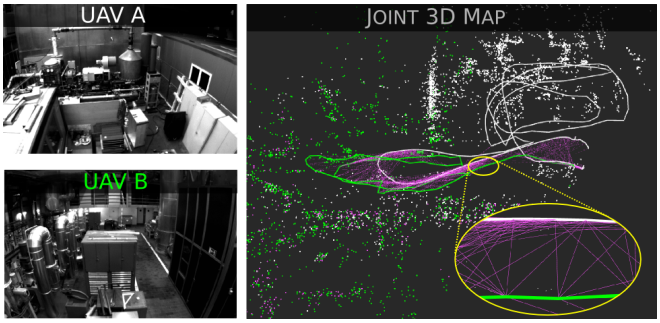

Fig. 1: A snapshot of the proposed system in a collaborative setup with two UAVs. On the left the viewpoints from each UAV are shown, while on the right is the joint 3D map built collaboratively.Trajectories and landmarks are colored in white and green for UAV A and UAV B, respectively. Landmarks that are shared across both UAVs are indicated in magenta, while covisibility edges, connecting keyframes accross the two UAVs are also in magenta, magnified in the inset for clarity.

Moving on from single-robot SLAM systems, the community started making the first steps towards investigating collaborative SLAM in multi-robot scenarios. While [22] for example, leverages the multi-camera setup with view overlap to perform SLAM in challenging dynamic scenes, [8] and [19] explore the advantages of employing multiple UAVs equipped with cameras for efficient mapping and collaborative SLAM, respectively. Due to the lack of metric measurements (e.g. inertial data), these systems can only provide estimates up to scale. Instead, the approach in [1] for collaborative stereo from two UAVs is capable of estimating the relative pose of two VI systems in simulation, albeit avoiding to address the global consistency of the estimation processes.

While the aforementioned open-sourced VIO systems have been very influential in robot navigation, their inevitable tendency to drift, limits their applicability in real scenarios, where global state estimation is required. In this spirit, we present a carefully designed back-end, which in combination with a nominal VIO system enables the generation of a globally consistent map at comparable accuracy with the state of the art VI-SLAM systems – at times even achieving error reduction of over 50%, solely considering the back-end optimization. Moreover, here we go a step further to illustrate the use of the proposed back-end with two UAVs to achieve collaborative mapping, while correcting for drift

upon loop-closures, both within each trajectory as well as across trajectories of different UAVs as shown in Figure 1. This paper outlines a new, complete back-end system in enough detail to enable reproducability of the proposed system, employable in combination with an off-the-shelf VIO system requiring only minimal modification. Furthermore, our evaluation on benchmarking datasets reveals that the proposed framework can achieve significant improvement in accuracy over the state of the art.

II. Preliminaries

A. Notation

In this paper, we use bold capital letters for matrices (e.g. $\boldsymbol{A}$), bold small letters for vectors (e.g $\boldsymbol{a}$), and capital letters for coordinate frames (e.g. A), while sets of variables are denoted by calligraphic letters (e.g. $\mathscr{A}$). A rigid body transformation from coordinate frame B to A is denoted by $\boldsymbol{T}_{AB}$, while the rotational part of any transformation $\boldsymbol{T}$ is denoted by $\boldsymbol{R}$ and the translational part by $\boldsymbol{t}$. A vector $\boldsymbol{x}$ expressed in coordinate frame A is written as ${}_A\boldsymbol{x}$. The origin (i.e. the inertial frame) of the system is denoted by W (also referred to as the world frame), the camera coordinate system by C, and the IMU body frame by S.

B. IMU Model and State Representaion

It is well known that readings from MEMS-IMUs do not capture the true acceleration and rotational velocity, but rather a biased version of them. While some errors, such as cross-couplings and scaling errors are constant and can be compensated for via factory calibration, other influences are time-variant and need to be estimated online. In order to model the IMU measurements, we use the standard measurement model, assuming that the accelerometer ${}_S\boldsymbol{a}(t)$ and the gyroscope measurements ${}_S\boldsymbol{\omega}_{WS}(t)$ are both corrupted by additive white noise $\boldsymbol{\eta}$ and have a sensor biases $\boldsymbol{b}$, which are assumed to be varying slowly over time (t), such that:

$$ {}_S\boldsymbol{a}(t) = \boldsymbol{R}_{WS}^{\mathsf{T}}(t)\left({}_W\hat{\boldsymbol{a}}(t) - {}_W\boldsymbol{g}\right) + \boldsymbol{b}_a(t) + \boldsymbol{\eta}_a(t) \ , \quad (1) $$

$$ {}_S\boldsymbol{\omega}_{WS}(t) = {}_S\hat{\boldsymbol{\omega}}_{WS}(t) + \boldsymbol{b}_g(t) + \boldsymbol{\eta}_g(t) \ . \quad (2) $$

The notation $\hat{\cdot}$ signifies the true values of the respective variables, while ${}_W\boldsymbol{g}$ is the gravity vector in the inertial frame. We differentiate the accelerometer-specific variables from the gyroscopic ones via the subscripts a and g, respectively.

Due to the characteristics of the IMU measurements, the state of the system $\boldsymbol{\Theta}$ includes the poses $\{\boldsymbol{R}_{WS}, \boldsymbol{t}_{WS}\}$ of all the keyframes (KF) in the trajectory, the positions ${}_W\boldsymbol{l}$ of all of the landmarks ever experienced, as well as the linear velocities ${}_W\boldsymbol{v}$ and bias terms $\boldsymbol{b}$:

$$ \boldsymbol{\Theta} \coloneqq \{\underbrace{\boldsymbol{R}_{WS}^k, \boldsymbol{t}_{WS}^k, {}_W\boldsymbol{v}^k, \boldsymbol{b}^k}_{\text{KF}_k}, {}_{S_r}\boldsymbol{l}^i\} \quad \forall k \in \mathscr{V}, \forall i \in \mathscr{L} \ , \ (3) $$

where $\mathscr{V}$ is the set of all keyframes and $\mathscr{L}$ is the set of all landmarks. Instead of expressing the landmarks in the global reference frame (W), we express them in local coordinates of a reference KF S_r as proposed in [2]. In combination with an inverse-depth parametrization [5], this aims at improving the conditioning of the problem during the optimization. However, for the sake of readability, we will treat the landmarks as if they were expressed in Euclidean coordinates. In this paper, we refer to individual state variables as θ_j.

III. Method

We consider the setup of two UAVs equipped with a monocular camera and an inertial sensor each, experiencing the world at the same time, while exhibiting an overlap in their fields of view. Following this paradigm, this section gives an overview of the proposed system to arrive to a joint, globally consistent map of the UAVs' surroundings and their relative poses within it.

A. System Overview

The proposed system, illustrated in Figure 2, employs a front-end Visual-Inertial Odometry (VIO) module onboard each UAV and then processes all information gathered from the UAVs to perform Landmark Matching and Mapping, Loop-Closure Detection, and Local and Global Bundle Adjustment (BA) on all estimates. VIO ensures a stable pose estimation of each UAV in six Degrees of Freedom (DoF) and is expected to drift, but can be used to safely stabilize the UAV. The decoupling of the VIO from the rest, the map management threads that can run on a ground station; VIO communicates keyframe (KF) messages to the back-end. While the absence of feedback from the global map to the VIO prohibits direct corrections of the VIO's state upon map changes, it enables the use of an off-the-shelf VIO with only minimal modifications. Furthermore, as transformation between the global map's and the VIO's coordinate can be easily estimated, the UAV would still be able to make use of corrections, as e.g. presented by [18].

During Landmark Matching and Mapping (in cyan in Figure 2), past observations get associated with the landmarks in the resulting, joint map from both UAVs, while new landmarks get initialized in this map. The map comprises of a set of 3D landmarks and KFs, where each KF consists of the corresponding UAV pose, a set of 2D observations and the landmarks visible from it. Each landmark in the map stores the KF-IDs that have observed it, an estimate of the local surface normal based on the viewing angles of all corresponding observations, as well as the most representative image descriptor for this landmark across all observations, as proposed in [16] – this aims to increase the re-detectability of the landmarks.

A Covisibility Graph is maintained throughout each session, with nodes corresponding to individual KFs. Two nodes share an edge if the corresponding KFs share a minimum number of landmark observations ($\alpha_{min} = 12$ in our implementation), and each edge is associated with a weight α reflecting the number shared landmark observations. An Essential Graph is also maintained (this notion was first introduced in [16]), which is of similar structure to the Covisibility Graph, only preserving the most essential information, by restricting edges even more (e.g. $\alpha_{min} = 100$). In addition to the purely spatial KF covisibility, we also keep track of

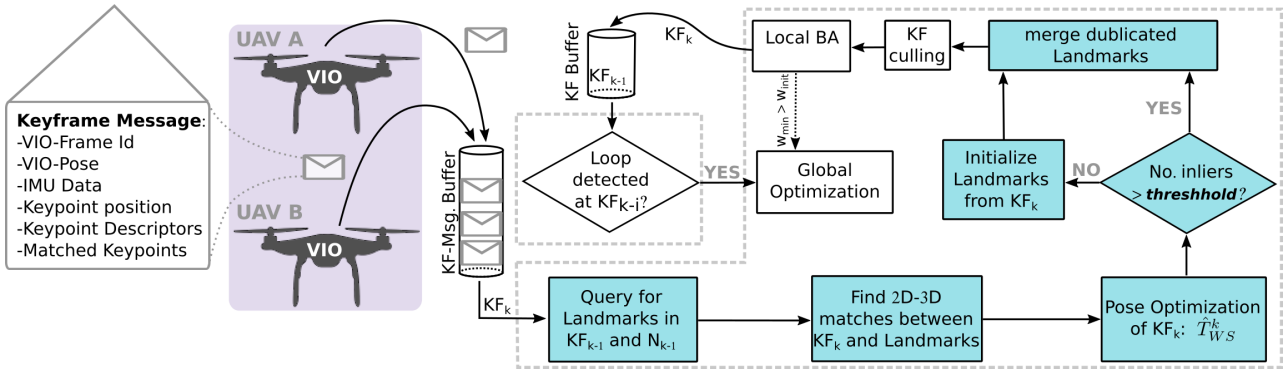

Fig. 2: A schematic of the proposed pipeline to fuse the experiences of multiple UAVs into a joint, globally consistent map, by reusing information obtained by the Visual Inertial Odometry (VIO) running onboard each UAV (in purple). At first, correspondences between keyframes and landmarks are established and new landmarks get initialized (boxes in cyan). The scene structure and UAVs' poses are then optimized frequently on a local scope and upon detection of loop closures, optimization is performed on a global scale.

the temporal predecessor of each KF, distinguishing the agent (i.e. here the UAV), from which the KF originates, as this necessary for the constraints used to obtain metric scale.

Since Loop-Closure Detection is mostly independent of Mapping and Local BA they run in separate threads. However, in case of a loop-closure, the system waits until the Local BA for the current KF has finished, and then triggers the loop correction, blocking the processing of new KFs until the map is updated with the result of the Global BA. At the core of the proposed system is the optimization of both the KF poses and scene structure (i.e. landmarks) simultaneously, including any IMU readings obtained between consecutive KF poses. This aims at recovering each UAV's trajectory in metric scale. Local BA is performed for more frequent small-scale corrections, while Global BA is used to optimize all poses and landmarks obtained from all participating agents.

B. Visual-Inertial-Odometry Input

The proposed system is generally independent of the choice of the VIO pipeline used, with the only requirement of providing metrically scaled current poses and corresponding 2D observations. For the experiments presented in this paper, the publicly available VIO system OKVIS [15] is employed with some adaptations to reuse the matching results. OKVIS performs a joint, non-linear optimization over a constant number of KFs, including inertial measurements.

Packing the relevant information for the current KF as provided by VIO into KF-messages, these then serve as input to the proposed back-end framework. Each KF-message encloses the current KF's pose, the IMU readings since the last KF, the locations of the current KF's kyepoints in the image, and their corresponding descriptors – thus, eliminating the need for sending full images. Each KF-message also includes a list of global identifiers of its associated keypoints, enabling tracing of the KF's keypoints back to older KFs that they were matched from, within the local window (of retained KFs) of VIO. This enables re-use of data associations as discussed in the next section. Note that this latter part of a KF-message is optional, as for example, filter based VIO systems (e.g. [3]) may not have this information available. As the KFs arrive at an unknown rate, we store arriving KF-messages within a first-in-first-out buffer, before processing them sequentially.

C. Landmark Matching and Mapping

Landmark matching consists of establishing correspondences of the current frame to the existing landmarks (via 3D-2D matching) and the creation of new correspondences (via 2D-2D matching) across different KFs. In order to establish matches to existing landmarks, every observation in the current KF is checked for correspondences with the past KFs via the global landmark identifiers listed in the KF-message. Before accepting a new correspondence, this is checked for consistency in terms of the reprojection error and the descriptor distance within the map. In order to establish additional 3D-2D correspondences, or in case the VIO system at hand does not provide matching information, the system uses the relative transformation $\boldsymbol{T}_{k-1,k}$ between the current keyframe (KF$_k$) and the previous KF (KF$_{k-1}$) stemming from the same agent, as estimated by the VIO. Given an estimate for the pose $\boldsymbol{T}^{k-1}_{WS}$ of KF$_{k-1}$, the system predicts the current pose ($\hat{\boldsymbol{T}}^k_{WS}$) as

$$\hat{\boldsymbol{T}}^k_{WS} = \boldsymbol{T}^{k-1}_{WS} \cdot \boldsymbol{T}_{k-1,k} \ . \tag{4}$$

As a result, all landmarks predicted to be visible in KF_k from KF$_{k-1}$ and its first-order neighbors (N$_{k-1}$) in the Covisiblity Graph, are projected in it. Similarly to [16], the search for matching observations is restricted within a radius around the predicted projection of a landmark, while a correspondence is established to the observation with the smallest descriptor distance. In case of multiple landmarks matching to the same 2D observation, only the correspondence to the landmark with the biggest number of observations is established, or the landmark with the smallest descriptor distance, if the first criterion is inconclusive. This process is performed first using a large radius (i.e. for coarse matching) followed by solving the P3P problem as in [13] for a number of RANSAC iterations (40 iterations) for outlier

filtering on the initial correspondences. The projection based matching is then repeated, using the pose obtained by the previous RANSAC step, with a more restrictive radius to find additional matches. Using all the established correspondences, the current KF pose is refined by minimizing the reprojection error of the matched landmarks in the current KF, while keeping the landmark position fixed.

Initialization of new landmarks is only performed when the UAV is in an *exploratory state*, which is determined by a minimum number on the 3D-2D inlier correspondences found (here 60). In order to initialize new landmarks, we first attempt to triangulate the remaining correspondences obtained by the VIO system (using their identifiers), for which no 3D association was found. At a second stage, new matches of unassociated observations are searched for. The candidate frames used for match searching are extracted again as the first-order neighbors of the previous keyframe (KF_{k-1}) in the Covisibility Graph. We only attempt to match observations corresponding to the same visual word as computed by the loop-closure detector, rendering the matching more efficient than brute force. All matches found are checked for consistency before inserting their correspondence as landmarks into the map. For every newly inserted landmark, we set its reference KF to the more recent one used to perform the triangulation. Due to this two-stage correspondence search, our system is capable of running without the need for matches obtained by the VIO system.

In a cleanup step, duplicated landmarks get merged by projecting all landmarks associated in KF_k to the covisible KFs and matches are searched for in the same fashion as for the initial 3D-2D matching. In case different landmarks are associated to one observation, they get merged into one landmark, i.e. the one with most observations associated to it. When there is no landmark associated with an observation, a new correspondence with that landmark is established.

D. Factor Graph Formulation

Keyframe-based VI-SLAM can be formulated as a factor graph [14], where the variable nodes θ_j represent the system state, and factor nodes f_i are given by the relation of measurements and the variables (observations). The factor graph defines the factorization of a function $f(\boldsymbol{\Theta})$ as

$$f(\boldsymbol{\Theta}) = \prod_i f_i(\mathscr{A}_i) \ , \tag{5}$$

where $\mathscr{A}_i$ represents the set of variable nodes affected by the factor f_i. The goal is to find the values of the variables $\boldsymbol{\Theta}^*$, which maximize the factorization function Equation (5). Under the usual assumption that observations are corrupted by zero-mean gaussian noise (gaussian measurement model), the problem can be stated as

$$\begin{aligned}
\boldsymbol{\Theta}^* &= \arg\max_{\boldsymbol{\Theta}} \{f(\boldsymbol{\Theta})\} = \arg\min_{\boldsymbol{\Theta}} \{-\log f(\boldsymbol{\Theta})\} \\
&= \arg\min_{\boldsymbol{\Theta}} \left\{ -\log \prod_i \exp\left(-\frac{1}{2} \| \boldsymbol{z}_i - h_i(\mathscr{A}_i) \|^2_{\boldsymbol{\Sigma}_i} \right) \right\} \\
&= \arg\min_{\boldsymbol{\Theta}} \left\{ \sum_i \| \boldsymbol{z}_i - h_i(\mathscr{A}_i) \|^2_{\boldsymbol{\Sigma}_i} \right\} \\
&= \arg\min_{\boldsymbol{\Theta}} \left\{ \sum_i \boldsymbol{e}_i^\intercal \boldsymbol{\Sigma}_i^{-1} \boldsymbol{e}_i \right\} = \arg\min_{\boldsymbol{\Theta}} \left\{ \sum_i \boldsymbol{e}_i^\intercal \boldsymbol{W}_i \boldsymbol{e}_i \right\} \ ,
\end{aligned} \tag{6}$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{x}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$ denotes the squared Mahalanobis distance, $\boldsymbol{e}_i$ represents the residual error, $\boldsymbol{\Sigma}_i$ the covariance matrix, and $\boldsymbol{W}_i$ the information matrix of the measurement i. In this paper, we use the residual notation to describe the objective function we are looking to minimize. Within VI-SLAM, we essentially use 3 different types of factors, which are introduced below based on the corresponding residual error terms for the factors.

Reprojection Factor. Given the position of a landmark ${}_{S_r}\boldsymbol{l}^j$ expressed in KF_r and the corresponding keypoint observation $\boldsymbol{z}^{k,j}$ in the image coordinates of KF_k, we define the reprojection error as

$$\boldsymbol{e}_r^{k,j} \coloneqq \boldsymbol{z}^{k,j} - h\left(\boldsymbol{K}^k \boldsymbol{T}_{CS} \boldsymbol{T}_{SW}^k \boldsymbol{T}_{WSS_r}^r \boldsymbol{l}^j \right) \ , \tag{7}$$

where $h(\cdot)$ converts homogeneous coordinates into image measurements and $\boldsymbol{K}$ is the camera matrix. Since we use undistorted keypoint coordinates the error function does not contain a distortion model.

IMU pre-integration Factor. Given a set of IMU (accelerometer and gyroscope) readings between two subsequent KFs, we can perform integration of the measurements with an initial estimate of the bias terms as in [7], which later can be optimized without the need to perform numerical re-integration of the raw measurements. With a given estimate of the pre-integration, the resulting residuals can be written as

$$\begin{aligned}
\boldsymbol{e}_{\Delta \boldsymbol{R}}^{k-1,k} &= \log\left(\left(\Delta\tilde{\boldsymbol{R}}_{k-1,k}(\bar{\boldsymbol{b}}_g^{k-1}) \exp\left(\frac{\partial \Delta\bar{\boldsymbol{R}}_{k-1,k}}{\partial \boldsymbol{b}_g} \delta\boldsymbol{b}_g \right) \right)^\intercal \right. \\
&\qquad \left. \boldsymbol{R}_{WS}^{k-1\intercal} \boldsymbol{R}_{k-1,k} \right) \\
\boldsymbol{e}_{\Delta \boldsymbol{v}}^{k-1,k} &= \Delta\boldsymbol{R}_{WS}^{k-1\intercal} \left({}_W\boldsymbol{v}^k - {}_W\boldsymbol{v}^{k-1} - {}_W\boldsymbol{g}\Delta t_{k-1,k} \right) \\
&\quad - \left(\Delta\tilde{\boldsymbol{v}}_{k-1,k}(\bar{\boldsymbol{b}}) + \frac{\partial \Delta\bar{\boldsymbol{v}}_{k-1,k}}{\partial \boldsymbol{b}_a} \delta\boldsymbol{b}_a + \frac{\partial \Delta\bar{\boldsymbol{v}}_{k-1,k}}{\partial \boldsymbol{b}_g} \right) \\
\boldsymbol{e}_{\Delta \boldsymbol{t}}^{k-1,k} &= \boldsymbol{R}_{WS}^{k-1,k\intercal} \left(\Delta\boldsymbol{t}_{k-1,k} - {}_W\boldsymbol{v}^{k-1}\Delta t_{k-1,k} - \frac{1}{2}\boldsymbol{g}\Delta t_{k-1,k}^2 \right) \\
&\quad - \left(\Delta\tilde{\boldsymbol{t}}_{k-1,k}(\bar{\boldsymbol{b}}^{k-1}) + \frac{\partial \Delta\bar{\boldsymbol{t}}_{k-1,k}}{\partial \delta\boldsymbol{b}_a} \delta\boldsymbol{b}_a + \frac{\partial \Delta\bar{\boldsymbol{t}}_{k-1,k}}{\partial \delta\boldsymbol{b}_g} \delta\boldsymbol{b}_g \right) \ ,
\end{aligned} \tag{8}$$

where $\tilde{\cdot}$ denotes values obtained by the current estimate of the pre-integration and $\bar{\cdot}$ denotes values obtained with the bias $\bar{\boldsymbol{b}}$ used at the time that the integration was performed. For more detailed explanation of pre-integration the reader is kindly referred to [7]. The scalar $\Delta t_{k-1,k}$ represents the integration time between KF_{k-1} and KF_k. As a result, the residual terms of Equation (8) are as follows

$$\boldsymbol{e}_a^{k-1,k} = \left[\boldsymbol{e}_{\Delta \boldsymbol{R}}^{k-1,k\intercal}, \boldsymbol{e}_{\Delta \boldsymbol{v}}^{k-1,k\intercal}, \boldsymbol{e}_{\Delta \boldsymbol{t}}^{k-1,k\intercal} \right]^\intercal \ . \tag{9}$$

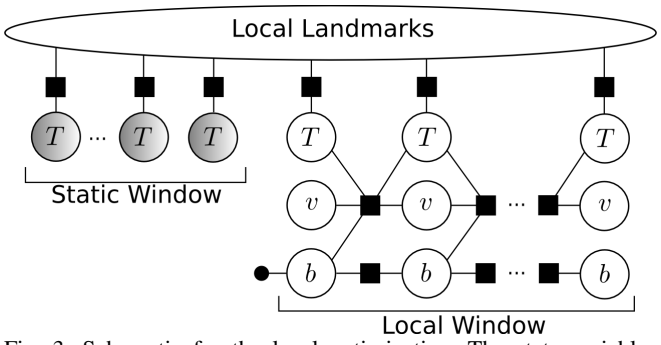

Fig. 3: Schematic for the local optimization. The state variables participating in the optimization are shown in clear circles, whereas static variables are shaded. The Factors in the optimization (residuals) are shown as squares and a prior on a variable is drawn as a small disc.

Prior Factor. Given prior knowledge of a variable θ at time t_k, the residual for a prior factor is the difference between the prior knowledge $\bar{\theta}^k$ and the estimate θ^k:

$$\boldsymbol{e}_\theta^k = \bar{\theta}^k - \theta^k \ . \tag{10}$$

Note that for non-Euclidean variables (i.e. rotations) the minus operation has to be adapted to the commonly used box-minus operator, as shown in [10].

E. Local Bundle Adjustment (BA)

Since a full BA quickly becomes computationally infeasible for real-time or close to real-time applications, local optimization is performed frequently as in most KF-based SLAM systems today [16], [20]. While for pure visual SLAM, it is well-established as shown by [16] that selecting the local optimization window based on covisibility is a reasonable choice, the situation in the case of VI-SLAM is different, as a temporal ordering of the keyframes is of crucial importance in order to obtain well defined constraints formed by the IMU cues.

In this work, we employ a strategy similarly to [17], where the the local optimization window of KFs is defined as the set of the N most recent KFs as illustrated in Figure 3. In the multi-agent case, we consider the last N KFs stemming from the same agent as KF_k. In addition to the KFs in the Local Window, KFs that share observations with the Local Landmarks visible in the Local Window are placed as fixed variables in the optimization (Static Window). For landmarks with only two observations, we check the KF within the Local Window whether it is the last one inside the window, in which case the landmark is completely deleted from the map, as it is unlikely to be re-detected. By doing so, landmark culling is performed by design without the need for further bookkeeping of the landmark observations.

Since the optimization of the bias terms is limited to the Local Window, we impose a prior on the N^{th} KF in order to constraint the variation of the bias. Therefore, the objective function for the local BA in terms of residuals can be written as

$$\begin{aligned} J(\boldsymbol{\Theta}) \coloneqq & \boldsymbol{e}_b^{N\intercal} \boldsymbol{W}_b^N \boldsymbol{e}_b^N + \sum_{k\in\mathscr{V}} \sum_{j\in\mathscr{L}(j)} \delta\left(\boldsymbol{e}_r^{k,j\intercal} \boldsymbol{W}_r^{k,j} \boldsymbol{e}_r^{k,j}\right) \\ & + \sum_{k-1,k\in\mathscr{V}\setminus\mathscr{V}_s} \boldsymbol{e}_a^{k-1,k\intercal} \boldsymbol{W}_a^{k-1,k} \boldsymbol{e}_a^{k-1,k} \\ & + \sum_{k-1,k\in\mathscr{V}\setminus\mathscr{V}_s} \boldsymbol{e}_b^{k-1,k\intercal} \boldsymbol{W}_b^{k-1,k} \boldsymbol{e}_b^{k-1,k} \ , \end{aligned} \tag{11}$$

where $\delta(\cdot)$ represents a robust cost function – here, the Cauchy loss function. The set of static KFs is denoted as $\mathscr{V}_s$. Optimization is performed using the Levenberg-Marquardt algorithm available in the optimization framework GTSAM[1], while we approximate the information matrix $\boldsymbol{W}_b^{N-1}$ for the bias prior in the next iteration by extracting the diagonal block of the Hessian matrix corresponding to $\boldsymbol{b}^{N-1}$, computed by linearizing Equation (11) at the updated state $\boldsymbol{\Theta}$.

F. IMU Bias Initialization

While we use the ability of the VIO system to accurately initialize the gravity direction and initial scale, the estimation of the IMU bias terms $\boldsymbol{b}$ is more sensitive, as all axes need to be sufficiently excited and therefore, the initialization is dependent on the movement. While the gyroscope bias usually can be estimated well after a few KFs, the accelerometer bias is more sensitive. When performing Local BA, the bias terms that are outside the Local Window are only re-adjusted following global optimization and therefore, usually at the beginning of a mission they are incorrect. We propose to perform an initial correction using an bundle adjustment triggered based on the uncertainty of the bias estimates. As described in the previous section, we compute the marginal an approximation information matrix of $\boldsymbol{b}^{N-1}$, which gives us an estimate of the uncertainty. As it is safe to assume that the accelerometer bias is problematic, we only look at the part of $\boldsymbol{W}_{\boldsymbol{b}}^{N-1}$ corresponding to $\boldsymbol{b}_a^{N-1}$ and extract

$$w_{min} \coloneqq \sqrt{\min\left(\operatorname{diag}\left\{\boldsymbol{W}_{\boldsymbol{b}_a}^{N-1}\right\}\right)} \ , \tag{12}$$

which approximates the minimal square root information on the accelerometer bias under the assumption that $\boldsymbol{W}_{\boldsymbol{b}}^{N-1}$ is predominant on the diagonal. The global optimization as in Section III-I is triggered as soon as w_{min} is above a threshold parameter w_{init}. We do the same procedure for both agents, however, if the second agent only has very few frames in the map, the initialization is postponed until a sufficient number of KFs form this agent are processed in order to avoid unstable results.

G. Keyframe Management

While inserting keyframes is a necessity during exploration, insertion of new KFs in a well mapped area is problematic in the sense that the number of error terms in Equation (11) grows unbounded causing the optimization to slow down. For purely visual SLAM, it is well established that this can be avoided by dropping KFs (culling) containing

[1]https://research.cc.gatech.edu/borg/gtsam

predominantely reduntant information [16]. Here we assume a KF to be redundant if more than 90 percent of its landmark are observed in at least 3 other KFs as well. When using IMU information, this approach is problematic, as the preintegrated IMU measurements form a weaker constraint the larger the integration time between two consecutive KFs gets. While [17] uses a fixed time-based threshold to limit the integration time between KFs, we propose to utilize the estimated uncertainty of the preintegrated measurement. Since the translational part of the preintegrated measurement is crucial to recover a trajectory of metric scale and is also the most sensitive value due to the double integration of the acceleration, we only use the sub-part of $\mathbf{\Sigma}_a$ corresponding to the translation. We only allow a KF k to be culled, if it is outside the Local Window of both agents and

$$\sigma_{min}^{k-1,k+1} \coloneqq \sqrt{\min\left(\operatorname{diag}\left\{\mathbf{\Sigma}_a^{k-1,k} + \mathbf{\Sigma}_a^{k,k+1}\right\}\right)} < \sigma_{\text{cull}} \tag{13}$$

holds. This results in a more generic threshold as a maximal integration time, since it accounts for the uncertainty of the bias used for the preintegration which changes over the trajectory and furthermore naturally considers the noise of the IMU measurements allowing to use the same threshold for different IMU measurement noise levels (i.e. for different sensors).

H. Loop-closure Detection & Frame Localization

In order to be able to correct accumulated drift over larger trajectories when going back to a previously mapped area, the need to recognize visited place arises. As OKVIS is a purely VIO system, it does not have an implementation of loop-closure detection nor correction. As a result, in our implementation we employ the bag of binary words approach [9] together with the appearance and geometric checks used in [16]. In brief, loop-closure candidates are accepted if the similarity score of an older matching keyframe (KF_m) is larger than the minimal similarity of the KFs sharing connections on the Covisibility graph with the current keyframe (KF_k). Once a suitable candidate (KF_l) is found, the KFs are matched via descriptor matching and a projective RANSAC is performed to filter outliers and finally decide upon the inlier observation whether the match is accepted as a loop-closure. In case a match is found, we transform the loop-closuring keyframe (KF_k) and its neighbors into the coordinate frame of the re-detected KF_l and search for additional matches before merging duplicated landmarks as described in Section III-C. After the merging step, the newly generated correspondences are inserted in the Covisibility Graph and a pose graph optimization followed by Global BA is triggered.

A similar routine is performed to initialize our multi-agent setup. Note that here, we assume that the first agent has already initialized the map and we try to localize any subsequent agent in this map. Since at this stage we do not have any covisibility information from the additional agent, we only start searching based on the descriptor similarity score, which we threshold in order to avoid tedious searching. To verify a candidate for initialization, we solve the P3P problem using [13] together with RANSAC to filter outliers. Once the initialization is performed, we directly associate the observations matched with landmarks and proceed with the normal mapping.

I. Global BA

During Global BA, a full optimization of both the structure and the KF states is performed. In our system, this optimization is carried out in three cases; when we detect a loop closure, when we trigger the initialization and also at the end of a mission. In the case of loop-closure, we fist perform an optimization of the Essential Graph as a 6DoF pose graph optimization without optimization of the velocity and bias terms. The second step of the global optimization, the Global BA, is identical for all of the three possible cases.

In the Global BA, we perform a full BA including the estimation of velocity and bias terms for all KFs. In contrast to the Local BA, all states variables are included in the optimization and we do not impose any prior on the bias terms. Therefore, the objective function to be minimize is expressed as

$$\begin{aligned} J(\mathbf{\Theta}) \coloneqq & \boldsymbol{e}_p^{0\intercal} \boldsymbol{W}_p^0 \boldsymbol{e}_p^0 + \sum_{k\in\mathscr{V}} \sum_{j\in\mathscr{L}(k)} \delta\left(\boldsymbol{e}_r^{k,j\intercal} \boldsymbol{W}_r^{k,j} \boldsymbol{e}_r^{k,j}\right) \\ & + \sum_{k-1,k\in\mathscr{V}} \boldsymbol{e}_a^{k-1,k\intercal} \boldsymbol{W}_a^{k-1,k} \boldsymbol{e}_a^{k-1,k} \\ & + \sum_{k-1,k\in\mathscr{V}} \boldsymbol{e}_b^{k-1,k\intercal} \boldsymbol{W}_b^{k-1,k} \boldsymbol{e}_b^{k-1,k} \ , \end{aligned} \tag{14}$$

where the first term of Equation (14) is a prior on the root KF, in order to remove the ambiguity arising by the choice of the reference coordinate system. Again, a Cauchy loss function is used on the reprojection terms. Since Global BA is only performed after a local optimization, we can expect only a very limited number of outliers, therefore we carry out the full optimization using the Levenberg-Marquardt algorithm and only perform an outlier removal afterwards.

IV. Experimental Results

A. Experimental Setup

In order to evaluate the proposed system, we perform experiments on the publicly available EuRoC dataset [4], consisting of different sequences recorded from a UAV flying different trajectories both in a smaller room (Vicon Room) as well as in a larger industrial environment (Machine-Hall), where we put our focus on the Machine-Hall sequences. This dataset is specifically selected to enable a direct and fair comparison of the proposed pipeline to the most relevant state of the art system in VI SLAM, namely VI-ORB-SLAM [17], as it was evaluated on this dataset and it is closed source. In order to conduct experiments in a multi-UAV setup, we run two different sequences from this dataset simultaneously, while treating each sequence as coming from a separate UAV.

Since the proposed system aims to achieve a globally consistent map and we only optimize KFs, we choose the Absolute Trajectory Error (ATE) as our evaluation metric for

comparison. Assuming an estimated trajectory of n KF poses $\boldsymbol{T}_{WS}^{1:n}$ and the corresponding trajectory in the ground truth $\boldsymbol{T}_{GS}^{1:n}$, where G is the origin of the ground truth poses, we can compute $\boldsymbol{T}_{GW}$ to transform the estimated trajectory into the origin of the ground truth, e.g. by using the method of Horn [11]. The error is computed as the Root Mean Squared Error (RMSE) of the translation ($trans(\cdot)$) for all poses as

$$RMSE(\boldsymbol{T}_{WS}^{1:n}) \coloneqq \sqrt{\frac{1}{n}\sum_{i=1}^{n} \| trans\left((\boldsymbol{T}_{GS}^{i})^{-1}\boldsymbol{T}_{GW}\boldsymbol{T}_{WS}^{i}\right)\|^2} \ . \tag{15}$$

The evaluation of our system was performed on an Intel Core i7-4710MQ running at 2.5 GHz with 16GB RAM.

B. Results

We first evaluate the system in a single UAV configuration and compare against VI-ORB-SLAM as shown in Table I. Note that the values for VI-ORB-SLAM are copied from [17] for reference, as there is no open-source implementation of this method. For our approach we report the mean value over three runs. In the smaller Vicon-Room sequences, the proposed system generally has a higher error level compared to VI-ORB-SLAM, although for the sequence *V1_02_medium* we perform slightly better. We attain this to the low-textured scene of this sequence, in which a tight coupling between front-end and back-end as employed by VI-ORB-SLAM is advantageous, enabling reaction to a low number of matches, e.g. triggering the detection of additional, weaker keypoints.

On the MH sequences, we are able to achieve over 50% reduction on the trajectory error compared to VI-ORB-SLAM for the well-textured sequences *MH_01- MH_03*. On *MH_04* and *MH_05*, which exhibit partially very bad illumination, we perform comparably to VI-ORB-SLAM with marginally bigger errors. Evidently, the proposed system achieves higher accuracy in feature rich sequences (i.e. well-textured scenes with sufficient illumination), which we attain to the following points. Compared to [17], we generally create fewer landmarks, allowing the inclusion of more KFs in our Local Window, therefore increasing the scope of the Local BA. Furthermore, the use of the inverse distance parametrization together with the local reference keyframe formulation generally results in a better conditioned optimization problem for larger trajectories. Additionally the inclusion of a soft prior in the local optimization results allows to adjust the bias terms more freely in the course of the local optimization, damping the diffusion of initial errors over the whole trajectory.

The fluctuations of the error, however, across the different sequences attest to the fact that the front-end is a crucial component in handling difficult scenarios e.g. with bad illumination and low-textured scenes. So while proposing a powerful back-end is shown to improve significantly the accuracy of the estimation processes, further investigation in interfacing it with the front-end promises to result to even further improvement. This is most evident for visually challenging Vicon-Room sequences, for which reason our evaluation was focused on the Machine Hall.

Evaluation of the complexity of the proposed system and the resulting timings is not straightforward, as real-time performance here depends on the rate at which KFs are processed rather than a fixed frame rate, therefore, we analyze complexity using the average KF-rate, as shown in Table I. Note that the KF-rate is both scene- and motion-dependent, and thus, it varies both across sequences and throughout one sequence. As a result, we consider the system to be real-time capable, if it is able to process the KFs faster than the average KF-rate. The average KF processing rate for each sequence is shown in Table I with the system achieving real-time capability across all sequences. A detailed breakdown of the execution time for *MH_03_medium* is shown in Figure 4. Since the execution time has relatively large fluctuations between KFs, we process the timings using a moving average filter. As it can be seen, the runtime is slowly increasing with a growing number of KFs, which is caused by the need for well defined IMU-constraints, prohibiting arbitrary KF culling and therefore, although the number of variables is approximately constant, the number of error terms contributing to the cost function increases (Static Window). The fluctuation within the sequence is attained to the fact that exploration generally tends to be cheaper, as the number of KFs in the Static Window decreases. Although the Global BA is the most expensive part of the system (included in the recorded average KF-processing rate), it is only sporadically triggered and therefore, the bottleneck for real-time operation is, on average, the Local BA including the computation of the prior information for the bias.

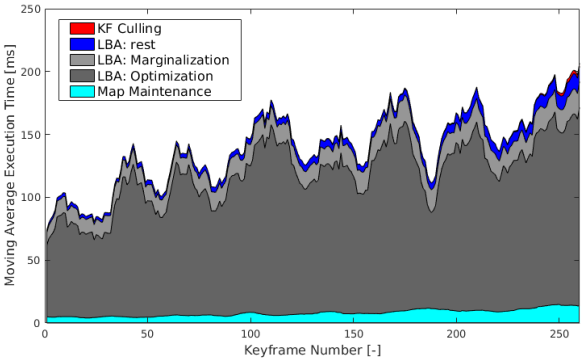

Fig. 4: Breakdown of the computation time of the proposed pipeline performed for every KF for the sequence *MH_03_medium*. To filter fluctuations between KFs,the timings presented are computed using a moving average filter over 10 KFs. The average incoming KF-rate for this sequence is approximately 4Hz (250 ms).

Evaluation using two UAVs was performed by combining different MH sequences. The trajectory error was computed by aligning the joint map to the ground-truth in the same fashion as for the single UAV setup. An overview of the results is in Table II, whereas the trajectory and landmarks for the combination *MH_02 & MH_03* is shown in Figure 1.

Although there are no IMU measurements between KFs from different UAVs to impose further constraints, it can be seen that the overall accuracy is maintained or increases compared to the single UAV case, indicating global consistency of the two trajectories in the common map frame. The advantages of collaborative sensing from two UAVs become evident especially in the difficult sequences. However, at this

	VI-ORB-SLAM			Proposed			
	RMSE [m]	Scale Err. [%]	RMSE* [m]	RMSE [m]	Scale Err. [%]	RMSE* [m]	KF-rate [Hz]
V1_01_easy	**0.023**	0.8	0.016	0.044	1.5	0.034	7.4
V1_02_medium	0.027	1.0	0.019	**0.021**	1.0	0.012	9.7
V1_03_difficult	X	X	X	**0.046**	2.0	0.034	10.0
MH_01_easy	0.068	0.3	0.068	**0.018**	0.2	0.015	6.5
MH_02_easy	0.073	0.4	0.072	**0.027**	0.4	0.020	6.5
MH_03_medium	0.071	0.1	0.071	**0.031**	0.2	0.030	6.3
MH_04_difficult	**0.087**	0.9	0.066	0.089	0.1	0.089	8.4
MH_05_difficult	**0.060**	0.2	0.060	0.070	0.5	0.054	8.1

TABLE I: The scale and RMSE errors of VI-ORB-SLAM [17] and the proposed monocular-inertial pipeline evaluated on the EuRoC dataset (averaged over 3 runs). The best RMSE performance in each sequence is indicated in bold. RMSE* records the error when performing the alignment to the ground-truth trajectory using a 7DoF transformation, indicating the error that would be achieved with perfect scale estimation.

UAV A:	*MH_01*	*MH_03*	*MH_04*
UAV B:	*MH_02*	*MH_02*	*MH_05*
RMSE [m]	0.021	0.026	0.059
Scale Err. [%]	0.3	0.05	0.1
RMSE* [m]	0.015	0.026	0.59

TABLE II: Average ATE for the proposed pipeline in the two-UAV setup. Different combinations of sequences are used to conduct experiments of different levels of difficulty. In Figure 1, the map as obtained by *MH_02* & *MH_03* is shown

stage we are only able to process the data close to real-time (factor of ~1.5), due to our sequential setup.

V. Conclusion

This work presents a back-end to monocular-inertial odometry from one or multiple agents, contributing towards achieving globally consistent SLAM, while resolving the scale ambiguity. The system considers the state of the bias estimate in both a local optimization and during the keyframe culling and is real-time capable in the single agent case. An evaluation on the EuRoC benchmarking dataset reveals over 50% improvement in accuracy at times over the state of the art. Finally, this system is demonstrated to achieve globally consistent collaborative VI mapping from two UAVs.

The significant reduction of the trajectory error in some of the test cases reveals the room for improvement still existing on the state of the art. However, the reported fluctuations emphasize the need for a tight integration between front-end and back-end in order to allow appropriate reactions to difficult conditions, such as low-textured scenes or bad illumination. Future work will aim at addressing this integration into the proposed system. Furthermore, appropriate methods to summarize IMU-constraints in order to expand the horizon for keyframe culling are essential towards the goal of life-long real-time SLAM.

References

[1] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart. Collaborative Stereo. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] J.-L. Blanco, J. González-Jiménez, and J.-A. Fernández-Madrigal. Sparser relative bundle adjustment (srba): constant-time maintenance and local optimization of arbitrarily large maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, may 2013.

[3] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. ROVIO: Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015.

[4] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. In *International Journal of Robotics Research (IJRR)*, 2016.

[5] J. Civera, A. J. Davison, and M. Montiel. Inverse Depth Parametrization for Monocular SLAM. 2008.

[6] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007.

[7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. In *IEEE Transactions on Robotics (T-RO)*, 2017.

[8] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2013.

[9] D. Galvez-López and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. In *IEEE Transactions on Robotics (T-RO)*, 2012.

[10] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representation through encapsulation of manifolds. *Information Fusion*, 2011.

[11] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. In *Journal of the Optical Society of America*, 1987.

[12] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[13] L. Kneip, D. Scaramuzza, and R. Siegwart. A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[14] F. R. Kschischang, B. Frey, and H.-A. Loeliger. Factor Graphs and the Sum-Product Algorithm. 2001.

[15] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, and R. Siegwart. Keyframe-based Visual-Inertial SLAM using Nonlinear Optimization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. In *IEEE Transactions on Robotics (T-RO)*, 2015.

[17] R. Mur-Artal and J. D. Tardós. Visual-Inertial Monocular SLAM with Map Reuse. In *IEEE Robotics and Automation Letters*, 2017.

[18] H. Oleynikova, M. Burri, S. Lynen, and R. Siegwart. Real-Time Visual-Inertial Localization for Aerial and Ground Robots. 2015.

[19] P. Schmuck and M. Chli. Multi-UAV Collaborative Monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[20] H. Strasdat, A. J. Davison, J. Montiel, and K. Konolige. Double Window Optimisation for Constant Time Visual SLAM. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[21] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart. Monocular Vision for Long-term MAV Navigation: A Compendium. *Journal of Field Robotics (JFR)*, 30:803–831, 2013.

[22] D. Zou and P. Tan. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.