This paper has been accepted for publication in *IEEE Robotics and Automation Letters*.

DOI: 10.1109/LRA.2017.2653359
IEEE Xplore: http://ieeexplore.ieee.org/document/7817784/

Visual-Inertial Monocular SLAM with Map Reuse

Raúl Mur-Artal and Juan D. Tardós

***Abstract*— In recent years there have been excellent results in Visual-Inertial Odometry techniques, which aim to compute the incremental motion of the sensor with high accuracy and robustness. However these approaches lack the capability to close loops, and trajectory estimation accumulates drift even if the sensor is continually revisiting the same place. In this work we present a novel tightly-coupled Visual-Inertial Simultaneous Localization and Mapping system that is able to close loops and reuse its map to achieve zero-drift localization in already mapped areas. While our approach can be applied to any camera configuration, we address here the most general problem of a monocular camera, with its well-known scale ambiguity. We also propose a novel IMU initialization method, which computes the scale, the gravity direction, the velocity, and gyroscope and accelerometer biases, in a few seconds with high accuracy. We test our system in the 11 sequences of a recent micro-aerial vehicle public dataset achieving a typical scale factor error of 1% and centimeter precision. We compare to the state-of-the-art in visual-inertial odometry in sequences with revisiting, proving the better accuracy of our method due to map reuse and no drift accumulation.**

***Index Terms*— SLAM, Sensor Fusion, Visual-Based Navigation**

I. INTRODUCTION

Motion estimation from onboard sensors is currently a hot topic in Robotics and Computer Vision communities, as it can enable emerging technologies such as autonomous cars, augmented and virtual reality, service robots and drone navigation. Among different sensor modalities, visual-inertial setups provide a cheap solution with great potential. On the one hand, cameras provide rich information of the environment, which allows to build 3D models, localize the camera and recognize already visited places. On the other hand, IMU sensors provide self-motion information, allowing to recover metric scale for monocular vision, and to estimate gravity direction, rendering absolute pitch and roll observable.

Visual-inertial fusion has been a very active research topic in the last years. The recent research is focused on tightly-coupled (i.e. joint optimization of all sensor states) visual-inertial odometry, using filtering [1]–[3] or keyframe-based non-linear optimization [4]–[8]. Nevertheless these approaches are only able to compute incremental motion and lack the capability to close loops and reuse a map of an already mapped environment. This implies that estimated trajectory accumulates drift without bound, even if the sensor is always moving in the same environment. This is due to the marginalization of past states to maintain a constant computational cost [1]–[3], [5], [6], or the use of full smoothing [4], [7], with an almost constant complexity in exploration but that can be as expensive as a batch method in the presence of loop closures. The filtering method [9], is able to close loops topologically and reuse its map, but global metric consistency is not enforced in real-time. The recent system [10] is able to reuse a given map, built offline, and perform visual-inertial tracking.

This work was supported by the Spanish government under Project DPI2015-67275, the Aragón regional governmnet under Project DGA T04-FSE and the Ministerio de Educación Scholarship FPU13/04175.

The authors are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain. Email: {raulmur,tardos}@unizar.es.

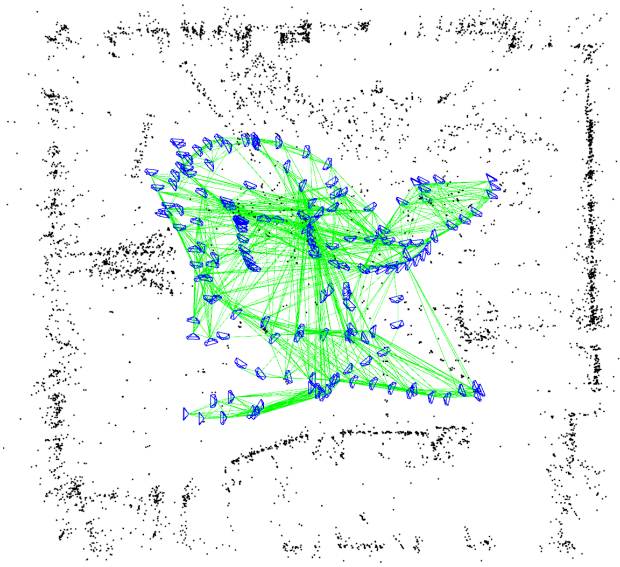

Fig. 1. Estimated map and keyframes by our Visual-Inertial ORB-SLAM in *V1_02_medium* from the EuRoC dataset [14]. The top view was rendered using the estimated gravity direction. The green lines connect keyframes that share more than 100 point observations and are a proof of the capability of the system to reuse the map, which, in contrast to visual-inertial odometry, allows zero-drift localization when continually revisiting the same place.

Building on the preintegration of Lupton and Sukkarieh [11], its application to the SO(3) manifold by Forster et al. [7], and its factor graph representation by Indelman et al. [4], we present in this paper Visual-Inertial ORB-SLAM, to the best of our knowledge the first keyframe-based Visual-Inertial SLAM that is able to metrically close loops in real-time and reuse the map that is being built online. Following the approach of ORB-SLAM [12], inspired by the work of Klein and Murray [13], our tracking optimizes the current frame assuming a fixed map, and our backend performs local Bundle Adjustment (BA), optimizing a local window of keyframes, including an outer window of fixed keyframes. In contrast to full smoothing, this approach allows constant time local BA, and by not marginalizing past states, we are able to reuse them. We detect large loops using place recognition and

correct them using a lightweight pose-graph optimization, followed by full BA in a separate thread, not to interfere with real-time operation. Fig. 1 shows a reconstruction by our system in a sequence with continuous revisiting.

Both our tracking and local BA fix states in their optimizations, which could potentially bias the solution. For this reason we need a reliable visual-inertial initialization that provides accurate state estimations before we start fixing states. To this end we propose to perform a visual-inertial full BA that provides the optimal solution for structure, camera poses, scale, velocities, gravity, and gyroscope and accelerometer biases. This full BA is a non-linear optimization that requires a good initial seed to converge. We propose in Section IV a divide and conquer approach to compute this initial solution. We firstly process a few seconds of video with our pure monocular ORB-SLAM [12] to estimate an initial solution for structure and several keyframe poses, up to an unknown scale factor. We then compute the bias of the gyroscope, which can be easily estimated from the known orientation of the keyframes, so that we can correctly rotate the accelerometer measurements. Then we solve scale and gravity without considering the accelerometer bias, using an approach inspired in [11]. To facilitate distinguishing between gravity and accelerometer bias, we use the knowledge of the magnitude of the gravity and solve for accelerometer bias, refining scale and gravity direction. At this point it is straightforward to retrieve the velocities for all keyframes. Our experiments validate that this is an efficient, reliable and accurate initialization method. Moreover it is general, could be applied to any keyframe-based monocular SLAM, does not assume any initial condition, and just require a movement of the sensor that make all variables observable [15]. While previous approaches [15]–[17] jointly solve vision and IMU, either ignoring gyroscope or accelerometer biases, we efficiently compute all variables by subdividing the problem in simpler steps.

II. VISUAL-INERTIAL PRELIMINARIES

The input for our Visual-Inertial ORB-SLAM is a stream of IMU measurements and monocular camera frames. We consider a conventional pinhole-camera model [18] with a projection function $\pi : \mathbb{R}^3 \rightarrow \Omega$, which transforms 3D points $\mathbf{X}_{\mathtt{C}} \in \mathbb{R}^3$ in camera reference $\mathtt{C}$, into 2D points on the image plane $\mathbf{x}_{\mathtt{C}} \in \Omega \subset \mathbb{R}^2$:

$$\pi(\mathbf{X}_{\mathtt{C}}) = \begin{bmatrix} f_u \frac{X_{\mathtt{C}}}{Z_{\mathtt{C}}} + c_u \\ f_v \frac{Y_{\mathtt{C}}}{Z_{\mathtt{C}}} + c_v \end{bmatrix}, \quad \mathbf{X}_{\mathtt{C}} = [X_{\mathtt{C}}\ Y_{\mathtt{C}}\ Z_{\mathtt{C}}]^T \tag{1}$$

where $[f_u\ f_v]^T$ is the focal length and $[c_u\ c_v]^T$ the principal point. This projection function does not consider the distortion produced by the camera lens. When we extract keypoints on the image, we undistort their coordinates so that they can be matched to projected points using (1).

The IMU, whose reference we denote with $\mathtt{B}$, measures the acceleration $\mathbf{a}_{\mathtt{B}}$ and angular velocity $\boldsymbol{\omega}_{\mathtt{B}}$ of the sensor at regular intervals Δt, typically at hundreds of Hertzs. Both measurements are affected, in addition to sensor noise, by slowly varying biases $\mathbf{b}_a$ and $\mathbf{b}_g$ of the accelerometer and gyroscope respectively. Moreover the accelerometer is subject to gravity $\mathbf{g}_{\mathtt{W}}$ and one needs to subtract its effect to compute the motion. The discrete evolution of the IMU orientation $\mathbf{R}_{\mathtt{WB}} \in \mathrm{SO}(3)$, position ${}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}$ and velocity ${}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}$, in the world reference $\mathtt{W}$, can be computed as follows [7]:

$$\begin{aligned} \mathbf{R}_{\mathtt{WB}}^{k+1} &= \mathbf{R}_{\mathtt{WB}}^{k} \operatorname{Exp}\left(\left(\boldsymbol{\omega}_{\mathtt{B}}^{k} - \boldsymbol{b}_g^k\right) \Delta t\right) \\ {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{k+1} &= {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{k} + \mathbf{g}_{\mathtt{W}} \Delta t + \mathbf{R}_{\mathtt{WB}}^{k} \left(\boldsymbol{a}_{\mathtt{B}}^{k} - \boldsymbol{b}_a^k\right) \Delta t \\ {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^{k+1} &= {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^{k} + {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{k} \Delta t + \frac{1}{2}\mathbf{g}_{\mathtt{W}} \Delta t^2 + \frac{1}{2}\mathbf{R}_{\mathtt{WB}}^{k} \left(\boldsymbol{a}_{\mathtt{B}}^{k} - \boldsymbol{b}_a^k\right) \Delta t^2 \end{aligned} \tag{2}$$

The motion between two consecutive keyframes can be defined in terms of the preintegration $\Delta\mathbf{R}$, $\Delta\mathbf{v}$ and $\Delta\mathbf{p}$ from all measurements in-between [11]. We use the recent IMU preintegration described in [7]:

$$\begin{aligned} \mathbf{R}_{\mathtt{WB}}^{i+1} &= \mathbf{R}_{\mathtt{WB}}^{i} \Delta\mathbf{R}_{i,i+1} \operatorname{Exp}\left(\left(\mathbf{J}_{\Delta R}^{g} \mathbf{b}_g^i\right)\right) \\ {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{i+1} &= {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{i} + \mathbf{g}_{\mathtt{W}} \Delta t_{i,i+1} \\ &\quad + \mathbf{R}_{\mathtt{WB}}^{i} \left(\Delta\mathbf{v}_{i,i+1} + \mathbf{J}_{\Delta v}^{g} \mathbf{b}_g^i + \mathbf{J}_{\Delta v}^{a} \mathbf{b}_a^i\right) \\ {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^{i+1} &= {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^{i} + {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{i} \Delta t_{i,i+1} + \frac{1}{2}\mathbf{g}_{\mathtt{W}} \Delta t_{i,i+1}^2 \\ &\quad + \mathbf{R}_{\mathtt{WB}}^{i} \left(\Delta\mathbf{p}_{i,i+1} + \mathbf{J}_{\Delta p}^{g} \mathbf{b}_g^i + \mathbf{J}_{\Delta p}^{a} \mathbf{b}_a^i\right) \end{aligned} \tag{3}$$

where the Jacobians $\mathbf{J}_{(\cdot)}^{a}$ and $\mathbf{J}_{(\cdot)}^{g}$ account for a first-order approximation of the effect of changing the biases without explicitly recomputing the preintegrations. Both preintegrations and Jacobians can be efficiently computed iteratively as IMU measurements arrive [7].

Camera and IMU are considered rigidly attached and the transformation $\mathbf{T}_{\mathtt{CB}} = [\mathbf{R}_{\mathtt{CB}}|_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}]$ between their reference systems known from calibration [19].

III. VISUAL-INERTIAL ORB-SLAM

The base of our visual-inertial system is ORB-SLAM [12], [20]. This system has three parallel threads for Tracking, Local Mapping and Loop Closing. The system is designed to work on large scale environments, by building a covisibility graph that allows to recover local maps for tracking and mapping, and by performing lightweight pose-graph optimizations at loop closure. In addition ORB-SLAM allows to build a map of an environment and switch to a less CPU-intensive *localization-only* mode (i.e. mapping and loop closing are disabled), thanks to the relocalization capability of the system. ORB-SLAM is open-source[1] and has been extensively evaluated on public datasets achieving top performing results. In this section we detail the main changes in the Tracking, Local Mapping and Loop Closing threads with respect to the original system. The visual-inertial initialization is presented in Section IV.

A. Tracking

Our visual-inertial tracking is in charge of tracking the sensor pose, velocity and IMU biases, at frame-rate. This allows us to predict the camera pose very reliably, instead of using an ad-hoc motion model as in the original monocular

[1] https://github.com/raulmur/ORB_SLAM2

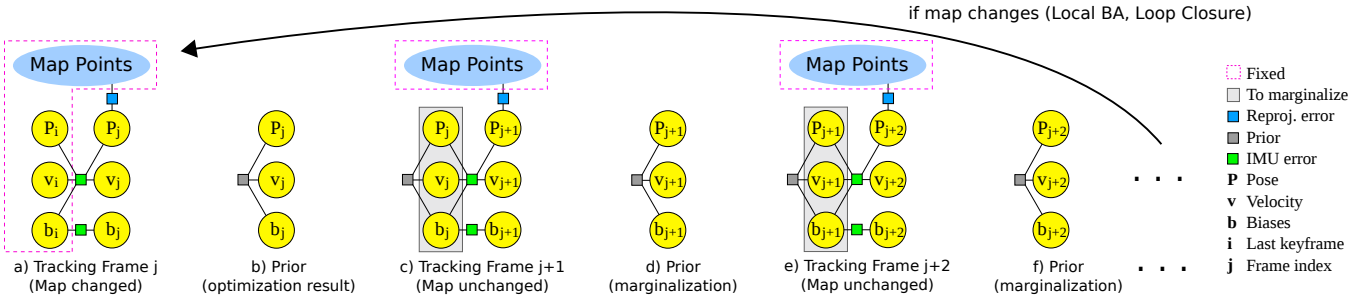

Fig. 2. Evolution of the optimization in the Tracking thread. (a) We start optimizing the frame j linked by an IMU constraint to last keyframe i. (b) The result of the optimization (estimation and Hessian matrix) serves as prior for next optimization. (c) When tracking next frame $j+1$, both frames j and $j+1$ are jointly optimized, being linked by an IMU constraint, and having frame j the prior from previous optimization. (d) At the end of the optimization, the frame j is marginalized out and the result serves as prior for following optimization. (e-f) This process is repeated until there is a map update from the Local Mapping or Loop Closing thread. In such case the optimization links the current frame to last keyframe discarding the prior, which is not valid after the map change.

system. Once the camera pose is predicted, the map points in the local map are projected and matched with keypoints on the frame. We then optimize current frame j by minimizing the feature reprojection error of all matched points and an IMU error term. This optimization is different depending on the map being updated or not by the Local Mapping or the Loop Closing thread, as illustrated in Fig. 2.

When tracking is performed just after a map update (Fig. 2a) the IMU error term links current frame j with last keyframe i:

$$\begin{aligned} \theta &= \left\{ \mathbf{R}^j_{\mathtt{WB}}, {}_{\mathtt{W}}\mathbf{p}^j_{\mathtt{B}}, {}_{\mathtt{W}}\mathbf{v}^j_{\mathtt{B}}, \mathbf{b}^j_g, \mathbf{b}^j_a \right\} \\ \theta^* &= \underset{\theta}{\operatorname{argmin}} \left(\sum_k \mathbf{E}_{\text{proj}}(k,j) + \mathbf{E}_{\text{IMU}}(i,j) \right) \end{aligned} \tag{4}$$

where the feature reprojection error $\mathbf{E}_{\text{proj}}$ for a given match k, is defined as follows:

$$\begin{aligned} \mathbf{E}_{\text{proj}}(k,j) &= \rho \left(\left(\mathbf{x}^k - \pi(\mathbf{X}^k_{\mathtt{C}}) \right)^T \boldsymbol{\Sigma}_{\boldsymbol{k}} \left(\mathbf{x}^k - \pi(\mathbf{X}^k_{\mathtt{C}}) \right) \right) \\ \mathbf{X}^k_{\mathtt{C}} &= \mathbf{R}_{\mathtt{CB}} \mathbf{R}^j_{\mathtt{BW}} \left(\mathbf{X}^k_{\mathtt{W}} - {}_{\mathtt{W}}\mathbf{p}^j_{\mathtt{B}} \right) + {}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}} \end{aligned} \tag{5}$$

where $\mathbf{x}^k$ is the keypoint location in the image, $\mathbf{X}^k_{\mathtt{W}}$ the map point in world coordinates, and $\boldsymbol{\Sigma}_{\boldsymbol{k}}$ the information matrix associated to the keypoint scale. The IMU error term $\mathbf{E}_{\text{IMU}}$ is:

$$\begin{aligned} \mathbf{E}_{\text{IMU}}(i,j) &= \rho \left(\left[\mathbf{e}^T_R \, \mathbf{e}^T_v \, \mathbf{e}^T_p \right] \boldsymbol{\Sigma}_I \left[\mathbf{e}^T_R \, \mathbf{e}^T_v \, \mathbf{e}^T_p \right]^T \right) \\ &\quad + \rho \left(\mathbf{e}^T_b \boldsymbol{\Sigma}_R \mathbf{e}_b \right) \\ \mathbf{e}_R &= \operatorname{Log} \left(\left(\Delta\mathbf{R}_{ij} \operatorname{Exp} \left(\mathbf{J}^g_{\Delta R} \mathbf{b}^j_g \right) \right)^T \mathbf{R}^i_{\mathtt{BW}} \mathbf{R}^j_{\mathtt{WB}} \right) \\ \mathbf{e}_v &= \mathbf{R}^i_{\mathtt{BW}} \left({}_{\mathtt{W}}\mathbf{v}^j_{\mathtt{B}} - {}_{\mathtt{W}}\mathbf{v}^i_{\mathtt{B}} - \mathbf{g}_{\mathtt{W}} \Delta t_{ij} \right) \\ &\quad - \left(\Delta\mathbf{v}_{ij} + \mathbf{J}^g_{\Delta v} \mathbf{b}^j_g + \mathbf{J}^a_{\Delta v} \mathbf{b}^j_a \right) \\ \mathbf{e}_p &= \mathbf{R}^i_{\mathtt{BW}} \left({}_{\mathtt{W}}\mathbf{p}^j_{\mathtt{B}} - {}_{\mathtt{W}}\mathbf{p}^i_{\mathtt{B}} - {}_{\mathtt{W}}\mathbf{v}^i_{\mathtt{B}} \Delta t_{ij} - \frac{1}{2} \mathbf{g}_{\mathtt{W}} \Delta t^2_{ij} \right) \\ &\quad - \left(\Delta\mathbf{p}_{ij} + \mathbf{J}^g_{\Delta p} \mathbf{b}^j_g + \mathbf{J}^a_{\Delta p} \mathbf{b}^j_a \right) \\ \mathbf{e}_b &= \mathbf{b}^j - \mathbf{b}^i \end{aligned} \tag{6}$$

where $\boldsymbol{\Sigma}_I$ is the information matrix of the preintegration and $\boldsymbol{\Sigma}_R$ of the bias random walk [7], and ρ is the Huber robust cost function. We solve this optimization problem with Gauss-Newton algorithm implemented in g2o [21]. After the optimization (Fig. 2b) the resulting estimation and Hessian matrix serves as prior for next optimization.

Assuming no map update (Fig. 2c), the next frame $j+1$ will be optimized with a link to frame j and using the prior computed at the end of the previous optimization (Fig 2b):

$$\begin{aligned} \theta &= \left\{ \mathbf{R}^j_{\mathtt{WB}}, \mathbf{p}^j_{\mathtt{W}}, \mathbf{v}^j_{\mathtt{W}}, \mathbf{b}^j_g, \mathbf{b}^j_a, \mathbf{R}^{j+1}_{\mathtt{WB}}, \mathbf{p}^{j+1}_{\mathtt{W}}, \mathbf{v}^{j+1}_{\mathtt{W}}, \mathbf{b}^{j+1}_g, \mathbf{b}^{j+1}_a \right\} \\ \theta^* &= \underset{\theta}{\operatorname{argmin}} \Big(\sum_k \mathbf{E}_{\text{proj}}(k,j+1) + \mathbf{E}_{\text{IMU}}(j,j+1) \\ &\qquad + \mathbf{E}_{\text{prior}}(j) \Big) \end{aligned} \tag{7}$$

where $\mathbf{E}_{\text{prior}}$ is a prior term:

$$\begin{aligned} \mathbf{E}_{\text{prior}}(j) &= \rho \left(\left[\mathbf{e}^T_R \, \mathbf{e}^T_v \, \mathbf{e}^T_p \, \mathbf{e}^T_b \right] \boldsymbol{\Sigma}_p \left[\mathbf{e}^T_R \, \mathbf{e}^T_v \, \mathbf{e}^T_p \, \mathbf{e}^T_b \right]^T \right) \\ \mathbf{e}_R &= \operatorname{Log} \left(\bar{\mathbf{R}}^j_{\mathtt{BW}} \mathbf{R}^j_{\mathtt{WB}} \right) \qquad \mathbf{e}_v = {}_{\mathtt{W}}\bar{\mathbf{v}}^j_{\mathtt{B}} - {}_{\mathtt{W}}\mathbf{v}^j_{\mathtt{B}} \\ \mathbf{e}_p &= {}_{\mathtt{W}}\bar{\mathbf{p}}^j_{\mathtt{B}} - {}_{\mathtt{W}}\mathbf{p}^j_{\mathtt{B}} \qquad \mathbf{e}_b = \bar{\mathbf{b}}^j - \mathbf{b}^j \end{aligned} \tag{8}$$

where $\bar{(\cdot)}$ and $\boldsymbol{\Sigma}_p$ are the estimated states and Hessian matrix resulting from previous optimization (Fig. 2b). After this optimization (Fig. 2d), frame j is marginalized out [5]. This optimization linking two consecutive frames and using a prior is repeated (Fig. 2e-f) until a map change, when the prior will be no longer valid and the tracking will link again the current frame to the last keyframe (Fig. 2a). Note that this is the optimization, Fig 2 (e-f), that is always performed in *localization-only* mode, as the map is not updated.

B. Local Mapping

The Local Mapping thread performs local BA after a new keyframe insertion. It optimizes the last N keyframes (local window) and all points seen by those N keyframes. All other keyframes that share observations of local points (i.e. are connected in the covisibility graph to any local keyframe), but are not in the local window, contribute to the total cost but are fixed during optimization (fixed window). The keyframe $N+1$ is always included in the fixed window as it constrains the IMU states. Fig. 3 illustrates the differences between

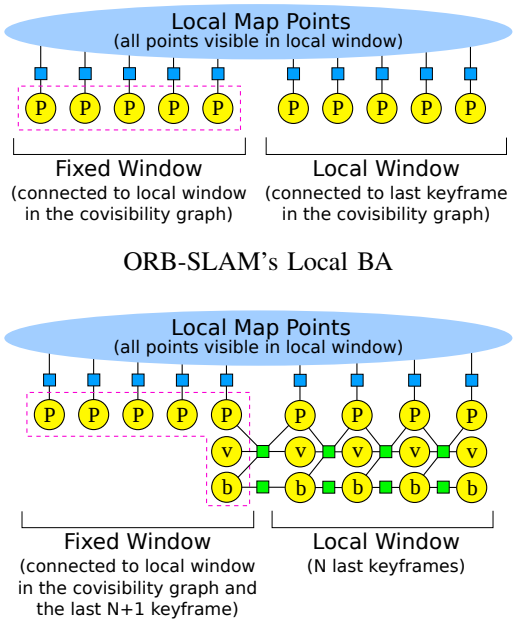

Fig. 3. Comparison of Local Bundle Adjustment between original ORB-SLAM (top) and proposed Visual-Inertial ORB-SLAM (bottom). The local window in Visual-Inertial ORB-SLAM is retrieved by temporal order of keyframes, while in ORB-SLAM is retrieved using the covisibility graph.

local BA in original ORB-SLAM and Visual-Inertial ORB-SLAM. The cost function is a combination of IMU error terms (6) and reprojection error terms (5). Note that the visual-inertial version, compared to the vision only, is more complex as there are 9 additional states (velocity and biases) to optimize per keyframe. A suitable local window size has to be chosen for real-time performance.

The Local Mapping is also in charge of keyframe management. The original ORB-SLAM policy discards redundant keyframes, so that map size does not grow if localizing in a well mapped area. This policy is problematic when using IMU information, which constrains the motion of consecutive keyframes. The longer the temporal difference between consecutive keyframes, the weaker information IMU provides. Therefore we allow the mapping to discard redundant keyframes, if that does not make two consecutive keyframes in the local window of local BA to differ more than 0.5s. To be able to perform full BA, after a loop closure or at any time to refine a map, we do not allow any two consecutive keyframes to differ more than 3s. If we switched-off full BA with IMU constraints, we would only need to restrict the temporal offset between keyframes in the local window.

C. Loop Closing

The loop closing thread aims to reduce the drift accumulated during exploration, when returning to an already mapped area. The place recognition module matches a recent keyframe with a past keyframe. This match is validated computing a rigid body transformation that aligns matched points between keyframes [22]. Finally an optimization is carried out to reduce the error accumulated in the trajectory. This optimization might be very costly in large maps, therefore the strategy is to perform a pose-graph optimization, which reduces the complexity, as structure is ignored, and exhibits good convergence as shown in [12]. In contrast to the original ORB-SLAM, we perform the pose-graph optimization on 6 Degrees of Freedom(DoF) instead of 7 DoF [23], as scale is observable. This pose-graph ignores IMU information, not optimizing velocity or IMU biases. Therefore we correct velocities by rotating them according to the corrected orientation of the associated keyframe. While this is not optimal, biases and velocities should be locally accurate to continue using IMU information right after pose-graph optimization. We perform afterwards a full BA in a parallel thread that optimizes all states, including velocities and biases.

IV. IMU INITIALIZATION

We propose in this section a method to compute an initial estimation for a visual-inertial full BA of the scale, gravity direction, velocity and IMU biases, given a set of keyframes processed by a monocular SLAM algorithm. The idea is to run the monocular SLAM for a few seconds, assuming the sensor performs a motion that makes all variables observable. While we build on ORB-SLAM [12], any other SLAM could be used. The only requirement is that any two consecutive keyframes are close in time (see Section III-B), to reduce IMU noise integration.

The initialization is divided in simpler subproblems: (1) gyroscope bias estimation, (2) scale and gravity approximation, considering no accelerometer bias, (3) accelerometer bias estimation, and scale and gravity direction refinement, and (4) velocity estimation.

A. Gyroscope Bias Estimation

Gyroscope bias can be estimated just from the known orientation of two consecutive keyframes. Assuming a negligible bias change, we optimize a constant bias $\mathbf{b}_g$, which minimizes the difference between gyroscope integration and relative orientation computed from ORB-SLAM, for all pairs of consecutive keyframes:

$$\underset{\mathbf{b}_g}{\operatorname{argmin}} \sum_{i=1}^{N-1} \left\| \operatorname{Log} \left(\left(\Delta\mathbf{R}_{i,i+1} \operatorname{Exp} \left(\mathbf{J}^g_{\Delta R} \mathbf{b}_g \right) \right)^T \mathbf{R}^{i+1}_{\mathtt{BW}} \mathbf{R}^{i}_{\mathtt{WB}} \right) \right\|^2 \tag{9}$$

where N is the number of keyframes. $\mathbf{R}^{(\cdot)}_{\mathtt{WB}} = \mathbf{R}^{(\cdot)}_{\mathtt{WC}} \mathbf{R}_{\mathtt{CB}}$ is computed from the orientation $\mathbf{R}^{(\cdot)}_{\mathtt{WC}}$ computed by ORB-SLAM and calibration $\mathbf{R}_{\mathtt{CB}}$. $\Delta\mathbf{R}_{i,i+1}$ is the gyroscope integration between two consecutive keyframes. We solve (9) with Gauss-Newton with a zero bias seed. Analytic jacobians for a similar expression can be found in [7].

B. Scale and Gravity Approximation (no accelerometer bias)

Once we have estimated the gyroscope bias, we can preintegrate velocities and positions, rotating correctly the acceleration measurements compensating the gyroscope bias.

The scale of the camera trajectory computed by ORB-SLAM is arbitrary. Therefore we need to include a scale

factor s when transforming between camera $\mathtt{C}$ and IMU $\mathtt{B}$ coordinate systems:

$$ {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}} = s\,{}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}} + \mathbf{R}_{\mathtt{WC}}\,{}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}} \tag{10} $$

Substituting (10) into the equation relating position of two consecutive keyframes (3), and neglecting at this point accelerometer bias, it follows:

$$ \begin{aligned} s\,{}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{i+1} = s\,{}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{i} + {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{i}\Delta t_{i,i+1} + \frac{1}{2}\mathbf{g}_{\mathtt{W}}\Delta t_{i,i+1}^{2} \\ + \mathbf{R}_{\mathtt{WB}}^{i}\Delta\mathbf{p}_{i,i+1} + \left(\mathbf{R}_{\mathtt{WC}}^{i} - \mathbf{R}_{\mathtt{WC}}^{i+1}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}} \end{aligned} \tag{11} $$

The goal is to estimate s and $\mathbf{g}_{\mathtt{W}}$ by solving a linear system of equations on those variables. To avoid solving for N velocities, and reduce complexity, we consider two relations (11) between three consecutive keyframes and use velocity relation in (3), which results in the following expression:

$$ \begin{bmatrix} \boldsymbol{\lambda}(i) & \boldsymbol{\beta}(i) \end{bmatrix} \begin{bmatrix} s \\ \mathbf{g}_{\mathtt{W}} \end{bmatrix} = \boldsymbol{\gamma}(i) \tag{12} $$

where, writing keyframes $i, i+1, i+2$ as $1, 2, 3$ for clarity of notation, we have:

$$ \begin{aligned} \boldsymbol{\lambda}(i) &= \left({}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{2} - {}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{1}\right)\Delta t_{23} - \left({}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{3} - {}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{2}\right)\Delta t_{12} \\ \boldsymbol{\beta}(i) &= \frac{1}{2}\mathbf{I}_{3\times 3}\left(\Delta t_{12}^{2}\Delta t_{23} + \Delta t_{23}^{2}\Delta t_{12}\right) \\ \boldsymbol{\gamma}(i) &= \left(\mathbf{R}_{\mathtt{WC}}^{2} - \mathbf{R}_{\mathtt{WC}}^{1}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{23} - \left(\mathbf{R}_{\mathtt{WC}}^{3} - \mathbf{R}_{\mathtt{WC}}^{2}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{12} \\ &\quad + \mathbf{R}_{\mathtt{WB}}^{2}\Delta\mathbf{p}_{23}\Delta t_{12} + \mathbf{R}_{\mathtt{WB}}^{1}\Delta\mathbf{v}_{12}\Delta t_{12}\Delta t_{23} \\ &\quad - \mathbf{R}_{\mathtt{WB}}^{1}\Delta\mathbf{p}_{12}\Delta t_{23} \end{aligned} \tag{13} $$

We stack then all relations of three consecutive keyframes (12) into a system $\mathbf{A}_{3(N-2)\times 4}\,\mathbf{x}_{4\times 1} = \mathbf{B}_{3(N-2)\times 1}$ which can be solved via Singular Value Decomposition (SVD) to get the scale factor s^{*} and gravity vector $\mathbf{g}_{\mathtt{W}}^{*}$. Note that we have $3(N-2)$ equations and 4 unknowns, therefore we need at least 4 keyframes.

C. Accelerometer Bias Estimation, and Scale and Gravity Direction Refinement

So far we have not considered accelerometer bias when computing scale and gravity. Just incorporating accelerometer biases in (12) will heavily increase the chance of having an ill-conditioned system, because gravity and accelerometer biases are hard to distinguish [15]. To increase observability we introduce new information we did not consider so far, which is the gravity magnitude G. Consider an inertial reference I with the gravity direction $\hat{\mathbf{g}}_{\mathtt{I}} = \{0, 0, -1\}$, and the already computed gravity direction $\hat{\mathbf{g}}_{\mathtt{W}} = \mathbf{g}_{\mathtt{W}}^{*}/\|\mathbf{g}_{\mathtt{W}}^{*}\|$. We can compute rotation $\mathbf{R}_{\mathtt{WI}}$ as follows:

$$ \begin{aligned} \mathbf{R}_{\mathtt{WI}} &= \mathrm{Exp}(\hat{\mathbf{v}}\theta) \\ \hat{\mathbf{v}} = \frac{\hat{\mathbf{g}}_{\mathtt{I}} \times \hat{\mathbf{g}}_{\mathtt{W}}}{\|\hat{\mathbf{g}}_{\mathtt{I}} \times \hat{\mathbf{g}}_{\mathtt{W}}\|}, &\quad \theta = \mathrm{atan2}\left(\|\hat{\mathbf{g}}_{\mathtt{I}} \times \hat{\mathbf{g}}_{\mathtt{W}}\|, \hat{\mathbf{g}}_{\mathtt{I}} \cdot \hat{\mathbf{g}}_{\mathtt{W}}\right) \end{aligned} \tag{14} $$

and express now the gravity vector as:

$$ \mathbf{g}_{\mathtt{W}} = \mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G \tag{15} $$

where $\mathbf{R}_{\mathtt{WI}}$ can be parametrized with just two angles around x and y axes in $\mathtt{I}$, because a rotation around z axis, which is aligned with gravity, has no effect in $\mathbf{g}_{\mathtt{W}}$. This rotation can be optimized using a perturbation $\boldsymbol{\delta\theta}$:

$$ \begin{aligned} \mathbf{g}_{\mathtt{W}} &= \mathbf{R}_{\mathtt{WI}}\mathrm{Exp}(\boldsymbol{\delta\theta})\,\hat{\mathbf{g}}_{\mathtt{I}}\,G \\ \boldsymbol{\delta\theta} &= \left[\boldsymbol{\delta\theta}_{\boldsymbol{xy}}^{\boldsymbol{T}}\ 0\right]^{T}, \quad \boldsymbol{\delta\theta}_{\boldsymbol{xy}} = \left[\delta\theta_x\ \delta\theta_y\right]^{T} \end{aligned} \tag{16} $$

with a first-order approximation:

$$ \mathbf{g}_{\mathtt{W}} \approx \mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G - \mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times} G\,\boldsymbol{\delta\theta} \tag{17} $$

Substituting (17) in (11) and including now the effect of accelerometer bias, we obtain:

$$ \begin{aligned} s\,{}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{i+1} = s\,{}_{\mathtt{W}}\mathbf{p}_{\mathtt{C}}^{i} + {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{i}\Delta t_{i,i+1} - \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times} G\Delta t_{i,i+1}^{2}\,\boldsymbol{\delta\theta} \\ + \mathbf{R}_{\mathtt{WB}}^{i}\left(\Delta\mathbf{p}_{i,i+1} + \mathbf{J}_{\Delta p}^{a}\mathbf{b}_{a}\right) + \left(\mathbf{R}_{\mathtt{WC}}^{i} - \mathbf{R}_{\mathtt{WC}}^{i+1}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}} \\ + \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G\Delta t_{i,i+1}^{2} \end{aligned} \tag{18} $$

Considering three consecutive keyframes as in (12) we can eliminate velocities and get the following relation:

$$ \begin{bmatrix} \boldsymbol{\lambda}(i) & \boldsymbol{\phi}(i) & \boldsymbol{\zeta}(i) \end{bmatrix} \begin{bmatrix} s \\ \boldsymbol{\delta\theta}_{\boldsymbol{xy}} \\ \mathbf{b}_{a} \end{bmatrix} = \boldsymbol{\psi}(i) \tag{19} $$

where $\boldsymbol{\lambda}(i)$ remains the same as in (13), and $\boldsymbol{\phi}(i)$, $\boldsymbol{\zeta}(i)$, and $\boldsymbol{\psi}(i)$ are computed as follows:

$$ \begin{aligned} \boldsymbol{\phi}(i) &= \left[\frac{1}{2}\mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times} G\left(\Delta t_{12}^{2}\Delta t_{23} + \Delta t_{23}^{2}\Delta t_{12}\right)\right]_{(:,1:2)} \\ \boldsymbol{\zeta}(i) &= \mathbf{R}_{\mathtt{WB}}^{2}\mathbf{J}_{\Delta p_{23}}^{a}\Delta t_{12} + \mathbf{R}_{\mathtt{WB}}^{1}\mathbf{J}_{\Delta v_{23}}^{a}\Delta t_{12}\Delta t_{23} \\ &\quad - \mathbf{R}_{\mathtt{WB}}^{1}\mathbf{J}_{\Delta p_{12}}^{a}\Delta t_{23} \\ \boldsymbol{\psi}(i) &= \left(\mathbf{R}_{\mathtt{WC}}^{2} - \mathbf{R}_{\mathtt{WC}}^{1}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{23} - \left(\mathbf{R}_{\mathtt{WC}}^{3} - \mathbf{R}_{\mathtt{WC}}^{2}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{12} \\ &\quad + \mathbf{R}_{\mathtt{WB}}^{2}\Delta\mathbf{p}_{23}\Delta t_{12} + \mathbf{R}_{\mathtt{WB}}^{1}\Delta\mathbf{v}_{12}\Delta t_{12}\Delta t_{23} \\ &\quad - \mathbf{R}_{\mathtt{WB}}^{1}\Delta\mathbf{p}_{12}\Delta t_{23} + \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G\Delta t_{ij}^{2} \end{aligned} \tag{20} $$

where $[]_{(:,1:2)}$ means the first two columns of the matrix. Stacking all relations between three consecutive keyframes (19) we form a linear system of equations $\mathbf{A}_{3(N-2)\times 6}\,\mathbf{x}_{6\times 1} = \mathbf{B}_{3(N-2)\times 1}$ which can be solved via SVD to get the scale factor s^{*}, gravity direction correction $\boldsymbol{\delta\theta}_{\boldsymbol{xy}}^{*}$ and accelerometer bias $\mathbf{b}_{a}^{*}$. In this case we have $3(N-2)$ equations and 6 unknowns and we need again at least 4 keyframes to solve the system. We can compute the condition number (i.e. the ratio between the maximum and minimum singular value) to check if the problem is well-conditioned (i.e. the sensor has performed a motion that makes all variables observable). We could relinearize (17) and iterate the solution, but in practice we found that a second iteration does not produce a noticeable improvement.

D. Velocity Estimation

We considered relations of three consecutive keyframes in equations (12) and (19), so that the resulting linear systems do not have the $3N$ additional unknowns corresponding to velocities. The velocities for all keyframes can now be computed using equation (18), as scale, gravity and bias are known. To compute the velocity of the most recent keyframe, we use the velocity relation in (3).

E. *Bias Reinitialization after Relocalization*

When the system relocalizes after a long period of time, using place recognition, we reinitialize gyroscope biases by solving (9). The accelerometer bias is estimated by solving a simplified (19), where the only unknown is the bias, as scale and gravity are already known. We use 20 consecutive frames localized with only vision to estimate both biases.

V. EXPERIMENTS

We evaluate the proposed IMU initialization method, detailed in Section IV and our Visual-Inertial ORB-SLAM in the EuRoC dataset [14]. It contains 11 sequences recorded from a micro aerial vehicle (MAV), flying around two different rooms and an industrial environment. Sequences are classified as *easy*, *medium* and *difficult*, depending on illumination, texture, fast/slow motions or motion blur. The dataset provides synchronized global shutter WVGA stereo images at 20Hz with IMU measurements at 200Hz and trajectory ground-truth. These characteristics make it a really useful dataset to test Visual-Inertial SLAM systems. The experiments were performed processing left images only, in an Intel Core i7-4700MQ computer with 8Gb RAM.

A. *IMU Initialization*

We evaluate the IMU initialization in sequence *V2_01_easy*. We run the initialization from scratch every time a keyframe is inserted by ORB-SLAM. We run the sequence at a lower frame-rate so that the repetitive initialization does not interfere with the normal behavior of the system. The goal is to analyze the convergence of the variables as more keyframes, i.e. longer trajectories, are processed by the initialization algorithm. Fig. 4 shows the estimated scale and IMU biases. It can be seen that between 10 and 15 seconds all variables have already converged to stable values and that the estimated scale factor is really close to its optimal value. This optimal scale factor is computed aligning the estimated trajectory with the ground-truth by a similarity transformation [22]. Fig. 4 also shows the condition number of (19), indicating that some time is required to get a well-conditioned problem. This confirms that the sensor has to perform a motion that makes all variables observable, especially the accelerometer bias. The last row in Fig. 4 shows the time spent by the initialization algorithm, which exhibits a linear growth. This complexity is the result of not including velocities in (12) and (19), which would have resulted in a quadratic complexity when using SVD to solve these systems. Subdividing the initialization in simpler subproblems, in contrast to [15], [17], results in a very efficient method.

The proposed initialization allows to start fusing IMU information, as gravity, biases, scale and velocity are reliably estimated. For the EuRoC dataset, we observed that 15 seconds of MAV exploration give always an accurate initialization. As a future work we would like to investigate an automatic criterion to decide when we can consider an initialization successful, as we observed that an absolute threshold on the condition number is not reliable enough.

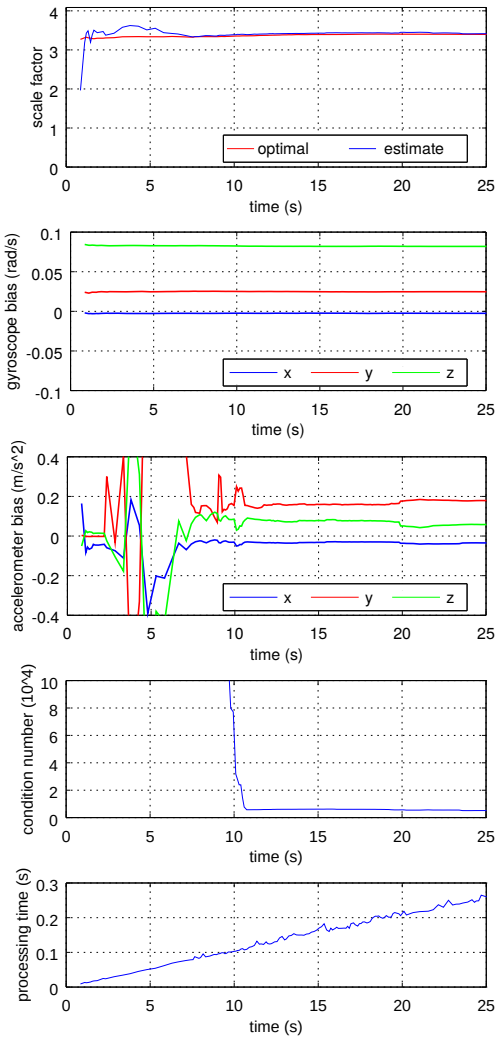

Fig. 4. IMU initialization in *V2_01_easy*.

B. *SLAM Evaluation and Comparison to State-of-the-Art*

We evaluate the accuracy of our Visual-Inertial ORB-SLAM in the 11 sequences of the EuRoC dataset. We start processing the sequences when the MAV starts exploring. The local window size for the local BA is set to 10 keyframes and the IMU initialization is performed after 15 seconds from monocular ORB-SLAM initialization. The system performs a full BA just after IMU initialization. Table I shows the translation Root Mean Square Error (RMSE) of the keyframe trajectory for each sequence, as proposed in [24]. We use the raw Vicon and Leica ground-truth as the post-processed one already used IMU. We observed a time offset between the visual-inertial sensor and the raw ground-truth of $-0.2s$ in the *Vicon Room 2* sequences and $0.2s$ in the *Machine Hall*, that we corrected when aligning both trajectories. We also measure the ideal scale factor that would align optimally the estimated trajectory and ground-truth. This scale factor can be regarded as the residual scale error of the trajectory and reconstruction. Our system successfully processes all these sequences in real-time, except *V1_03_difficult*, where the

TABLE I

KEYFRAME TRAJECTORY ACCURACY IN EUROC DATASET (RAW GROUND-TRUTH)

	Visual-Inertial ORB-SLAM						Monocular ORB-SLAM	
	No Full BA			Full BA			No Full BA	Full BA
	RMSE (m)	Scale Error (%)	RMSE(m) *GT scale**	RMSE (m)	Scale Error (%)	RMSE (m) *GT scale**	RMSE(m) *GT scale**	RMSE(m) *GT scale**
V1_01_easy	0.027	0.9	0.019	0.023	0.8	0.016	0.015	0.015
V1_02_medium	0.028	0.8	0.024	0.027	1.0	0.019	0.020	0.020
V1_03_difficult	X	X	X	X	X	X	X	X
V2_01_easy	0.032	0.2	0.031	0.018	0.2	0.017	0.021	0.015
V2_02_medium	0.041	1.4	0.026	0.024	0.8	0.017	0.018	0.017
V2_03_difficult	0.074	0.7	0.073	0.047	0.6	0.045	X	X
MH_01_easy	0.075	0.5	0.072	0.068	0.3	0.068	0.071	0.070
MH_02_easy	0.084	0.8	0.078	0.073	0.4	0.072	0.067	0.066
MH_03_medium	0.087	1.5	0.067	0.071	0.1	0.071	0.071	0.071
MH_04_difficult	0.217	3.4	0.081	0.087	0.9	0.066	0.082	0.081
MH_05_difficult	0.082	0.5	0.077	0.060	0.2	0.060	0.060	0.060

**GT scale*: the estimated trajectory is scaled so that it perfectly matches the scale of the ground-truth. These columns are included for comparison purposes but do not represent the output of a real system, but the output of an *ideal* system that could estimate the true scale.

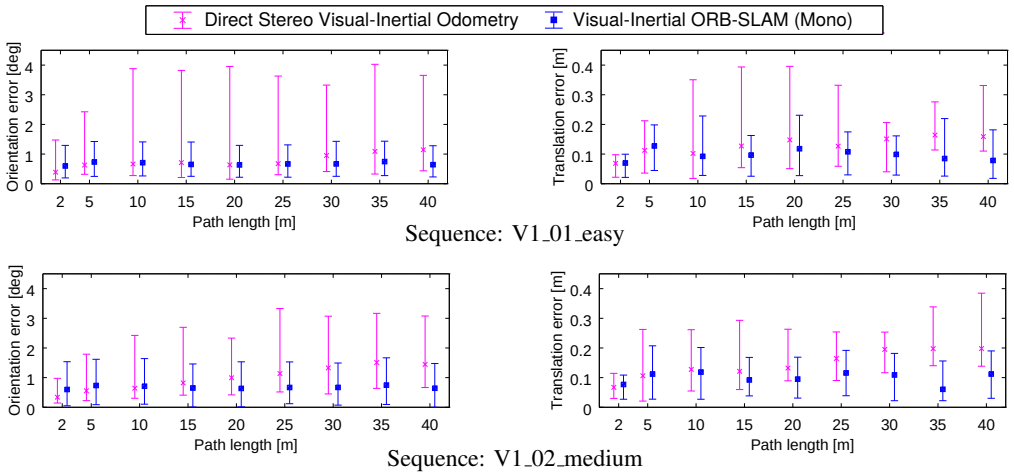

Fig. 5. Relative Pose Error [25] comparison between our approach and the state-of-the-art direct stereo visual-inertial odometry [6]. The error for our SLAM system does not grow for longer paths, due to map reuse, in contrast to the visual-inertial odometry method where drift cannot be compensated. Note that [6] uses stereo, while our results are monocular.

movement is too extreme for the monocular system to survive 15 seconds. Our system is able to recover motion with metric scale, with a scale error typically below 1%, achieving a typical precision of 3cm for 30m^2 room environments and of 8cm for 300m^2 industrial environments. To show the loss in accuracy due to scale error, we also show the RMSE if the system would be able to recover the true scale, see *GT scale* columns. We also show that the precision and scale estimation can be further improved by performing a visual-inertial full BA at the end of the execution, see Full BA columns. The reconstruction for sequence *V1_02_medium* can be seen in Fig. 1, and in the accompanying video[2].

To contextualize our results, we include as baseline the results of our vision-only system in Table I. Our visual-inertial system is more robust as it can process *V2_03_difficult*, it is able to recover metric scale and does not suffer scale drift. The accuracy of the visual-inertial system is similar to the accuracy that would obtain the vision-only version if it could ideally recover the true scale. However the visual-inertial bundle adjustment is more costly, as explained in Section III-B, and the local window of the local BA has to be smaller that in the vision-only case. This explains the slightly worse results of the *GT scaled* visual-inertial results without full BA. In fact the visual-inertial full BA typically converges in 15 iterations in 7 seconds, while the vision-only full BA converges in 5 iterations in less than 1 second.

In order to test the capability of Visual-Inertial ORB-SLAM to reuse a previous map, we run in a row all sequences of the same environment. We process the first sequence and perform a full BA. Then we run the rest of the sequences, where our system relocalizes and continue doing SLAM. We

[2]https://youtu.be/rdR5OR8egGI

then compare the accumulated keyframe trajectory with the ground-truth. As there exists a previous map, our system is now able to localize the camera in sequence *V1_03_difficult*. The RMSE in meters for V1, V2 and MH environments are 0.037, 0.027 and 0.076 respectively, with an scale factor error of 1.2%, 0.1% and 0.2%. A final full BA has a negligible effect as we have already performed a full BA at the end of the first sequence. These results show that there is no drift accumulation when revisiting the same scene, as the RMSE for all sequences is not larger than for individual sequences.

Finally we have compared Visual-Inertial ORB-SLAM to the state-of-the-art direct visual-inertial odometry for stereo cameras [6], which also showed results in *Vicon Room 1* sequences, allowing for a direct comparison. Fig. 5 shows the Relative Pose Error (RPE) [25]. To compute the RPE for our method, we need to recover the frame trajectory, as only keyframes are optimized by our backend. To this end, when tracking a frame we store a relative transformation to a reference keyframe, so that we can retrieve the frame pose from the estimated keyframe pose at the end of the execution. We have not run a full BA at the end of the experiment. We can see that the error for the visual-inertial odometry method grows with the traveled distance, while our visual-inertial SLAM system does not accumulate error due to map reuse. The stereo method [6] is able to work in *V1_03_difficult*, while our monocular method fails. Our monocular SLAM successfully recovers metric scale, and achieves comparable accuracy in short paths, where the advantage of SLAM is negligible compared to odometry. This is a remarkable result of our feature-based monocular method, compared to [6] which is direct and stereo.

VI. CONCLUSIONS

We have presented a novel tightly coupled Visual-Inertial SLAM system, that is able to close loops in real-time and localize the sensor reusing the map in already mapped areas. This allows to achieve a *zero-drift* localization, in contrast to visual odometry approaches where drift grows unbounded. The experiments show that our monocular SLAM recovers metric scale with high precision, and achieves better accuracy than the state-of-the-art in stereo visual-inertial odometry when continually localizing in the same environment. We consider this *zero-drift* localization of particular interest for virtual/augmented reality systems, where the predicted user viewpoint must not drift when the user operates in the same workspace. Moreover we expect to achieve better accuracy and robustness by using stereo or RGB-D cameras, which would also simplify IMU initialization as scale is known. The main weakness of our proposed IMU initialization is that it relies on the initialization of the monocular SLAM. We would like to investigate the use of the gyroscope to make the monocular initialization faster and more robust.

ACKNOWLEDGMENT

We thank the authors of [14] for releasing the EuRoC dataset and the authors of [6] for providing their data for the comparison in Fig. 5.

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE Int. Conf. Robot. and Autom. (ICRA)*, 2007.

[2] K. Wu, A. Ahmed, G. Georgiou, and S. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Robot.: Sci. and Syst. (RSS)*, 2015.

[3] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst. (IROS)*, 2015.

[4] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. and Auton. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.

[5] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[6] V. Usenko, J. Engel, J. Stueckler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *IEEE Int. Conf. Robot. and Autom. (ICRA)*, 2016.

[7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," in *IEEE Trans. Robot.*, 2016, DOI: 10.1109/TRO.2016.2597321 (to appear).

[8] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *IEEE Int. Conf. Robot. and Autom. (ICRA)*, 2016.

[9] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Robot. Res.*, vol. 30, no. 4, pp. 407–430, 2011.

[10] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robot.: Sci. and Syst. (RSS)*, 2015.

[11] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, 2012.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM Int. Symp. Mixed and Augmented Reality (ISMAR)*, 2007.

[14] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[15] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *Int. J. Comput. Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[16] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration," *IEEE Trans. Autom. Sci. and Engineering*, 2016, DOI: 10.1109/TASE.2016.2550621 (to appear).

[17] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robot. and Autom. Lett.*, vol. 2, no. 1, pp. 18–25, 2017.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Comput. Vision*, 2nd ed. Cambridge University Press, 2004.

[19] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst. (IROS)*, 2013.

[20] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," *arXiv preprint arXiv:1610.06475*, , 2016.

[21] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Int. Conf. Robot. and Autom. (ICRA)*, 2011.

[22] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Society America A*, vol. 4, no. 4, pp. 629–642, 1987.

[23] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM." in *Robot.: Sci. and Syst. (RSS)*, 2010.

[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM system.," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst. (IROS)*, 2012.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2012.