2022 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Santa Maria da Feira, Portugal – April 29-30, 2022

2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 978-1-6654-8217-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICARSC55462.2022.9784783

Sensors Calibration and Filter Initialization in Visual Inertial Odometry

André Pereira Nogueira
Institute for Systems and Robotics, LARSyS
Instituto Superior Técnico, University of Lisbon
Email: anogueira@isr.tecnico.ulisboa.pt

José António Gaspar
Institute for Systems and Robotics, LARSyS
Instituto Superior Técnico, University of Lisbon
Email: jag@isr.tecnico.ulisboa.pt

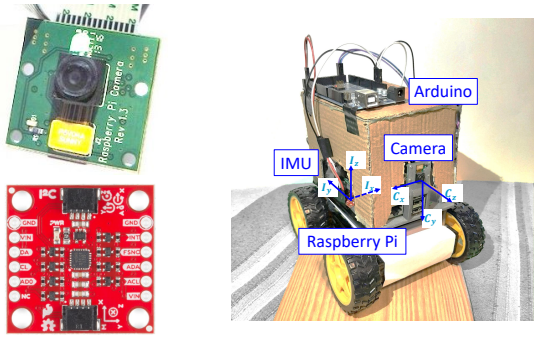
***Abstract*—Visual inertial odometry (VIO) is an enabling technology for mobile home robots which is possible nowadays due to novel low-cost cameras and inertial measurement units (IMUs). This work analyzes how the system-hardware calibration and system-state initialization, followed by the dynamical propagation of IMU readings, involving intrinsic bias, influence the VIO. Then we analyze the synergies between these two sensors in the context of pose estimation. The calibration of the system, namely the rigid transformation between camera and IMU is performed with a free, public domain, tool. The visual and inertial data is processed by means of an Unscented Kalman Filter with Lie group embedding for state representation. We propose a state initialization for this filter that enables matching the integrated IMU readings with the tracked visual features. Experiments and results show the importance of successfully matching the feature tracks in the first images with the starting integration of IMU readings, as significant errors in initial bias estimations may preclude sensors fusion and filter convergence. Results show also that the operation of the fusion filter allows synergies between the two sensors, while the IMU provides instantaneous and reliable estimations of translation and rotation speeds, the visual component provides IMU bias correction.**

***Index Terms*—Visual Inertial Odometry (VIO), Filter Initialization, Sensor Calibration.**

I. Introduction

Visual Inertial Odometry (VIO) combines imaging (video) with inertial (IMU) information to continuously estimate the pose of a robot. The IMU specializes in tracking the movement and orientation of objects. It is often used in navigation applications like aerial vehicles. However, its newfound accessibility in current times has facilitated its integration into low-cost systems.

Combining cameras and IMUs in the context of pose estimation has multiple advantages, the main of which being the possibility of creating synergies between these sensors. In practice, this means the visual information can help calibrate the IMU (bias), while the IMU provides motion scale to the visual information. In this work we develop a low-cost system that can perform visual inertial odometry, in order to analyze the synergy between sensors. Sensors calibration and filter initialization are key aspects for obtaining an effective navigation system.



(a) Main sensors camera and IMU

(b) Complete setup reference frames

Fig. 1. Wheeled mobile robot with camera and IMU mounted on fixed positions.

In the literature, handling precisely with IMU bias is critical point, when building navigation systems using low-cost IMUs [1]. In general, works as [2] show it is important estimating and removing bias from the IMU readings, independently of their price tag.

Simultaneous Location and Mapping (SLAM) involves pose (position and orientation) estimation, and therefore provides an output intersecting with VIO. [3] compares the performance of visual SLAM (vSLAM) with visual inertial SLAM (viSLAM), and concludes that typically viSLAM has better performance. The intrinsic difficulty of vSLAM to capture motion scale, its lesser robustness and accuracy, are the main reasons affecting performance. Works as [4] have set out to implement already proven algorithms on low-cost ground robots. They show it may be necessary to compromise filter performance due to the lack of computational power of low-cost robots.

In an opposite direction, of accepting computational complexity, [5] proposes a state of the art versatile monocular visual inertial state estimator by means of a nonlinear optimization-based method. The approach [5] is based on the alignment of a visual structure obtained by means of a structure from motion strategy with the results of an IMU preintegration. Work [5], and the previously referred ones, show that it is worthwhile to pursue the idea of creating a low-cost visual inertial system dedicated to odometry. They also indicate that the process of IMU bias removal has a significant impact on the overall odometry performance.

In this work, we study inertial sensors combined with cameras, focusing on IMU bias estimation and removal, and vision / IMU synergies. We consider a setup based on low-cost sensors, IMU and camera, mounted on a small mobile robot (see Figure 1). The visual and inertial data allow obtaining VIO, more precisely pose estimation along time, using an Unscented Kalman Filter (UKF) based on Lie groups [6]. To perform sensor data fusion all data must be represented in a single frame, which can be achieved through calibration, as detailed in this work. One specific challenge tackled in this work is filter initialization, namely in the aspect of significant inaccuracy on IMU bias estimates precluding the matching of visual features, and therefore deterioration cycle which leads to VIO filter divergence.

II. Background and State of the Art

In this work we consider a robot with two sensors: an RGB camera and a 9 DoF IMU. Each sensor captures data in a referential (frame) it defines internally. We consider the pinhole model for the camera associated with the projection $P = K[R\,t]$, where R and t are the extrinsic parameters of the camera (i.e., the rigid transformation that map its referential to the world frame), and K is the intrinsic parameter's matrix and P is the projection matrix that maps world points to an image plane using homogeneous coordinates (scale factor λ) $\lambda\mathbf{x} = P\mathbf{X}$.

One critical aspect of monocular visual only odometry is that a single camera cannot capture metric information, only captures 3D information up-to a scale factor [7]. In other words, one has ambiguity in the motion scale, the estimated trajectory is unitless. This problem can be solved by assuming priors, as known camera-to-floor height, or using additional sensors, as IMUs. IMUs, unlike RGB cameras, are able to capture motion scale [5].

As for the IMU, we use a 9 DoF IMU (accelerometer, magnetometer and gyroscope), encompassing bias and additive (white) measurement noise. We represent the IMU measurements as angular rate $\omega = [\omega_x\ \omega_y\ \omega_z]^T \in \mathbb{R}^3$, linear acceleration $a = [a_x\ a_y\ a_z]^T \in \mathbb{R}^3$, and magnetic field $m = [m_x\ m_y\ m_z]^T \in \mathbb{R}^3$. In Equation 2 we have (embedded) the complete measurement model of the accelerometer and gyroscope measurements.

Visual inertial odometry can be implemented with various alternative frameworks, e.g. using Kalman filters. The original Kalman Filter, as presented in [8], is a statistically optimal method of data fusion. This means the weighted fusion between the prediction and the correction of the next state is statistically optimal. This filter can only be applied to linear systems. Non-optimal extensions were developed like the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF, see [9]).

ORB-SLAM3 [10] is a state of the art SLAM implementation. It includes incremental odometry estimation within mapping and localization of the robot in the map. ORB-SLAM3 can perform Visual, Visual Inertial and Multi Map SLAM. The supported camera setups are monocular, stereo and RGBD, and is also possible to choose between pinhole and fisheye lens models. Experimentally, ORB-SLAM3 is shown to be as robust as the best available systems and significantly more accurate.

Fusion18 [6] performs VIO with a monocular camera and an IMU. It compares favorably, in terms of precision and accuracy, to ORB-SLAM2 (ORB-SLAM3 was published some years later, improving mostly on mapping). [6] integrates Lie groups with a Square-Root Unscented Kalman Filter (SR-UKF) to compute the state evolution. This approach is the culmination of several works (i) the Lie group structure of SLAM advocated in the field of invariant filtering, see e.g. [11]–[13] and (ii) the UKF on Lie Groups (UKF-LG), whose general methodology has been introduced in [14].

While *Fusion18* is state of the art on the computation of the state evolution, aspects as sensors calibration and filter initialization still require work when considering novel low-cost IMUs.

III. Mobile Robot Setup and VIO Filter

In order to acquire visual inertial data, we developed a mobile robot based on a Multi-Chassis 4WD Kit controlled by a Raspberry Pi and equipped with a Pi board camera and a 9 DoF IMU (IMU Breakout ICM-20948 Qwiic SEN-15335). Figure 1 shows the robot including the coordinate systems of each sensor.

The camera and the IMU do not have their coordinate systems aligned (Figure 1). We consider three frames: the world frame W, the body (IMU) frame B and the camera frame C. After using Matlab's Camera Calibrator application and the Kalibr Allan Matlab toolbox[1] to perform camera and IMU calibration, respectively, we performed the IMU-Camera calibration ($C \leftrightarrow B$) with the Kalibr OpenCV framework[2]. It estimates the extrinsic parameters $(\mathbf{R}_{C\to B}, \mathbf{t}_{C\to B})$ and $(\mathbf{R}_{B\to C}, \mathbf{t}_{B\to C})$.

In the following we do a detailed introduction to the base VIO filter, *Fusion18* by M. Brossard et al. [6], used in our setup.

The state structure includes the robot pose, which is formed by its position $\mathbf{x} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in SO(3)$, plus the velocity $\mathbf{v} \in \mathbb{R}^3$, the IMU biases $b_\omega \in \mathbb{R}^3$ and $b_a \in \mathbb{R}^3$, and the 3D position of p landmarks $\mathbf{p}_1, \ldots, \mathbf{p}_p \in \mathbb{R}^3$ in W. The state is the pair (χ, b), where χ is as square matrix

$$\chi = \begin{bmatrix} \mathbf{R} & \mathbf{v} & \mathbf{x} & \mathbf{p}_1 \cdots \mathbf{p}_p \\ \mathbf{0}_{(p+2)\times 3} & & \mathbf{I}_{(p+2)\times(p+2)} & \end{bmatrix} \quad (1)$$

with size $(3+2+p)\times(3+2+p)$ and with $\mathbf{0}_m$ and $\mathbf{I}_m$ as, respectively, zero matrix and identity matrix with sizes m. The bias vector is defined as $b = [b_\omega^T\ b_a^T]^T \in \mathbb{R}^6$.

Even though our hardware setup is different, our models are based on the ones present in [6]. We have a grounded body navigating on the ground equipped with an IMU. Despite having a car like robot, the model representing a full 3D

[1]https://github.com/rpng/kalibr_allan
[2]https://github.com/ethz-asl/kalibr

pose is advantageous because, for our case, as it allows us to anticipate scenarios like floors with inclinations. We can model the system as

$$\text{body state} \begin{cases} \dot{\mathbf{R}} = \mathbf{R}(\omega - b_\omega + n_\omega)_\times \\ \dot{\mathbf{v}} = \mathbf{R}(a - b_a + n_a) - g \\ \dot{\mathbf{x}} = \mathbf{v} \end{cases}, \tag{2}$$

$$\text{IMU biases} \begin{cases} \dot{b_\omega} = n_{b_\omega} \\ \dot{b_a} = n_{b_a} \end{cases}, \tag{3}$$

$$\text{landmarks} \left\{ \dot{\mathbf{p}}_i = 0, i = 1, \ldots, p \right. , \tag{4}$$

where $(\omega)_\times$ portrays the skew-symmetric matrix related with the cross product with vector $\omega \in \mathbb{R}^3$. We can group the multiple noises as $n = [n_\omega^T \; n_a^T \; n_{b_\omega}^T \; n_{b_a}^T]^T \sim \mathcal{N}(0, Q)$.

Using the Euler method we can discretize Equations (2) to (4), not including rotation. For a small time step Δt, we get

$$\begin{aligned} &\mathbf{R}_{t+\Delta t} = \mathbf{R}_t \mathbf{exp}\left[(\omega_t - b_{\omega,t})\Delta t + \mathbf{Cov}(n_\omega)^{1/2} g\sqrt{\Delta t}\right]_\times \\ &\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + (\mathbf{R}_t(a_t - b_{a,t}) - g)\Delta t, \quad \mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{v}_t \Delta t. \\ &b_{\omega,t+\Delta t} = b_{\omega,t}, \quad b_{a,t+\Delta t} = b_{a,t}, \quad \mathbf{p}_{i,t+\Delta t} = \mathbf{p}_{i,t} \end{aligned} \tag{5}$$

Along with the IMU, we have the calibrated monocular camera that provides visual information and observes and tracks p landmarks in the visual scene. The camera observes landmark $\mathbf{p}_i$ through pinhole model as $\mathbf{y}_i = [y_u^i \; y_v^i]^T + n_\mathbf{y}^i$, where y_i is the result of projection:

$$\lambda[y_u^i \; y_v^i \; 1]^T = K[\mathbf{R}_{B\to C}^T \; (\mathbf{R}^T \; (\mathbf{p}_i - \mathbf{x}) - \mathbf{t}_{B\to C})], \tag{6}$$

with λ as the scale factor. This is a transformation from frame W to current image plane using the state's pose, $\mathbf{R}$ and $\mathbf{x}$ ($W \to B$), the IMU-camera extrinsic parameters, $\mathbf{R}_{B\to C}$ and $\mathbf{t}_{B\to C}$ ($B \to C$), and the camera intrinsic parameters, K ($C \to$ image plane).

The filter compares the projection to the expected position of the landmark in the image. This process invalidates the 3D landmarks that are too distant to the respective expected 2D feature.

IV. VIO Filter State Initialization

As referred, we use *Fusion18* [6] as the base VIO filter. We propose methods for (i) filter initialization and (ii) features management. See Figure 2 and details in the following.

A. Feature Management, Limiting the Deterioration Cycle

Before detailing our methodologies for feature management, it is important to detail the cycle of sensing deterioration that occurs due to calibration or initialization inaccuracies.

At the start of each filter iteration, the state is propagated using the IMU data. Each propagation implies an integration of IMU data which causes the state to drift from reality due to IMU bias. When the first image is captured, the state's pose will be used to project to that image the 3D landmarks initially available. A drifted pose results in a noisy / biased projection, which causes landmarks invalidation (removal from state) if they land off an established window. If all landmarks get invalidated, the filter will disregard the visual information completely and therefore does not estimate IMU biases.

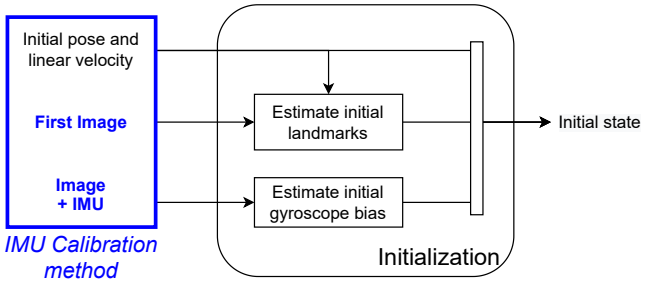


Fig. 2. Proposed initialization methodology.

If IMU biases are not estimated and removed from the measurements, they keep being integrated, creating a deterioration cycle that will result in filter divergence. We term the segment where the deterioration cycle may occur as the *initialization period*.

Our use of features management seeks to improve the robustness of the initial pose estimation in order to avoid the deterioration cycle. The most critical aspect to control is the initial number of state propagations before the first landmark observation. One wants to upper-bound, limit, the number of state propagations without visual updates. In other words, one wants the predicted projections of the 3D landmarks do not step larger than the visual feature's registration window.

In summary, as compared to *Fusion18* [6], which relies on an auxiliary SLAM filter for 3D landmarks initialization and resetting while showing it is possible to obtain more accurate and precise pose estimations (Section II), we propose an embedded process based on features tracking and 3D reconstruction scaled from IMU motion data. The embedded process allows us to select more frequently landmarks replacement, which is helpful for scenarios lacking texture at some areas.

In addition, we propose initializing the VIO with a batch optimization process to scale the visual reconstruction and to tune bias estimation, to be detailed in the next section.

B. Initialization

The initialization is responsible for establishing the initial state (χ, b) of the system (Equation 1). We use zero initial values for the robot pose and linear velocity (other values can be used, e.g. from a past or parallel VIO process, or from ground truth to help assessment). That leaves the initial 3D landmarks and the IMU biases to be initialized.

a) Landmark Initialization: During the filter initialization, new 3D landmarks are found by triangulation of features detected on multiple frames. In addition we need to capture a bank of features whose reconstruction can be used to replace the initial landmarks. This bank of features can be obtained with an algorithm as the Harris corner detector.

To estimate the 3D positions of detected features that initialize the state, we convert pixel coordinates, (u, v), to

metric coordinates, $[X,\ Y]^T = [(u-c_x)/fs_x,\ (v-c_y)/fs_y]^T$. Depth is arbitrarily set as $Z = 1$ for all landmarks, until the second frame, and the following ones, together with IMU readings, provide information to update those values. Given the 3D landmarks on the camera frame, we want to transform them to a frame that allows integrating IMU readings. This can be done using the the camera-IMU extrinsic parameters, $\mathbf{R}_{C\to B}$ and $\mathbf{t}_{C\to B}$ ($C \to B$), and the initial state pose, $\mathbf{R}$ and $\mathbf{x}$ ($B \to W$).

b) Gyroscope Bias Estimation: As mentioned, the removal of bias from IMU measurements is one of the most critical aspects of developing a VIO system. Let us now detail the estimation of the gyroscope bias, which was inspired in the work [5]. To perform this estimation we have an initial time period where the visual and inertial data is acquired (VIO is not running at this time). Let us consider I and J as the total number of images and IMU samples in the dataset, respectively. The first step is to estimate the orientation of all image frames, q_i (quaternions with $i = 1, \ldots, I$), using only the visual information. To achieve these orientation estimates we use the MonoSLAM algorithm of [15].

Secondly (or in parallel), we preintegrate the inertial data in the initial camera frame C (which may be a trivial zero rotation and zero position) in order to estimate the rotation matrices between each image frame, $\hat{\gamma}_i$, $i = 1, 2, \ldots, I$. We consider the starting bias to be null $[0\ 0\ 0]^T$, or a 3×1 vector of values previously obtained by a calibration method. For j as a discrete moment corresponding to a IMU sample within $[t_i, t_{i+1}]$, we can discretely integrate the gyroscope measurements as

$$\gamma_{j+1} = \gamma_j \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\omega_j - b_{\omega_j})\delta t \end{bmatrix}, \tag{7}$$

with δt as the time interval between IMU samples j and $j+1$. In order to get the rotation between image frames $i-1$ and i, we need to accumulate the rotations within that interval by doing the quaternion multiplication between consecutive rotations following the correct order.

We now have all the camera orientations estimated from the visual data, and have all rotation constrains between camera poses estimated from the inertial data. Due to bias and noise, a frame with orientation q_i rotated by $\hat{\gamma}_{i+1}$ is not equal to the frame with orientation q_{i+1}. To combine all the data we minimize a cost function:

$$\min_{\delta b_\omega} \sum_{i\in I} \left\| q_{i+1}^{-1} \otimes q_i \otimes \gamma_{i+1} \right\|^2 \tag{8}$$

where γ_{i+1} is the orientation constrain with first-order correction with respect to the gyroscope bias, $\gamma_{i+1} \approx \hat{\gamma}_{i+1} \otimes [1 \quad \frac{1}{2} J_{b_\omega}^{\gamma}\, \delta b_\omega]^T$, and $J_{b_\omega}^{\gamma}$ is a part of the first-order Jacobian, J_{i+1}, of the covariance matrix, P_{i+1}, obtained in the IMU preintegration (see [5]).

By initializing both starting orientations (MonoSLAM and preintegration) as an identity quaternion (null rotation) we are guaranteeing both methods to estimate orientation in the same reference since we have both body B and camera C frames fixed in relation to each other. Because we are only interested in rotations, and the whole systems rotates as one, all estimations stay coherent. With that done, we have now estimated the initial bias. It is possible to repropagate the rotations between frames as (8) with the estimated bias and repeat the process.

V. Experiments and Results

In this section we present experimental results on the continuous estimation of the robot pose using the methodologies proposed in Section IV.

A. Estimated Pose Error Metrics

In order to assess experimental results, we use two metrics: the root-mean-square error (RMSE) metric that evaluates the residuals in a point to point basis, and a Procrustes based metric that aligns the estimates to the ground truth using the Procrustes problem to find the rigid transformation (with scale) that most closely performs this alignment. The first metric will be used to evaluate all orientation estimates, while the second will be used for trajectories estimated with our acquired datasets.

B. IMU and Video Datasets

Two datasets were acquired with our mobile robot, with an approximately constant visual sampling rate, while the inertial sampling rate is kept varying in order to maximize the throughput of the sensor and the microcontroller [3]. These datasets start and end with a few seconds of no motion to help calibration and ground truth acquisition.

Movement Curve *(mc_01)* Ground plane curves implying rotations over the IMU's z axis. At the end of the path, robot already stopped, the visual information shows significant luminosity changes. Acquisition rates of 10 Hz (average) and 5 Hz for the inertial and visual data, respectively.

Movement Forward Pause Forward *(mfpf_02)* Two straight line forward movements, Figure 4(a), separated by a resting period. Acquisition rates of 10 Hz (average) and 5 Hz for the inertial and visual data, respectively.

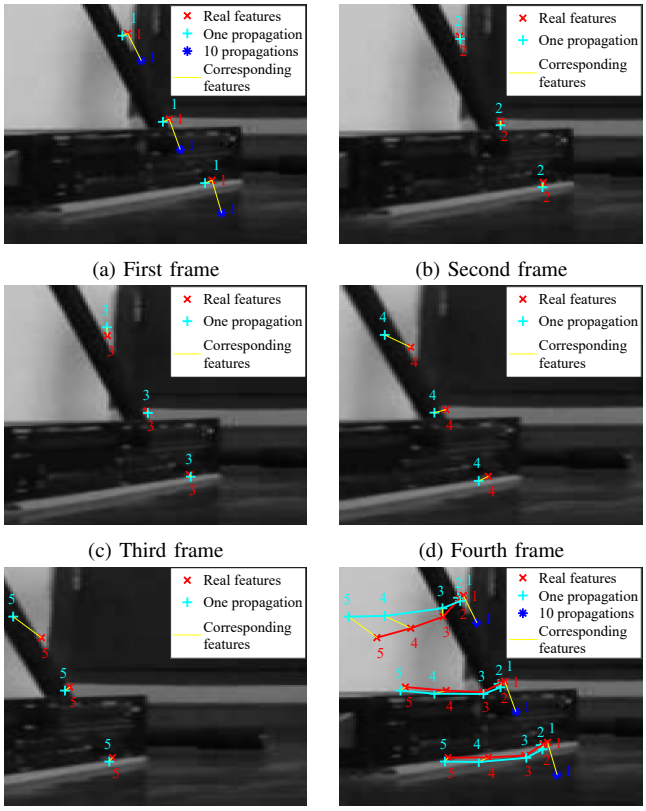
The datasets are complemented with ground truth. We video-record the car with a fixed camera and mark key positions on an homography making an orthographic view of the ground plane. From the key positions we interpolate the complete trajectory. The orientation is acquired directly from the IMU as it provides precise information through the embedded Digital Motion Processor (DMP).

C. Initial Frames Fusion with IMU Readings

First we detail an experiment to validate the methodologies presented in Section IV and then we detail navigation experiments on datasets acquired by the developed mobile robot.

The first experiment shows prevention of deterioration cycle, allowing the filter to track the real pose. We use the filter with our proposed methodologies and compare the 3D

[3]Data access source code https://github.com/josegaspar999/data_access/tree/master/datasets/AndreNogueira . To download and display data run `data_download('mfpf_02'); sentmove2_data_tst(40);`



(a) First frame (b) Second frame

(c) Third frame (d) Fourth frame

(e) Fifth frame (f) Summary, all time samples

Fig. 3. Feature tracking on first 5 frames with and without IMU sample limitation. Features: red are real (tracked) features; cyan and blue are estimated features with and without IMU sample limitation, respectively;

landmark projections for the first 5 frames with and without the IMU propagations limitation. The results are shown in Figure 3. For clarity, we refer to the colors of the markers indicating the features. The red features are the real (tracked) features, cyan features are the estimated features with IMU sample limitation (one propagation), and dark blue features are the estimated features without IMU sample limitation (10 propagations).

The reason why the blue features only appear in the first frame is because they are invalidated due to their reprojection error (they are too far from the red features). The filter renders these features as invalid so they would not deteriorate the state estimation. An additional experiment consisting in hijacking the filter and demanding it to further estimate the state with these invalid landmarks resulted in reprojections outside the bounds of the image.

Without an adequate initialization and feature management methodologies, as described in Section IV-A, all initial landmarks would get invalidated. Then, the estimated IMU biases would grow and result into VIO filter divergence. As we limited state propagation until sufficient visual information was acquired, we ensured the filter to converge, as shown by successful features tracking (cyan) in Figure 3(f).

D. Navigation Assessment

Navigation assessment is conducted on dataset *mc_01*. We perform different experiments by configuring the process noise covariance, which is related to the IMU measurements, and by default resulted from the IMU calibration (IMU intrinsic parameters). Figure 4 ((b), (c) and (d)) shows the Procrustes error metric on the robot position.

Estimated trajectories are similar for both configurations of the process noise. Figure 4(c) shows the filter is not being able to obtain a synergy of the sensor readings as the RMSE is constantly raising. Increasing the process noise covariance allows the filter to attenuate the negative effects of an initially too large IMU bias. Figure 4(d) shows a synergy between the sensors is present, namely the visual data regulates the IMU bias and the IMU provides a motion scaling for the monocular vision. The RMSE is kept low for most of the time.

About time $t = 25\,\mathrm{s}$ the RMSE has an almost vertical increase, which is the time where one observes unstable illumination leading to detection and tracking of multiple unreliable features within shadows. The vision part of the filter perceives a wrong movement. Since we increased the process noise covariance the filter trusts more on visual than IMU measurements. The process noise configuration explains why the illumination problem does not appear to exist in Figure 4(c). Transient problems with visual measurements indicate the need to re-initialize (reset) the filter.

E. Filter Reset using the Past State

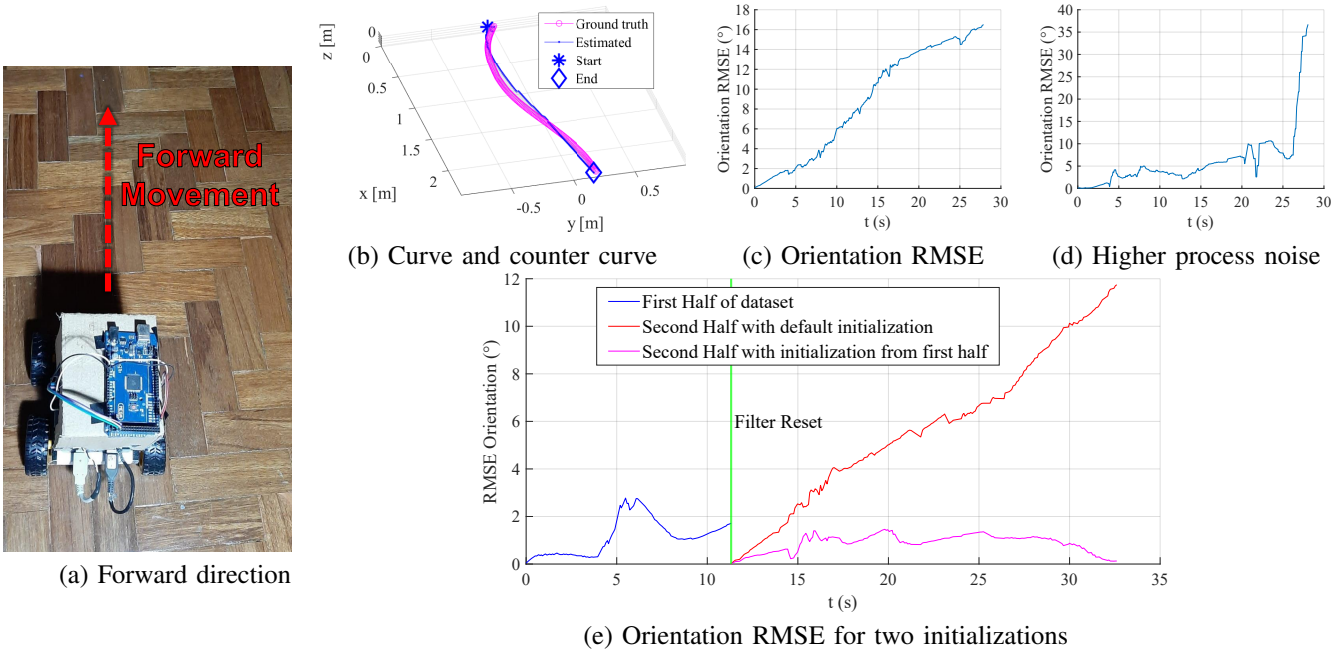
The steps for this experiment are (1) run the filter on half of a dataset; (2) save the final state along with all detected features and landmarks; and (3) run the filter on second half of the dataset but using the previous run final state as initialization data. The variables we are considering in this initialization are (i) both sensor biases, (ii) the 3D landmark estimations or (iii) the reserved features along with their sighting in past images. We initialize the pose and linear velocity to zero values.

More in detail, we run the filter on dataset *mfpf_02*, encompassing two similar trajectories, that are separated by a resting phase, on which we make the filter reset. We perform this experiment using the initialization detailed in Section IV-B to initialize the state after the reset. This allows us to compare our generic and (mostly) online initialization to this calibration like approach.

Figure 4(e) shows the RMSE just on orientation since it is clearer than assessing the full poses. Running an initialization, after resetting, not using the knowledge of the previous state, shows a linearly increasing error which is the result of a failed initialization process (red line in Figure 4(e)). The run after resetting, doing an initialization taking advantage of the previous state, shows a short, or almost non existing initialization period due to accurate initialization data (magenta line in Figure 4(e)).

VI. Conclusions

In this work we considered a low-cost mobile robot, that acquires visual and inertial data, and studied the sensor fusion / synergies in the context of pose estimation. We proposed a



(a) Forward direction

(b) Curve and counter curve

(c) Orientation RMSE

(d) Higher process noise

(e) Orientation RMSE for two initializations

Fig. 4. Mobile robot on a forward trajectory (a). Orientation RMSE when running dataset *mc_01* with default process noise (c) and higher noise (d), plus representative trajectory (b). Orientation RMSE when running the two halves of dataset *mfpf_02* with a reset in the middle (e). After the reset: red with default initialization and magenta with calibration initialization.

state initialization for a Unscented Kalman Filter based on Lie groups that experimentally showed promising results.

A critical aspect observed is the necessarily fast IMU bias estimation, as otherwise the IMU measurements cause the state to diverge. Significantly incorrect IMU bias estimations imply invalidating visual landmark observations. This would start a cycle of deterioration between both sensors.

Given approximately correct initializations, one obtains the desired cycle of cooperation between sensors. The bias estimation coming from the visual information makes the landmark projection more accurate, which in term would improves the bias estimation.

As future work, we consider allowing each sensor to operate independently, for some time, in order to handle cases of insufficient illumination or IMU bias momentarily incorrect. In the particular, we consider the recent Madgwick filter [16] as an effective way for operating the IMU in case of momentarily missing the visual data.

ACKNOWLEDGMENT

This work was supported by the Portuguese Foundation for Science (FCT) with the LARSyS - FCT Project UIDB/50009/2020.

REFERENCES

[1] Y. Liu, N. Noguchi, and K. Ishii, "Development of a Low-cost IMU by Using Sensor Fusion for Attitude Angle Estimation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4435–4440, 2014.

[2] G. G. Scandaroli and P. Morin, "Nonlinear filter design for pose and IMU bias estimation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4524–4530, 2011.

[3] S. M. Potirakis, M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *Journal of Sensors*, 02 2021.

[4] L. Bo, L. Li, and H. Liu, "SoC Implementation of Visual-inertial Odometry for Low-cost Ground Robots," *Journal of Physics: Conference Series*, vol. 1453, p. 012091, 01 2020.

[5] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[6] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented Kalman Filtering on Lie Groups for Fusion of IMU and Monocular Vision," *International Conference on Robotics and Automation (ICRA)*, 2017.

[7] A. J. Davison, R. I., N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[8] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[9] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI* (I. Kadar, ed.), vol. 3068, pp. 182 – 193, International Society for Optics and Photonics, SPIE, 1997.

[10] C. Campos, R. Elvira, J. Gómez, J. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *arXiv preprint arXiv:2007.11898*, 2020.

[11] A. Barrau and S. Bonnabel, "An EKF-SLAM algorithm with consistency properties," 2016.

[12] A. Barrau and S. Bonnabel, "Invariant Kalman Filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, 05 2018.

[13] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and Consistency Analysis for A 3D Invariant-EKF SLAM," *CoRR*, 2017.

[14] M. Brossard, S. Bonnabel, and J. Condomines, "Unscented Kalman Filtering on Lie Groups," in *IROS 2017, EEE/RSJ International Conference on Intelligent Robots and Systems*, (Vancouver, Canada), IEEE/RSJ, Sept. 2017.

[15] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[16] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, pp. 113–118, 2010.