2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Fast and Robust Initialization for Visual-Inertial SLAM

Carlos Campos, José M.M. Montiel and Juan D. Tardós

***Abstract*— Visual-inertial SLAM (VI-SLAM) requires a good initial estimation of the initial velocity, orientation with respect to gravity and gyroscope and accelerometer biases. In this paper we build on the initialization method proposed by Martinelli [1] and extended by Kaiser et al. [2], modifying it to be more general and efficient. We improve accuracy with several rounds of visual-inertial bundle adjustment, and robustify the method with novel observability and consensus tests, that discard erroneous solutions. Our results on the EuRoC dataset show that, while the original method produces scale errors up to 156%, our method is able to consistently initialize in less than two seconds with scale errors around 5%, which can be further reduced to less than 1% performing visual-inertial bundle adjustment after ten seconds.**

I. INTRODUCTION

Visual-Inertial SLAM stands for those techniques able to build a map and simultaneously locate an agent inside the map, using as input a camera and an Inertial Measurement Unit (IMU) [3] [4] [5]. Both sensors require to be initialized or calibrated before using them. While camera is calibrated just once as it does not evolve with time, IMU has to be initialized before every use. Inertial initialization aims to compute values for the initial velocity, gravity direction and gyroscope and accelerometer biases that can be used as initial seed for the Visual-Inertial SLAM estimation process, guaranteeing its convergence. Once these values are found, inertial measurements can be used to improve tracking, making it more robust, as well as to find the true scale of the map, which cannot be obtained with pure monocular systems.

Some techniques initialize first the monocular SLAM part, building a map up-to-scale, and then, enough time later for having good observability of all variables, compute scale factor, gravity direction and IMU biases. This technique, first proposed by Mur-Artal and Tardós [5] and latter adapted in [6], gives very accurate results. In particular, the true scale of the environment can be estimated with an error around 1%. However, this approach has two main drawbacks:

- Dependency on visual initialization. It requires to have an initial map up-to-scale with enough map-points. If the visual part is not able to initialize, or if it gets lost within a short period of time, the inertial system will not initialize.
- Initialization takes too long. The method requires to run monocular SLAM for around fifteen seconds to safely find the correct inertial parameters for the visual-inertial bundle adjustment (BA). This time is too long for many applications.

This work was supported in part by the Spanish government under grants DPI2015-67275 and DPI2017-91104-EXP, the Aragón government under grant DGA_T45-17R, and by Huawei under grant HF2017040003.

The authors are with Instituto de Investigación en ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain campos@unizar.es; josemari@unizar.es; tardos@unizar.es

Martinelli [1] proposed a very interesting closed form method that only requires to track a few points, and solves jointly for the camera motion, the environment structure, and the inertial parameters. The paper included a theoretical analysis of the conditions for the problem to be solvable and presented simulated results. The recent work of Kaiser et al. [2] extends the method to also estimate gyroscope bias, and presents results with real data from a drone, showing that it is possible to initialize in less than 3 seconds, with relative errors around 15% on speed and gravity. This initialization method has several limitations:

- It assumes that all features are tracked in all frames, and that all tracks provided are correct. In case of spurious tracks, it can provide arbitrarily bad solutions.
- The initialization accuracy is low, compared to [5]. To improve it, a lot of tracks and frames are needed, increasing its computational cost, and making it unfeasible in real time.
- With noisy sensors, trajectories that are close to the unsolvable cases analyzed in [1] give weak observability of some of the variables. The method lacks robust criteria to decide when an initialization is accurate enough.

In this paper we build on the *Martinelli-Kaiser solution* [1] [2] (or simply *MK-solution*), modifying it to be more general, efficient, robust and precise. The main novelties of our initialization algorithm are:

1) **Generality**: we generalize the method to use partial tracks and to take into account the camera-IMU relative pose.
2) **Efficiency**: we reduce the running time by using a fixed number of m features and n keyframes carefully chosen, and adopting the preintegration method proposed in [7] [8].
3) **Observability test**: after *MK-solution*, we perform visual-inertial BA with the m points and n keyframes, and apply a novel observability test to check the accuracy of the solution. If the test fails, the initialization is discarded.
4) **Consensus test**: we try to initialize additional tracked features by triangulating the 3D point and checking the reprojection error. If enough consensus is found, we perform a second visual-inertial BA with all points and keyframes, and initialize the SLAM map with them. Otherwise, the initialization is discarded.

We present exhaustive initialization tests on the EuRoC

dataset [9] showing that the method is able to consistently initialize in less than 2s with a scale error of 5%. This error converges to 1% performing visual-inertial BA after 10s, when the trajectory is long enough to give good observability of all variables.

II. Initial Solution

A. Feature extraction and tracking

We use the multiscale ORB extractor [10] to find M uniformly distributed points to be tracked. As points with high resolution are preferable, we only extract ORB points at the three finest image scales, using a scale factor of 1.2. We have found that the extraction of FAST features and matching with ORB descriptors is not stable enough to obtain long tracks, as most of the features were lost in some of the frames. Hence, to solve this problem, we have performed feature tracking using the pyramidal implementation of the Lucas-Kanade algorithm [11] available in OpenCV.

However, simple tracking is not enough since there could be some tracks which are not correct, particularly in areas of repetitive or low texture. To attack this problem, the geometric consistency of these tracks is checked by finding a fundamental matrix using RANSAC. The combination of ORB keypoints, Lucas-Kanade tracking (KLT), and the fundamental matrix check, leads to long tracks for a high number of features. When some track is lost or rejected by the RANSAC algorithm, new ORB points are extracted and start to be tracked, in order to keep a constant number of M tracks. Each time we detect that there are at least m tracked points with a track length of at least l pixels, we launch our initialization method. This first test to decide whether to attempt initialization or not is called *track-length test*.

B. Modified Martinelli-Kaiser solution

In this part we extend the method proposed in [1] [2] to be able to deal with features not seen by all cameras, and computing terms in an efficient way using inertial pre-integration proposed by [7] [8]. Let m be the number of tracked features used for initialization, placed at $(\mathbf{x}_1 \dots \mathbf{x}_m)$ in the global reference frame, and $\mathcal{C} = \{\mathcal{C}_1 \dots \mathcal{C}_n\}$ the set of n cameras indexes used for initialization, also referred to as keyframes, which are chosen to be uniformly distributed along time. Let's also call $\mathcal{C}^i = \{\mathcal{C}_{1_i} \dots \mathcal{C}_{n_i}\}$ the set of n_i cameras indexes seeing feature i-th. Here we remark that, in contrast to [1], in our formulation not all features have to be observed by all cameras, and the transformation from camera to body (IMU), obtained from calibration, is taken into account. Without loss of generality we use as global reference the first body pose used for initialization.

From figure 1, we can write the set of equalities:

$$\begin{aligned} \mathbf{p}_{1_i} + \mathbf{R}_{1_i}\mathbf{t}_{BC} + \lambda^i_{1_i}\mathbf{u}^i_{1_i} = \mathbf{p}_j + \mathbf{R}_j\mathbf{t}_{BC} + \lambda^i_j\mathbf{u}^i_j \\ i = 1 \dots m, \quad \forall j \in \mathcal{C}^i \setminus 1_i \end{aligned} \tag{1}$$

where:

- $\mathbf{p}_j \in \mathbb{R}^3$: position of the j-th body in the global reference frame.

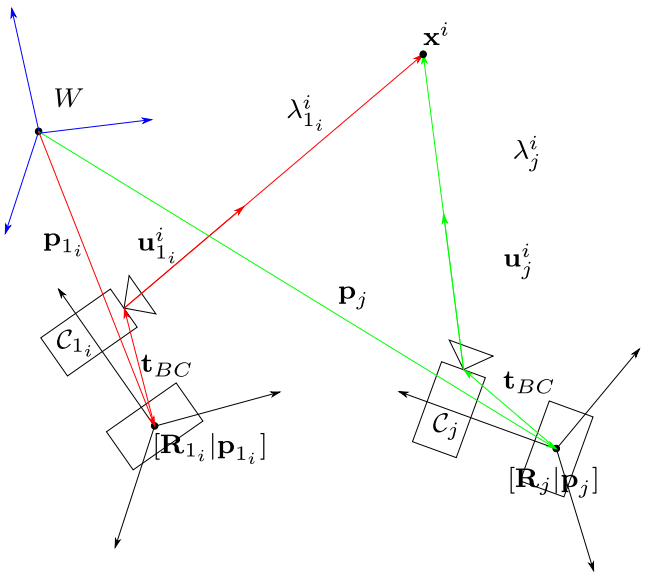

Fig. 1: Relationships between two Body (IMU) and Camera poses observing the same feature.

- $\mathbf{R}_j \in \mathrm{SO}(3)$: rotation matrix from j-th body to global reference frame.
- λ^i_j: distance between i-th feature and j-th camera.
- $\mathbf{u}^i_j$: unitary vector from j-th camera to i-th feature in the global reference frame. It can be computed as $\mathbf{u}^i_j = \mathbf{R}_j\mathbf{R}_{BC}\,{}_{c_j}\mathbf{u}^i_j$, being ${}_{c_j}\mathbf{u}^i_j$ the unitary vector in the j-th camera reference frame for the i-th feature, which is directly obtained from the tracked images.
- $[\mathbf{R}_{BC}|\mathbf{t}_{BC}]$: transformation from Camera to Body (IMU).

$\mathbf{R}_j$ and $\mathbf{p}_j$ can be computed by integrating the inertial measurements. Considering constant biases during the initialization time, it leads to [8]:

$$\mathbf{R}_{1,j}(\mathbf{b}^g) = \prod_{k=1}^{t_j-1} \mathrm{Exp}((\boldsymbol{\omega}^m_k - \mathbf{b}^g)\Delta t) \tag{2}$$

$$\mathbf{v}_j(\mathbf{b}) = \mathbf{v}_1 + \mathbf{g}\Delta t_{1,j} + \sum_{k=1}^{t_j-1} \mathbf{R}_{1,k}(\mathbf{a}^m_k - \mathbf{b}^a)\Delta t \tag{3}$$

$$\mathbf{p}_j(\mathbf{b}) = \mathbf{p}_1 + \sum_{k=1}^{t_j-1} \left(\mathbf{v}_k\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \frac{1}{2}\mathbf{R}_{1,k}(\mathbf{a}^m_k - \mathbf{b}^a)\Delta t^2 \right) \tag{4}$$

where:

- Exp stands for the exponential map $\mathrm{Exp} : \mathbb{R}^3 \to \mathrm{SO}(3)$
- $\mathbf{a}^m_k$ and $\boldsymbol{\omega}^m_k$ are k-th acceleration and angular velocity IMU measurement
- $\mathbf{b} = (\mathbf{b}^a, \mathbf{b}^g)$ are their corresponding accelerometer and gyro biases
- $\mathbf{g}$ stands for gravity in the global frame
- $\mathbf{v}_j$ is the velocity at the j-th body.
- $\Delta t_{i,j}$ denotes time difference between i-th and j-th poses.

However, the above expressions have an important drawback: each time that $\mathbf{b}^g$ or $\mathbf{b}^a$ are modified, all the IMU integration requires to be recomputed, which is very time consuming. To solve this problem we have adopted the linear preintegration correction from [7] [8], splitting these expressions in bias dependent and non-dependent terms, using delta terms $\Delta\mathbf{R}_{1,j}$, $\Delta\mathbf{v}_{1,j}$, $\Delta\mathbf{p}_{1,j}$, which are directly computed from IMU measurements and which are defined as follows:

$$\Delta\mathbf{R}_{1,j} \triangleq \mathbf{R}_1^T\mathbf{R}_j \tag{5}$$

$$\Delta\mathbf{v}_{1,j} \triangleq \mathbf{R}_1^T(\mathbf{v}_j - \mathbf{v}_1 - \mathbf{g}\Delta t_{1,j}) \tag{6}$$

$$\Delta\mathbf{p}_{1,j} \triangleq \mathbf{R}_1^T(\mathbf{p}_j - \mathbf{p}_1 - \mathbf{v}_1\Delta t_{1,j} - \frac{1}{2}\mathbf{g}\Delta t_{1,j}^2) \tag{7}$$

If during intermediate steps $\mathbf{b}^g$ changes more than 0.2 rad/sec from the value used for preintegration, the preintegration is recomputed with this new bias, otherwise, delta terms are directly updated using their Jacobians w.r.t. biases ($\frac{\partial\Delta\mathbf{R}_{1,j}}{\partial\mathbf{b}^g}$, $\frac{\partial\Delta\mathbf{v}_{1,j}}{\partial\mathbf{b}^g}$, $\frac{\partial\Delta\mathbf{v}_{1,j}}{\partial\mathbf{b}^a}$, $\frac{\partial\Delta\mathbf{p}_{1,j}}{\partial\mathbf{b}^g}$, $\frac{\partial\Delta\mathbf{p}_{1,j}}{\partial\mathbf{b}^a}$). In this way, we relinearize each time we get too far from the linearization point. These Jacobians can be found in [8]. We rewrite eq. (1) using expressions (5), (6) and (7):

$$\begin{aligned}\lambda_{1_i}^i\mathbf{u}_{1_i}^i - \lambda_j^i\mathbf{u}_j^i - \mathbf{v}_1\Delta t_{1_i,j} - \mathbf{g}\left(\frac{\Delta t_{1,j}^2 - \Delta t_{1,1_i}^2}{2}\right) = \\ \Delta\mathbf{p}_{1,j} - \Delta\mathbf{p}_{1,1_i} + (\Delta\mathbf{R}_{1,j} - \Delta\mathbf{R}_{1,1_i})\mathbf{t}_{BC} \\ \forall i = 1\dots m, \quad \forall j \in \mathcal{C}^i \setminus 1_i\end{aligned} \tag{8}$$

Now, we can add the gravity magnitude information. Instead of adding it as a constraint of the gravity magnitude as done in [2], we prefer to model the gravity by mean of a rotation matrix parametrized by only two angles (α, β) (rotation around z-axis has no effect) and the vector $\mathbf{g}_I = (0, 0, -g)$, thus we remain in an unconstrained problem:

$$\mathbf{g} = \mathrm{Exp}(\alpha, \beta, 0)\mathbf{g}_I \tag{9}$$

Equation (8) becomes:

$$\lambda_{1_i}^i\mathbf{u}_{1_i}^i - \lambda_j^i\mathbf{u}_j^i - \mathbf{v}_1\Delta t_{1_i,j} = \mathbf{s}_{1_i,j}(\mathbf{b}^g, \mathbf{b}^a, \alpha, \beta) \tag{10}$$

where:

$$\begin{aligned}\mathbf{s}_{1_i,j}(\mathbf{b}^g, \mathbf{b}^a, \alpha, \beta) = \mathrm{Exp}(\alpha, \beta, 0)\mathbf{g}_I\left(\frac{\Delta t_{1,j}^2 - \Delta t_{1,1_i}^2}{2}\right) + \\ \Delta\mathbf{p}_{1,j} - \Delta\mathbf{p}_{1,1_i} + (\Delta\mathbf{R}_{1,j} - \Delta\mathbf{R}_{1,1_i})\mathbf{t}_{BC}\end{aligned} \tag{11}$$

Now, each time biases are updated, we do not need to preintegrate again all the measurements, but only to update them by means of Jacobians. Neglecting accelerometer bias as in [2], the only unknowns are λ_j^i ($\forall i = 1\dots m$, $j \in \mathcal{C}^i \setminus 1_i$), $\mathbf{v}_1$, α, β and $\mathbf{b}^g$. Stacking equations for all possible values of i and j we build an overdetermined sparse linear system, with only three non-zero elements per row, such as:

$$\mathbf{A}(\mathbf{b}^g)\mathbf{x} = \mathbf{s}(\mathbf{b}^g, \alpha, \beta) \tag{12}$$

where $\mathbf{x} = (\mathbf{v}_1, \{\lambda_j^i\})$ is the unknown vector with linear dependence. To jointly find linear and no-linear dependent parameters, we solve the next unconstrained minimization problem:

$$(\mathbf{b}^g, \alpha, \beta) \triangleq \underset{\mathbf{b}^g,\alpha,\beta}{\arg\min}\left(\min_{\mathbf{x}}\|\mathbf{A}(\mathbf{b}^g)\mathbf{x} - \mathbf{s}(\mathbf{b}^g, \alpha, \beta)\|_2^2\right) \tag{13}$$

Cost function $c(\mathbf{b}^g, \alpha, \beta) = \min_{\mathbf{x}}\|\mathbf{A}(\mathbf{b}^g)\mathbf{x} - \mathbf{s}(\mathbf{b}^g, \alpha, \beta)\|_2^2$ is evaluated for each $(\mathbf{b}^g, \alpha, \beta)$ using the following scheme:

1) **Update $\Delta\mathbf{R}_{1,j}$ and $\Delta\mathbf{p}_{1,j}$**: Using [8], we don't need to reintegrate all IMU measurements each time that $\mathbf{b}^g$ changes. We simply update delta terms using their Jacobians w.r.t. bias. That supposes an important computational saving.
2) **Compute $\mathbf{A}(\mathbf{b}^g)$ and $\mathbf{s}(\mathbf{b}^g, \alpha, \beta)$** and build the linear system.
3) **Solve $\mathbf{x} = \mathbf{A}(\mathbf{b}^g)\backslash\mathbf{s}(\mathbf{b}^g, \alpha, \beta)$** using conjugate gradient, which is suitable for sparse systems.
4) **Compute** $c(\mathbf{b}^g, \alpha, \beta) = \|\mathbf{A}(\mathbf{b}^g)\mathbf{x} - \mathbf{s}(\mathbf{b}^g, \alpha, \beta)\|_2^2$.

The computational cost of evaluating $c(\mathbf{b}^g, \alpha, \beta)$ comes, first, from solving a sparse system with no more than $3 + n \times m$ unknowns and $3 \times (n-1) \times m$ equations and, second, from integrating inertial measurement along the initialization time. However, using formulation from [8], we can avoid reintegration, integrating IMU measures only once, and updating preintegrated terms by means of a linear approximation.

To optimize $c(\mathbf{b}^g, \alpha, \beta)$ and find the correct gyro bias and gravity direction we use Levenberg–Marquardt algorithm, where Jacobians of the cost function are computed numerically. As result, not only IMU initialization parameters are found ($\mathbf{g}$, $\mathbf{b}^g$ and $\mathbf{v}_1$) but also the position of tracked points ($\lambda_j^i\mathbf{u}_j^i$). We highlight that not all M tracked features have been used during this initialization, but only a small set of m features, aiming to reduce computational complexity. However, the solutions found after this step are not accurate enough to launch the system, and further intermediate stages are required.

III. Improved Solution

A. First BA and observability test

After finding the initial parameters ($\mathbf{g}$, $\mathbf{b}^g$, $\mathbf{v}_1$, $\{\lambda_j^i\}$) we build a visual-inertial BA problem with the same n body poses and m points from the previous step (see figure 2). We set the z axis in the estimated gravity direction. All body poses have six optimizable variables $(\phi, \mathbf{t}) \in \mathfrak{se}(3)$ except the first one, which has only two (pitch and roll) since translation and yaw have been fixed in order to remove the four gauge freedoms inherent to the visual inertial problem (initial position and yaw). Body velocities are also included in the optimization task, and they evolve according to the inertial measurements. Initial estimations for each vertex are added using results from the *MK-solution*. In addition,

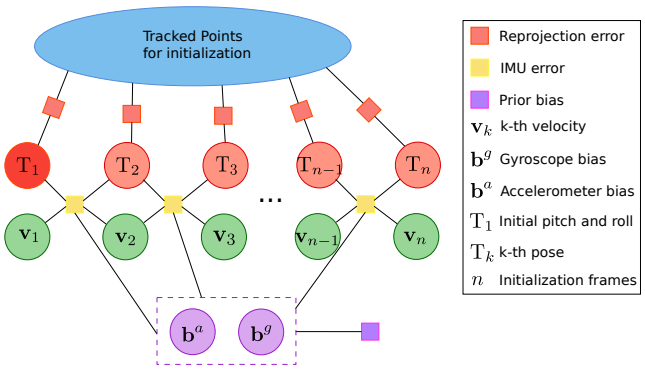

Fig. 2: Graph for the first visual-inertial BA. The body poses and points included in the optimization are the same used in the initial solution.

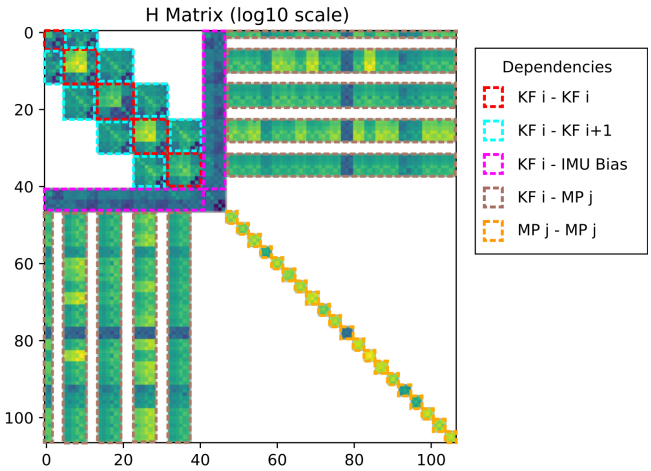

Fig. 3: Example of Hessian matrix for an initial map with 5 keyframes (KF) and 20 map points (MP). One can distinguish different blocks, outlined with dashed lines. In the top-left part, we have the diagonal blocks of each keyframe (red), blocks relating consecutive keyframes, due to the IMU measurements (blue), and blocks relating keyframes and IMU biases (pink). In the bottom-right part, there are only the diagonal blocks of the map points (orange). Out-of-diagonal terms relate map points with the keyframes that observe them (brown). In this example all cameras observe all features.

accelerometer bias $\mathbf{b}^a$ is included in this optimization, but similarly to $\mathbf{b}^g$ it is assumed to be constant for all frames. Previous $\mathbf{b}^g$ estimation from *MK-solution* step is included by means of a prior, as well as $\mathbf{b}^a$ is forced to be close to zero. We call this optimization *first BA* or simply *BA1*. Analytic expression for Jacobians, found in [8], are used for IMU residuals, while Jacobians for the reprojection error have been derived analytically, taking into account that we are optimizing body pose and not camera pose.

Usually this optimization provides a better initialization solution. However, if the motion performed gives low observability of the IMU variables, the optimization can converge to arbitrarily bad solutions. For example this happens in case of pure rotational motion or non-accelerated motions [1]. In order to detect these failure cases we propose an *observability test*, where we analyze the uncertainty associated to estimated variables. This could be done by analyzing the covariance matrix of the estimated variables and checking if its singular values are small enough. However, this would require to invert the information matrix, i.e. the Hessian matrix from *first BA*, which has high dimensions $(3m + 6 + 9n - 4)$, being computationally too expensive. Instead, we perform the observability test imposing a minimal threshold to all singular values of the Hessian matrix associated to our *first BA*. The Hessian can be built from the Jacobian matrices associated to each edge in the graph, as explained next.

Denote $\{\mathbf{x}_1 \dots \mathbf{x}_p\}$ the set of p states, and $\{\mathbf{e}_1 \dots \mathbf{e}_q\}$ the set of q measurements which appear in the *first BA*. Let's call $\mathcal{E}_i$ the set of measurement where state i is involved. The Hessian block matrix for states i and j, taking a first order approximation, can be built as follows:

$$\mathbf{H}_{i,j} \approx \sum_{\mathbf{e} \in \mathcal{E}_i \cap \mathcal{E}_j} \mathbf{J}_{i,\mathbf{e}}^T \Omega_{\mathbf{e}} \mathbf{J}_{j,\mathbf{e}} \tag{14}$$

where $\Omega_{\mathbf{e}}$ stands for the information matrix of the $\mathbf{e}$ measurement, and $\mathbf{J}_{i,\mathbf{e}}$ for the Jacobian of the $\mathbf{e}$ measurement w.r.t. i-th state. In order to have a non-zero (i, j) block matrix, there must to be an edge between i and j node in the graph (measurement depending on both variables) as shown in figure 3.

Applying the SVD decomposition to $\mathbf{H}$ and looking at the smallest singular value one can determine if the performed motion guarantees observability of all the IMU variables. Hence, we discard all initializations where the smallest Hessian singular value falls below a threshold denoted by t_{obs}. If this *observability test* is not passed, we discard the initialization attempt. Examples of a successful and a rejected case are shown in figure 4.

B. Consensus test and second BA

As we have noted before, not all M tracked features have been used in *MK-solution* and *first BA* steps, but only m features. To take advantage of these extra unused tracked points, we propose to perform a *consensus test* in order to detect initializations which have been performed using spurious data, such as bad tracked features.

First, the 3D point position of each unused track is triangulated between the two most distant frames which saw the point, by mean of Least-Squares triangulation using a SVD decomposition [12]. Only tracks with parallax greater than 0.01 radians are used. Then we re-project each 3D point into all the frames which observe it, compute the residual re-projection error, and perform a $\chi^2(95\%)$ test with $2n_i - 3$ degrees of freedom, where n_i is the number of frames which observe this point. The *consensus test* is performed counting the percentage of inliers: if it is bigger than a threshold t_{cons} we consider that the proposed solution is accurate, if not, we discard the initialization attempt.

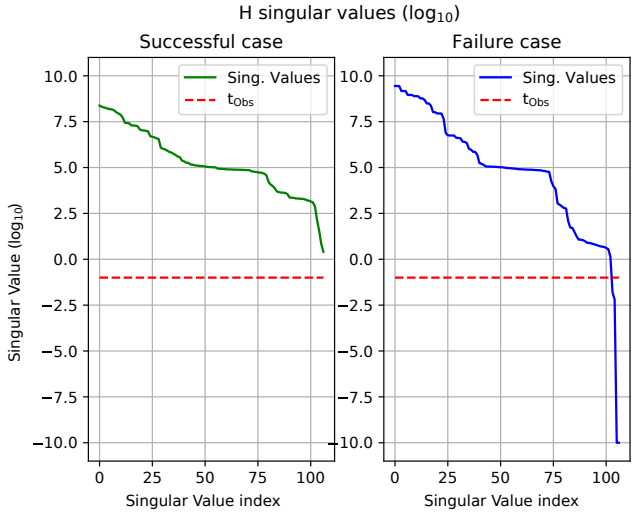

Fig. 4: Singular values of the information matrix for a successful initialization and a failure case on the EuRoC V103 sequence. The successful case has a RMSE ATE error of 3.16 % in the initialization trajectory, and corresponds to a translation and rotation motion. The failure case has an error of 64.99 % and corresponds to an almost pure rotational motion. We draw the observability threshold used $t_{obs} = 0.1$

If the *consensus test* is successful, we perform a *second BA* (or simply *BA2*) including the m points used in the initial solution plus all the points which have been triangulated and detected as inliers, having a total of M' points. The graph for this optimization is similarly built than in case of *BA1* but with more points.

C. Map initialization

After this second BA, the keyframe poses are accurate enough, but we only have a few points to initialize the map. Before launching the whole ORB-SLAM visual-inertial system, we triangulate new points aiming to densify the point cloud and to ease the posterior tracking operation. Since we already have the keyframe poses, we extract ORB features in each keyframe and perform an epipolar search in each other, using the ORB descriptor. All these new points, together with the M' points from *BA2*, are promoted to map points, and the n frames used for initialization are promoted to map keyframes. The covisiblity graph [13] of this new map is also created, taking into account the observations of points.

IV. Experiments

The most important parameters of our method are shown in table I. Our implementation uses ORB-SLAM visual-inertial [5] with its three threads for tracking, mapping and loop closing. Initialization is performed in a parallel thread, thus it has no effect in the real time tracking thread. For *MK-solution* we use Eigen C++ library, while for graph optimization of *BA1* and *BA2* we use g2o C++ library [14]. Experiments have been run in V1 dataset from EuRoC [9] using a Intel Core i7-7700 computer with 32 GB of memory.

TABLE I: Parameters of our initialization algorithm

Total number of tracks	M	200
Track-length test (in pixels)	l	200
Tracks used for *MK-solution*	m	20
Keyframes used for *MK-solution*	n	5
Observability test: Singular value threshold	t_{obs}	0.1
Consensus test: Inlier threshold	t_{cons}	90%

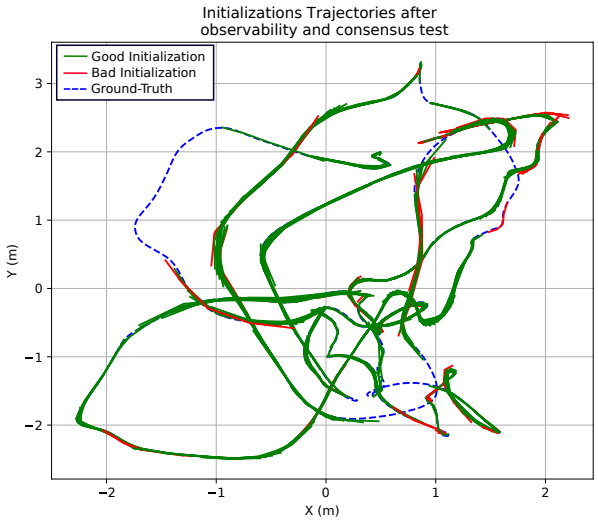

Fig. 5: Initializations found along the EuRoC V101 trajectory, after the *observability and consensus tests*. In blue, ground truth trajectory; in green, estimated initialization trajectories whose RMSE ATE error is lower than 5%; in red, those with a bigger error. Our method was able to find 511 correct initializations along the whole trajectory, running in real time.

A. Results

EuRoC dataset provides stereo images and synchronized IMU measures for three different indoor environments, with different complexity. We have tested our method for environment V1 from EuRoC at three difficulty levels. We run two different experiments.

In a first experiment, we try to initialize as often as possible in real time. Along the whole trajectory, every time the tracking thread has m tracks with length l, if the initialization thread is idle, a new initialization attempt is launched. Figure 5 shows the initializations found for trajectory V101 after the *observability and consensus test*. We show in red trajectories which have a RMSE ATE [15] error bigger than 5% of the initialization trajectory length. We can see in the figure that our initialization algorithm is successful almost along all the trajectory. The parts without initializations are due to rejection from observability or consensus test.

In table II we show the main numerical results of these experiments with the three V1 sequences. RMSE ATE [15] is expressed in percentage over the length of the initialization trajectory. Below each sequence name we show successful initializations over the total number. First thing to notice

TABLE II: Results of exhaustive initialization tests over the three V1 EuRoC sequences.

V1 EuRoC Dataset							
		After *track-length test*		After *Observ + Cons. test*			
Seq. Name		RMSE ATE (%)	Scale error (%)	RMSE ATE (%)	Scale error (%)	CPU time (ms)	Trajectory time (s)
V101 (511/728)	*MK-solution*	9.176	32.998	7.749	25.104	95.082	2.235
	MK-solution + BA1	3.977	10.719	2.352	6.471	104.114	2.235
	MK-solution + BA1&2	3.270	8.816	**2.036**	**5.496**	120.983	2.235
V102 (101/395)	*MK-solution*	12.025	156.751	6.760	48.926	60.285	0.968
	MK-solution + BA1	6.338	25.252	2.541	7.195	70.963	0.968
	MK-solution + BA1&2	5.149	20.341	**1.935**	**5.497**	84.443	0.968
V103 (71/336)	*MK-solution*	47.928	128.008	6.634	21.691	62.160	1.070
	MK-solution + BA1	71.774	28.160	2.475	6.836	73.301	1.070
	MK-solution + BA1&2	71.068	24.556	**1.870**	**5.259**	84.676	1.070

is the large number of initialization attempts. For example, in sequence V101 which lasts 130 seconds, up to 728 initializations are computed, and 511 of them have passed the *observability and consensus test*. The table shows that the original Martinelli-Kaiser solution obtains average scale errors between 32.9% and 156.7% on these sequences. This error can be reduced until 8.8% to 24.5% applying the two rounds of visual-inertial BA proposed here. More interestingly, applying the novel *observability and consensus tests*, inaccurate initializations are consistently rejected, and the average scale error is reduced to around 5% for all sequences, a very significant improvement over the original method. The ATE error is also drastically reduced after both tests.

Considering the initialization time we see an evident difference between V101, that requires initialization trajectories of 2.2 seconds in average, and V102 and V103 where 1 second is enough. In these two last sequences motion is faster and the *track-length test* is satisfied in less time than in the first sequence, where the quad-copter is flying at low speed.

Regarding the computational cost, the average CPU required to solve the initialization is less than 85ms for sequences V102 and V103, and around 121ms for V101, due to the longer preintegration period. In all cases, the *MK-solution* step takes around 75% of the total initialization CPU time.

In table III we show computational times for our method which uses preintegration with first order bias correction from [8]. Compared with using the original formulation from Martinelli and Kaiser, computing time is reduced by 60%.

In a second experiment, we launch visual-inertial ORB-SLAM [5] and we retrieve the RMSE ATE and the scale error just after the proposed initialization, and after performing full visual-inertial BA at 5 seconds and 10 seconds from the first keyframe timestamp. We can see in table IV that all three sequences converge to scale error smaller than 1% after 10 seconds, confirming that our initialization method is accurate enough to launch visual-inertial SLAM. An example of Visual-Inertial ORBSLAM [5] using our proposed initialization can be found in the accompanying video.

TABLE III: Comparison of running time for *MK Solution+BA1+BA2* repeating IMU integration in each iteration and using preintegration with first order bias correction [8].

V101 EuRoC Dataset			
	CPU time (ms)		
	Mean	Std	Max
Reintegrating each time	301.302	91.974	678.886
Using first order correction	120.983	27.609	214.989

TABLE IV: Results of VI-SLAM using our initialization (average errors on five executions are shown).

V1 EuRoC Dataset						
	After initialization		After BA 5s		After BA 10s	
Seq. Name	RMSE ATE (m)	Scale error (%)	RMSE ATE (m)	Scale error (%)	RMSE ATE (m)	Scale error (%)
V1_01_easy	0.0183	4.99	0.0200	1.85	**0.0170**	**0.84**
V1_02_medium	0.0364	7.38	0.0076	3.67	**0.0162**	**0.71**
V1_03_difficult	0.0043	4.80	0.0129	2.50	**0.0120**	**0.27**

Compared with the initialization method proposed in [5], our method requires trajectories of 1 or 2 seconds instead of 15 seconds, uses less CPU time, and is able to successfully initialize in sequence V103, where the previous method failed.

V. Conclusions

We have proposed a fast joint monocular-inertial initialization method, based on the work of Martinelli [1] and Kaiser et al. [2]. We have adapted it to be more general, allowing incomplete feature tracks, and more computationally efficient using the IMU preintegration method of Forster et al. [8]. Our results show that the original Martinelli-Kaiser technique does not provide a good enough initialization in most practical scenarios, hence we have proposed two visual-inertial BA steps to improve the solution and two novel tests to detect bad initializations. These techniques have proven to be worth it, reducing scale error down to 5% and rejecting bad initializations. Solutions found after those steps is good enough to launch Visual-Inertial ORBSLAM [5] and converge to very accurate maps.

In summary, we have developed a fast method for joint initialization of monocular-inertial SLAM, using trajectories of 1 to 2 seconds, that is much more accurate and robust than the original technique [2], with a maximum computational cost of 215ms.

As future work we would like to investigate the adaptation of the initialization method to the stereo case, taking into account that scale is directly observable from the images. We are also interested in taking profit of gyroscope readings for tracking, even before the initialization has been performed. Finally, we would like to test the initialization performance in in more difficult scenarios with our own acquired sequences.

REFERENCES

[1] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

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

[3] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[5] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[6] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4225–4232.

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

[8] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems*, 2015.

[9] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[11] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," in *Int. Joint. Conf. on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.

[12] R. Szeliski, *Computer vision: algorithms and applications*. Springer Verlag, London, 2011.

[13] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[15] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.