2018 IEEE International Symposium on Mixed and Augmented Reality

Visual-Inertial SLAM Initialization: A General Linear Formulation and a Gravity-Observing Non-Linear Optimization

Javier Domínguez-Conti* Jianfeng Yin† Yacine Alami‡ Javier Civera§

Abstract

The initialization is one of the less reliable pieces of Visual-Inertial SLAM (VI-SLAM) and Odometry (VI-O). The estimation of the initial state (camera poses, IMU states and landmark positions) from the first data readings lacks the accuracy and robustness of other parts of the pipeline, and most algorithms have high failure rates and/or initialization delays up to tens of seconds. Such initialization is critical for AR systems, as the failures and delays of the current approaches can ruin the user experience or mandate impractical guided calibration.

In this paper we address the state initialization problem using a monocular-inertial sensor setup, the most common in AR platforms. Our contributions are 1) a general linear formulation to obtain an initialization seed, and 2) a non-linear optimization scheme, including gravity, to refine the seed. Our experimental results, in a public dataset, show that our approach improves the accuracy and robustness of current VI state initialization schemes.

Index Terms: Visual-Inertial SLAM, Visual-Inertial Initialization, Visual-Inertial Localization, Visual-Inertial Mapping, Sensor Fusion

1 Introduction

Visual and Visual-Inertial SLAM systems are a key component in many Augmented Reality (AR) applications, as they provide real-time estimates of the 6 degrees-of-freedom camera motion and the 3D structure of the scene. Due to several scientific and technological advances in the last two decades, SLAM robustness and accuracy is now sufficient for use in production applications in multiple settings [11, 14, 20]

However, even the most advanced Visual-Inertial SLAM (VI-SLAM) systems have not reached the readiness level required for some other important, demanding applications, and there are still several challenges. Among them, SLAM initialization stands up as one of their weakest parts [1, 3, 7, 27]. In particular, the speed and reliability of SLAM initialization has a direct impact on application reliability and user experience, for mass-scale Augmented Reality (AR), Virtual Reality (VR), and Robotics applications.

The monocular-inertial configuration is of particular interest for SLAM due to the complementary nature of both sensors. The single lens monocular camera, as an exteroceptive sensor, can estimate a visual map and avoid the proprioceptive drift. On the other hand, the inertial sensing provides the metric scale to the scene and camera motion and adds extra short-term robustness. On a more practical side both are small, cheap and low-power sensors, and hence very convenient for AR, VR and Robotics.

In this work we improve the robustness and accuracy of VI-SLAM and VI-O state initialization, with three specific contributions:

- A general linear formulation for the monocular-inertial state initialization. Differently from previous approaches, the feature matches do not have to be common to all frames. We demonstrate that our general formulation outperforms the state-of-the-art algorithms.
- A non-linear optimization that refines the initial state. Differently from other approaches, our approach estimates the gravity direction, which allows to correct initial deviations and improves the geometric accuracy.
- An exhaustive and thorough evaluation of our proposal in a public dataset [2], in order to characterize the achievable performance in the most complete and repeatable manner.

The rest of the paper is organized as follows. Section 2 details the related papers. Section 3 presents our general linear formulation for initialization, and Section 4 our non-linear optimization scheme including the gravity. Section 5 contains our experimental evaluation, and section 6 the conclusions and lines for future work.

2 Related Work

2.1 Visual-Inertial Initialization

In spite of its relevance in practical applications, the initialization of VI-SLAM and VI-O has been largely unaddressed in the literature. To name a few examples, notice how it is not discussed on [6, 13, 23] or how it is only briefly mentioned in [9, 12, 22] but not thoroughly evaluated nor analyzed. When mentioned, most of the works only report the heuristics or assumptions made to set the initial state and that it converged for most or all of their experiments.

It is only recently that VI-SLAM initialization has been addressed and evaluated in a thorough manner. In general, the most recent works on VI-SLAM contain more details on the specific initialization techniques, some of them explicitly acknowledging it as a challenge (e.g., [25, 29]). The recent related work can be broadly grouped into two categories.

2.1.1 Progressive initialization

[21, 24], among others, estimate the different parts of the visual-inertial state following a loosely-coupled scheme. The details vary, but broadly the rotation and translation direction are estimated using only vision. After that, the gyroscope bias can be estimated from the visual rotation and gyroscope measurements. Finally, the gravity, scale, initial velocity and accelerometer bias are extracted using the accelerometer data and previous results.

These approaches inherit all the limitations of pure monocular initialization, as they rely on it in the first step. [21], for example, reports initialization delays up to 10 seconds. Their robustness and accuracy might also be lower because of this reason.

2.1.2 Closed-form initialization

These algorithms [10, 17, 18] determine a closed-form solution for the state using all the visual and inertial data up to a certain time step. Our approach falls into this category. Using all the data at once for initialization has the potential to achieve a better performance. As an example, the initialization delays in [10] are around 2 seconds, much lower than the progressive approaches.

[*] I3A, Universidad de Zaragoza, Spain. jdoco@unizar.es

[†] Geomagical Labs, Inc., 444 Castro Street, Suite 710, Mountain View, CA 94041 USA.jianfeng@geomagical.com

[‡] Geomagical Labs, Inc., 444 Castro Street, Suite 710, Mountain View, CA 94041 USA.yacine@geomagical.com

[§] I3A, Universidad de Zaragoza, Spain. jcivera@unizar.es

978-1-5386-7459-8/18/$31.00 ©2018 IEEE
DOI 10.1109/ISMAR.2018.00027

One of the limitations of these closed-form solutions is that they assume common matches in all the frames. This is difficult to achieve, especially in low-texture environments with imperfect lighting or for aggressive camera motions. Our formulation overcomes this limitation, allowing partial matches in the sequence, that improves the robustness and accuracy.

As an additional limitation, the evaluation of the initialization performance is limited in the literature. We present the first thorough evaluation of these techniques in a public dataset, to facilitate the comparisons.

2.2 Visual-Inertial Non-Linear Optimization

Classic VI-SLAM systems were based on filtering, e.g., [15, 19]. However, parallel tracking and mapping approaches, based on non-linear optimization, have shown a higher accuracy in the most recent comparisons [5, 12].

VI-SLAM and odometry systems based on non-linear optimization (e.g., [4, 6, 12, 16, 21, 24]) do not include the gravity in the state, some of them explicitly stating that they use a seed from initialization. In this work we propose a formulation that includes the gravity in the non-linear optimization of the visual-inertial states. Our experiments demonstrate a reasonable convergence from the initial linear seed.

3 State Initialization

The input to our algorithm are the M initial video frames, and their associated IMU readings, of a visual-inertial sequence. Our aim is to extract an initial seed for the camera motion and the depth of a set of tracked salient visual features. This initial seed will be refined by the non-linear optimization described in Section 4.

3.1 Image Processing

We extract corner features [26] in the first frame and use Kanade-Lucas-Tomasi (KLT) tracking [28] to find their correspondences in the next frames. As the camera moves and new scene areas appear in the image we extract new corner features and track them, to maintain a sufficient number of feature correspondences spreading over the whole image.

Specifically, in the experiments reported, we always extract the most salient features up to a maximum of 200. Some of these corners might not be successfully tracked, and hence the number of tracked features is usually smaller. Every time one third of the image at its top, bottom, left or right parts does not contain tracked corners, we extract new ones. We skip one frame from every two, for efficiency reasons. We did not observe spurious feature tracks in our experiments, so we did not add an outlier rejection scheme. Although we used this particular setting, we observed that the results do not change substantially for reasonable variations of it.

3.2 Initialization

We generalize the visual-inertial constraint proposed in [18], removing the requirement that all features must be matched to a single reference frame. In our approach, then, the use of partial feature correspondences between any pair of frames $\mathscr{I}_j$ and $\mathscr{I}_k$ is allowed, and there is no need for them to be tracked for the whole sequence.

For each tracked point i ($1 \leqslant i \leqslant N_{jk}$) that is seen in *any* two views j and k ($1 \leqslant j < k \leqslant M$), and assuming that the inertial and camera frames are coincident, the following equation holds:

$$-\mathbf{g}_j \frac{t_{jk}^2}{2} - \mathbf{v}_j t_{jk} + \lambda_j^i \boldsymbol{\mu}_j^i - \mathtt{R}_{jk} \lambda_k^i \boldsymbol{\mu}_k^i = \mathbf{s}_{jk} \qquad (1)$$

where

- $\mathscr{C}_j$ and $\mathscr{C}_k$ are the two camera poses corresponding to the images $\mathscr{I}_j$ and $\mathscr{I}_k$, taken at times t_j and t_k

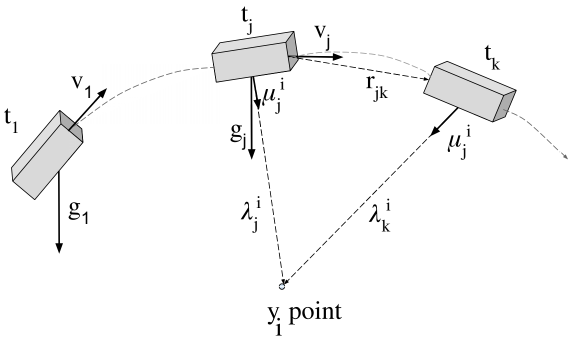


Figure 1: Illustration of the most relevant notation used throughout the paper.

- N_{jk} is the number of point correspondences between the images $\mathscr{I}_j$ and $\mathscr{I}_k$
- M is the total number of frames
- $t_{jk} = t_k - t_j$ is the time interval between time instants t_j and t_k
- $\boldsymbol{\mu}_j^i$ and $\boldsymbol{\mu}_k^i$ are the unit-norm directional vectors from the optical centers of the j^{th} and k^{th} cameras, respectively, to the 3D point feature $\mathbf{y}_i$ in their respective local camera frames
- λ_j^i and λ_k^i are the distances to the point $\mathbf{y}_i$ from the optical center of the j^{th} and k^{th} cameras respectively
- $\mathbf{v}_j$ is the velocity of the j^{th} camera in its local reference frame
- $\mathbf{g}_j$ is the gravity vector in the reference frame of the j^{th} camera, that can be transformed to the reference frame of the 1^{st} camera as $\mathbf{g}_j = \mathtt{R}_{j1}\mathbf{g}_1$
- $\mathtt{R}_{jk}$ is the relative rotation matrix between the camera poses $\mathscr{C}_j$ and $\mathscr{C}_k$
- $\mathbf{s}_{jk}$ is the preintegration of the accelerometer data from t_j to t_k

See Figure 1 for an illustration of the problem and its most relevant magnitudes. The geometric interpretation of Equation 1 is as follows. $\lambda_j^i \boldsymbol{\mu}_j^i$ is the vector from the optical center of camera $\mathscr{C}_j$ to the point feature $\mathbf{y}_i$ in the local frame of camera $\mathscr{C}_j$. $\mathtt{R}_{jk}\lambda_k^i \boldsymbol{\mu}_k^i$ is the vector from the optical center of camera $\mathscr{C}_k$ to $\mathbf{y}_i$ in the local frame of camera $\mathscr{C}_j$. Subtracting both vectors gives us the translation $\mathbf{r}_{jk}$ between cameras j and k

$$\mathbf{r}_{jk} = \lambda_j^i \boldsymbol{\mu}_j^i - \mathtt{R}_{jk} \lambda_k^i \boldsymbol{\mu}_k^i \qquad (2)$$

$\mathbf{r}_{jk}$ can also be extracted from the following kinematic equation [18]

$$\mathbf{r}_{jk} = \mathbf{v}_j t_{jk} + \int_{t_j}^{t_k} (t_k - \tau) \mathtt{R}_{j\tau} \mathbf{a}_\tau d\tau \qquad (3)$$

where $\mathbf{a}_\tau$ is the linear acceleration in the local frame at time τ and $\mathtt{R}_{j\tau}$ the relative rotation between time instants t_j and τ. The acceleration measurements from the IMU, $\tilde{\boldsymbol{a}}_\tau$, are affected by the sensor bias $\mathbf{b}_a$ (assumed constant for small time intervals), the gravity $\mathbf{g}_\tau$ and the noise $\boldsymbol{\eta}_a$ ($\tilde{\boldsymbol{a}}_\tau = \boldsymbol{a}_\tau + \mathbf{b}_a - \mathbf{g}_\tau + \boldsymbol{\eta}_a$). Equation 3 is then transformed as follows.

$$\mathbf{r}_{jk} = \mathbf{v}_j t_{jk} + \mathbf{g}_j \frac{t_{jk}^2}{2} + \mathbf{s}_{jk} \tag{4}$$

where we approximate $\mathbf{s}_{jk} = \int_{t_j}^{t_k} (t_k - \tau)\mathtt{R}_{j\tau}(\tilde{\mathbf{a}}_\tau - \mathbf{b}_a)d\tau$ as the preintegration of the rotated acceleration measurements (we assume $\mathbf{b}_a \ll \tilde{\mathbf{a}}_\tau$). The combination of Equation 2 and Equation 4 gives us Equation 1.

Let now define the world reference frame as the local reference frame of the first camera $\mathscr{C}_1$. And, in order to simplify the notation for relative motions, let skip the subindex 1 in the rest of the paper (*e.g.*, $\mathtt{R}_j \equiv \mathtt{R}_{1j}$).

The rotation $\mathtt{R}_j$ from camera $\mathscr{C}_1$ to $\mathscr{C}_j$ can be obtained from the integration of the angular velocities $\boldsymbol{\omega}_\tau$

$$\mathtt{R}_j = \mathrm{Exp}\left(\int_{t_1}^{t_j} \boldsymbol{\omega}_\tau d\tau\right) \tag{5}$$

The discrete gyroscope readings $\tilde{\boldsymbol{\omega}}_\tau$ are affected from the bias $\mathbf{b}_g$ (assumed constant for small time intervals) and noise $\boldsymbol{\eta}_g$, so $\tilde{\boldsymbol{\omega}}_\tau = \boldsymbol{\omega}_\tau + \mathbf{b}_g + \boldsymbol{\eta}_g$. The above integral can be approximated by the product

$$\mathtt{R}_j \approx \prod_{n=1}^{j} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_n - \mathbf{b}_g\right)\Delta t_n\right) \tag{6}$$

The velocity $\mathbf{v}_j$ of the j^{th} camera can be related to the velocity of the first camera $\mathbf{v}_1$ by integrating the linear acceleration:

$$\mathbf{v}_j = \mathbf{v}_1 + \int_{t_1}^{t_j} \mathtt{R}_\tau \mathbf{a}_\tau d\tau \tag{7}$$

We can rewrite the velocity $\mathbf{v}_j$ as a function of the velocity $\mathbf{v}_1$ and the gravity $\mathbf{g}_1$ (both in the world frame) by adding the accelerometer model and assuming constant acceleration within the time interval Δt_n

$$\mathbf{v}_j \approx \mathtt{R}_j^\top (\mathbf{v}_1 + t_{1j}\mathbf{g}_1 + \sum_{n=1}^{j} (\mathtt{R}_n(\tilde{\mathbf{a}}_n - \mathbf{b}_a)\Delta t_n)) \tag{8}$$

In the practical case of the inertial unit and the camera being related by a transformation $\mathtt{T}_{IC} = (\mathtt{R}_{IC}, \mathbf{p}_{IC})$, and referring the point directions $\boldsymbol{\mu}_j^i$ and $\boldsymbol{\mu}_k^i$ to the reference frame of the inertial sensor, Equation 1 is modified as follows

$$\begin{aligned} -\mathtt{R}_j^\top \mathbf{g}_1 \frac{t_{jk}^2}{2} - \mathbf{v}_j t_{jk} + \mathtt{R}_{IC}\lambda_j^i \boldsymbol{\mu}_j^i - \mathtt{R}_{jk}\mathtt{R}_{IC}\lambda_k^i \boldsymbol{\mu}_k^i = \\ = \mathbf{s}_{jk} + \mathtt{R}_{jk}\mathbf{p}_{IC} - \mathbf{p}_{IC} \end{aligned} \tag{9}$$

By combining Equation 8 and Equation 9 we obtain

$$\begin{aligned} -\mathtt{R}_j^\top \mathbf{g}_1 t_{jk}(t_{1j} + \frac{t_{jk}}{2}) - \mathtt{R}_j^\top \mathbf{v}_1 t_{jk} + \\ + \mathtt{R}_{IC}\lambda_j^i \boldsymbol{\mu}_j^i - \mathtt{R}_{jk}\mathtt{R}_{IC}\lambda_k^i \boldsymbol{\mu}_k^i = \\ = \mathbf{s}_{jk} + \mathtt{R}_{jk}\mathbf{p}_{IC} - \mathbf{p}_{IC} + \mathtt{R}_j^\top t_{jk} \sum_{n=1}^{j} (\mathtt{R}_n(\mathbf{a}_n - \mathbf{b}_a)\Delta t_n) \end{aligned} \tag{10}$$

Equation 10 gives a linear constraint for the unknowns $\mathbf{g}_1$, $\mathbf{v}_1$, λ_j^i and λ_k^i. Grouping the constraints for all the points correspondences and all the image pairs lead to a linear system of equations of the form

$$\mathtt{A}\mathbf{X} = \mathbf{D} \tag{11}$$

that can be decomposed as follows

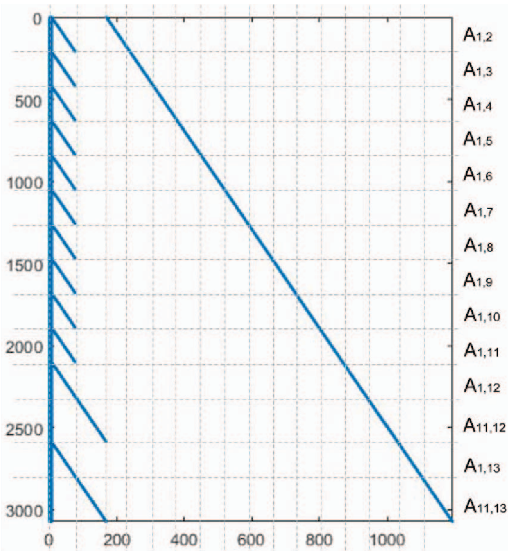


Figure 2: Illustrative example of the structure of the matrix $\mathtt{A}$, for a sequence of 13 frames and feature extraction at frames #1 and #11. Non-zero elements are displayed in blue. For each submatrix $\mathtt{A}_{jk}$ the left diagonal corresponds to $\boldsymbol{\mu}_j^i$ and the right one to $\boldsymbol{\mu}_k^i$.

$$\begin{pmatrix} \mathtt{A}_{12} \\ \vdots \\ \mathtt{A}_{jk} \\ \vdots \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{v}_1 \\ \boldsymbol{\Lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_{12} \\ \vdots \\ \mathbf{d}_{jk} \\ \vdots \end{pmatrix} \tag{12}$$

where $\mathbf{g}_1$ and $\mathbf{v}_1$ are, respectively, the gravity and the velocity of the first camera in its local reference frame and $\boldsymbol{\Lambda} = \begin{pmatrix}\lambda_1^1 \ldots \lambda_j^i \ldots \lambda_k^i \ldots\end{pmatrix}^\top$ the vector containing the distances from viewpoints $\{1, \ldots, j, \ldots, k, \ldots, M\}$ to features $\{1, \ldots, i, \ldots, N\}$. $\mathbf{d}_{jk}$ is as follows

$$\mathbf{d}_{jk} = \begin{pmatrix} \mathbf{s}_{jk} + \mathtt{R}_{jk}\mathbf{p}_{IC} - \mathbf{p}_{IC} + \mathtt{R}_j^\top t_{jk} \sum_{n=1}^{j} (\mathtt{R}_n(\mathbf{a}_n - \mathbf{b}_a)\Delta t_n) \\ \vdots \\ \mathbf{s}_{jk} + \mathtt{R}_{jk}\mathbf{p}_{IC} - \mathbf{p}_{IC} + \mathtt{R}_j^\top t_{jk} \sum_{n=1}^{j} (\mathtt{R}_n(\mathbf{a}_n - \mathbf{b}_a)\Delta t_n) \end{pmatrix} \tag{13}$$

Each sub-matrix $\mathtt{A}_{jk}$ refers to the visual-inertial constraints between t_j and t_k, and its specific structure is as follows

$$\mathtt{A}_{jk} = \left(\begin{array}{c|c|c|c|c|c|c|c|c} \mathtt{T}_{jk}^g & \mathtt{T}_{jk}^v & \ldots & \mathtt{R}_{IC}\mu_j^1 & 0 & \ldots & -\mathtt{R}_{jk}\mathtt{R}_{IC}\mu_k^1 & 0 & 0 \\ \mathtt{T}_{jk}^g & \mathtt{T}_{jk}^v & \ldots & 0 & \mathtt{R}_{IC}\mu_j^2 & \ldots & 0 & -\mathtt{R}_{jk}\mathtt{R}_{IC}\mu_k^2 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ \mathtt{T}_{jk}^g & \mathtt{T}_{jk}^v & \ldots & 0 & 0 & \ldots & \ldots & \ldots & \ldots \end{array}\right) \tag{14}$$

where $\mathtt{T}_{jk}^g = -\mathtt{R}_j^\top t_{jk}(t_{1j} + \frac{t_{jk}}{2})$ and $\mathtt{T}_{jk}^v = -\mathtt{R}_j^\top t_{jk}$. Note that, in the case $j = 1$ (no feature addition), our formulation is equivalent to the one in [10].

Figure 2 illustrates the sparsity pattern of the matrix $\mathtt{A}$ with a particular example of our experiments.

3.3 Gyroscope and Accelerometer Bias Estimation

The IMU preintegration in Equation 6 and Equation 8, among others, involves the accelerometer and gyroscope biases $\mathbf{b}_a$ and $\mathbf{b}_g$. These

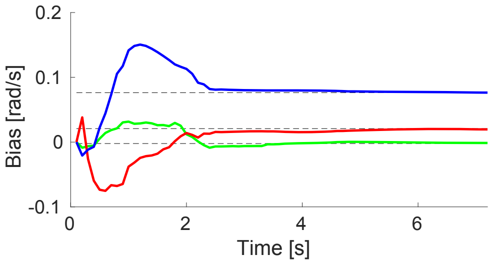


Figure 3: Estimation of the gyroscope bias by minimizing the cost in Equation 15. The ground truth bias is $\mathbf{b}_g = [-0.00222, 0.02082, 0.07632]$ rad/s.

biases might not be known at the initialization time. We adopt the same approach than [10], with minor modifications, to estimate their values.

As [10], we observed that the accelerometer bias does not have a significant influence on the results of the closed-form initialization. We then set $\mathbf{b}_a = [0,0,0]^\top$.

To estimate the gyroscope bias we find the value that minimizes the residual of the following cost function:

$$\hat{\mathbf{b}}_g = \underset{\mathbf{b}_g}{\arg\min} \|\mathtt{A}\mathbf{X} - \mathbf{D}\|^2 \tag{15}$$

We take as initial seed $\mathbf{b}_g = [0,0,0]^\top$, since the real value of the bias is usually very small. Figure 3 shows the bias estimation results for one of our experiments. Similarly to [10], we observed that the gyroscope bias converges to an accurate estimation in $2-3$ seconds.

It is important to take into account that the bias can diverge as the integration time increases. [10] proposes a formulation to avoid this divergence by using a priori knowledge of the gyroscope bias. Since we assume that there is no knowledge of the real bias at any time, we constraint our initialization sequences to be between 3 and 5.5 seconds. In our experiments the gyroscope bias values were accurate for these sequence lengths.

3.4 Efficiency Considerations

Each submatrix $\mathtt{A}_{jk}$ (Equation 14), corresponding to a pair of different views, contains $3N_{jk}$ equations and $3+3+2N_{jk}$ unknowns (3 for the gravity vector $\mathbf{g}_1$, 3 for the velocity $\mathbf{v}_1$ and the two distances λ_j^i and λ_k^i per correspondence). For the full system defined in Equation 11 we can sum over all the image pairs P and we have $3\sum_P N_{jk}$ equations and $3+3+2\sum_P N_{jk}$ unknowns.

Notice in Figure 2 that the matrix $\mathtt{A}$ is sparse. Every row has up to 8 non-zero values depending on the submatrix. The smallest sparsity ratio (zero elements over the total) is $1-\sum_P \frac{8}{N_{jk}}$, which is close to 1, as typical values of pairwise matches N_{jk} are in the hundreds. Using standard linear algebra packages (like Eigen [8]) the system can be solved very efficiently.

4 Non-Linear Optimization

4.1 State Definition

Each camera pose $\mathscr{C}_j$ is defined by the tuple $\left(\mathtt{T}_j, \boldsymbol{\delta\phi}_j, \boldsymbol{\delta}\mathbf{p}_j\right)$; where $\mathtt{T}_j = \left(\mathtt{R}_j, \mathbf{p}_j\right)$ belongs to the Special Euclidean Group SE(3) and $\left(\boldsymbol{\delta\phi}_j, \boldsymbol{\delta}\mathbf{p}_j\right) \in \mathbb{R}^6$ to its associated tangent space $\mathfrak{se}(3)$ at $\mathtt{T}_j$. We use the following retraction, taken from [6], to obtain the camera pose

$$\mathscr{R}_{\mathtt{T}}(\boldsymbol{\delta\phi}_j, \boldsymbol{\delta}\mathbf{p}_j) = (\mathtt{R}_j \operatorname{Exp}(\boldsymbol{\delta\phi}_j^\wedge),\ \mathbf{p}_j + \mathtt{R}_j \boldsymbol{\delta}\mathbf{p}_j) \tag{16}$$

where Exp(.) is the exponential map, and the *hat* operator $(.)^\wedge$ is used to convert $\boldsymbol{\delta\phi}_j = \left(\delta\phi_{j,1}, \delta\phi_{j,2}, \delta\phi_{j,3}\right)^\top \in \mathbb{R}^3$ to the space of skew symmetric matrices of the Lie algebra. Specifically, using the following three basis vectors

$$\mathtt{G}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \mathtt{G}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \mathtt{G}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{17}$$

the $\operatorname{Exp}(\boldsymbol{\delta\phi}_j^\wedge)$ operation results

$$\operatorname{Exp}(\boldsymbol{\delta\phi}_j^\wedge) = \delta\phi_{j,1}\mathtt{G}_1 + \delta\phi_{j,2}\mathtt{G}_2 + \delta\phi_{j,3}\mathtt{G}_3 \tag{18}$$

As detailed in [6], this representation is convenient in order to frame the optimization problem in the locally Euclidean tangent space, using a minimal parameterization.

We model the gravity with the tuple $\left(\mathtt{R}_g, \boldsymbol{\delta\phi}_g, g\right)$. g is the gravity modulus. $\left(\mathtt{R}_g, \boldsymbol{\delta\phi}_g\right)$ represents the rotation between the global reference frame, defined as the local frame of the first camera $\mathscr{C}_1$, and the gravity reference frame $\mathscr{G}$. If we define the gravity vector in its local frame $\mathscr{G}$ as $\mathbf{g}_{\mathscr{G}} = (0,0,-g)^\top$, it follows that

$$\mathbf{g}_1 = \mathtt{R}_{\mathscr{G}}(\mathtt{R}_g, \boldsymbol{\delta\phi}_g)\mathbf{g}_{\mathscr{G}} \tag{19}$$

where the relative rotation between the gravity and the world reference frame is

$$\mathtt{R}_{\mathscr{G}}(\mathtt{R}_g, \boldsymbol{\delta\phi}_g) = \mathtt{R}_g \operatorname{Exp}(\boldsymbol{\delta\phi}_g^\wedge), \boldsymbol{\delta\phi}_g = \begin{pmatrix} \boldsymbol{\delta\phi}_{go} \\ 0 \end{pmatrix} \tag{20}$$

Notice that the third component of $\delta\phi_g$ is 0, as it corresponds to rotations around the z-axis in the gravity frame, that are not observable. The other two angles $\boldsymbol{\delta\phi}_{go} = \left(\delta\phi_{go,1}, \delta\phi_{go,2}\right)^\top$ correspond to the observable part and are the ones that are estimated.

The state vector to optimize is then as follows

$$\mathbf{x} = \begin{pmatrix} \boldsymbol{\delta\phi}_{go}^\top & \boldsymbol{\delta}\mathbf{b}_g^\top & \boldsymbol{\delta}\mathbf{b}_a^\top & \mathbf{c}_1^\top & \dots & \mathbf{c}_M^\top & \mathbf{y}_1^\top & \dots & \mathbf{y}_N^\top \end{pmatrix}^\top \tag{21}$$

where $\mathbf{c}_j, \forall j > 1$ contains the camera-inertial state at time j, composed of its rotation and translation increments in the tangent space $\boldsymbol{\delta\phi}_j$ and $\boldsymbol{\delta}\mathbf{p}_j$, and its linear velocity $\mathbf{v}_j$

$$\mathbf{c}_j = \left(\boldsymbol{\delta\phi}_j^\top, \boldsymbol{\delta}\mathbf{p}_j^\top, \mathbf{v}_j^\top\right)^\top \tag{22}$$

and $\mathbf{c}_1 = \mathbf{v}_1$ only contains the initial velocity.

4.2 Initial seed

The visual-inertial state defined in Section 4.1 can be extracted from the solution of the linear system in Section 3 as follows.

Let $\boldsymbol{\mu}_{g1} = \frac{\mathbf{g}_1}{g}$ be the gravity direction in the reference frame of the first camera. The columns of $\mathtt{R}_g$ are the unit basis vectors of the gravity reference frame $\mathscr{G}$ in the reference frame of the first camera. $\boldsymbol{\mu}_{g1}$ is hence the third column of $\mathtt{R}_g$, as it models the unit vector pointing in the gravity direction in the world frame

$$\mathtt{R}_g = \begin{pmatrix} \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 & \boldsymbol{\mu}_{g1} \end{pmatrix} \tag{23}$$

From the constraint $\boldsymbol{\mu}_1 \perp \boldsymbol{\mu}_{g1}$, and choosing for convenience $\boldsymbol{\mu}_1 = \left(\mu_{1,x}, 0, \mu_{1,z}\right)^\top$ –a rotation around $\boldsymbol{\mu}_{g1}$ does not change the gravity direction– we can obtain $\mu_{1,x}$ and $\mu_{1,z}$

$$\mu_{1,z} = \sqrt{\frac{1}{1 + \left(\frac{\mu_{g1,z}}{\mu_{g1,x}}\right)^2}};\ \ \mu_{1,x} = -\mu_{1,z}\frac{\mu_{g1,z}}{\mu_{g1,x}} \tag{24}$$

We set $\boldsymbol{\mu}_2$ as the cross product of $\boldsymbol{\mu}_{g1}$ and $\boldsymbol{\mu}_1$.

The rotation $\mathtt{R}_j$ for the monocular-inertial sensor at time t_j can be initialized from the preintegration of the gyroscope. A seed for the translation $\mathbf{p}_j$ can be extracted by averaging the subtraction of common point vectors. For example, if there are N_{1j} common points between frame 1 and j

$$\mathbf{p}_j = \frac{1}{N_{1j}} \sum_{i=1}^{N_{1j}} \left(\lambda_1^i \boldsymbol{\mu}_1^i - \mathtt{R}_j \lambda_j^i \boldsymbol{\mu}_j^i \right) \tag{25}$$

Each point $\mathbf{y}_i$ is represented by its Euclidean coordinates, that can be obtained in the reference frame of the j^{th} camera as $\mathbf{y}_i = \lambda_j^i \boldsymbol{\mu}_j^i$. As the linear initialization is affected by the IMU drift, the seed for the points has a large variation depending on the camera considered. We observed that averaging the point positions leads to a better convergence, hence

$$\mathbf{y}_i = \sum_{j=1}^{M} \mathtt{R}_j \lambda_j^i \boldsymbol{\mu}_j^i \tag{26}$$

4.3 Residuals

We minimize a joint residual vector $\mathbf{r}$ composed of inertial ($\mathbf{r}_{\Delta\mathtt{R}_{j,j+1}}$, $\mathbf{r}_{\Delta\mathrm{v}_{j,j+1}}$, $\mathbf{r}_{\Delta\mathrm{p}_{j,j+1}}$) and visual ($\mathbf{r}_{\Delta\mathbf{z}_{i,k}}$) terms.

$$\mathbf{r} = \left(\mathbf{r}_{\Delta\mathtt{R}_{1,2}}^\top \; \mathbf{r}_{\Delta\mathbf{v}_{1,2}}^\top \; \mathbf{r}_{\Delta\mathbf{p}_{1,2}}^\top \; \cdots \; \mathbf{r}_{\Delta\mathtt{R}_{M-1,M}}^\top \; \mathbf{r}_{\Delta\mathbf{v}_{M-1,M}}^\top \; \mathbf{r}_{\Delta\mathbf{p}_{M-1,M}}^\top \; \mathbf{r}_{\Delta\mathbf{z}_{1,1}}^\top \; \cdots \; \mathbf{r}_{\Delta\mathbf{z}_{i,k}}^\top \; \cdots \right)^\top \tag{27}$$

The inertial residuals between consecutive frames j and $j+1$ ($j \leqslant M-1$) are taken from [6]

$$\begin{aligned}
\mathbf{r}_{\Delta\mathtt{R}_{j,j+1}} =& \mathrm{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{j,j+1} \mathrm{Exp}\left(\frac{\partial \Delta\tilde{\mathtt{R}}_{j,j+1}}{\partial \mathbf{b}_g} \delta\mathbf{b}_g \right) \right)^\top \right. \\
& \left. \mathtt{R}_j^\top \mathtt{R}_{j+1} \right) && (28) \\
\mathbf{r}_{\Delta\mathbf{v}_{j,j+1}} =& \mathtt{R}_j^\top \left(\mathbf{v}_{j+1} - \mathbf{v}_j - \mathbf{g}\Delta t \right) - \\
& - \left(\Delta\tilde{\mathbf{v}}_{j,j+1} + \frac{\Delta\tilde{\mathbf{v}}_{j,j+1}}{\partial \mathbf{b}_g} \delta\mathbf{b}_g + \frac{\Delta\tilde{\mathbf{v}}_{j,j+1}}{\partial \mathbf{b}_a} \delta\mathbf{b}_a \right) && (29) \\
\mathbf{r}_{\Delta\mathbf{p}_{j,j+1}} =& \mathtt{R}_j^\top \left(\mathbf{p}_{j+1} - \mathbf{p}_j - \mathbf{v}_j \Delta t - \frac{1}{2}\mathbf{g}\Delta t^2 \right) - \\
& - \left(\Delta\tilde{\mathbf{p}}_{j,j+1} + \frac{\Delta\tilde{\mathbf{p}}_{j,j+1}}{\partial \mathbf{b}_g} \delta\mathbf{b}_g + \frac{\Delta\tilde{\mathbf{p}}_{j,j+1}}{\partial \mathbf{b}_a} \delta\mathbf{b}_a \right) && (30)
\end{aligned}$$

where $\Delta\tilde{\mathtt{R}}_{j,j+1}$, $\Delta\tilde{\mathbf{v}}_{j,j+1}$ and $\Delta\tilde{\mathbf{p}}_{j,j+1}$ are the preintegrated inertial measurements between frames j and $j+1$. As proposed in [6], we approximate the effect of the biases by its first order propagation. If the bias increments are large, we preintegrate again to avoid the linearization errors. As we are considering a small number of frames, the overhead is negligible.

The visual reprojection error $\mathbf{r}_{\Delta\mathbf{z}_{i,k}}$ for a point $\mathbf{y}_i$ in a camera $\mathscr{C}_k$ is

$$\mathbf{r}_{\Delta\mathbf{z}_{i,k}} = \mathbf{z}_{i,k} - \pi\left(\mathbf{y}_i, \mathtt{R}_k, \mathbf{p}_k\right) \tag{31}$$

where $\pi\left(\mathbf{y}_j, \mathtt{R}_k, \mathbf{p}_k\right)$ is the pinhole projection model and $\mathbf{z}_{i,k}$ the tracked image feature corresponding to the point $\mathbf{y}_i$ in the image taken by camera $\mathscr{C}_k$.

We use Gauss-Newton, starting from the linear seed detailed in Section 4.2. The Jacobians of the inertial residuals –except for the ones referred to gravity– can be found in [6].

	MH_01_easy	MH_02_easy	MH_03_medium
Init	900	800	400
End	3600	2900	2500
	MH_04_difficult	MH_05_difficult	V1_01_easy
Init	500	500	100
End	1900	2100	2700
	V1_02_medium	V1_03_difficult	V2_01_easy
Init	100	200	100
End	1600	2000	2100
	V2_02_medium	V2_03_difficult	
Init	100	100	
End	2200	1800	

Table 1: Initial and final frames used in our evaluation results for each EuRoC sequence.

5 Experimental Results

5.1 Experimental Setup

For our experimental analysis, we have used the publicly available EuRoC MAV dataset[1] [2]. The dataset contains 11 stereo-inertial streams, recorded with an aerial vehicle, and with accurate ground truth for position and orientation.

The sequences image 2 scenes (*Vicon Room* and *Machine Hall*) and are classified into *Easy*, *Medium* and *Difficult* according to the illumination conditions and aggressiveness of the motion. The different scene types and challenge graduation will be very useful to analyze our contribution in different conditions.

We used the left camera of the stereo and the inertial data for our evaluation. We only considered the sequence fragments where the aerial vehicle is moving –we detail the initial and final frames of our evaluation in Table 1. Within such fragments we evaluate our initialization algorithm every 20 frames, resulting in a corpus of 1,080 test cases containing a wide variety of motions and degrees of image texture.

5.2 General Initialization Formulation

First of all, we will illustrate qualitatively the most relevant aspects of our proposal. Figure 4 shows, with an example of our experiments, the accuracy of the initialization depending on the number of frames used. We can observe three regions. In the first one, for a low number of frames, the error is high due to small camera translation (low parallax) and low IMU excitation. In the second region, as the camera moves and more frames are added, the initialization error reduces. Finally, in the third region, the IMU drift accumulates and the error grows. Notice that the minimum-error region depends on the motion and scene depth, and hence it is difficult to predict.

Figure 4 shows how our algorithm (denoted as *Adding features*, in blue and green); extends the low-error initialization region compared to the baseline [10]. This is highly convenient: As it is difficult to set in advance an optimal number of frames, a larger low-error region increases the chances a successful initialization.

Secondly, Figure 5 shows several frames and tracked features in one of our experiments. None of the features extracted in the first frame (in red in Figure 5a) can be tracked until the last frame (Figure 5d) due to the fast camera motion. Our algorithm is able to add new features to track twice (in blue in Figure 5b and Figure 5c). The robustness is improved, as we can relate a higher number of frames. And also the accuracy, due to a better spreading of the tracking features throughout all the views.

Table 2 shows the quantitative results. Our metrics are the number of initialization failures over the total number of cases, and the

[1] http://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets

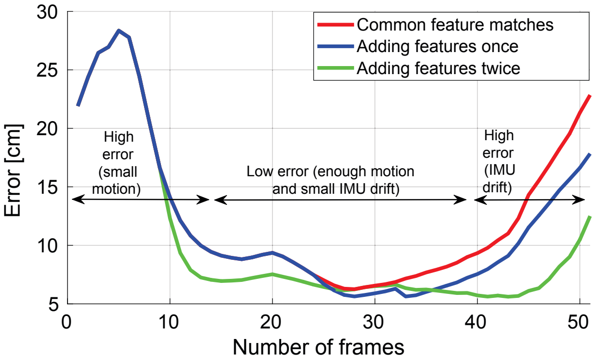
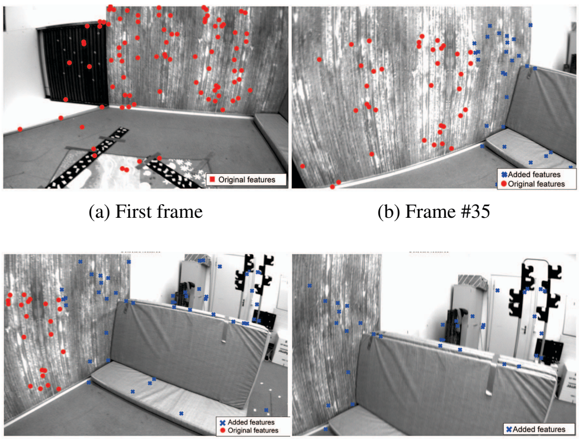


Figure 4: Example of error dependence with respect to the number of frames and added features. Notice the high error areas, at the beginning due to small motion, and at the end due to the drift. Notice also the improvement obtained by adding features.



(a) First frame

(b) Frame #35

(c) Frame #78

(d) Frame #104

Figure 5: Feature addition example. (a) shows, in red, the features extracted in the first frame. In (b), after 35 frames, new scene areas appear in the right part of the image, and we initialize new features there (in blue). In (c) we add features again in the right image part. Notice how after 104 frames none of the original (red) features is seen.

camera position error. We do not evaluate the rotation error, as it mainly corresponds to the preintegration of the gyroscope data. We define failure as loss of tracked features for a particular number of frames. For the position error, we use the Normalized Root Mean Square Error (*NRMSE*):

$$NRMSE = \frac{RMSE}{\sum_{i=1}^{M-1} |\mathbf{p}_{i+1} - \mathbf{p}_i|} \tag{32}$$

where *RMSE* is the standard root mean square error, and the denominator $\sum_{i=1}^{M-1} |\mathbf{p}_{i+1} - \mathbf{p}_i|$ accounts for the trajectory length. We use the normalized version to compare errors of trajectories with different lengths in a fair manner. For a more complete analysis, the table separates the results of the *Easy*, *Medium* and *Difficult* sequences, and also the *Vicon Room* and *Machine Hall*. For each sequence, we also evaluate several sequence lengths.

Observe first the drastic reduction of failures of our algorithm. [10] has a small failure rate for short sequences, but it grows very rapidly with the number of frames and sequence difficulty, to unacceptable numbers. The failure rate for our proposal is almost zero in all cases.

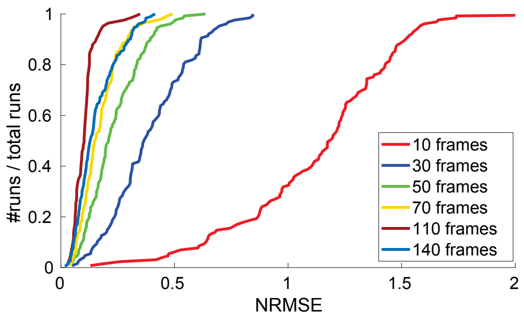


Figure 6: Cumulative error for different sequence lengths. In this dataset the error decreases from 10 frames until 110 frames, and increases after that.

Furthermore, notice that the *NRMSE* is also unacceptably high for small sequence lengths, both in the baseline [10] and our proposal. For example, for 10 frames in *Vicon Room* is over 100%, meaning that the error is similar to the trajectory length. The errors are acceptable only after $2-3$ seconds (this is also reported in [10]). It is precisely in this regime that our algorithm shows the most valuable improvement, as it is able to reduce the high failure rates of [10]. In any case, we improve the *NRMSE* for all the cases due to a better spreading of the salient features.

The dependence of the *NRMSE* with the number of frames can be better appreciated in Figure 6, that shows a histogram of cumulative errors for several sequence lengths. Notice how the error greatly reduces from 10 to 30 frames, keeps reducing at a slower rate until 110 frames, and grows again after that due to the IMU drift. The specific values might vary for other sensors and sequences, but we believe the qualitative aspects of the graph should be generalizable. As said before: Large errors for small sequences correspond to low parallax and low IMU excitation; and large errors for large sequences to IMU drift.

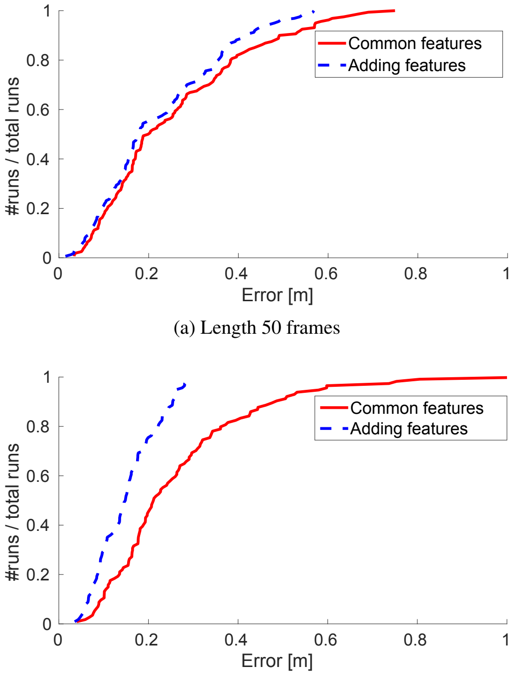
From Figure 6 and Table 2, we conclude that the desirable operation modes for initialization are two. The first, one, initializing as soon as the error is reasonably low ($\sim$ 2.5 seconds) for time-critical applications. And the second one, initializing when the error is smaller if we can wait ($\sim$ 5.5 seconds). Figure 7 details the cumulative error for these two relevant cases. Our algorithm outperforms [10], more noticeably in the second case. In these two plots, differently from the previous ones, we report the non-normalized *RMSE* errors, to give a better intuition of the magnitudes.

Finally, we would like to remark the differences between the two scenes and the three difficulty levels in Table 2. Notice that our *NRMSE* improvement is more noticeable in the *Vicon Room* sequences. The reason is that the *Vicon Room* is a smaller place than the *Machine Hall* (See Figure 8 for sample images of both). For similar motions, then, there are higher chances that the tracked features go out of the camera field of view, which is the case that our algorithm is able to address.

It is also worth remarking the different NRMSE magnitudes between the *Machine Hall* and *Vicon Room* sequences. The reason is the distance of the tracked points to the camera: in the *Vicon Room*, the maximum scene depth is around 6 meters, while in the *Machine Hall* the maximum depth can be up to 20 meters (again, see Figure 8 for sample images). The parallax in the *Vicon Room* is higher than in the *Machine Hall* and hence the estimated depths λ_j^i are more

			Vicon Room				Machine Hall			
	Length		Failures (All runs)		NRMSE (%) (Successful runs)		Failures (All runs)		NRMSE (%) (Successful runs)	
	Frames	Sec.	[10]	Ours	[10]	Ours	[10]	Ours	[10]	Ours
Easy sequences	10	0.5	0 (0%)	0 (0%)	118.8	118.8	0 (0%)	0 (0%)	120.8	120.8
	30	1.5	5 (2.2%)	**0 (0%)**	36.7	**35.3**	0 (0%)	0 (0%)	48.0	48.0
	50	2.5	9 (3.9%)	**0 (0%)**	21.0	**17.8**	1 (0.4%)	**0 (0%)**	30.1	**30.0**
	60	3	14 (6.0%)	**0 (0%)**	17.5	**16.4**	1 (0.4%)	**0 (0%)**	20.6	**20.0**
	70	3.5	24 (10.3%)	**0 (0%)**	15.0	**12.9**	1 (0.4%)	**0 (0%)**	22.1	**21.9**
	90	4.5	40 (17.2%)	**0 (0%)**	11.5	**9.4**	6 (2.5%)	**0 (0%)**	20.1	**18.7**
	110	5.5	82 (35.3%)	**0 (0%)**	10.2	**6.8**	14 (5.9%)	**0 (0%)**	18.4	**18.1**
Medium sequences	10	0.5	9 (5.1%)	**0 (0%)**	100.7	**99.6**	7 (6.6%)	**0 (0%)**	131.2	131.2
	30	1.5	43 (24.3%)	**0 (0%)**	30.1	**28.2**	8 (7.5%)	**0 (0%)**	51.2	**51.1**
	50	2.5	87 (47.8%)	**0 (0%)**	17.6	**16.8**	27 (25.5%)	**0 (0%)**	28.4	**28.1**
	60	3	93 (51.1%)	**0 (0%)**	14.6	**11.0**	36 (34.0%)	**0 (0%)**	25.2	**25.0**
	70	3.5	108 (59.4%)	**0 (0%)**	13.7	**11.5**	37 (34.9%)	**0 (0%)**	22.7	**22.4**
	90	4.5	141 (77.5%)	**0 (0%)**	8.0	**7.1**	40 (37.7%)	**0 (0%)**	12.9	**12.6**
	110	5.5	157 (88.7%)	**1 (0.6%)**	5.2	**3.8**	54 (50.1%)	**1 (0.9%)**	8.7	**7.5**
Difficult sequences	10	0.5	79 (43.4%)	**0 (0%)**	98.8	**98.6**	0 (0%)	0 (0%)	142.3	142.3
	30	1.5	132 (72.5%)	**0 (0%)**	32.7	**32.6**	2 (1.6%)	**0 (0%)**	71.0	**70.7**
	50	2.5	150 (82.4%)	**0 (0%)**	17.3	**16.7**	5 (3.3%)	**0 (0%)**	52.4	**52.3**
	60	3	162 (89.0%)	**1 (0.5%)**	15.0	**12.9**	19 (12.5%)	**0 (0%)**	47.2	**45.7**
	70	3.5	165 (90.7%)	**1 (0.5%)**	10.5	**10.4**	22 (14.5%)	**0 (0%)**	41.3	**40.6**
	90	4.5	172 (94.5%)	**2 (1.1%)**	5.9	**5.7**	51 (33.6%)	**0 (0%)**	35.1	**33.1**
	110	5.5	177 (97.3%)	**3 (1.6%)**	3.7	**3.5**	71 (46.7%)	**0 (0%)**	28.5	**27.9**

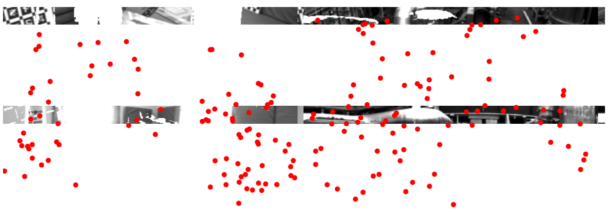
Table 2: Number of failures and NRMSE(%) of the initialization proposed in [10] against ours trying to initialize every 20 frames on the Vicon Room and Machine Hall sequences of the EuRoC dataset. Notice that the NRMSE is evaluated only in the experiments successfully ran on the two compared algorithms.



(a) Length 50 frames

(b) Length 110 frames

Figure 7: Effect of the addition of features on the pose error. The error decrease is more noticeable in longer sequences due to the feature addition of our proposal.



(a) Vicon Room Frame

(b) Machine Hall Frame

Figure 8: Sample images (with tracked features) from the EuRoC dataset.

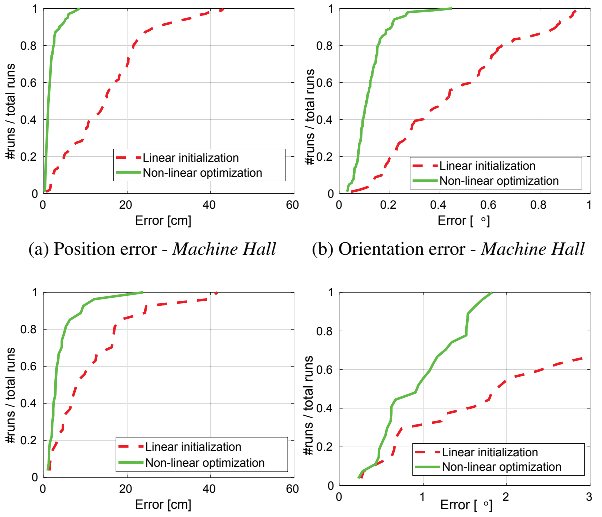
accurate in the *Vicon Room* than in the *Machine Hall* sequences.

Notice how the failure ratio of [10] is higher as the difficulty increases. The reason is that *Medium* and, particularly, *Difficult* sequences contain aggresive camera motions, and salient points can only be tracked for a small number of frames. Our formulation is able to address that issue successfully by adding new features.

5.3 Gravity-observing non-linear optimization

We have evaluated the performance of our proposed non-linear optimization. We take as initial seed the result of the closed-form solution. The sequence length in these experiments is 3 seconds, considered approximately a minimum length to obtain a reasonable accuracy (see Table 2).

Figure 9 shows the reduction or the error in the camera pose after the proposed non-linear optimization in the *Machine Hall* and *Vicon Room* sequences. Notice that a non-linear visual-inertial bundle adjustment always reduces the error with respect to the linear initialization. The improvement is large for the biggest initialization



(a) Position error - *Machine Hall* (b) Orientation error - *Machine Hall*

(c) Position error - *Vicon Room* (d) Orientation error - *Vicon Room*

Figure 9: Cumulative error graphs before and after the non-linear optimization, in red and green respectively. The results are divided into position and orientation error in the *Machine Hall* sequences ((a) and (b) respectively) and position and orientation error in the *Vicon Room* sequences ((c) and (d) respectively).

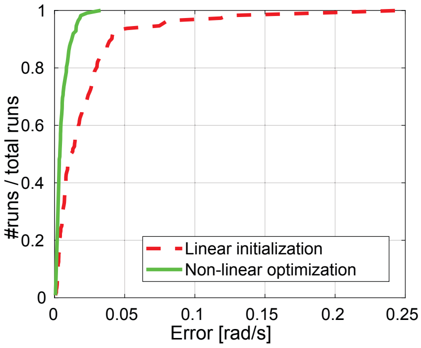
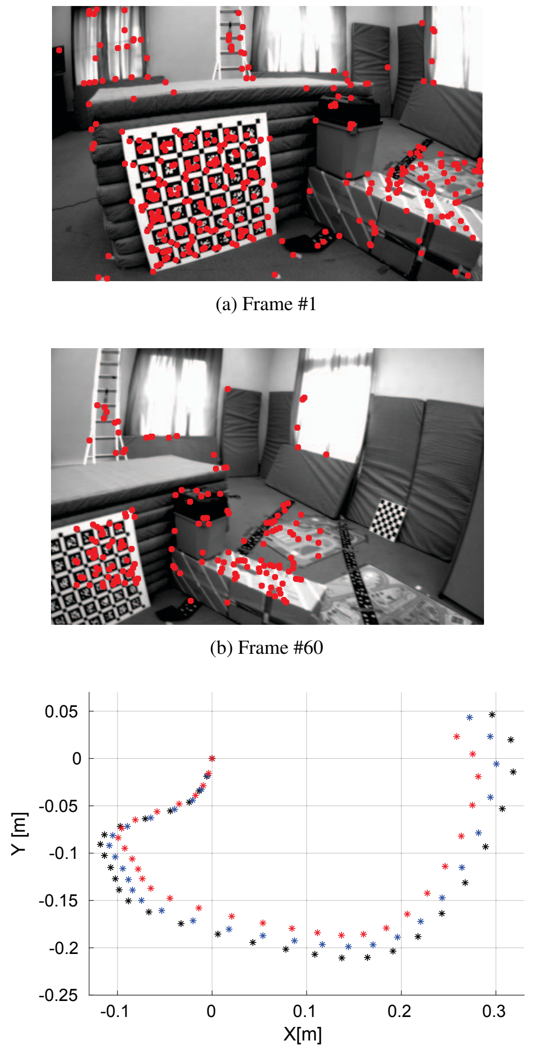


Figure 10: Cumulative error for the estimated gyroscope bias, for the linear seed (dotted red) and (solid green) non-linear optimization.

errors (for the worst seed of the *Machine Hall*, from 108 cm to 22 cm), and it is smaller as the initialization seed is better.

Figure 10 shows the cumulative error for the estimation of the gyroscope bias. As before, this estimation also improves after the non-linear optimization. The errors for the accelerometer bias, that we do not show, are bigger due to two main reasons. First, the sensitivity of the errors to the accelerometer biases is small. And second, for this short sequences, the accelerometer bias is highly coupled with the gravity.

Finally, Figure 11 shows, with illustrative purposes: The first and last frames of one of our experiments (with the tracked features), and the ground truth (in black) and estimated camera trajectories. Notice how the solution of the linear seed (in red) is already rather accurate, and it is further refined by the non-linear optimization (in blue).



(a) Frame #1

(b) Frame #60

(c) Top view of the 3D estimated trajectory (ground truth in black, linear seed in red, non-linear optimization in blue)

Figure 11: Sample experiment, with the first and last image and the estimated 3D trajectory.

6 Conclusions

In this paper we have proposed an algorithm for a robust and accurate initialization of the state of a monocular-inertial sensing setup. Our specific contributions are mainly two: First, a general linear formulation to obtain an initial seed for the state, that overcomes the assumption of common matches in all the frames. And second, a state model including gravity for non-linear optimization.

We have made a thorough evaluation in the public dataset EuRoC, that shows how our proposal improves significantly the accuracy and robustness of the state of the art. We show how our proposal reduces the failure rate to almost zero in the wide variety of textures and motions in the dataset. We report the camera position errors, that are also smaller than the state of the art. Finally, we demonstrate the convergence of our gravity-observing non-linear optimization algorithm starting from the initial solution given by the closed-form approach.

Acknowledgments

This work was supported by Geomagical Labs, Inc., the Spanish government (project DPI2015-67275) and the Aragón regional government (Grupo DGA-T45_17R/FSE).

References

[1] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant outdoor localization and SLAM initialization from 2.5D maps. *IEEE transactions on visualization and computer graphics*, 21(11):1309–1318, 2015.

[2] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[3] J. Civera, A.J. Davison, and JMM Montiel. Interacting multiple model monocular SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3704–3709, 2008.

[4] Alejo Concha, Giuseppe Loianno, Vijay Kumar, and Javier Civera. Visual-inertial direct slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1331–1338, 2016.

[5] Jeffrey Delmerico and Davide Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[6] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[7] Steffen Gauglitz, Chris Sweeney, Jonathan Ventura, Matthew Turk, and Tobias Hollerer. Live tracking and mapping from both general and rotation-only camera motion. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 13–22. IEEE, 2012.

[8] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[9] Vadim Indelman, Stephen Williams, Michael Kaess, and Frank Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, 2013.

[10] Jacques Kaiser, Agostino Martinelli, Flavio Fontana, and Davide Scaramuzza. Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robotics and Automation Letters*, 2(1):18–25, 2017.

[11] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[12] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[13] Mingyang Li and Anastasios I Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[14] Haomin Liu, Guofeng Zhang, and Hujun Bao. Robust keyframe-based monocular SLAM for augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–10, 2016.

[15] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2012.

[16] Lu Ma, Juan M Falquez, Steve McGuire, and Gabe Sibley. Large scale dense visual inertial SLAM. In *Field and Service Robotics*, pages 141–155. Springer, 2016.

[17] Agostino Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44–60, 2012.

[18] Agostino Martinelli. Closed-form solution of visual-inertial structure from motion. *International journal of computer vision*, 106(2):138–152, 2014.

[19] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *2007 IEEE international conference on Robotics and automation*, pages 3565–3572, 2007.

[20] Raúl Mur-Artal and Juan D Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 2017.

[21] Raúl Mur-Artal and Juan D Tardós. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[22] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal of Computer Vision*, 113(3):208–219, 2015.

[23] Pedro Piniés, Todd Lupton, Salah Sukkarieh, and Juan D Tardós. Inertial aiding of inverse depth slam using a monocular camera. In *2007 IEEE International Conference on Robotics and Automation*, pages 2797–2802. IEEE, 2007.

[24] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *arXiv preprint arXiv:1708.03852*, 2017.

[25] Tong Qin and Shaojie Shen. Robust initialization of monocular visual-inertial estimation on aerial robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4225–4232, 2017.

[26] Jianbo Shi et al. Good features to track. In *IEEE Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[27] Chengzhou Tang, Oliver Wang, and Ping Tan. GlobalSLAM: Initialization-robust Monocular Visual SLAM. *arXiv preprint arXiv:1708.04814*, 2017.

[28] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, CMU, 1991.

[29] Zhenfei Yang and Shaojie Shen. Monocular visual–inertial state estimation with online initialization and camera–imu extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1):39–51, 2017.