Deterministic Initialization of Metric State Estimation Filters for Loosely-Coupled Monocular Vision-Inertial Systems

Deterministic Initialization of Metric State Estimation Filters for Loosely-Coupled Monocular Vision-Inertial Systems

Laurent Kneip, Stephan Weiss, and Roland Siegwart
Autonomous Systems Lab, ETH Zurich

Abstract— **In this work, we present a novel, deterministic closed-form solution for computing the scale factor and the gravity direction of a moving, loosely-coupled, and monocular vision-inertial system. The methodology is based on analysing delta-velocities. On one hand, they are obtained from a differentiation of the up-to-scale camera pose computation by a visual odometry or visual SLAM algorithm. On the other hand, they can also be retrieved from the gravity-affected short-term integration of acceleration signals. We derive a method for separating the gravity contribution and recovering the metric scale factor of the vision algorithm. The method thus also recovers the offset in roll and pitch angles of the vision reference frame with respect to the direction of the gravity vector. It uses only a single inertial integration period, and no absolute orientation information is required. For optimal sensor-fusion and metric scale-estimation filters in the loosely-coupled case, it has been shown that the convergence of the fusion of an up-to-scale pose information with inertial measurements largely depends on the availability of a good initial value for the scale factor. We show how this problem can be tackled by applying the method presented in this paper. Finally, we present results in simulation and on real data, demonstrating the suitability of the method in real scenarios.**

I. Introduction

Over the last few years, the set of robotic applications using visual sensors for simultaneous localization and mapping (SLAM) has been growing steadily, due to the generality and descriptability of cameras. Alternatives such as ultrasonic sensors, laser rangefinders, or time-of-flight cameras are too sparse in information content or bulky. The ratio between the information content given by ordinary cameras and the corresponding sensor size or weight is unmatched by any other sensor type. Especially, compact solutions such as small inspection robots or micro aerial vehicles tend towards using vision more and more. However, using only a single camera for localization of the robot poses great research challenges.

The problem with monocular vision is that cameras only provide bearing information about features, and no depth information. The latter may be recovered by triangulating matched features from multiple views [1]—called structure from motion—, more generally resulting in visual odometry [2] or visual SLAM [3], [4] approaches. However, both the camera velocity and the 3D map are only computed up to an unknown scale factor, and the orientation with respect to the vertical direction remains unknown. While this does not pose a serious problem for most computer vision applications, it certainly does in robotics, where the control stability of the vehicle depends on measurements in absolute scale. This problem can be, however, handled in different manners. In this paper, we focus on the combination of monocular vision and inertial measurements for computing the ego-motion of the sensor-carrying platform in absolute scale, and determining the orientation with respect to the gravity direction.

A good introduction on inertial and visual sensing can be found in [5]. One of the first advantages of inertial sensors compared to cameras is that, in static systems, they can provide directly the vertical direction (i.e. the vector of the gravity force). Several works have used the direction of the gravity vector for boosting ground-plane estimation and structure-from-motion [6], [7]. The difficulty when working with the vertical direction is that the gravity force cannot be read directly from the inertial sensors if the system is not static. On the other hand, the additional measurements of body-accelerations provide a useful cue for retrieving the absolute scale in the monocular case.

As analyzed in [5], there are two different ways to combine inertial and vision measurements for absolute scale structure and motion estimation, which are called *loosely-coupled* and *tightly-coupled*. The loosely-coupled approach treats the inertial and vision units as two separate filters running at different rates and exchanging information, while the tightly-coupled approach combines them into a single, statistical filter. Among the loosely-coupled approaches are the works of [8], [9], [10], [11], [12], [13], [14], [15], [16], while among the tightly-coupled ones are those of [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27].

Of special interest to us are the works of [17], [23], [24], [25], [26], [27], [15], [16], since they use only a monocular vision-inertial system without known 3D points or any additional sensors. Strelow and Singh [17] implemented a tightly-coupled approach by combining the inertial readings into an EKF-based monocular SLAM. Their implementation is similar to the popular monocular SLAM of Davison et al. [4], but they replaced the constant velocity motion model by a centripetal acceleration motion model including the inertial readings as updating measurements. In [23], [24], Lupton and Sukkarieh implemented a filter to estimate both the absolute scale of monocular SLAM and the biases of the IMU. Similarly, Kelly and Sukatme [25], [26] demonstrated that a monocular EKF-SLAM approach may also be used for simultaneous camera-to-IMU calibration. They also presented an observability analysis of the problem. More recently, Jones and Soatto [27] proposed a similar approach to solve camera self-calibration and monocular motion estimation using inertial measurements. They also characterized

the conditions under which the system is observable and described the efficient implementation of an EKF. Finally, Nützi et al. [15] and Weiss and Siegwart [16] also used a monocular SLAM, but they implemented the data fusion in a loosely-coupled approach. For the visual SLAM, they used the keyframe-based monocular SLAM of Klein and Murray [3], which outputs camera poses and 3D points up-to-scale. The absolute scale of the motion was finally estimated by fusing the monocular and inertial measurements through an EKF by adding the scale factor of the monocular SLAM as an additional variable to the state.

The drawback with all the monocular visual-SLAM-based solutions mentioned so far is that they use a filter-based approach to data fusion (such as EKF or UKF), which typically requires good initialization values for a quick convergence of the filter. They combine the IMU data by means of Kalman-filter-based approaches in order to keep track of the absolute scale and thus of the absolute velocity of the system, but as stressed in the work of Weiss and Siegwart [16] where the scale factor is held as an additional variable in the state, the convergence behavior of the latter is strongly dependent on proper initialization of the filter.

A generic way for initializing the scale without having to rely on any further knowledge is given by recent advances in deterministic scale computation approaches. Kneip et al. [28] and Martinelli [29] have developed deterministic ways for computing the scale and the gravity direction through short term integration of inertial measurements and tracking of single features. The drawback of these methods, however, is that they require double integration of the acceleration data over three or more camera observation points, thus leading to high errors. Furthermore, the work of [28] depends on preprocessing the acceleration data in order to remove the gravity component, and [29] has only shown results in simulation.

In this paper, we propose a new deterministic, closed-form solution for computing the absolute scale and the gravity direction from visual pose computation and inertial data. In contrast to [28] and [29], the approach presented is able to work with only a single IMU integration period, and does no longer depend on absolute orientation information. Furthermore, the acceleration data is only subjected to single integration instead of double integration. The approach uses only information captured inside the IMU/vision compound, and is able to deliver a reasonable value for the scale factor and the gravity vector, as well as their standard deviations within approximatively one second of operation time. We show the benefit of using the result for initializing a metric state estimation filter, and present successful results on a real dataset. Section II starts with presenting the theoretical derivations leading to a unique closed form solution. Section III presents a thorough validation of the method on synthetic data, which helps identifying critical motion sequences. We furthermore show the impact on the convergence behavior of a metric state estimation filter. Section IV then presents the results on real data, finally showing that the method is applicable to real-world scenarios.

II. Theory

A. Assumptions

We assume to have a calibrated camera and a computer vision algorithm delivering orientation and up-to-scale position information with respect to an initial frame. The initial camera frame has an orientation offset with respect to the inertial frame. The scale factor and the orientation offset are drifting over time. The algorithm is assumed to perform relative scale propagation, and hence the drifts of the unknown scale factor and orientation offset are bounded and can be regarded as constant over short observation periods. Furthermore, we assume for the moment to have a calibrated IMU (Inertial Measurement Unit) delivering a bias-free but gravity-affected measurement of the body acceleration, as well as the angular velocity. The extrinsic calibration between the camera and the IMU is assumed to be known. In the real-world case, a good intrinsic and extrinsic calibration can be obtained using off-the-shelf toolboxes like [30], [31].

B. Approach

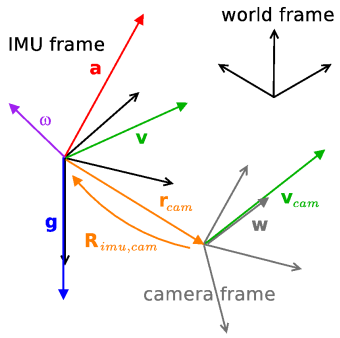

Fig. 1. Velocity $\mathbf{v}$, acceleration $\mathbf{a}$, and rotational velocity ω of the IMU. $\mathbf{g}$ denotes the gravity vector, $\mathbf{r}_{cam}$ and $\mathbf{R}_{imu,cam}$ the extrinsic parameters between the IMU and the camera, and $\mathbf{v}_{cam}$ and $\mathbf{w}$ the scaled and unscaled camera velocity in the IMU and camera frame, respectively.

As indicated in Figure 1, we define the IMU velocity $\mathbf{v}$ and acceleration $\mathbf{a}$ in the IMU frame. $\mathbf{a}$ represents the measured IMU acceleration and is a combination of the true body acceleration $\mathbf{a}'$ and the gravity vector $\mathbf{g}$—still expressed in the IMU frame—, namely $\mathbf{a} = \mathbf{a}' - \mathbf{g}$. The relative rotation from IMU frame k to $k+x$ is indicated with $\mathbf{R}_{k+x,k}$, and the sampling rate of the IMU is denoted with T. Given these definitions, a time discrete formulation—following a semi-implicit Euler scheme—of the change of the IMU velocity between frames k and $k+1$ is given by

$$\begin{aligned}
\mathbf{v}[k+1] &= \mathbf{R}_{k+1,k}\mathbf{v}[k] + T\frac{\mathbf{a}'[k+1] + \mathbf{R}_{k+1,k}\mathbf{a}'[k]}{2} \\
&= \mathbf{R}_{k+1,k}\mathbf{v}[k] + T\frac{\mathbf{g}[k+1] + \mathbf{R}_{k+1,k}\mathbf{g}[k]}{2} \\
&\quad + T\frac{\mathbf{a}[k+1] + \mathbf{R}_{k+1,k}\mathbf{a}[k]}{2} \\
&= \mathbf{R}_{k+1,k}\mathbf{v}[k] + T\mathbf{g}[k+1] \\
&\quad + T\frac{\mathbf{a}[k+1] + \mathbf{R}_{k+1,k}\mathbf{a}[k]}{2}. \qquad (1)
\end{aligned}$$

By writing out the expression for $\mathbf{v}[k+2]$ (substituting k by $k+1$), and again replacing $\mathbf{v}[k+1]$ by (1), we obtain

$$\begin{aligned}\mathbf{v}[k+2] &= \mathbf{R}_{k+2,k}\mathbf{v}[k] + 2T\mathbf{g}[k+2] \\ &\quad + T\left(\frac{\mathbf{a}[k+2] + \mathbf{R}_{k+2,k+1}\mathbf{a}[k+1]}{2}\right) \\ &\quad + T\left(\frac{\mathbf{R}_{k+2,k+1}\mathbf{a}[k+1] + \mathbf{R}_{k+2,k}\mathbf{a}[k]}{2}\right). \end{aligned} \quad (2)$$

It can be easily verified that (1) and (2) follow the simple rule

$$\mathbf{v}[k] = \mathbf{R}_{k,k-l}\mathbf{v}[k-l] + l\cdot T\mathbf{g}[k]$$
$$+\frac{T}{2}\sum_{n=0}^{l-1}\left(\mathbf{R}_{k,k-l+n+1}\mathbf{a}[k-l+n+1] + \mathbf{R}_{k,k-l+n}\mathbf{a}[k-l+n]\right). \quad (3)$$

In the real-world case, the relative rotations of the acceleration between successive IMU frames can be generated from short-term integrations of the gyroscopic measurements. IMUs commonly effectuate this integration in a complementary filter, such that the orientation change can be safely recovered directly from the IMU angles.

The velocity of the camera in the IMU frame is given by

$$\mathbf{v}_{cam}[k] = \mathbf{v}[k] + \omega[k]\times\mathbf{r}_{cam}, \quad (4)$$

where ω and $\mathbf{r}_{cam}$ represent the rotational velocity and the camera position in the IMU frame, respectively. The velocity of the camera can also be expressed using the unscaled velocity $\mathbf{w}_{cam}$ retrieved from a differentiation of the pose computation of the vision algorithm in the camera frame. It is given by

$$\mathbf{v}_{cam}[k] = q\cdot\mathbf{R}_{imu,cam}\cdot\mathbf{w}_{cam}[k], \quad (5)$$

where q represents the scale factor and $\mathbf{R}_{imu,cam}$ the rotation from the camera to the IMU frame. Replacing (5) and (4) in (3), we finally obtain

$$\begin{aligned} &q\left(\mathbf{R}_{imu,cam}\mathbf{w}_{cam}[k] - \mathbf{R}_{k,k-l}\mathbf{R}_{imu,cam}\mathbf{w}_{cam}[k-l]\right) - lT\mathbf{g}[k] \\ &= \frac{T}{2}\sum_{n=0}^{l-1}\left(\mathbf{R}_{k,k-l+n+1}\mathbf{a}[k-l+n+1] + \mathbf{R}_{k,k-l+n}\mathbf{a}[k-l+n]\right) \\ &\quad + \omega[k]\times\mathbf{r}_{cam} - \mathbf{R}_{k,k-l}\left(\omega[k-l]\times\mathbf{r}_{cam}\right). \end{aligned} \quad (6)$$

By defining

$$\Delta\mathbf{w}_{k,k-l} = \mathbf{R}_{imu,cam}\mathbf{w}_{cam}[k] - \mathbf{R}_{k,k-l}\mathbf{R}_{imu,cam}\mathbf{w}_{cam}[k-l] \quad (7)$$

$$\begin{aligned}\Delta\mathbf{a}_{k,k-l} &= \frac{T}{2}\sum_{n=0}^{l-1}\left(\mathbf{R}_{k,k-l+n+1}\mathbf{a}[k-l+n+1] + \mathbf{R}_{k,k-l+n}\mathbf{a}[k-l+n]\right) \\ &\quad + \omega[k]\times\mathbf{r}_{cam} - \mathbf{R}_{k,k-l}\left(\omega[k-l]\times\mathbf{r}_{cam}\right), \end{aligned} \quad (8)$$

we obtain,

$$q\Delta\mathbf{w}_{k,k-l} - lT\mathbf{g}[k] = \Delta\mathbf{a}_{k,k-l}. \quad (9)$$

This equation can be reformulated into

$$\left[\Delta\mathbf{w}_{k,k-l}\,\middle|\,\begin{matrix}-lTg & 0 & 0\\ 0 & -lTg & 0\\ 0 & 0 & -lTg\end{matrix}\right]\cdot\begin{pmatrix} q \\ \mathbf{n}_g[k]\end{pmatrix}$$
$$= \mathbf{A}_{k,k-l}\cdot\begin{pmatrix} q \\ \mathbf{n}_g[k]\end{pmatrix} = \Delta\mathbf{a}_{k,k-l}, \quad (10)$$

where matrix $\mathbf{A}_{k,k-l}$ intuitively contains the delta-velocity observation $\Delta\mathbf{w}_{k,k-l}$ retrieved from the vision algorithm, and vector $\Delta\mathbf{a}_{k,k-l}$ the gravity-affected one from the IMU. g represents the norm of the gravity vector and $\mathbf{n}_g[k]$ the direction of the gravity vector in the IMU frame at $t[k]$. The system of equations represents three constraints for four unkowns, and with the additional non-linear constraint $||\mathbf{n}_g[k]|| = 1$, a complete solution for computing the scale factor and the gravity direction is finally obtained. If pose estimates and the corresponding differentials are available at time instants $t[k]$ and $t[k-l]$ (note that the camera might run at a different rate), a single period of single IMU integration is thus sufficient for determination of these drifting terms. Moreover, no absolute orientation information is required.

C. Duality of Solution

The set of solutions to the underdetermined system of equations (10) is given by

$$\begin{pmatrix} q \\ \mathbf{n}_g[k]\end{pmatrix} = \mathbf{A}^{+}\cdot\Delta\mathbf{a}_{k,k-l} + \lambda\cdot\mathscr{N}(\mathbf{A}), \quad (11)$$

where $\mathbf{A}^{+}$ represents the pseudoinverse and $\mathscr{N}(\mathbf{A})$ the nullspace vector of the 3×4 matrix $\mathbf{A}$. Using the norm constraint $||\mathbf{n}_g[k]|| = 1$, we obtain the constraint

$$||[\mathbf{0}_{3\times1}\ I_3]\left(\mathbf{A}^{+}\cdot\Delta\mathbf{a}_{k,k-l} + \lambda\cdot\mathscr{N}(\mathbf{A})\right)|| = 1, \quad (12)$$

which finally leads to a second order polynomial in λ, and thus with (11) two possible solutions for the scale factor and the gravity direction.

This result raises the question whether two physically correct solutions can coexist, or whether we are missing additional constraints in order to always obtain a unique solution. The answer is given in Figure 2, which shows a simplified planar scenario. Intuitively, the algorithm is trying to determine the direction of the integrated gravity vector $lT\mathbf{g}[k]$ such that $\Delta\mathbf{a}'_{k,k-l} = \Delta\mathbf{a}_{k,k-l} + lT\mathbf{g}[k]$ becomes parallel to the delta-velocity vector $\Delta\mathbf{w}_{k,k-l}$. The condition for a physically meaningful solution is given by

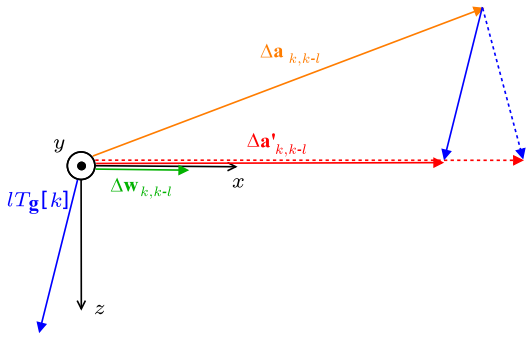

Fig. 2. Ambiguous construction of the true delta-velocity $\Delta\mathbf{a}'_{k,k-l}$ from the observed delta-velocity $\Delta\mathbf{a}_{k,k-l}$ and the gravity contribution $lT\mathbf{g}[k]$.

$$\frac{\Delta\mathbf{a}'_{k,k-l}\cdot\Delta\mathbf{w}_{k,k-l}}{||\Delta\mathbf{a}'_{k,k-l}||\cdot||\Delta\mathbf{w}_{k,k-l}||}=1, \tag{13}$$

meaning that the delta-velocities $\Delta\mathbf{a}'_{k,k-l}$ and $\Delta\mathbf{w}_{k,k-l}$ are not only parallel, but also pointing to the same direction. As we can see in Figure 2, there exist situations with two physically correct solutions fulfilling this condition. Moreover, it can be observed that the two solutions come closer to each other the more orthogonal the velocity change is with respect to the gravity direction, which then in practice makes the disambiguation a challenging problem.

D. Unique solution and robustness against noise

In the real world case, the noise in the acceleration signal and the numerically differentiated visual position estimation leads to the fact that the determination of the scale and the gravity direction through a single IMU integration period is prone to errors. Our solution to tackle this problem consists in considering multiple delta-velocity samples over an extended observation window of duration $l_{max}T$. Moreover, in order to ensure higher signal-to-noise ratios and noise cancelling effects in the acceleration integration, we also define a minimal integration time $l_{min}T$. As illustrated in Figure 3, this leads to a finite number of delta-velocity samples that can be taken over the entire observation window with different IMU integration times reaching from $l_{min}T$ to $l_{max}T$. If an unscaled camera velocity $\mathbf{w}$ is delivered in regular intervals of nT, and l_{max} and l_{min} are multiples of n, this leads to $\frac{1}{2}(\frac{l_{max}-l_{min}}{n}+2)(\frac{l_{max}-l_{min}}{n}+1)$ possible samples.

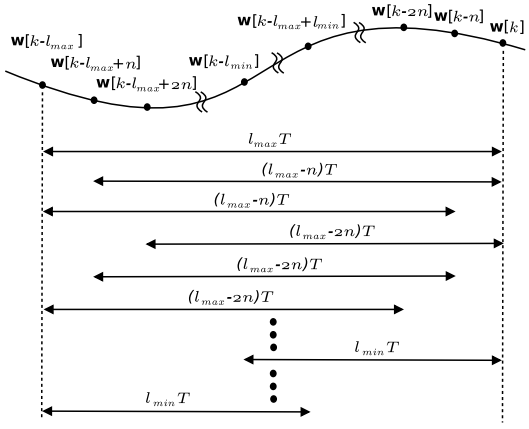

Fig. 3. Different possibilities of creating delta-velocity samples in an observation window of duration $l_{max}T$. The IMU integration times for the different samples reach from $l_{min}T$ to $l_{max}T$.

Let's assume that the two camera velocities for a given sample are $\mathbf{w}[k-i]$ and $\mathbf{w}[k-i-l]$. In order to have all samples returning a measurement value for the gravity direction in the most recent frame of the complete observation window—namely $\mathbf{n}_g[k]$—, we need to introduce the rotation between older IMU frames at $t[k-i]$ and the most recent one at $t[k]$ into (10), which results in

$$\left[\Delta\mathbf{w}_{k-i,k-i-l}\left|\begin{pmatrix}-lTg & 0 & 0\\ 0 & -lTg & 0\\ 0 & 0 & -lTg\end{pmatrix}\mathbf{R}_{k-i,k}\right.\right]\cdot\begin{pmatrix}q\\ \mathbf{n}_g[k]\end{pmatrix} = \Delta\mathbf{a}_{k-i,k-i-l}. \tag{14}$$

As indicated under II-C, each delta-velocity sample will then return two solutions using the norm constraint on $\mathbf{n}_g[k]$.

Our approach to find the best solution over the entire observation window then consists in checking the level of agreement of every solution with all other delta-velocity constraints, and then selecting the best one. The error function for a certain solution is simply given by

$$\sum_{l=l_{min}}^{l_{max}}\sum_{i=0}^{l_{max}-l}||\mathbf{A}_{k-i,k-i-l}\cdot\begin{pmatrix}q\\ \mathbf{n}_g[k]\end{pmatrix}-\Delta\mathbf{a}_{k-i,k-i-l}||, \tag{15}$$

where l and i are incremented in steps of n. In the general case, this approach allows to solve the solution ambiguity, since—in contradiction to the "true" solution of each sample—the "wrong" solution typically fulfills other delta-velocity constraints from the observation window to a significantly smaller degree. However, in critical motion situations, the ambiguity can still persist over the entire observation period. This happens for instance when the motion is rectilinear with more or less constant acceleration over the entire window, and can be identified by analysing the results of the error function. In this case, only a consideration of previous scale measurements can help to identify the best "true" solution. This is allowed based on the assumption that the scale factor is drifting only slowly.

Note that it is possible to stack multiple constraints in order to find a least-squares solution. While this theoretically even allows the determination of $||\mathbf{g}||$, for the sake of robustness, in practice it is better to keep the dimensionality of the solution vector as small as possible, therefore fix $||\mathbf{g}||$ to the known value via the non-linear norm constraint. Moreover, the least-squares solution would require an additional outlier rejection step.

III. Simulation results

A. Noise-free case

For a proper test of our algorithm, we selected helicoidal signals in all three directions with phase-shifts of 120° in between. This is done in order to emulate a camera trajectory with variable dynamics and variable dominant direction of acceleration with respect to the gravity vector. The signal amplitude is $\pm 1m$, and the frequency corresponds to $\frac{\pi}{6}\frac{rad}{s}$. After deriving the body accelerations, the scale factor of the position is modified with a bias drifting quickly from 2 to 3 over the entire duration of the experiment ($30s$). The rotational offset of the vision reference frame with respect to the gravity direction is then varied between 0.1 and 0.3 *rad* in roll, and -0.3 and -0.1 *rad* in pitch. Respecting the changing offset of the initial vision frame, the vertical gravity vector is finally transformed from the inertial frame to the body frame, and added to the acceleration signals. The sampling

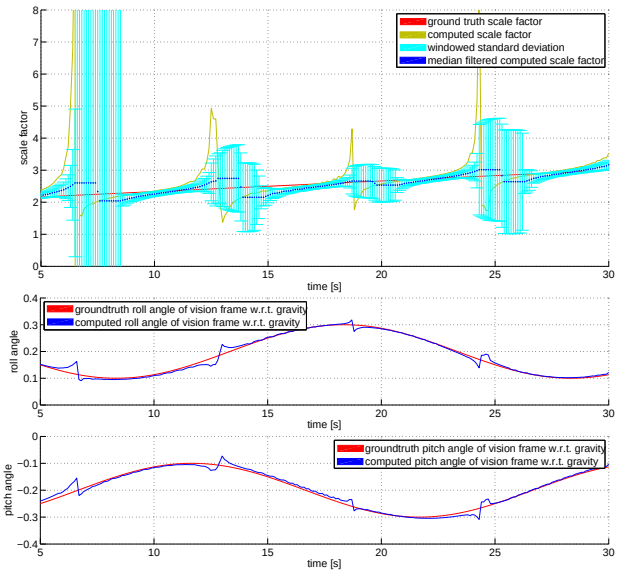

Fig. 4. Results of the scale and gravity computation on a simulated and noise-free dataset. Both the scale factor and the roll and pitch offsets of the vision frame are continuously varied over time.

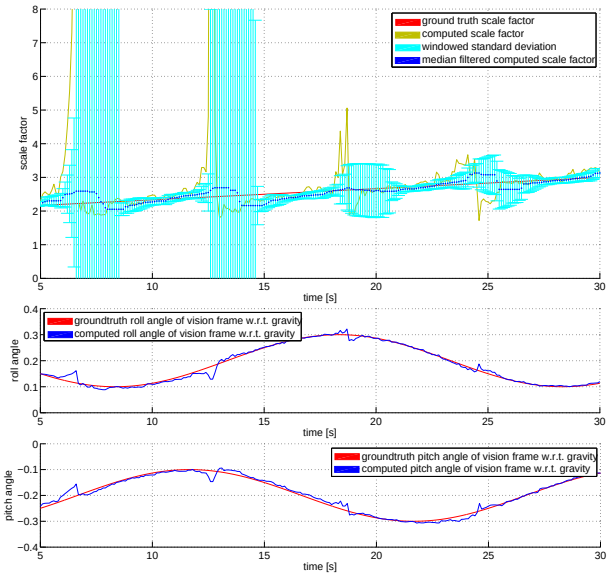

Fig. 5. Results of the scale and gravity computation on a simulated and noisy dataset.

rates of IMU and camera are simulated with 100 and 10 Hz, respectively.

As shown in Figure 4, the algorithm is able to continuously recover both the metric scale factor as well as the offset of the vision reference frame with respect to the direction of gravity. For this experiment, we set $l_{max} = 120$ and $l_{min} = 80$, meaning an observation window of $1.2s$, and integration times varying between 0.8 and $1.2s$. The blue dots represent the result of an additional windowed median filter over $2s$.

It can be observed that the error of the scale factor computation is increased in regular intervals of $6s$, thus clearly related to the instantaneous motion characteristics. This also leads to temporary biases in the windowed median and the estimated covariance of the signal. We found that errors occur when the delta-velocities are almost orthogonal to the gravity vector. It can also be proven mathematically that the accuracy of the algorithm is depending on the orthogonality between $\Delta\mathbf{a}_{k,k-l}$ or $\Delta\mathbf{w}_{k,k-l}$ and $\mathbf{n}_g[k]$. We have

$$q = \frac{||\Delta\mathbf{a}'_{k,k-l}||}{||\Delta\mathbf{w}_{k,k-l}||} = \frac{||\Delta\mathbf{a}_{k,k-l} + lTg\mathbf{n}_g[k]||}{||\Delta\mathbf{w}_{k,k-l}||}.$$

When chosing $\Delta\mathbf{a}_{k,k-l} = \begin{pmatrix}\Delta a_{k,k-l} & 0 & 0\end{pmatrix}^t$, $\Delta\mathbf{w}_{k,k-l} = \begin{pmatrix}\Delta w_{k,k-l} & 0 & 0\end{pmatrix}^t$, and $\mathbf{n}_g[k] = \begin{pmatrix}\cos\theta & 0 & \sin\theta\end{pmatrix}^t$— meaning delta-velocities along the x-direction and variable orthogonality of the gravity direction—, we obtain

$$\begin{aligned} q &= \frac{||\begin{pmatrix}(\Delta a_{k,k-l} + lTg\cos\theta) & 0 & lTg\sin\theta\end{pmatrix}^t||}{|\Delta w_{k,k-l}|} \\ &= \frac{\sqrt{\Delta a_{k,k-l}^2 + 2\Delta a_{k,k-l} lTg\cos\theta + (lTg)^2}}{|\Delta w_{k,k-l}|}. \end{aligned}$$

Partial differentiation with respect to θ leads to

$$\frac{\partial q}{\partial\theta} = \frac{-\Delta a_{k,k-l} lTg\sin\theta}{|\Delta w_{k,k-l}|\sqrt{\Delta a_{k,k-l}^2 + 2\Delta a_{k,k-l} lTg\cos\theta + (lTg)^2}}.$$

This result basically proves that an error in the computed gravity direction has biggest impact on the accuracy of the scale when the gravity direction is orthogonal to the delta-velocity ($\theta = \pm 90°$).

The mentioned errors also underline the statements from Section II-C: The algorithm is performing better when the delta-velocity components are aligned with the gravity vector, thus easing the disambiguation of the two solutions. Moreover, since we are having observation windows of $1.2s$, the effect gets amplified significantly depending on the amount of drift in the scale factor. Especially in critical cases, we are depending on former values for doing the disambiguation. These operations are actually based on the assumption that the drift is not too intense.

B. Noise-affected case

In order to analyse the effect of noise on the quality of the results, we repeated the same experiment with white Gaussian noise added to the measured camera position (standard deviation: $0.01\frac{m}{s}$) and acceleration signals (standard deviation: $0.1\frac{m}{s\sqrt{h}}$[1]). Due to noise-cancelling effects, the impact on the integration of the acceleration signals remains minor. The impact on the differentiated camera position, however, is significant. In order to obtain stable results, one has to differentiate the inexact position in a numerically stable way, which means based on a regression or spectral decomposition method. We opted for the first one due to its simplicity, and selected a window over 10 past samples ($1s$)

[1] $\frac{m}{s\sqrt{h}}$ is the standard unit to express the error of an IMU (m is meters, s is seconds, and h is hours)

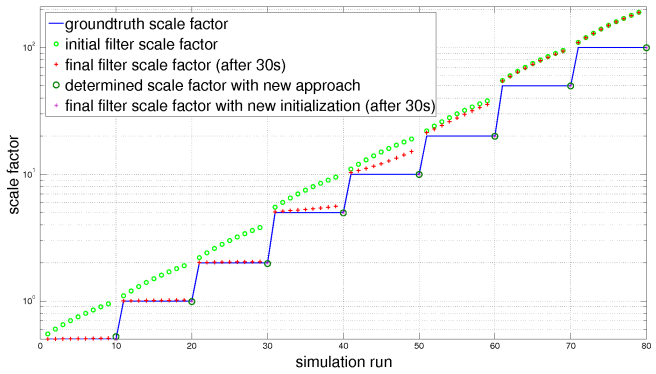

Fig. 6. Effects on the scale estimation filter convergence with initial scale factors reaching from 110% to 190% of the groundtruth value. The final scale factor of the filter represents the value after 30 seconds of operation time. The plot also shows that the filter convergence is guaranteed, if using the scale factor determined by our new approach.

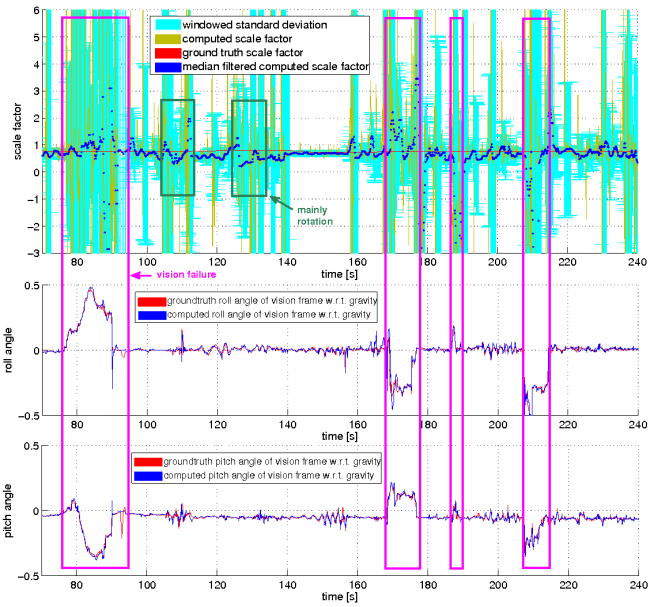

Fig. 7. Results of the scale factor and roll/pitch-offset computation on a real dataset. The algorithm is performing well apart from cases where visual SLAM estimated are erroneous (purple blocks) and critical motion sequences (green blocks).

for performing third-order polynomial regression. The scale and gravity computation using this numerical differentiation can finally be observed in Figure 5. It clearly shows that the quality of the results remains comparable to the noise-free case.

C. Improvement of scale estimation filter convergence

As mentioned in the introduction, the obvious benefit of the proposed method is its ability to deliver a good initial filter-value for the scale factor in the loosely-coupled monocular case, where the scale factor is maintained as an additional variable in the state vector. The sensor-fusion filter used is the one presented in the work of Weiss and Siegwart [16]. In their work, they demonstrated the impact of a bad initial value on the convergence behavior of the filter, so here we reproduce a similar experiment in order to show how an initial value proposed by our algorithm is able to boost the filter convergence. The results are presented in Figure 6, where initial scale factor values ranging from 110% to 190% of the ground truth scale factor are tested. It can be observed that the filter is converging worse for higher scale factors than for lower ones, which suggests that the important criterion for good convergence is the absolute deviation from the true scale factor, and not the relative one. In case of too large errors, the filter is not able to converge at all anymore. The good thing, however, is that the absolute error of the scale factor returned by our deterministic solution behaves in a similar way, meaning that it stays more or less independent of the absolute value of the groundtruth scale factor. It is able to return satisfying results and enable the filter convergence for all tested groundtruth scale factors (up to values of 100), and it has been verified that the absolute errors are constantly remaining in the same order of magnitude.

IV. Results on Real Data

In order to have a common basis of comparison with state-of-the-art methods, we tested the new approach on the same dataset as the one used in [28] and [16]. The dataset consists of inertial and monocular vision data, and ground truth data has been gathered using the Vicon motion capture system. The monocular camera-IMU setup consists of a uEye UI-122xLE—a small monochrome USB-camera gathering 752×480 images with global shutter at a rate of 15 Hz—and a Crossbow VG400CC-200 IMU providing measurement updates at a rate of 75 Hz. The noise contained in the acceleration measurements amounts to $0.5\frac{m}{s\sqrt{h}}$. The field of view of the camera is 150°. The extrinsic calibration of the IMU is realized using the camera-inertial calibration toolkit by Lobo [31]. The intrinsic camera calibration and visual pose estimates have been generated by applying the PTAM-framework (Parallel Tracking and Mapping), a visual SLAM implementation by Klein and Murray [3]. The settings for l_{max} and l_{min} (maximum and minimum number of IMU samples for integration windows) have been left unchanged in comparison to Section III.

As illustrated in Figure 7, the algorithm is well able to continuously determine an approximative value for the scale factor and the rotational offset with respect to the gravity vector. Occasional absences of good estimates are due to tracking failures in the vision algorithm. Even though the determined scale contains a significant amount of noise, the absolute error of the median-filtered value is steadily remaining below 0.5. There are two additional sequences during which the absolute error is slightly elevated, however corresponding to motion periods that consist mainly of pure rotations. The method depends on the a priori knowledge of the bias that is typically contained in real acceleration measurements, which can be obtained by averaging the acceleration values in a static pre-initialization phase. Note that, in analogy to the work of [16], a failure of the visual tracking algorithm can be easily detected by analysing the

continuity of the estimated roll and pitch offsets between the vision and the inertial frame.

Even though the method is obviously not able to compete with the statistical estimation of the scale factor (denoted as the ground truth value in Figure 7), one has to bare in mind that the initial value of the scale factor for the statistical solution has been set by hand, and that our approach thus provides very important complementary information for good initialization of the filter state and covariance. In comparison to [28], the approach is more robust since using the support of an entire vision algorithm instead of only single feature observations.

V. Conclusion

In this paper, we present a novel approach for loosely-coupled monocular vision-inertial systems to deterministically compute the scale factor and the gravity vector. The method is based on analysing delta-velocity observations obtained both from a stable numerical differentiation of up-to-scale visual pose estimates, and short-term integrations of the gravity-affected acceleration signal. An ambiguity in the solution has been identified along with ways to resolve it via a noise-resistant approach. The resulting approach for scale computation can prove very useful in generating good initial values for a motion- and scale-estimation filter, which is crucial for successful convergence in the estimation process. Reasonable performance could be demonstrated on real data. Limitations are given by rectilinear trajectories or zero-acceleration motion sequences.

VI. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly) and from the Swiss National Science Foundation under grant agreement n. 200021 125017/1.

References

[1] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003.

[2] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 1, pages 652–659, 2004.

[3] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, 2007.

[4] A. Davison, D. Reid, D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):1052–1067, 2007.

[5] P. Corke, J. Lobo, and J Dias. An introduction to inertial and visual sensing. *International Journal of Robotics Research*, 26(6):519–535, 2007.

[6] J. Lobo and J. Dias. Inertial sensed ego-motion for 3d vision. *Journal of Robotic Systems*, 21(1):3–12, 2004.

[7] T. Vieville, E. Clergue, and P.E.D.S. Facao. Computation of ego motion using the vertical cue. In *Machine Vision Applications*, volume 8, pages 41–52. Springer, 1995.

[8] P. Corke. An inertial and visual sensing system for a small autonomous helicopter. *International Journal of Robotics Systems*, 21(2):43–51, 2004.

[9] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. In *Proceedings of The IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004.

[10] L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *Int. J. Rob. Res.*, 26(6):577–589, 2007.

[11] P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous motion and structure estimation by fusion of inertial and vision data. *The International Journal of Robotics Research*, 26(6):591–605, 2007.

[12] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, pages 4326–4333, Washington D.C., 2002.

[13] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Roma, 2007.

[14] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Andrew E. Johnson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[15] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. In *International Conference on Unmanned Aerial Vehicles*, Dubai, 2010.

[16] S. Weiss and R. Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[17] D. Strelow and S. Singh. Online motion estimation from image and inertial measurements. In *Workshop on Integration of Vision and Inertial Sensors (INERVIS)*, Coimbra, Portugal, 2003.

[18] D. Strelow. Motion estimation from image and inertial measurements. *PhD thesis*, 2004.

[19] S.G. Chroust and M. Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotic Systems*, 21(2):73–83, 2004.

[20] A. Huster, E. W. Frew, and S. M. Rock. Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements. In *Proceedings of The Oceans Conference*, volume 3, pages 1857–1864, Biloxi, 2002. MTS/IEEE.

[21] G. Qian, R. Chellappa, and Q. Zheng. Bayesian structure from motion using inertial information. In *International Conference on Image Processing*, Rochester, New York, USA, 2002.

[22] G. Baldwin, R. Mahony, and J. Trumpf. A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Kobe, 2009.

[23] T. Lupton and S. Sukkarieh. Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In *International Conference on Robotics and Automation*, Pasadena, California, USA, 2008.

[24] T. Lupton and S. Sukkarieh. Efficient integration of inertial observations into visual SLAM without initialization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009.

[25] J. Kelly and G. S. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In *Proc. IEEE International Conference on Computational Intelligence in Robotics and Automation*, Korea, 2009.

[26] J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, 2010.

[27] E. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *International Journal of Robotics Research*, 2010.

[28] L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and Siegwart R. Closed-form solution for absolute scale velocity determination combining intertial measurements and a single feature correspondence. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[29] A. Martinelli. Closed-form solution for attitude and speed determination by fusing monocular vision and inertial measurements. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[30] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[31] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575, 2007.