Online temporal calibration for camera–IMU systems: Theory and algorithms

Mingyang Li and Anastasios I. Mourikis

The International Journal of Robotics Research published online 1 May 2014
DOI: 10.1177/0278364913515286

Article

The International Journal of Robotics Research
1–18

DOI: 10.1177/0278364913515286
ijr.sagepub.com

Online temporal calibration for camera–IMU systems: Theory and algorithms

Mingyang Li and Anastasios I. Mourikis

Abstract

When fusing visual and inertial measurements for motion estimation, each measurement's sampling time must be precisely known. This requires knowledge of the time offset that inevitably exists between the two sensors' data streams. The first contribution of this work is an online approach for estimating this time offset, by treating it as an additional state variable to be estimated along with all other variables of interest (inertial measurement unit (IMU) pose and velocity, biases, camera-to-IMU transformation, feature positions). We show that this approach can be employed in pose-tracking with mapped features, in simultaneous localization and mapping, and in visual–inertial odometry. The second main contribution of this paper is an analysis of the identifiability of the time offset between the visual and inertial sensors. We show that the offset is locally identifiable, except in a small number of degenerate motion cases, which we characterize in detail. These degenerate cases are either (i) cases known to cause loss of observability even when no time offset exists, or (ii) cases that are unlikely to occur in practice. Our simulation and experimental results validate these theoretical findings, and demonstrate that the proposed approach yields high-precision, consistent estimates, in scenarios involving either known or unknown features, with both constant and time-varying offsets.

Keywords

Vision-aided inertial navigation, online temporal calibration, time-offset estimation, identifiability analysis

1. Introduction

Autonomous vehicles such as aerial vehicles and ground robots require accurate 3D pose estimates. We here focus on vision-aided inertial navigation methods, which provide such estimates by fusing measurements from a camera and an inertial measurement unit (IMU). In recent years, several algorithms of this kind have been proposed, tailored to different applications. For instance, if features with known coordinates are available, map-based localization algorithms can be used to provide absolute-pose estimates (e.g. Wu et al., 2005; Trawny et al., 2007). In an unknown environment, simultaneous localization and mapping (SLAM) methods can be used for jointly estimating the vehicle's 3D motion and the positions of visual landmarks (e.g. Jones and Soatto, 2011; Hesch et al., 2012). Finally, if estimates for the vehicle's motion are needed but no map-building is required, visual–inertial odometry methods can be employed (e.g. Mourikis and Roumeliotis, 2007; Li and Mourikis, 2012).

For the estimation algorithms to perform well in any of these cases, both the spatial and the temporal relationship between the camera and IMU must be accurately modeled. The first of these problems, often termed *extrinsic sensor calibration*, has been addressed by several authors (see e.g. Mirzaei and Roumeliotis, 2008; Kelly and Sukhatme, 2010; Jones and Soatto, 2011; Weiss et al., 2012b). By contrast, the problem of temporal calibration between the data streams of the camera and the IMU has largely been left unexplored, and is the main focus of this work.

To enable the processing of the sensor measurements in an estimator, a timestamp is typically obtained for each camera image and IMU sample. This timestamp is taken either from the sensor itself, or from the operating system (OS) of the computer receiving the data. These timestamps, however, are typically inaccurate. Due to the time needed for data transfer, sensor latency, and OS overhead, a delay (different for each sensor) exists between the actual sampling of a measurement and its timestamp. Additionally, if different clocks are used for timestamping (e.g. on different sensors), these clocks may suffer from clock skew. As a result, an unknown time offset t_d typically exists between the timestamps of the camera and the IMU. If this time

Department of Electrical Engineering, University of California, Riverside, USA

Corresponding author:
Anastasios I. Mourikis, Department of Electrical Engineering, University of California, Riverside, CA 92521, USA.
Email: mourikis@ee.ucr.edu

offset is not estimated and accounted for, it will introduce unmodeled errors in the estimation process, and reduce its accuracy.

With the exception of the work of Kelly and Sukhatme (2010), discussed in Section 2, previous literature on vision-aided inertial navigation has not addressed the problem of estimating t_d. Presumably, algorithm developers either determine this offset using hardware-specific knowledge, or develop offline methods for estimating t_d on a case-by-case basis, or assume that t_d is sufficiently small, so it can be ignored. However, these solutions are not general enough, and in the case where t_d varies over time (e.g. due to clock skew) they can lead to eventual failure of the estimator. In contrast to these methods, we here present a methodology for estimating t_d online during vision-aided inertial navigation, and a theoretical analysis of its properties.

Specifically, the first contribution of this work is a formulation for the online estimation of the time offset in extended Kalman filter (EKF)-based algorithms. Our approach relies on explicitly including t_d in the EKF state vector, and estimating it concurrently with all other states of interest. This method is applicable in both known and unknown environments (i.e. in both map-based estimation, and in SLAM/visual–inertial odometry), and with both feature-based and pose-based EKF formulations. We here present EKF estimators for all these cases. These estimators jointly estimate (i) the IMU state, comprising the IMU position, velocity, orientation, and biases, (ii) the transformation between the camera and IMU frames, (iii) the time offset between the sensors' data streams, and (iv) the positions of visual features, when EKF-SLAM is performed. Compared to the 'standard' algorithms, which require the time offset to be perfectly known in advance, the proposed approach only incurs a minimal computational overhead, as it only requires one additional scalar variable to be included in the filter state.

The second main contribution of this work is the analysis of the identifiability of the time offset between the camera and IMU. In particular, we present the analysis for the case where the time offset, feature positions, camera-to-IMU transformation, and IMU biases are all unknown, and need to be estimated along with the IMU trajectory. For this (most general) scenario, we prove that the time offset t_d is locally identifiable in general trajectories. Moreover, we characterize the critical trajectories that cause loss of identifiability, and show that these are either (i) cases that are known to be degenerate even when t_d is perfectly known (e.g. constant-velocity motion), or (ii) cases that are unlikely to occur in practice. Thus, including t_d in the estimated state vector does not introduce new, practically significant critical trajectories for the overall system's observability.

These theoretical results have direct practical implications. They prove that, when the time offset between the camera and IMU is not known in advance, it *can* be estimated online by the proposed EKF-based algorithms, together with all the other quantities of interest. The identifiability properties of t_d guarantee that this estimation will be successful, even if t_d is drifting over time. Our experimental and simulation results confirm that the proposed methods yield high-precision, consistent state estimates, in scenarios involving either known or unknown features, with both constant and time-varying offsets. Importantly, we show that the accuracy obtained when t_d is estimated online is almost indistinguishable from the precision we would obtain if t_d was perfectly known in advance. These results, together with the fact that the inclusion of t_d in the filter's state vector causes minimal increase in the estimator's complexity, demonstrate the practical advantages of the online temporal calibration of camera–IMU systems.

2. Related work

Sensor latency is a common problem, and therefore the topic of state estimation with time-delayed and time-offset measurements has been studied in several contexts. The vast majority of existing approaches focus on the problem of using delayed sensor measurements for estimation, when the delay is perfectly known in advance (see e.g. Bak et al., 1998; Bar-Shalom, 2002; Zhang et al., 2005, and references therein). The vision-aided inertial navigation method of Weiss et al. (2012a) belongs to this category. Moreover, a number of methods have been proposed for the case where the time offset is only approximately known (Julier and Uhlmann, 2005; Choi et al., 2009). However, all these algorithms use the time offset between the sensors as an input: they do not attempt to estimate it, or to improve a prior estimate using additional data, and are thus not applicable to the problem we address.

To the best of our knowledge, the first work addressing the problem of time-offset estimation in camera–IMU systems in a principled manner is that of Kelly and Sukhatme (2010). In that work, rotation estimates from each individual sensor are first computed, and then temporally aligned via batch ICP-like registration in the space of rotations. This technique can be applied to sensors beyond just cameras and IMUs (Tungadi and Kleeman, 2010). While this approach addresses the presence of a time offset and its estimation in a rigorous manner, it has two limitations. First, being offline in nature, it cannot operate in the presence of time-varying time offsets. Moreover, since only the rotation measurements are used, this method does not utilize all the available measurement information.

We note that the standard way to estimate the time-shift between two signals is by determining the peak of the cross-correlation between them (see e.g. Fertner and Sjolund, 1986; Giovanni and Scarano, 1993, and references therein). This technique could be used to estimate the time offset between the camera and IMU, by correlating the rotation estimates computed by the two sensors. Moreover, a number of approaches exist that determine the timing between sensors using low-level data, such as direct measurements

of network response times (see e.g. Harrison and Newman, 2011, and references therein). Conceivably, any of these methods could be used online, in parallel to the state estimator, to provide estimates for the time offset t_d. To deal with time-varying offsets, the methods could run periodically. However, this would lead to a more complex implementation than the solution proposed in this work, and would not be able to properly model the uncertainty of t_d in the estimator.

In contrast to the methods discussed above, the approach proposed in our work allows for online estimation of t_d, by treating it as an additional state to be estimated. This idea, which has recently also been proposed in the context of GPS-based navigation (Skog and Haendel, 2011), makes it possible to use all the measurements from the camera, gyroscope, and accelerometer in a tightly coupled formulation. Moreover, it models the uncertainty in the estimate of t_d, and its impact on the accuracy of motion estimation, in a natural way via the EKF covariance matrix. The formulation of the EKF-based estimators for online temporal calibration first appeared in an earlier conference version of this paper (Li and Mourikis, 2013a). Compared to that publication, we here present additional experimental and simulation results, as well as a detailed analysis of the time offset's identifiability (Section 6). To the best of our knowledge, this is the first time such an analysis has been presented for 3D motion estimation.

Note that, even though the observability properties of vision-aided inertial navigation have been studied in great detail in the past, all prior work assumes perfect knowledge of the time offset between sensors (Mirzaei and Roumeliotis, 2008; Jones and Soatto, 2011; Kelly and Sukhatme, 2011; Martinelli, 2012). Our identifiability analysis makes it possible to extend the results of the prior work to the unknown-t_d case. Specifically, in Section 6 we prove that even if t_d is unknown, it can be determined based on the sensor data, except in a small set of degenerate cases. Therefore, unless the trajectory is one of the degenerate ones, we can view the problem as one where t_d is known, and the results of Jones and Soatto (2011) and Kelly and Sukhatme (2011) apply.

3. Time-offset definition

Consider a system comprising a camera and IMU, in which each sensor provides measurements at a constant frequency, known at least to a good approximation. As described in Section 1, an unknown time offset, t_d, generally exists between the two sensors' reported timestamps. Figure 1 illustrates how a time offset can arise due to difference in the sensors' latency, but t_d may also arise due to synchronization errors, missed data, and clock skew. Note that, depending on the system at hand, t_d may have a positive or negative value. For instance, if the offset is caused by sensor latency, then t_d will be positive when the IMU has a longer latency than the camera, and negative in the opposite case.

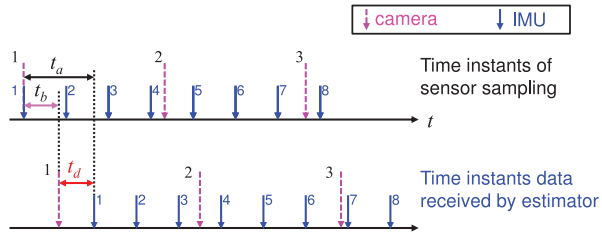

Fig. 1. Example of time offset arising due to latency in the sensor data. In this case the IMU data arrives with a latency t_a, while the camera data have a latency t_b, both of which are unknown. Since t_a and t_b are different, the estimator receives measurements that were recorded simultaneously (e.g. the first IMU and camera measurement) with timestamps that are offset by $t_d = t_a - t_b$.

In our formulation, we use the 'IMU time' as the time reference by convention. Therefore, $\boldsymbol{\zeta}(t)$ denotes the value of a quantity $\boldsymbol{\zeta}$ at the time instant the IMU measurement with timestamp t was recorded. On the other hand, if an image with timestamp t is received from the camera, this image was actually captured at time $t + t_d$. We point out that in our formulation it is possible (as proven in Section 6) to identify the time offset t_d, but not the individual latencies of the sensors. However, these latencies cannot be determined unless additional, external state information is available. By using the 'IMU time' as the time reference we circumvent this difficulty, and obtain equations that only involve the time offset t_d, which we can estimate using only the camera and IMU measurements.

4. Map-based pose estimation

4.1. State vector formulation

We first consider the case of pose estimation in an environment containing features with known 3D coordinates. Our objective is to estimate the 3D pose of the system with respect to a global coordinate frame, $\{G\}$, while concurrently performing temporal and spatial calibration between the two sensors. To this end we employ an EKF estimator, whose state vector comprises the IMU state and the time offset t_d, as well as the transformation, ${}^C_I\mathbf{T}$, between the IMU frame, $\{I\}$, and the camera frame, $\{C\}$:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_I^{\mathrm{T}}(t) & {}^C_I\bar{\mathbf{q}}^{\mathrm{T}} & {}^C\mathbf{p}_I^{\mathrm{T}} & t_d \end{bmatrix}^{\mathrm{T}} \tag{1}$$

where[1] $\mathbf{x}_I(t)$ is the IMU state at time t, and ${}^C_I\mathbf{T}$ is described by the unit quaternion ${}^C_I\bar{\mathbf{q}}$ and the translation vector ${}^C\mathbf{p}_I$. Following standard practice, we define the IMU state as the 16×1 vector

$$\mathbf{x}_I = \begin{bmatrix} {}^I_G\bar{\mathbf{q}}^{\mathrm{T}} & {}^G\mathbf{p}_I^{\mathrm{T}} & {}^G\mathbf{v}_I^{\mathrm{T}} & \mathbf{b}_{\mathbf{g}}^{\mathrm{T}} & \mathbf{b}_{\mathbf{a}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{2}$$

where the 4×1 unit quaternion ${}^I_G\bar{\mathbf{q}}$ describes the rotation from the global frame to the IMU frame, ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the IMU's position and velocity expressed in the global frame, and $\mathbf{b}_{\mathbf{g}}$ and $\mathbf{b}_{\mathbf{a}}$ are the IMU's gyroscope and accelerometer

biases. These are modeled as random walk processes driven by zero-mean white Gaussian noise vectors $\mathbf{n_{wg}}$ and $\mathbf{n_{wa}}$, respectively. Using (2), the filter state vector becomes

$$\mathbf{x} = \begin{bmatrix} {}^I_G\bar{\mathbf{q}}^{\mathrm{T}} & {}^G\mathbf{p}_I^{\mathrm{T}} & {}^G\mathbf{v}_I^{\mathrm{T}} & \mathbf{b}_\mathbf{g}^{\mathrm{T}} & \mathbf{b}_\mathbf{a}^{\mathrm{T}} & {}^C_I\bar{\mathbf{q}}^{\mathrm{T}} & {}^C\mathbf{p}_I^{\mathrm{T}} & t_d \end{bmatrix}^{\mathrm{T}}$$

Based on this state vector, we obtain the following 22×1 error-state vector for the EKF:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\boldsymbol{\theta}}^{\mathrm{T}} & {}^G\tilde{\mathbf{p}}_I^{\mathrm{T}} & {}^G\tilde{\mathbf{v}}_I^{\mathrm{T}} & \tilde{\mathbf{b}}_\mathbf{g}^{\mathrm{T}} & \tilde{\mathbf{b}}_\mathbf{a}^{\mathrm{T}} & \tilde{\boldsymbol{\phi}}^{\mathrm{T}} & {}^C\tilde{\mathbf{p}}_I^{\mathrm{T}} & \tilde{t}_d \end{bmatrix}^{\mathrm{T}} \quad (3)$$

where for the position, velocity, and bias states, as well as for the time offset t_d, the standard additive error definition has been used (e.g. ${}^G\mathbf{v}_I = {}^G\hat{\mathbf{v}}_I + {}^G\tilde{\mathbf{v}}_I$). On the other hand, for the orientation errors we use a minimal 3D representation, defined by the equations (Li and Mourikis, 2012, 2013b)

$${}^I_G\bar{\mathbf{q}} \simeq {}^I_G\hat{\bar{\mathbf{q}}} \otimes \begin{bmatrix} \frac{1}{2}\tilde{\boldsymbol{\theta}} \\ 1 \end{bmatrix} \text{ and } {}^C_I\bar{\mathbf{q}} \simeq {}^C_I\hat{\bar{\mathbf{q}}} \otimes \begin{bmatrix} \frac{1}{2}\tilde{\boldsymbol{\phi}} \\ 1 \end{bmatrix} \quad (4)$$

4.2. EKF propagation

In the EKF, the IMU measurements are used to propagate the state and covariance estimates. Specifically, the gyroscope and accelerometer measurements are described respectively by the equations

$$\boldsymbol{\omega}_m(t) = {}^I\boldsymbol{\omega}(t) + \mathbf{b_g}(t) + \mathbf{n_r}(t) \quad (5)$$

$$\mathbf{a}_m(t) = {}^I_G\mathbf{R}(t)\left({}^G\mathbf{a}(t) - {}^G\mathbf{g}\right) + \mathbf{b_a}(t) + \mathbf{n_a}(t) \quad (6)$$

where ${}^I\boldsymbol{\omega}$ is the IMU's rotational velocity, ${}^G\mathbf{g}$ is the gravitational acceleration, and $\mathbf{n_r}$ and $\mathbf{n_a}$ are zero-mean white Gaussian noise processes. Using these measurements, we can write the dynamics of the state vector as

$${}^I_G\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\boldsymbol{\Omega}\Big(\boldsymbol{\omega}_m(t) - \mathbf{b_g}(t) - \mathbf{n_r}(t)\Big){}^I_G\bar{\mathbf{q}}(t) \quad (7)$$

$${}^G\dot{\mathbf{v}}(t) = {}^I_G\mathbf{R}(t)^{\mathrm{T}}\left(\mathbf{a}_m(t) - \mathbf{b_a}(t) - \mathbf{n_a}(t)\right) + {}^G\mathbf{g} \quad (8)$$

$${}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t) \quad (9)$$

$$\dot{\mathbf{b}}_\mathbf{g}(t) = \mathbf{n_{wg}}(t), \quad \dot{\mathbf{b}}_\mathbf{a}(t) = \mathbf{n_{wa}}(t) \quad (10)$$

$${}^C_I\dot{\bar{\mathbf{q}}}(t) = \mathbf{0}, \qquad {}^C\dot{\mathbf{p}}_I(t) = \mathbf{0} \quad (11)$$

$$\dot{t}_d(t) = 0 \quad (12)$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} \lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ \boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix} \quad (13)$$

Equations (7) to (9) describe the dynamics of the IMU motion, (10) describes the random-walk processes that model the biases' slowly time-varying nature, and (11) describes the fact that ${}^C_I\mathbf{T}$ remains constant, while the last line expresses the fact that the time offset between the camera and the IMU also remains constant. If the time offset is known to be time-varying, we can model it as a random-walk process by replacing the last line of the dynamics with $\dot{t}_d(t) = n_d(t)$, where $n_d(t)$ is a white Gaussian noise process, whose power spectral density expresses the variability of t_d.

Equations (7) to (12) describe the continuous-time evolution of the true states. For propagating the state estimates in a discrete-time implementation, we follow the approach described in Li and Mourikis (2013b). Specifically, for propagating the orientation from time instant t_k to t_{k+1}, we numerically integrate the differential equation

$${}^I_G\dot{\hat{\bar{\mathbf{q}}}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_\mathbf{g}(t_k)\right){}^I_G\hat{\bar{\mathbf{q}}}(t) \quad (14)$$

in the interval $t \in [t_k, t_{k+1}]$, assuming that $\boldsymbol{\omega}_m(t)$ is changing linearly between the samples received from the IMU at t_k and t_{k+1}. The velocity and position estimates are propagated by

$${}^G\hat{\mathbf{v}}_{k+1} = {}^G\hat{\mathbf{v}}_k + {}^G_I\hat{\mathbf{R}}(t_k)\,\hat{\mathbf{s}}_k + {}^G\mathbf{g}\Delta t \quad (15)$$

$${}^G\hat{\mathbf{p}}_{k+1} = {}^G\hat{\mathbf{p}}_k + {}^G\hat{\mathbf{v}}_k\Delta t + {}^G_I\hat{\mathbf{R}}(t_k)\,\hat{\mathbf{y}}_k + \frac{1}{2}{}^G\mathbf{g}\Delta t^2 \quad (16)$$

where $\Delta t = t_{k+1} - t_k$, and

$$\hat{\mathbf{s}}_k = \int_{t_k}^{t_{k+1}} {}^{I_k}_I\hat{\mathbf{R}}(\tau)\left(\mathbf{a}_m(\tau) - \hat{\mathbf{b}}_\mathbf{a}(t_k)\right)d\tau \quad (17)$$

$$\hat{\mathbf{y}}_k = \int_{t_k}^{t_{k+1}}\int_{t_k}^{s} {}^{I_k}_I\hat{\mathbf{R}}(\tau)\left(\mathbf{a}_m(\tau) - \hat{\mathbf{b}}_\mathbf{a}(t_k)\right)d\tau\, ds \quad (18)$$

The above integrals are computed using Simpson integration, assuming a linearly changing $\mathbf{a}_m$ in the interval $[t_k, t_{k+1}]$. Besides the IMU position, velocity, and orientation, all other state estimates remain unchanged during propagation.

In addition to the state estimate, the EKF propagates the state covariance matrix as follows:

$$\mathbf{P}(t_{k+1}) = \boldsymbol{\Phi}(t_{k+1}, t_k)\,\mathbf{P}(t_k)\,\boldsymbol{\Phi}(t_{k+1}, t_k)^{\mathrm{T}} + \mathbf{Q}_d \quad (19)$$

where $\mathbf{P}$ is the state-covariance matrix, $\mathbf{Q}_d$ is the covariance matrix of the process noise, and $\boldsymbol{\Phi}(t_{k+1}, t_k)$ is the error-state transition matrix, given by

$$\boldsymbol{\Phi}(t_{k+1}, t_k) = \begin{bmatrix} \boldsymbol{\Phi}_I(t_{k+1}, t_k) & \mathbf{0}_{15\times 7} \\ \mathbf{0}_{7\times 15} & \mathbf{I}_{7\times 7} \end{bmatrix} \quad (20)$$

with $\boldsymbol{\Phi}_I(t_{k+1}, t_k)$ being the 15×15 error-state transition matrix for the IMU state, derived in Li and Mourikis (2012, 2013b).

4.3. EKF updates

We now describe how the camera measurements are employed for EKF updates. Note that, if no time offset existed (or, equivalently, if it was perfectly known a priori), the EKF update would present no difficulty. The complications arise from the fact that the image received by the filter at time t was in fact recorded at time $t + t_d$, where t_d is now a random variable.

Let us consider the observation of the ith feature in the image timestamped at t. Assuming an intrinsically calibrated camera, this is described by

$$\mathbf{z}_i(t) = \mathbf{h}\big({}^C\mathbf{p}_{f_i}(t+t_d)\big) + \mathbf{n}_i(t+t_d) \quad (21)$$

$$= \frac{1}{{}^Cz_i(t+t_d)}\begin{bmatrix} {}^Cx_i(t+t_d) \\ {}^Cy_i(t+t_d)\end{bmatrix} + \mathbf{n}_i(t+t_d) \quad (22)$$

where $\mathbf{h}(\cdot)$ is the perspective camera model, $\mathbf{n}_i$ is the measurement noise vector, modeled as zero-mean Gaussian with covariance matrix $\sigma_{im}^2\mathbf{I}_{2\times 2}$, and ${}^C\mathbf{p}_{f_i}(t+t_d)$ is the position of the feature with respect to the camera at the time the image was sampled:

$${}^C\mathbf{p}_{f_i}(t+t_d) = {}^C_I\mathbf{R}\,{}^I_G\mathbf{R}(t+t_d)\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I(t+t_d)\right) + {}^C\mathbf{p}_I \quad (23)$$

In this equation ${}^G\mathbf{p}_{f_i}$ is the position of the ith feature in the global frame, which in this section is assumed to be known.

To use $\mathbf{z}_i(t)$ for an EKF update, we must formulate the residual between the actual measurement and the measurement expected based on the filter's estimates (Maybeck, 1982):

$$\mathbf{r}_i = \mathbf{z}_i(t) - \mathbf{h}\big(\widehat{{}^C\mathbf{p}_{f_i}(t+t_d)}\big) \quad (24)$$

where $\widehat{{}^C\mathbf{p}_{f_i}(t+t_d)}$ denotes the estimate of ${}^C\mathbf{p}_{f_i}(t+t_d)$. To zero-order approximation (as dictated by the EKF paradigm), this estimate is given by

$$\widehat{{}^C\mathbf{p}_{f_i}(t+t_d)} = {}^C_I\hat{\mathbf{R}}\,{}^I_G\hat{\mathbf{R}}(t+\hat{t}_d)\left({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t+\hat{t}_d)\right) + {}^C\hat{\mathbf{p}}_I$$

The above equation shows that, in order to compute the residual $\mathbf{r}_i$, we must have access to the estimates of the state at time $t+\hat{t}_d$. Therefore, to process the measurement $\mathbf{z}_i(t)$, we propagate using the IMU measurements up to $t+\hat{t}_d$, at which point we compute $\mathbf{r}_i$, and perform an EKF update. For this update, the Jacobian of $\mathbf{h}\big({}^C\mathbf{p}_{f_i}(t+t_d)\big)$ with respect to the filter state is necessary. This is given by

$$\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d) = \begin{bmatrix}\mathbf{H}_{\boldsymbol{\theta},i} & \mathbf{H}_{\mathbf{p},i} & \mathbf{0}_{2\times 9} & \mathbf{H}_{\boldsymbol{\phi},i} & \mathbf{H}_{\mathbf{p}_c,i} & \mathbf{H}_{t_d,i}\end{bmatrix} \quad (25)$$

where the nonzero blocks are the Jacobians with respect to the IMU rotation, IMU position, camera-to-IMU rotation, camera-to-IMU translation, and time offset, respectively. These are computed as

$$\begin{aligned}
\mathbf{H}_{\boldsymbol{\theta},i} &= \mathbf{J}_i\,{}^C_I\hat{\mathbf{R}}\,{}^I_G\hat{\mathbf{R}}(t+\hat{t}_d)\lfloor({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t+\hat{t}_d))\times\rfloor \\
\mathbf{H}_{\mathbf{p},i} &= -\mathbf{J}_i\,{}^C_I\hat{\mathbf{R}}\,{}^I_G\hat{\mathbf{R}}(t+\hat{t}_d) \\
\mathbf{H}_{\boldsymbol{\phi},i} &= \mathbf{J}_i\,{}^C_I\hat{\mathbf{R}}\lfloor{}^I_G\hat{\mathbf{R}}(t+\hat{t}_d)({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t+\hat{t}_d))\times\rfloor \\
\mathbf{H}_{\mathbf{p}_c,i} &= \mathbf{J}_i \\
\mathbf{H}_{t_d,i} &= \mathbf{H}_{\boldsymbol{\theta},i}\,{}^I_G\hat{\mathbf{R}}^{\mathrm{T}}(t+\hat{t}_d)\,{}^I\hat{\boldsymbol{\omega}}(t+\hat{t}_d) + \mathbf{H}_{\mathbf{p},i}\,{}^G\hat{\mathbf{v}}_I(t+\hat{t}_d)
\end{aligned} \quad (26)$$

where $\mathbf{J}_i$ is the Jacobian of the perspective model,

$$\mathbf{J}_i = \left.\frac{\partial\mathbf{h}(\mathbf{f})}{\partial\mathbf{f}}\right|_{\mathbf{f}=\widehat{{}^C\mathbf{p}_{f_i}(t+t_d)}} = \frac{1}{{}^C\hat{z}_i}\begin{bmatrix}1 & 0 & -\frac{{}^C\hat{x}_i}{{}^C\hat{z}_i} \\ 0 & 1 & -\frac{{}^C\hat{y}_i}{{}^C\hat{z}_i}\end{bmatrix} \quad (27)$$

Note that all the matrices shown above are computed using the EKF state estimates available at time $t+\hat{t}_d$. In addition, the Jacobian with respect to the time offset, $\mathbf{H}_{t_d,i}$, requires an estimate of the rotational velocity vector, which is computed using the IMU measurements as ${}^I\hat{\boldsymbol{\omega}}(t+\hat{t}_d) = \boldsymbol{\omega}_m(t+\hat{t}_d) - \hat{\mathbf{b}}_{\mathbf{g}}$. We thus see that all the Jacobians can be computed in closed form, using quantities available to the filter at $t+\hat{t}_d$. Using the above expression for $\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)$, we can now proceed to carry out the EKF update. Specifically, the state and covariance matrix are updated as

$$\hat{\mathbf{x}}(t+\hat{t}_d) \leftarrow \hat{\mathbf{x}}(t+\hat{t}_d) + \mathbf{K}_i\mathbf{r}_i \quad (28)$$

$$\mathbf{P}(t+\hat{t}_d) \leftarrow \mathbf{P}(t+\hat{t}_d) - \mathbf{K}_i\mathbf{S}_i\mathbf{K}_i^{\mathrm{T}} \quad (29)$$

where

$$\mathbf{K}_i = \mathbf{P}(t+\hat{t}_d)\,\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)^{\mathrm{T}}\,\mathbf{S}_i^{-1}, \quad \text{with} \quad (30)$$

$$\mathbf{S}_i = \mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)\,\mathbf{P}(t+\hat{t}_d)\,\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)^{\mathrm{T}} + \sigma_{im}^2\mathbf{I} \quad (31)$$

If more than one feature is observed in the same image, their residuals can be processed in the same manner.

A few interesting comments can be made at this point. We start by noting that the camera measurement was recorded at time $t+t_d$, but it is being processed at $t+\hat{t}_d$. Since the estimate of the time offset will inevitably contain some error, the measurement will inevitably be processed at a slightly incorrect time instant. However, the EKF explicitly accounts for this fact. Specifically, since t_d is included in the estimated state vector, the filter keeps track of the uncertainty in $\hat{t}_d$, via the state-covariance matrix $\mathbf{P}$. Therefore, when computing the covariance matrix of the residual ($\mathbf{S}_i$ in (31)) the uncertainty in the time offset is explicitly modeled, and is accounted for in the computation of the state update. As a result, we are able to obtain both more accurate pose estimates and a better characterization of their uncertainty.

It is worth pointing out that, in some cases, the camera timestamps may be affected by a random-noise component ('jitter'), in addition to a systematic time offset. In the proposed formulation, it is straightforward to model these random effects in the estimator's equations. Specifically, if each image timestamp is affected by an independent, zero-mean, random error with standard deviation σ_t, then, instead of (31), the covariance matrix of the residual is computed as

$$\mathbf{S}_i = \mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)\,\mathbf{P}(t+\hat{t}_d)\,\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)^{\mathrm{T}} + \sigma_{im}^2\mathbf{I} + \sigma_t^2\mathbf{H}_{t_d,i}\mathbf{H}_{t_d,i}^{\mathrm{T}} \quad (32)$$

This modification makes it possible to model the additional timestamp uncertainty, and to account for it in the EKF update equations.

4.4. Implementation

The algorithm for concurrent pose estimation and temporal calibration described in the preceding sections can be

implemented in a number of different ways. We have opted for a multi-threaded approach, as it allows for increased flexibility and extensibility. The high-level architecture described here is employed both in map-based localization and in localization with unknown features, discussed in Section 5. In our implementation, each sensor's readings are managed by a separate thread. Two queues are maintained (one for the IMU measurements and one for the camera images), and each new measurement is timestamped and placed in the appropriate queue as it arrives.

One thread is used for implementing the EKF equations. This thread waits until the image queue has at least one image, at which point it begins using the measurements in the IMU queue to propagate the state estimate and its covariance matrix, as described in Section 4.2. If the timestamp of the first image in the image queue is t, then IMU measurements are used to propagate up to time $t+\hat{t}_d$. After propagation is completed, the EKF update is performed, as described in Section 4.3. We note here that, in general, the time instant $t+\hat{t}_d$ will fall between two consecutive sample times of the IMU. Therefore, we employ linear interpolation to obtain an inferred IMU measurement at $t+\hat{t}_d$, used for propagation as well as in computing the estimated rotational velocity in (26).

Finally, we point out that if the camera images are delayed relative to the IMU (i.e. if $t_d < 0$), the EKF thread will produce state estimates with a latency, as it waits until an image is available before processing the IMU measurements. If estimates of the current state are required at the IMU sample rate, these can be computed by a separate thread running in parallel. Specifically, this thread can use the latest available state estimate from the EKF, as well as all the measurements in the IMU queue, to compute the estimate for the current system state via propagation.

5. Motion estimation with unknown features

The preceding section describes time-offset estimation when the feature positions in the world are known a priori. In many cases, however, we are interested in motion estimation in previously unknown environments, for which it is not possible to have a feature map. Several algorithms have been developed for vision-aided inertial navigation in this type of application. Broadly, these methods use the visual measurements in one of two ways (Williams et al., 2011): on the one hand, feature-based methods include feature positions in the state vector being estimated, as in EKF-SLAM, and employ the feature observations directly for state updates (see e.g. Pinies et al., 2007; Jones and Soatto, 2011). On the other hand, pose-based methods do not include feature positions in the state vector, and instead maintain a state vector containing a number of poses (e.g. Roumeliotis et al., 2002; Bayard and Brugarolas, 2005; Mourikis and Roumeliotis, 2007; Shkurti et al., 2011; Li and Mourikis, 2012). In these methods, the feature measurements are first used to define constraints between two or more poses (e.g. to estimate the relative motion between consecutive images), and these constraints are subsequently used for filter updates.

As we explain in what follows, the proposed approach for time-offset estimation by explicitly including t_d in the filter state vector can be readily applied in both types of method.

5.1. Feature-based methods

In typical feature-based EKF algorithms (often referred to as EKF-SLAM algorithms) the state vector of the filter contains the current state of the IMU, as well as the positions of the features detected by the camera. In order to perform online spatial and temporal calibration, we here additionally include t_d and the camera-to-IMU transformation in the EKF state vector. Thus, this state vector is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^{\mathrm{T}} & {}^{C}_{I}\bar{\mathbf{q}}^{\mathrm{T}} & {}^{C}\mathbf{p}_I^{\mathrm{T}} & t_d & \mathbf{f}_1^{\mathrm{T}} & \mathbf{f}_2^{\mathrm{T}} & \cdots & \mathbf{f}_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

where $\mathbf{f}_i$, $i = 1, \ldots N$, are the features, which can be parameterized in a number of different ways (e.g. Cartesian (xyz) position, inverse depth (Montiel et al., 2006), homogeneous coordinates (Sola, 2010)).

With this augmented state vector, the feature measurements, $\mathbf{z}_i$, can be directly employed for EKF updates. The process followed in order to address the presence of the time offset t_d is analogous to that described in Section 4.3. Specifically, if a measurement $\mathbf{z}_i(t)$ is received, we employ the IMU measurements for propagation up to time $t+\hat{t}_d$, and perform an EKF update at that time. The residual (24) is computed, as well as its Jacobian with respect to the state, given by

$$\mathbf{H}_{\mathbf{x},i}^{\mathrm{SLAM}} = \begin{bmatrix} \mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d) & \mathbf{0} & \cdots & \mathbf{H}_{\mathbf{f},i}(t+\hat{t}_d) & \cdots & \mathbf{0} \end{bmatrix}$$

where $\mathbf{H}_{\mathbf{x},i}(t+\hat{t}_d)$ is defined in (25), and $\mathbf{H}_{\mathbf{f},i}(t+\hat{t}_d)$ is the Jacobian with respect to the ith feature state, whose exact form will depend on the chosen feature parameterization. With this Jacobian, the EKF update can proceed according to the standard EKF equations, with no further modification.

5.2. Pose-based methods

In pose-based EKF algorithms, the state vector typically contains the current IMU state, as well as M poses (with $M \geq 1$), corresponding to time instants images were recorded. For instance, in Mourikis and Roumeliotis (2007), Shkurti et al. (2011), Kottas et al. (2012), and Li and Mourikis (2012) the state vector is formulated as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^{\mathrm{T}} & \mathbf{c}_1^{\mathrm{T}} & \cdots & \mathbf{c}_M^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{33}$$

where $\mathbf{c}_j$, $j = 1, ..., M$, is the camera pose at the time the jth image was recorded:

$$\mathbf{c}_j = \begin{bmatrix} {}^{C}_{G}\mathbf{q}_j^{\mathrm{T}} & {}^{G}\mathbf{p}_{C_j}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{34}$$

Every time a new image is received, the state vector is augmented to include a copy of the current camera pose, and the oldest pose is removed. The features are tracked for up to M images, and are used for deriving constraints between the camera poses.

In order to estimate the extrinsic calibration and time offset between the camera and IMU in this setting, we can include these parameters in the state vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^{\mathrm{T}} & {}^{C}_{I}\bar{\mathbf{q}}^{\mathrm{T}} & {}^{C}\mathbf{p}_I^{\mathrm{T}} & t_d & \mathbf{c}_1^{\mathrm{T}} & \cdots & \mathbf{c}_M^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{35}$$

To account for these additional parameters, one can compute the Jacobians of the feature measurements with respect to them, similarly to what was described in the previous cases. However, in pose-based methods, an alternative, and simpler, approach exists. Specifically, the only filter operation that needs to be changed, compared to the original methods, is state augmentation: when a new image is received with timestamp t, we augment the state vector to include an estimate of the camera pose at time $t+t_d$ (instead of time t, as in the original method). We therefore use the IMU measurements to propagate up to $t+\hat{t}_d$, at which point we augment the state with the estimate of the camera pose at $t+t_d$:

$$\hat{\mathbf{c}}_{new} = \begin{bmatrix} \widehat{{}^{C}_{G}\mathbf{q}(t+t_d)} \\ \widehat{{}^{G}\mathbf{p}_C(t+t_d)} \end{bmatrix} = \begin{bmatrix} {}^{C}_{I}\hat{\mathbf{q}} \otimes {}^{I}_{G}\hat{\bar{\mathbf{q}}}(t+\hat{t}_d) \\ {}^{G}\hat{\mathbf{p}}_I(t+\hat{t}_d) + {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{\mathrm{T}}\, {}^{I}\hat{\mathbf{p}}_C \end{bmatrix}$$

The filter covariance matrix is also augmented, as

$$\mathbf{P}(t+\hat{t}_d) \leftarrow \begin{bmatrix} \mathbf{P}(t+\hat{t}_d) & \mathbf{P}(t+\hat{t}_d)\,\mathbf{J}_{new}^{\mathrm{T}} \\ \mathbf{J}_{new}\mathbf{P}(t+\hat{t}_d) & \mathbf{J}_{new}\mathbf{P}(t+\hat{t}_d)\,\mathbf{J}_{new}^{\mathrm{T}} \end{bmatrix} \tag{36}$$

where $\mathbf{J}_{new}$ is the Jacobian of $\mathbf{c}_{new}$ with respect to the state vector. This matrix has the following structure:

$$\mathbf{J}_{new} = \begin{bmatrix} \mathbf{J}_I & \mathbf{J}_{IC} & \mathbf{J}_t & \mathbf{0} \end{bmatrix}$$

where $\mathbf{J}_I$ is the Jacobian with respect to the IMU state,

$$\mathbf{J}_I = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 9} \\ -\lfloor {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{\mathrm{T}}\, {}^{I}\hat{\mathbf{p}}_C \times \rfloor & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 9} \end{bmatrix}$$

$\mathbf{J}_{IC}$ is the Jacobian with respect to the camera-to-IMU transformation,

$$\mathbf{J}_{IC} = \begin{bmatrix} {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{\mathrm{T}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{\mathrm{T}} \end{bmatrix}$$

and $\mathbf{J}_t$ is the Jacobian with respect to t_d,

$$\mathbf{J}_t = \begin{bmatrix} {}^{I}_{G}\hat{\mathbf{R}}^{\mathrm{T}}(t+\hat{t}_d)\,{}^{I}\hat{\boldsymbol{\omega}}(t+\hat{t}_d) \\ {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{\mathrm{T}}\lfloor {}^{I}\hat{\boldsymbol{\omega}}(t+\hat{t}_d)\times \rfloor\, {}^{I}\hat{\mathbf{p}}_C + {}^{G}\hat{\mathbf{v}}_I(t+\hat{t}_d) \end{bmatrix} \tag{37}$$

Compared to the methods of Mourikis and Roumeliotis (2007) and Li and Mourikis (2012), the above equations differ in that additional Jacobians are computed with respect to the parameters of ${}^{C}_{I}\mathbf{T}$, and with respect to the time offset t_d. This is the only change that is needed: after the augmentation has been performed in this fashion, the feature measurements can be used for EKF updates as in Mourikis and Roumeliotis (2007) and Li and Mourikis (2012), with no further alterations. Since the dependence of the camera poses on t_d has been modeled (via the Jacobian $\mathbf{J}_t$), when the EKF update is performed t_d will also be updated, as normal in the EKF.

As a final remark, we note that up to now, we have only discussed EKF-based methods, and we have shown that the time offset between the camera and IMU can be explicitly included in the state vector and estimated online. The key to achieving this is to model the dependence of the measurements on the time offset, by computing the appropriate Jacobians with respect to t_d (see (26) and (37)). It is important to point out that a similar approach can be followed in methods that employ iterative minimization for pose estimation, either in batch (Dellaert and Kaess, 2006) or in incremental form (Kaess et al., 2008). By including t_d in the estimated state vector, and computing Jacobians with respect to it, we can estimate it jointly with all other states of interest.

6. Identifiability analysis

In the preceding sections we presented online algorithms for estimating the time offset between the camera and IMU. However, for these algorithms to be able to obtain meaningful results, this time offset must be identifiable (equivalently, observable)[2] given the available measurements. In this section, we examine the identifiability of t_d in the case where the feature positions, time offset, camera-to-IMU transformation, and IMU biases are all unknown, and need to be estimated along with the IMU trajectory. We show that even in this most general (and most challenging) case, t_d is in general locally identifiable (Bellman and Astrom, 1970). Clearly, if more information were available (e.g. known feature positions, and/or known ${}^{C}_{I}\mathbf{T}$) the local identifiability of t_d would also hold.

6.1. Camera measurement model

We begin by examining the type of information provided by the camera measurements. When a camera is observing (a sufficient number of) unknown features, the feature measurements can be used to compute (i) the orientation of the camera with respect to the initial camera frame, and (ii) the position of the camera with respect to the initial camera frame, up to an unknown scale factor s (Hartley and Zisserman, 2000). That is, by processing the visual measurements in the time interval $[0, t]$, we can compute a measurement of the camera rotation in the time interval $[t_d, t+t_d]$, and a scaled measurement of the camera translation in the same time interval:

$$\mathbf{R}_c(t) = {}^{C_o}_{C}\mathbf{R}(t+t_d)\, e^{\lfloor \mathbf{n}_{cr}(t)\times \rfloor} \tag{38}$$

$$\mathbf{p}_c(t) = s^{C_o}\mathbf{p}_C(t+t_d) + \mathbf{n}_{cp}(t) \tag{39}$$

where $\mathbf{n}_{cr}(t)$ and $\mathbf{n}_{cp}(t)$ are noise vectors, and $\{C_o\}$ is the initial camera frame, that is, the camera frame at the time instant t_d. Note that equivalently, we can state that the visual measurements can be used to compute (i) a measurement, $\boldsymbol{\omega}_c$, of the camera's rotational velocity, and (ii) a scaled measurement, $\mathbf{v}_c$, of the camera's translational velocity,

$$\boldsymbol{\omega}_c(t) = {}^{C}\boldsymbol{\omega}(t+t_d) + \mathbf{n}_{c\omega}(t) \tag{40}$$

$$\mathbf{v}_c(t) = s\,{}^{C_o}\mathbf{v}_C(t+t_d) + \mathbf{n}_{cv}(t) \tag{41}$$

where $\mathbf{n}_{c\omega}(t)$ and $\mathbf{n}_{cv}(t)$ are the measurement noise vectors.

We will carry out our analysis by using the measurement models described above, instead of the 'raw' feature measurements. We stress that (38)–(39) or (40)–(41) contain all the information that the visual feature measurements provide for the camera motion, and therefore we can use these, instead of the feature measurements, for our analysis. The advantage of using the 'abstracted' camera measurement models shown above (an approach also employed in Kelly and Sukhatme, 2011) is that their use leads to a significantly simplified analysis.

6.2. Overview of the approach

In previous work, the question of which states of the vision-aided inertial navigation system can be estimated and under what conditions has been addressed by performing an observability analysis. Specifically, Jones and Soatto (2011) and Kelly and Sukhatme (2011) examined the case where the feature positions, camera-to-IMU transformation, and IMU biases are unknown, but the time offset between the camera and IMU is known. For this scenario it was shown that, in general, the following quantities are (locally weakly) observable:

(O1) The trajectory expressed in the initial camera frame.
(O2) The gravity vector expressed in the initial camera frame.
(O3) The gyroscope and accelerometer biases.
(O4) The camera-to-IMU transformation.
(O5) The positions of the features expressed in the camera frame.

The quantities O1–O5 are *generally* observable, that is, unless the camera follows 'degenerate' trajectories such as constant-velocity motion or rotation about a single axis (see Jones and Soatto, 2011; Kelly and Sukhatme, 2011, for a complete characterization). On the other hand, four degrees of freedom are always unobservable: three corresponding to the system's position in the global frame, and one to the rotation about gravity (i.e. the yaw).

The approach we follow in order to examine the local identifiability of t_d can be explained, at a high level, as follows. We start by defining a vector $\boldsymbol{\xi}$, which contains the time offset t_d, as well as all the additional quantities needed in order to compute O1–O5. Specifically, we define $\boldsymbol{\xi}$ as

$$\boldsymbol{\xi} = \begin{bmatrix} {}^{C_o}\mathbf{v}_o^{\mathrm{T}} & {}^{C_o}\mathbf{g}^{\mathrm{T}} & \mathbf{b}_{\mathbf{g}}^{\mathrm{T}} & \mathbf{b}_{\mathbf{a}}^{\mathrm{T}} & {}^{C}\mathbf{p}_I^{\mathrm{T}} & {}^{C}_{I}\mathbf{q}^{\mathrm{T}} & t_d & s \end{bmatrix}^{\mathrm{T}}$$

where ${}^{C_o}\mathbf{v}_o$ is the IMU velocity at time instant t_d, expressed in $\{C_o\}$, and ${}^{C_o}\mathbf{g}$ is the gravity vector expressed in the same frame. We stress that the elements of $\boldsymbol{\xi}$ describe all the potentially observable quantities in the system, and none of the quantities that are known to be unobservable. To see why, we first note that $\boldsymbol{\xi}$ contains the time offset t_d, as well as O2, O3, and O4 explicitly. Moreover, using the elements of $\boldsymbol{\xi}$ and the measurements, we can compute the velocity of the camera at any time instant, expressed in the frame $\{C_o\}$ (as shown in (47)). Integrating this velocity yields the camera trajectory in $\{C_o\}$, which is quantity O1. Finally, once the trajectory of the camera is known, then the features' positions (quantity O5) can be computed by triangulation using the camera measurements.

The above discussion shows that if $\boldsymbol{\xi}$ is locally identifiable given the measurements, then t_d and O1–O5 are locally weakly observable. To determine whether $\boldsymbol{\xi}$ is locally identifiable, we employ the camera and IMU measurements, along with the system dynamics, to derive constraint equations that $\boldsymbol{\xi}$ must satisfy. These constraints have the general form

$$\mathbf{c}_i\left(\boldsymbol{\xi}, \mathbf{z}_{[0,t]}, \boldsymbol{\omega}_{m[0,t]}, \mathbf{a}_{m[0,t]}\right) = \mathbf{0} \tag{42}$$

where $\mathbf{z}_{[0,t]}, \boldsymbol{\omega}_{m,[0,t]}$, and $\mathbf{a}_{m,[0,t]}$ are the camera, gyroscope, and accelerometer measurements in the time interval $[0, t]$. Once these constraints are formulated, we can check the local identifiability of $\boldsymbol{\xi}$ by examining the rank of the Jacobian matrices of these constraints with respect to $\boldsymbol{\xi}$.

Since the conditions for the observability of O1–O5 have been derived in previous work, we focus here on examining the identifiability of t_d. The main result, proven in Lemma 6.2, is that t_d is locally identifiable, except in a small number of degenerate motion cases that can be characterized in detail. This result validates the use of the EKF-based algorithms described in the preceding sections. Moreover, it provides us with a way to examine the observability of the entire system (including t_d and O1–O5), by leveraging the results of Jones and Soatto (2011) and Kelly and Sukhatme (2011). Specifically, when t_d is identifiable, we can use the measurements alone to determine its value; therefore in terms of observability, the system can be treated as one with a known t_d in this case. Put differently, when t_d is identifiable, the observability properties of O1–O5 are identical to those derived in previous work for the known-t_d case. Thus, to certify that all the quantities of interest (t_d and O1–O5) are observable, one can use Lemma 6.2 to verify that t_d is locally identifiable, and then employ the results of Jones and Soatto (2011) and Kelly and Sukhatme (2011) to verify the observability of O1–O5.

Our analysis of the identifiability of t_d consists of two steps. First, we show that, even if only the rotational components of the motion (i.e. rotational constraints) are considered, the time offset is in general locally identifiable, and we obtain a concrete description of the cases that cause loss of identifiability. We then show that if all the available information is utilized (which is the case in the algorithms

described in the preceding sections), the set of degenerate cases is further restricted. The details of the analysis are presented in what follows.

6.3. Derivation of the constraints for $\boldsymbol{\xi}$

We here ignore the noise (as standard in an identifiability/observability analysis) and derive two constraint equations: one based on the rotational velocity measurements, and one based on the accelerometer and (scaled) velocity measurements. We begin by using (40) and (5), and the identity ${}^{C}\boldsymbol{\omega}(t) = {}^{C}_{I}\mathbf{R}\,{}^{I}\boldsymbol{\omega}(t)$, to obtain

$$\boldsymbol{\omega}_m(t) = {}^{C}_{I}\mathbf{R}^{\mathrm{T}}\boldsymbol{\omega}_c(t-t_d)+\mathbf{b_g} \tag{43}$$

which we can write in the form of (42) as follows:

$$\mathbf{c}_1(\boldsymbol{\xi},t) = {}^{C}_{I}\mathbf{R}^{\mathrm{T}}\boldsymbol{\omega}_c(t-t_d)+\mathbf{b_g}-\boldsymbol{\omega}_m(t) = \mathbf{0} \tag{44}$$

This constraint involves the known functions of the IMU and camera rotational velocity measurements, as well as the unknown parameters t_d, ${}^{C}_{I}\mathbf{R}$ (equivalent to ${}^{C}_{I}\bar{\mathbf{q}}$), and $\mathbf{b_g}$.

To obtain the second constraint, we express $\mathbf{v}_c(t)$ as a function of $\boldsymbol{\xi}$, $\mathbf{R}_c(t)$, and $\mathbf{a}_m(t)$. We start by writing ${}^{C_o}\mathbf{v}_C(t+t_d)$ as

$$\begin{aligned} {}^{C_o}\mathbf{v}_C(t+t_d) &= {}^{C_o}_{G}\mathbf{R}\,{}^{G}\mathbf{v}_C(t+t_d) && (45)\\ &= {}^{C_o}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_I(t+t_d)-{}^{G}_{C}\dot{\mathbf{R}}(t+t_d)\,{}^{C}\mathbf{p}_I\right) && (46)\end{aligned}$$

Next, we note that the IMU velocity at time instant $t+t_d$ can be computed as

$${}^{G}\mathbf{v}_I(t+t_d) = {}^{G}\mathbf{v}_I(t_d)+\int_0^t {}^{G}_{I}\mathbf{R}(\tau+t_d)\,{}^{I}\mathbf{a}(\tau+t_d)\,d\tau$$

Using (6), we can write the last equation as

$$\begin{aligned} {}^{G}\mathbf{v}_I(t+t_d) = {}^{G}\mathbf{v}_I(t_d)&+\int_0^t {}^{G}_{I}\mathbf{R}(\tau+t_d)\,\mathbf{a}_m(\tau+t_d)\,d\tau\\ &+{}^{G}\mathbf{g}\,t-\int_0^t {}^{G}_{I}\mathbf{R}(\tau+t_d)\,d\tau\,\mathbf{b_a}\end{aligned}$$

Substituting the last equation in (46), and using the identity $\mathbf{R}_c(\tau) = {}^{C_o}_{I}\mathbf{R}(\tau+t_d)\,{}^{C}_{I}\mathbf{R}^{\mathrm{T}}$ (see (38)), we obtain

$$\begin{aligned} {}^{C_o}\mathbf{v}_C(t+t_d) = {}^{C_o}\mathbf{v}_o+{}^{C_o}\mathbf{g}\,t&+\int_0^t \mathbf{R}_c(\tau)\,{}^{C}_{I}\mathbf{R}\,\mathbf{a}_m(\tau+t_d)\,d\tau\\ &-\int_0^t \mathbf{R}_c(\tau)\,d\tau\,{}^{C}_{I}\mathbf{R}\,\mathbf{b_a}-\dot{\mathbf{R}}_c(t)\,{}^{C}\mathbf{p}_I \end{aligned} \tag{47}$$

where we have used the notation ${}^{C_o}\mathbf{v}_o = {}^{C_o}_{G}\mathbf{R}\,{}^{G}\mathbf{v}_I(t_d)$. Finally, substitution in (41) yields the following constraint (ignoring the noise):

$$\begin{aligned} \mathbf{c}_2(\boldsymbol{\xi},t) = s\Bigg({}^{C_o}\mathbf{v}_o+{}^{C_o}\mathbf{g}\,t&+\int_0^t \mathbf{R}_c(\tau)\,{}^{C}_{I}\mathbf{R}\,\mathbf{a}_m(\tau+t_d)\,d\tau\\ &-\int_0^t \mathbf{R}_c(\tau)\,d\tau\,{}^{C}_{I}\mathbf{R}\,\mathbf{b_a}-\dot{\mathbf{R}}_c(t)\,{}^{C}\mathbf{p}_I\Bigg)-\mathbf{v}_c(t) = \mathbf{0}\end{aligned} \tag{48}$$

This equation is the second constraint we sought: it involves terms that are known via the IMU and camera measurements, as well as the elements of $\boldsymbol{\xi}$. In what follows, we show how we can employ (44) and (48) to determine the identifiability of t_d.

6.4. Identifiability of t_d based on the rotational-velocity constraint

We now prove the following result.

Lemma 6.1. *The time offset t_d is locally identifiable based on the rotational constraint* (44) *alone, if no vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ exist, such that the rotational velocity of the IMU satisfies the following differential equation:*

$${}^{I}\dot{\boldsymbol{\omega}}(t) = \lfloor \mathbf{k}_2\times\rfloor\,{}^{I}\boldsymbol{\omega}(t)+\mathbf{k}_1 \tag{49}$$

Proof. To examine the local identifiability of t_d based on the constraint (44), we compute the derivative of $\mathbf{c}_1(\boldsymbol{\xi},t)$ with respect to the elements of $\boldsymbol{\xi}$ that appear in it:

$$\mathbf{D}_1(t) = \begin{bmatrix} \frac{\partial\,\mathbf{c}_1}{\partial\,\mathbf{b_g}} & \frac{\partial\,\mathbf{c}_1}{\partial\,\tilde{\boldsymbol{\phi}}} & \frac{\partial\,\mathbf{c}_1}{\partial\,t_d}\end{bmatrix} \tag{50}$$

where

$$\frac{\partial\,\mathbf{c}_1}{\partial\,\mathbf{b_g}} = \mathbf{I}_{3\times 3} \tag{51}$$

$$\frac{\partial\,\mathbf{c}_1}{\partial\,\tilde{\boldsymbol{\phi}}} = -\lfloor {}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\boldsymbol{\omega}_c(t-t_d)\times\rfloor \tag{52}$$

$$\frac{\partial\,\mathbf{c}_1}{\partial\,t_d} = -{}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\dot{\boldsymbol{\omega}}_c(t-t_d) \tag{53}$$

Note that, since (44) must hold for any t, we can generate an infinite number of constraints, by choosing different values of t. A sufficient condition for $\mathbf{b_g}$, ${}^{C}_{I}\bar{\mathbf{q}}$, and t_d to be locally identifiable based on these constraints is that there exists a set of time instants, $\mathcal{S} = \{t_1, t_2, \ldots t_s\}$, such that the matrix

$$\begin{bmatrix} \mathbf{D}_1(t_1)\\ \mathbf{D}_1(t_2)\\ \vdots\\ \mathbf{D}_1(t_s)\end{bmatrix} \tag{54}$$

has full column rank (Doren et al., 2009). Equivalently, a sufficient condition is that there exists no nonzero vector $\mathbf{k} = [\mathbf{k}_1^{\mathrm{T}}\ \ \mathbf{k}_2^{\mathrm{T}}\ \ k_3]^{\mathrm{T}}$ such that, for all $t > 0$, the condition $\mathbf{D}_1(t)\,\mathbf{k} = \mathbf{0}$ holds. Note that, since we are interested in detecting cases in which t_d is not identifiable, we can restrict our attention to the vectors $\mathbf{k}$ in which k_3 is nonzero. Thus, we can set $k_3 = 1$ (the scaling of $\mathbf{k}$ is arbitrary). Using (50)–(53), the sufficient condition for t_d to be locally identifiable is that there exist no vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ such that

$$\begin{aligned} &\mathbf{k}_1-\lfloor {}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\boldsymbol{\omega}_c(t-t_d)\times\rfloor\,\mathbf{k}_2-{}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\dot{\boldsymbol{\omega}}_c(t-t_d) = \mathbf{0}\Rightarrow\\ &\mathbf{k}_1+\lfloor \mathbf{k}_2\times\rfloor\,{}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\boldsymbol{\omega}_c(t-t_d)-{}^{C}_{I}\mathbf{R}^{\mathrm{T}}\,\dot{\boldsymbol{\omega}}_c(t-t_d) = \mathbf{0},\forall t>0\end{aligned}$$

Using the identity ${}^C_I\mathbf{R}^{\mathrm{T}}\,\boldsymbol{\omega}_c(t-t_d) = {}^I\boldsymbol{\omega}(t)$ in the above equation yields (49).

□

Note that Lemma 6.1 provides a sufficient condition for the local identifiability of t_d: if (49) does not hold for any $\mathbf{k}_1$ and $\mathbf{k}_2$, then t_d is locally identifiable. We can in fact show that this condition is also a necessary one, by showing that if it is not met, there exists at least one indistinguishable family of solutions for t_d and the remaining unknown parameters. To this end, we start by noting that if (49) holds for some $\mathbf{k}_1$ and $\mathbf{k}_2$, then ${}^I\boldsymbol{\omega}(t)$ is given by

$$ {}^I\boldsymbol{\omega}(t) = e^{\lfloor \mathbf{k}_2 \times \rfloor t}\mathbf{k}_o + \int_0^t e^{\lfloor \mathbf{k}_2 \times \rfloor (t-\tau)}\, d\tau \cdot \mathbf{k}_1 \qquad (55) $$

where $\mathbf{k}_o$ is the initial value of the rotational velocity. We can now prove, by substitution in (5) and (40), that for any scalar δ, the sets $\{t_d, {}^C_I\mathbf{R}, \mathbf{b_g}, {}^I\boldsymbol{\omega}(t)\}$ and $\{t'_d, {}^C_I\mathbf{R}', \mathbf{b}'_\mathbf{g}, {}^I\boldsymbol{\omega}'(t)\}$, where

$$ \begin{aligned} t'_d &= t_d + \delta \\ {}^C_I\mathbf{R}' &= {}^C_I\mathbf{R}e^{-\lfloor \mathbf{k}_2 \times \rfloor \delta} \\ \mathbf{b}'_\mathbf{g} &= \mathbf{b_g} + \int_0^\delta e^{\lfloor \mathbf{k}_2 \times \rfloor (\delta-\tau)}\, d\tau \mathbf{k}_1 \\ {}^I\boldsymbol{\omega}'(t) &= {}^I\boldsymbol{\omega}(t) - \int_0^\delta e^{\lfloor \mathbf{k}_2 \times \rfloor (\delta-\tau)}\, d\tau \mathbf{k}_1 \end{aligned} $$

yield exactly the same measurements $\boldsymbol{\omega}_m(t)$ and $\boldsymbol{\omega}_c(t)$, for all t. This means that t_d and t'_d are indistinguishable, and thus t_d is not locally identifiable.

An important question to answer is whether the cases in which t_d becomes unidentifiable are practically relevant. Examination of (55) shows that this general functional form for ${}^I\boldsymbol{\omega}(t)$ encompasses the cases of no rotation, when $\mathbf{k}_0 = \mathbf{k}_1 = \mathbf{0}$, and of constant rotational velocity, for example when $\mathbf{k}_1 = \mathbf{k}_2 = \mathbf{0}$ (in fact, in these cases the indistinguishable families contain more than one free parameter). Besides these two cases, which can potentially arise in real-world trajectories, we believe that all other degenerate situations are unlikely to occur in practice. We should note however, that the cases of zero or constant rotational velocity result in loss of observability even if the time offset between the camera and IMU is perfectly known (Jones and Soatto, 2011; Kelly and Sukhatme, 2011). In this sense, we see that having an unknown t_d does not appear to cause loss of observability in additional, practically significant situations.

6.5. Identifiability of t_d based on both constraints

In the preceding section we only considered the rotational constraints provided by the camera and IMU. We now also take into account the constraints arising from the velocity and acceleration measurements, and show that the set of degenerate motion cases that lead to an unidentifiable t_d can be restricted further. Specifically, we prove the following result.

Lemma 6.2. *The time offset between the camera and IMU is locally identifiable, if no* $\mathbf{k}_1$, $\mathbf{k}_2$, k_3, $\mathbf{k}_4$, $\mathbf{k}_5$, *and* $\mathbf{k}_6$ *exist, such that (i) the rotational velocity of the IMU satisfies the differential equation* (49)*, and (ii) the accelerometer measurements satisfy the differential equation*

$$ \begin{aligned} \dot{\mathbf{a}}_m(t) = {} & (\lfloor \mathbf{k}_2 \times \rfloor - k_3\mathbf{I}_3)\,\mathbf{a}_m(t) - \mathbf{k}_4 \\ & - \left(\lfloor {}^I\boldsymbol{\omega}(t) \times \rfloor^2 + \lfloor {}^I\dot{\boldsymbol{\omega}}(t) \times \rfloor\right)\mathbf{k}_5 - {}^{I_o}_I\mathbf{R}(t)^{\mathrm{T}}\,\mathbf{k}_6 \end{aligned} \qquad (56) $$

Proof. The proof of this result follows a course analogous to that of Lemma 6.1, and is given in the Appendix. □

The above result shows that, in general, t_d is locally identifiable, and it may become unidentifiable only when both the orientation and the position of the platform follow specific motion patterns. As discussed, equation (49) yields only few critical cases for the orientation that are likely to occur in practice. Similarly, by examining the nature of the solutions to (56) for different choices of $\mathbf{k}_i$, we find only few cases of practical significance. The most important one is that of constant accelerometer measurements, which can arise for instance when the platform is accelerating at a constant rate with no rotation, or when it is moving at constant velocity and rotations occur only about the direction of gravity. However, these scenarios would result in loss of observability even if t_d was perfectly known. Thus, we once again see that estimating t_d online does not appear to result in any new degenerate trajectories with practical significance.

As a final remark, we point out that Lemma 6.2 provides a means of certifying the local identifiability of t_d only. As discussed in Section 6.2, the identifiability/observability of the remaining states must be checked separately. Situations can arise where t_d is locally identifiable, but some of the states O1–O5 defined in Section 6.2 are not. For example, if the rotational velocity of the platform is given by ${}^I\boldsymbol{\omega}(t) = [\sin(t) \quad 0 \quad 0]^{\mathrm{T}}$, t_d is locally identifiable, as ${}^I\boldsymbol{\omega}(t)$ does not satisfy an equation of the form (49). However, since the rotation happens only about the axis $[1 \quad 0 \quad 0]^{\mathrm{T}}$, in this case the orientation of the camera with respect to the IMU is not identifiable (Mirzaei and Roumeliotis, 2008; Jones and Soatto, 2011; Kelly and Sukhatme, 2011). Thus, as explained in Section 6.2, in order to certify that the entire system is locally weakly observable, one should certify that (49) and (56) do not simultaneously hold, *and* that none of the critical trajectories identified in Jones and Soatto (2011) and Kelly and Sukhatme (2011) occurs.

7. Experiments

7.1. Real-world experiments

Our real-world experiments involve indoor localization in an office environment, and outdoor localization during

Fig. 2. The area where the indoor experiment took place.

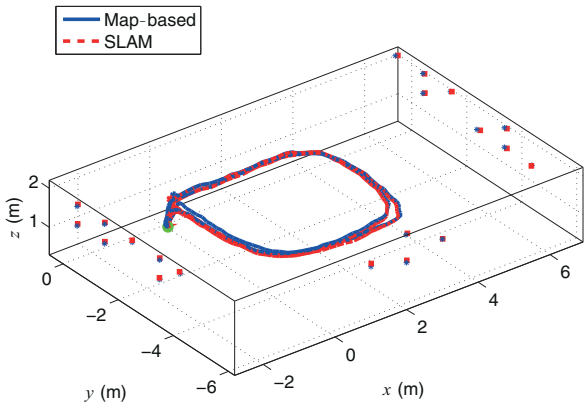

Fig. 3. Indoor experiment: trajectory estimates and feature positions. The blue line is the map-based trajectory estimate and the blue asterisks represent the known LED feature positions. The red dashed line is the SLAM trajectory estimate, and the red dots are the SLAM estimates for the positions of the LEDs.

urban driving. The visual–inertial system used in our experiments consists of a PointGrey Bumblebee2 stereo pair (only a single camera was used) and an Xsens MTI-G unit. The IMU reported inertial measurements at 100 Hz, while the camera captured images at 20 Hz.

7.1.1. Map-based indoor localization. The lab environment where the indoor-localization experiment took place is shown in Figure 2. In this area 20 blue LED lights with accurately known positions exist, and these are used as the mapped visual features for localization. During the experiment, the sensor platform started from a known initial position, and was moved in two loops around the room, returning to its initial location after each one. Since no high-precision ground truth was otherwise available, this motion pattern gives us three time instants in the trajectory, for which the estimation errors can be computed. Figure 3 shows the trajectory estimated by the algorithm described in Section 4 (blue line) and the positions of the mapped

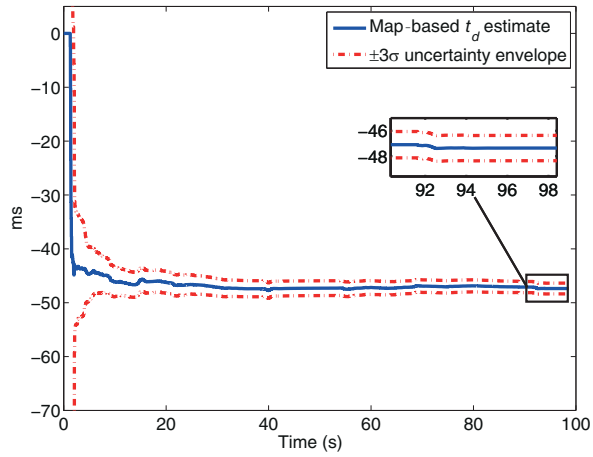

Fig. 4. Map-based experiment: time-offset estimate and corresponding $\pm 3\sigma$ uncertainty envelope computed from the covariance matrix reported by the estimator.

features (blue asterisks). For the known positions in the trajectory, the maximum estimation error was found to be 4.6 cm, which is commensurate with the covariance matrix reported by the filter at these locations. In Figure 4 we plot the estimate of the time offset, t_d, as well as the uncertainty envelope defined by ± 3 times the standard deviation reported by the EKF. We can see that within the first few seconds the estimate converges very close to its final value, and that the uncertainty in the estimate drops rapidly. We point out that the standard deviation of t_d at the end of the experiment is only 0.40 ms, showing the high precision attainable by the proposed online estimation method.

To examine the effects of online time-offset estimation on the reported estimates and their precision, we re-processed the same dataset, but using the final estimate for t_d as an input and disabling its online estimation. In Figure 5(a) we plot the difference of the trajectory estimates computed by the two methods (online estimation of t_d vs a priori known t_d), while Figure 5(b) shows the standard deviation reported by the two approaches. These plots demonstrate that performing estimation of t_d online yields results that are almost identical to those we would obtain if t_d was known in advance. We thus see that using the proposed online approach for determining t_d, instead of an offline calibration procedure, would result in no significant loss of performance.

7.1.2. EKF-SLAM. To test the performance of the online estimation of t_d in EKF-SLAM, we used the same dataset as for map-based localization, which provides a baseline for comparison. The features used for EKF-SLAM consisted of the 20 LEDs for which ground truth is available, as well as any additional features that were detected in the images via the Shi-Tomasi algorithm (Shi and Tomasi, 1994). Once the LED features were first detected and initialized, they were kept in the state vector for the remainder of the experiment. The re-observations of these 'persistent' features

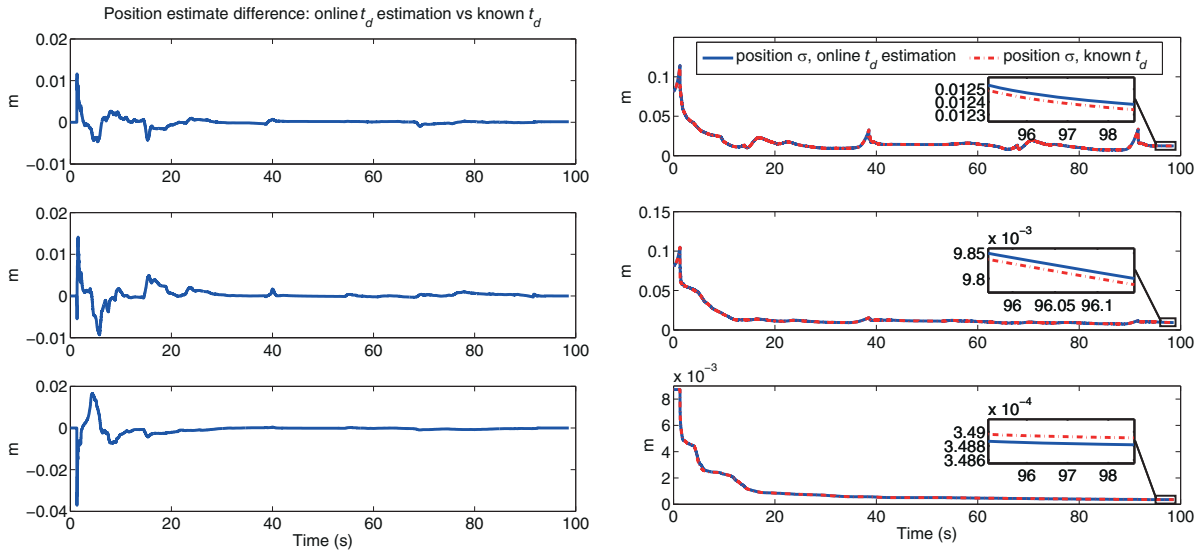

Fig. 5. Map-based localization: comparison of concurrent localization and t_d estimation vs localization with known t_d. (a) The difference in the position estimates computed in the two cases. (b) The filter's reported standard deviation in the two cases.

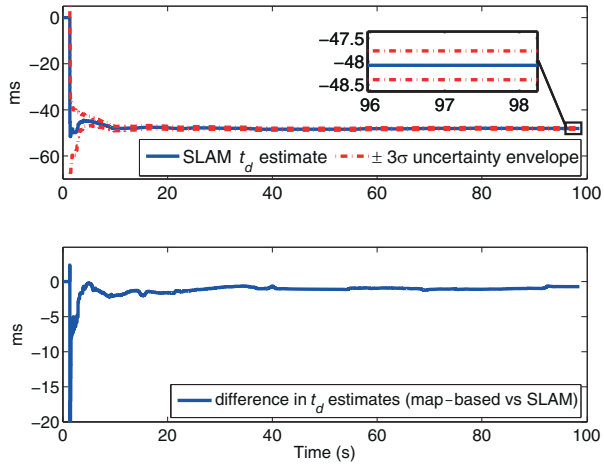

Fig. 6. Top plot: t_d estimate computed by EKF-SLAM, and the corresponding $\pm 3\sigma$ uncertainty envelope. Bottom plot: difference in the t_d estimates of map-based and EKF-SLAM estimation.

after each loop provided correction of position drift. On the other hand, each of the Shi-Tomasi features was kept in the state vector only for as long as it was visible after its initialization. When the feature dropped out of the camera field of view, it was removed from the state, to maintain the computational cost of the algorithm within real-time constraints. On average, in each image 65 'temporary' Shi-Tomasi features were tracked.

For the SLAM features the inverse-depth parametrization is used initially, and it is then converted to the Cartesian (xyz) parametrization to reduce computation as suggested by Civera et al. (2008). In addition to the features, the state vector of the filter contains the time offset t_d and ${}^C_I\mathbf{T}$, as described in Section 5.1. To ensure consistent estimation, the modified-Jacobian approach described in Li and Mourikis (2013b) is employed.

The trajectory estimate computed by EKF-SLAM with concurrent estimation of both ${}^C_I\mathbf{T}$ and t_d is shown in Figure 3 (red dashed line), along with the estimated positions for the LED features (red dots). We can observe that the trajectories estimated by the map-based method and EKF-SLAM are very similar. The maximum error for EKF-SLAM at the known locations of the trajectory is 5.7 cm, while for the persistent features the maximum error is 7.0 cm. Moreover, the top plot of Figure 6 shows the t_d estimate computed by EKF-SLAM, as well as the corresponding $\pm 3\sigma$ uncertainty envelope. The bottom plot of Figure 6 shows the difference between the values of t_d computed in the map-based and EKF-SLAM experiments. After the first 3 s in the trajectory, the two estimates are within 2.5 ms of each other, while the final estimates differ by 0.7 ms. The fact that the state estimates by the two different methods are so similar is a direct consequence of the identifiability of t_d in both cases, and indicates the robustness of the proposed algorithms.

Similarly to the previous experiment, we re-processed the data in EKF-SLAM using the final t_d estimate as an input and disabling its online estimation (note that ${}^C_I\mathbf{T}$ is still being estimated online). The difference in the trajectory estimates of the two approaches, as well as the reported position uncertainty, are shown in Figure 7(a) and Figure 7(b), respectively. These plots show that the difference in the trajectory estimates is very small (within one σ of the reported covariance for most of the trajectory), while the reported uncertainty is almost indistinguishable after the first few seconds of the experiment. We see that, due to the identifiability of t_d even without known feature positions,

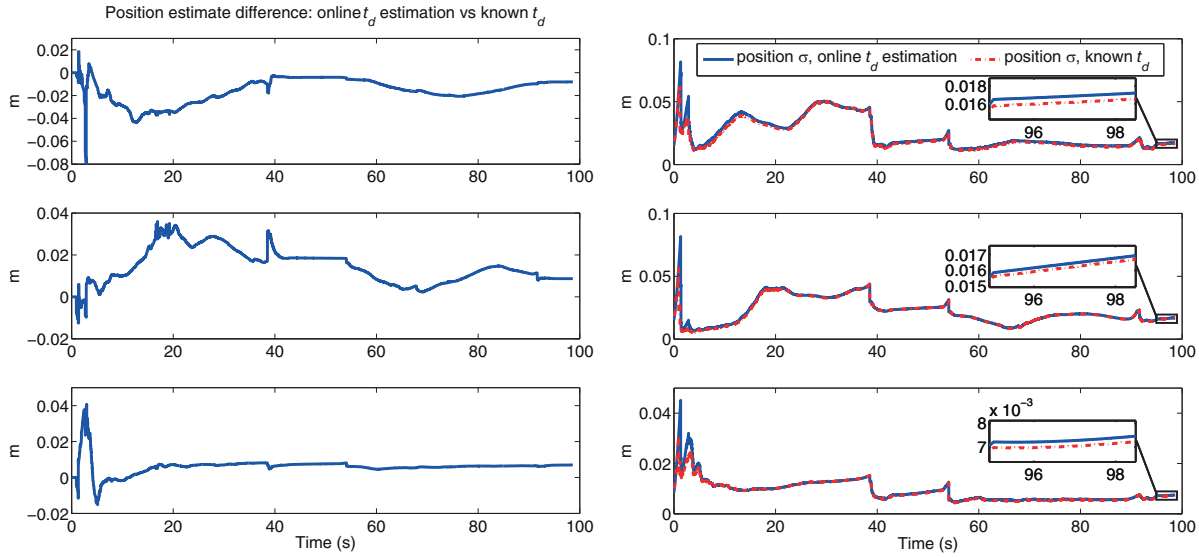

Fig. 7. EKF-SLAM localization: comparison of concurrent localization and t_d estimation vs localization with known t_d. (a) The difference in the position estimates computed in the two cases. (b) The filter's reported standard deviation in the two cases.

the online estimation of t_d does not incur any significant loss of performance.

7.1.3. Visual–inertial odometry. In addition to the indoor experiment, we also carried out a larger, outdoor driving experiment, to test the performance of visual–inertial odometry with concurrent estimation of t_d and ${}^C_I\mathbf{T}$. In this experiment, the camera–IMU system was mounted on the roof of a car driving in Riverside, CA, covering approximately 7.3 km in 11 min. The algorithm used for estimation is the multi-state constraint Kalman filter (MSCKF) 2.0 algorithm of Li and Mourikis (2013b), with the addition of the time offset in the state vector, as described in Section 5.2. Shi-Tomasi features are extracted in the images, and matched by normalized cross-correlation. Figure 8 shows (i) the trajectory estimate computed by the proposed method, (ii) the estimate obtained using the final estimate of t_d as a known input, and (iii) the estimate computed without online estimation of ${}^C_I\mathbf{T}$ and t_d (for ${}^C_I\mathbf{T}$ manual measurements were used, and $t_d = 0$ was assumed in this case). For this experiment, ground truth was obtained by a GPS–INS system, and is shown in black in Figure 8.

Figure 9 shows the orientation (yaw) and horizontal position errors for the three algorithms tested. On this plot, we also show the predicted uncertainty envelope computed as ± 3 times the standard deviation reported by the proposed method. We can observe that the proposed method yields accurate estimates (the error remains below 0.5% of the traveled distance), which are within the reported uncertainty envelope, indicating consistency. Similarly to what we observed in the cases of map-based estimation and SLAM, the results of the algorithm that performs online estimation of t_d are very similar to those obtained with a known t_d. Moreover, the proposed method is considerably more accurate than the case where t_d and ${}^C_I\mathbf{T}$ are assumed to have 'nominal' values, and are not estimated online.

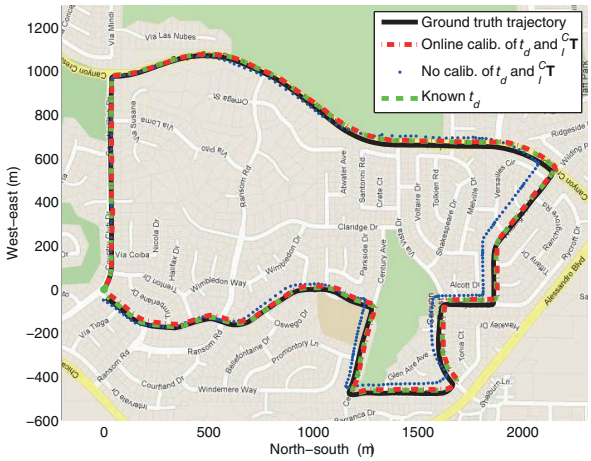

Fig. 8. Visual–inertial odometry: trajectory estimates by the three methods tested, plotted vs ground truth on a map of the area.

7.2. Simulations

In addition to the real-world experiments presented above, we conducted Monte-Carlo simulation tests, to examine the accuracy and consistency of the estimates produced by the proposed algorithms, and to verify whether the results of our real-world testing are typical.

7.2.1. Map-based motion estimation. For the simulations of map-based localization, we generated trajectories with a camera/IMU system moving on a sinusoidal trajectory. In each image six landmarks with known locations, with

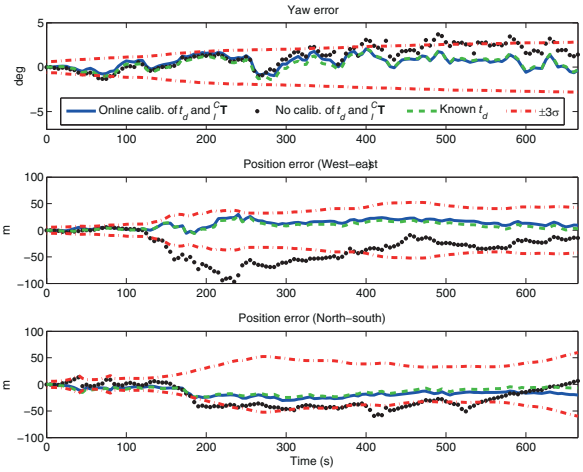

Fig. 9. Visual–inertial odometry: yaw and horizontal position errors by the three methods tested in the real-world experiment.

Table 1. RMSE in the map-based simulations.

${}^G\mathbf{p}_I$	${}^I_G\bar{\mathbf{q}}$	${}^G\mathbf{v}_I$	${}^C\mathbf{p}_I$	${}^C_I\bar{\mathbf{q}}$	t_d
0.096 m	0.10°	0.021 m/s	0.088 m	0.036°	1.519 ms

depths uniformly distributed between 5 and 20 m, were visible. The sensor noise parameters were chosen to be identical to those of the sensors we used for the real-world experiment described in Section 7.1.1. The IMU provided measurements at 100 Hz, while the images were recorded at 10 Hz.

To examine the statistical properties of our proposed algorithm, we carried out 50 Monte-Carlo trials. In each trial, the rotation and translation between the IMU and the camera were set equal to known nominal values, with the addition of random errors $\delta\mathbf{p}$ and $\delta\boldsymbol{\phi}$. In each trial, $\delta\mathbf{p}$ and $\delta\boldsymbol{\phi}$ were randomly drawn from zero-mean Gaussian distributions with standard deviations equal to $\sigma_p = 0.1$ m and $\sigma_\theta = 1.0°$ along each axis, respectively. In addition, t_d was randomly drawn from the Gaussian distribution $\mathcal{N}(0, \sigma_t^2)$, with $\sigma_t = 50$ ms, and kept constant for the duration of the trial. Time offsets in the order of tens of milliseconds are typical of most systems in our experience.

Table 1 shows the root mean squared error (RMSE) for the IMU position, orientation, and velocity, as well as for the camera-to-IMU transformation and the time offset. The values shown are averages over all Monte-Carlo trials, and over the second half of the trajectory (i.e. after the estimation uncertainty has reached steady state). This table shows that the proposed approach allows for precise estimation of all the variables of interest, including the time offset t_d.

Additionally, to examine the consistency of the state estimates we computed the normalized estimation error squared (NEES) for the IMU state, $^C_I\mathbf{T}$ and t_d, each averaged over all Monte-Carlo trials and all timesteps. For a variable $\mathbf{a}$, the NEES at timestep k of a given trial is computed as $\tilde{\mathbf{a}}_k^{\mathrm{T}}\mathbf{P}_{\mathbf{a}_k}^{-1}\tilde{\mathbf{a}}_k$, where $\tilde{\mathbf{a}}_k$ is the estimation error and $\mathbf{P}_{\mathbf{a}_k}$ is the covariance matrix reported by the filter. If the estimator

Table 2. RMSE in the EKF-SLAM simulations.

${}^G\mathbf{p}_I$	${}^I_G\bar{\mathbf{q}}$	${}^G\mathbf{v}_I$	${}^C\mathbf{p}_I$	${}^C_I\bar{\mathbf{q}}$	${}^G\mathbf{p}_f$	t_d
0.078 m	0.26°	0.017 m/s	0.01 m	0.07°	0.094 m	0.1 ms

is consistent, that is, if it reports an appropriate covariance matrix for its state estimates, the NEES should have an average value equal to the dimension of $\mathbf{a}$ (the NEES for a consistent estimator is a χ^2-distributed random variable with dim($\mathbf{a}$) degrees of freedom) (Bar-Shalom et al., 2001). In our tests, the average NEES values were 15.32 for the IMU state, 6.6 for $^C_I\mathbf{T}$, and 0.99 for t_d, close to the expected values of 15, 6, and 1, respectively. This indicates that the estimator is consistent, and that the covariance matrix reported by the EKF is an accurate description of the actual uncertainty of the estimates.

7.2.2. EKF-SLAM. For the SLAM simulations, we generated trajectories that were similar to the trajectory of the real-world experiment presented in Section 7.1.2. The simulated IMU-camera system moved in a $7 \times 12 \times 5$ m room for 90 s, at an average velocity of 0.37 m/s. The sensor characteristics were the same as in the real-world experiment described in Section 7.1.2. A total of 50 persistent features were placed on the walls of the room, and in each image we generated 100 additional temporary features, with depths uniformly distributed between 1.5 and 10 m.

Table 2 shows the RMSE for the IMU position, orientation, and velocity, as well as for the camera-to-IMU transformation, feature positions, and the time offset, averaged over all Monte-Carlo trials, and over the second half of the trajectory. Similarly to what was observed in the case of map-based localization, the time offset between the sensors can be very accurately estimated, owing to its identifiability. The average NEES values have been computed as 17.00 for the IMU state, 6.66 for $^C_I\mathbf{T}$, and 0.87 for t_d. Again, these are close to the theoretically expected values of 15, 6, and 1, respectively. For the feature position the average NEES is 4.48, above the theoretically expected value of 3. This slight increase in the feature NEES is expected, as EKF-SLAM is known to be sensitive to the nonlinearity of the measurement models.

7.2.3. Visual–inertial odometry. To obtain realistic simulation environments for visual–inertial odometry, we generated the simulation data based on a real-world dataset. Specifically, the ground truth trajectory (position, velocity, orientation) for the simulations is generated from the ground truth of a real-world dataset, which was about 13 min, 5.5 km long. Using this trajectory, we subsequently generated IMU measurements corrupted with noise and biases, as well as visual feature tracks with characteristics matching those in the real-world data. For each trial the camera-to-IMU transformation and the time offset were

Table 3. RMSE and NEES values in the Monte-Carlo simulations of EKF-based visual-inertial odometry.

Scenario		Imprecise			Precise
${}^C_I\mathbf{T}$ estimation		on	off	on	N/A
t_d estimation		off	on	on	N/A
IMU pose RMSE	north (m)	54.60	18.39	8.11	7.93
	east (m)	81.82	13.50	5.18	5.00
	down (m)	14.53	45.07	0.64	0.53
	roll (^{o})	0.39	0.18	0.06	0.06
	pitch (^{o})	0.33	0.18	0.05	0.05
	yaw (^{o})	1.19	1.22	0.70	0.69
IMU state NEES		85.4	2046	14.6	14.5
Calibration RMSE	${}^C\mathbf{p}_I$ (m)	0.07	N/A	0.01	N/A
	${}^C_I\bar{\mathbf{q}}$ (^{o})	0.31	N/A	0.05	N/A
	t_d (ms)	N/A	0.28	0.25	N/A

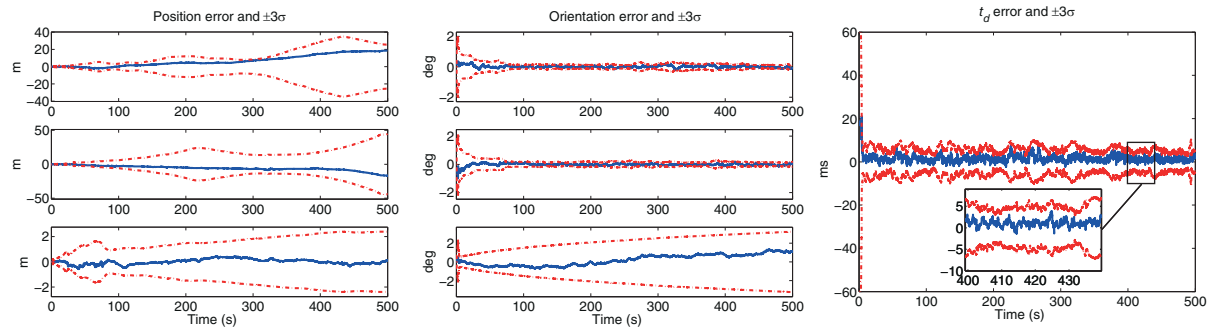

Fig. 10. Visual–inertial odometry with unknown features and drifting time-offset t_d: estimation errors (blue lines) and associated $\pm 3\sigma$ envelopes (red dash-dotted lines). (a) The IMU position errors in the north–east–down directions, (b) the IMU orientation errors in roll–pitch–yaw, (c) the error in the estimate of t_d. Note that the position and yaw uncertainty gradually increases, as normal in visual–inertial odometry without any known landmarks.

generated as in the map-based simulations, by perturbing known nominal values.

In the tests presented here, we compare the performance of visual–inertial odometry using the MSCKF-based approach described in Section 5.2, in four cases: (i) online ${}^C_I\mathbf{T}$ estimation enabled, but t_d estimation disabled, (ii) t_d estimation enabled, but ${}^C_I\mathbf{T}$ estimation disabled, (iii) t_d and ${}^C_I\mathbf{T}$ estimation enabled (i.e. the proposed approach), and (iv) the case where t_d and ${}^C_I\mathbf{T}$ are perfectly known and not estimated. In the first three cases (termed the 'imprecise' ones), the exact values of ${}^C_I\mathbf{T}$ and t_d are not known (only their nominal values are known). When a particular parameter is not estimated, it is assumed to be equal to the nominal value. By comparing these three cases, we can evaluate the necessity and effectiveness of the online estimation of individual parameters. Moreover, by comparing against case (iv), where all parameters are perfectly known (the 'precise' scenario), we can assess the loss of accuracy incurred due to the uncertainty in the knowledge of these parameters.

Table 3 shows the average RMSE and NEES for the four cases, averaged over 50 Monte-Carlo trials. For clarity, the position errors are reported in the north–east–down (NED) frame, and IMU orientation in roll–pitch–yaw. We see that, to be able to accurately estimate the IMU's motion, both the frame transformation and the time offset between the camera and IMU must be estimated. If either of these is falsely assumed to be perfectly known, the estimation accuracy and consistency are considerably degraded (see the first two data columns in Table 3). Moreover, by comparing the third and fourth data columns, we can see that the accuracy obtained by our online estimation approach is very close to that obtained when both ${}^C_I\mathbf{T}$ and t_d are perfectly known. This result, which was also observed in the real-world experiments, is significant from a practical standpoint: it shows that the proposed online approach, initialized with rough estimates, can provide pose estimates almost indistinguishable to what we would get if offline calibration was performed in advance.

7.2.4. Time-varying t_d. For all the results presented up to now, a constant time offset was used. We here also examine the case of a time-varying t_d in visual–inertial odometry. Instead of presenting Monte-Carlo simulation results (which look similar to those in Table 3), it is interesting to show the results of a single representative trial. In this

trial, the time offset varies linearly from 20 ms at the start, to 520 ms at after 500 s, modeling a severe clock drift of 0.5 s in 8.3 min. Figure 10 presents the estimation errors and associated ± 3 standard deviations for the IMU position, the IMU orientation, and the time offset. We can see that even in this challenging situation (unknown features, uncertain camera-to-IMU transformation, large and time-varying offset) the estimates remain consistent. We stress that this is due to the identifiability of t_d, which allows us to track its value closely as it drifts over time.

8. Conclusion

In this paper we have proposed an online approach for estimating the time offset, t_d, between the camera and IMU during EKF-based vision-aided inertial navigation. The key component of our formulation is that the variable t_d is explicitly included in the EKF state vector, and estimated jointly with all other variables of interest. This makes it possible to track time-varying offsets, to characterize the uncertainty in the estimate of t_d, and to model the effect of the imperfect knowledge of t_d on the accuracy of the estimates, in a natural way. Moreover, we have shown that t_d is identifiable in general trajectories, which guarantees the effectiveness of the proposed online estimation approach. A detailed characterization of the critical motion cases that lead to loss of identifiability of t_d reveals that they are either (i) cases that are known to cause loss of observability even with a perfectly known t_d, or (ii) cases that are unlikely to occur in practice. Our simulation and experimental results indicate that the proposed approach leads to high-precision estimates for the system motion as well as for the temporal and spatial alignment between the camera and IMU.

Funding

This work was supported by the National Science Foundation (grant numbers IIS-1117957 and IIS-1253314), the UC Riverside Bourns College of Engineering, and the Hellman Family Foundation (Hellman Fellowship).

Notes

1. Notation: the preceding superscript for vectors (e.g. X in ${}^X\mathbf{a}$) denotes the frame of reference with respect to which the vector is expressed. ${}^X_Y\mathbf{R}$ is the rotation matrix rotating vectors from frame $\{Y\}$ to $\{X\}$, and ${}^X_Y\bar{\mathbf{q}}$ is the corresponding unit quaternion (Trawny and Roumeliotis, 2005); ${}^X\mathbf{p}_Y$ denotes the position of the origin of frame $\{Y\}$, expressed in $\{X\}$, and $\otimes$ denotes quaternion multiplication. Further, $\lfloor \mathbf{c} \times \rfloor$ is the skew-symmetric matrix corresponding to vector $\mathbf{c}$, and $\mathbf{0}_3$ and $\mathbf{I}_3$ are the 3×3 zero and identity matrices, respectively. Finally, $\hat{a}$ is the estimate of a variable a, and $\tilde{a} \doteq a - \hat{a}$ is the error of the estimate.
2. Note that t_d can be viewed as either a fixed parameter to be estimated or as a state of a dynamical system, with zero dynamics. Therefore, we can use the terms identifiability and observability interchangeably for t_d.

References

Bak M, Larsen T, Norgaard M, et al. (1998) Location estimation using delayed measurements. In: *Proceedings of the IEEE International workshop on advanced motion control*, Coimbra, Portugal.

Bar-Shalom Y (2002) Update with out-of-sequence-measurements in tracking: Exact solution. *IEEE Transactions on Aerospace and Electronic Systems* 38(3): 769–778.

Bar-Shalom Y, Li XR and Kirubarajan T (2001) *Estimation with Applications to Tracking and Navigation*. New York, NY: John Wiley & Sons.

Bayard DS and Brugarolas PB (2005) An estimation algorithm for vision-based exploration of small bodies in space. In: *Proceedings of the American control conference*, Portland, OR, pp. 4589–4595.

Bellman R and Astrom K (1970) On structural identifiability. *Mathematical Biosciences* 7: 329–339.

Choi M, Choi J, Park J, et al. (2009) State estimation with delayed measurements considering uncertainty of time delay. In: *Proceedings of the IEEE International conference on robotics and automation*, Kobe, Japan, pp. 3987–3992.

Civera J, Davison A and Montiel J (2008) Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics* 24(5): 932–945.

Dellaert F and Kaess M (2006) Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research* 25(12): 1181–1203.

Doren JV, Douma S, den Hof PV, et al. (2009) Identifiability: From qualitative analysis to model structure approximation. In: *Proceedings of the 15th IFAC symposium on system identification*, Saint-Malo, France, pp. 664–669.

Fertner A and Sjolund A (1986) Comparison of various time delay estimation methods by computer simulation. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34(5): 1329–1330.

Giovanni G and Scarano G (1993) Discrete-time techniques for time-delay estimation. *IEEE Transactions on Signal Processing* 42(2): 525–533.

Harrison A and Newman P (2011) TICSync: Knowing when things happened. In: *Proceedings of the IEEE International conference on robotics and automation*, Shanghai, China, pp. 356–363.

Hartley R and Zisserman A (2000) *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press.

Hesch JA, Kottas DG, Bowman SL, et al. (2012) Towards consistent vision-aided inertial navigation. In: *Proceedings of the International workshop on the algorithmic foundations of robotics*, Cambridge, MA.

Jones E and Soatto S (2011) Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research* 30(4): 407–430.

Julier SJ and Uhlmann JK (2005) Fusion of time delayed measurements with uncertain time delays. In: *Proceedings of the American control conference*, Portland, OR, pp. 4028–4033.

Kaess M, Ranganathan A and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.

Kelly J and Sukhatme G (2011) Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research* 30(1): 56–79.

Kelly J and Sukhatme GS (2010) A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In: *Proceedings of the International symposium of experimental robotics*, New Delhi, India.

Kottas DG, Hesch JA, Bowman SL, et al. (2012) On the consistency of vision-aided inertial navigation. In: *Proceedings of the International symposium on experimental robotics*, Quebec City, Canada.

Li M and Mourikis AI (2012) Improving the accuracy of EKF-based visual-inertial odometry. In: *Proceedings of the IEEE International conference on robotics and automation*, St Paul, MN, pp. 828–835.

Li M and Mourikis AI (2013a) 3-D motion estimation and online temporal calibration for camera-IMU systems. In: *Proceedings of the IEEE International conference on robotics and automation*, Karlsruhe, Germany, pp. 5689–5696.

Li M and Mourikis AI (2013b) High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research* 32(6): 690–711.

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

Maybeck PS (1982) *Stochastic Models, Estimation and Control* (*Mathematics in Science and Engineering*, volume 141, part 2). London: Academic Press.

Mirzaei FM and Roumeliotis SI (2008) A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics* 24(5): 1143–1156.

Montiel J, Civera J and Davison A (2006) Unified inverse depth parametrization for monocular SLAM. In: *Proceedings of robotics: Science and systems*, Philadelphia, PA, pp. 81–88.

Mourikis AI and Roumeliotis SI (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proceedings of the IEEE International conference on robotics and automation*, Rome, Italy, pp. 3565–3572.

Pinies P, Lupton T, Sukkarieh S, et al. (2007) Inertial aiding of inverse depth SLAM using a monocular camera. In: *Proceedings of the IEEE/RSJ International conference on intelligent robots and systems*, Rome, Italy, pp. 2797–2802.

Roumeliotis SI, Johnson AE and Montgomery JF (2002) Augmenting inertial navigation with image-based motion estimation. In: *Proceedings of the IEEE International conference on robotics and automation*, Washington DC, pp. 4326–4333.

Shi J and Tomasi C (1994) Good features to track. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, Seattle, WA, pp. 593–600.

Shkurti F, Rekleitis I, Scaccia M, et al. (2011) State estimation of an underwater robot using visual and inertial information. In: *Proceedings of the IEEE/RSJ International conference on intelligent robots and systems*, San Francisco, CA, pp. 5054–5060.

Skog I and Haendel P (2011) Time synchronization errors in loosely coupled GPS-aided inertial navigation systems. *IEEE Transactions on Intelligent Transportation Systems* 12(4): 1014–1023.

Sola J (2010) Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations. In: *Proceedings of the IEEE International conference on robotics and automation*, Anchorage, AK, pp. 3513–3518.

Trawny N and Roumeliotis SI (2005) *Indirect Kalman filter for 6D pose estimation.* Technical report number 2005-002, Department of Computer Science and Engineering, University of Minnesota, MN.

Trawny N, Mourikis AI, Roumeliotis SI, et al. (2007) Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. *Journal of Field Robotics* 24(5): 357–378.

Tungadi F and Kleeman L (2010) Time synchronisation and calibration of odometry and range sensors for high-speed mobile robot mapping. In: *Proceedings of the Australasian conference on robotics and automation*, Canberra, Australia.

Weiss S, Achtelik M, Chli M, et al. (2012a) Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV. In: *Proceedings of the IEEE International conference on robotics and automation*, St Paul, MN, pp. 31–38.

Weiss S, Achtelik M, Lynen S, et al. (2012b) Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environment. In: *Proceedings of the IEEE International conference on robotics and automation*, St Paul, MN, pp. 957–964.

Williams B, Hudson N, Tweddle B, et al. (2011) Feature and pose constrained visual aided inertial navigation for computationally constrained aerial vehicles. In: *Proceedings of the IEEE International conference on robotics and automation*, Shanghai, China, pp. 5655–5662.

Wu A, Johnson E and Proctor A (2005) Vision-aided inertial navigation for flight control. *AIAA Journal of Aerospace Computing, Information, and Communication* 2(9): 348–360.

Zhang K, Li XR and Zhu Y (2005) Optimal update with out-of-sequence-measurements. *IEEE Transactions on Signal Processing* 53(6): 1992–2005.

Appendix: Proof of Lemma 6.2

We start by performing a change of variables, defining $\bar{\mathbf{v}} = s^{C_o}\mathbf{v}_o$, $\bar{\mathbf{g}} = s^{C_o}\mathbf{g}$, $\bar{\mathbf{b}} = -s^C_I\mathbf{R}\mathbf{b}_\mathbf{a}$, and $\bar{\mathbf{p}} = -s^C\mathbf{p}_I$, to obtain

$$\mathbf{c}_2(\bar{\boldsymbol{\xi}}, t) = \bar{\mathbf{v}} + \bar{\mathbf{g}}t + s\int_0^t \mathbf{R}_c(\tau)\,{}^C_I\mathbf{R}\,\mathbf{a}_m(\tau + t_d)\,d\tau + \int_0^t \mathbf{R}_c(\tau)\,d\tau\,\bar{\mathbf{b}} + \dot{\mathbf{R}}_c(t)\,\bar{\mathbf{p}} - \mathbf{v}_c(t) = \mathbf{0} \quad (57)$$

with

$$\bar{\boldsymbol{\xi}} = \begin{bmatrix}\bar{\mathbf{v}}^\mathrm{T} & \bar{\mathbf{g}}^\mathrm{T} & \bar{\mathbf{b}}^\mathrm{T} & \mathbf{b}_\mathbf{g}{}^\mathrm{T} & \bar{\mathbf{p}}^\mathrm{T} & {}^C_I\mathbf{q}^\mathrm{T} & t_d & s\end{bmatrix}^\mathrm{T}$$

Note that since t_d, $\mathbf{b}_\mathbf{g}$, and ${}^C_I\bar{\mathbf{q}}$ remain the same in the change from $\boldsymbol{\xi}$ to $\bar{\boldsymbol{\xi}}$, the condition $\mathbf{c}_1$ in (44) also holds for $\bar{\boldsymbol{\xi}}$ with no modification.

Clearly, $\boldsymbol{\xi}$ is locally identifiable if and only if $\bar{\boldsymbol{\xi}}$ is. A sufficient condition for identifiability is that there exists a set of time instants, such that if we evaluate the matrix containing the Jacobians of $\mathbf{c}_1$ and $\mathbf{c}_2$ at these time instants, the matrix has full column rank. In turn, a sufficient condition for this is that there exists no nonzero vector $\mathbf{m}$ such that, for all $t > 0$,

$$\begin{bmatrix}\mathbf{D}_1(t)\\ \mathbf{D}_2(t)\end{bmatrix}\mathbf{m} = \mathbf{0} \;\Leftrightarrow\; \mathbf{D}_1(t)\,\mathbf{m} = \mathbf{0} \text{ and } \mathbf{D}_2(t)\,\mathbf{m} = \mathbf{0} \quad (58)$$

where $\mathbf{D}_1(t)$ is the Jacobian of $\mathbf{c}_1$ with respect to $\bar{\boldsymbol{\xi}}$, and $\mathbf{D}_2(t)$ is the Jacobian of $\mathbf{c}_2$ with respect to $\bar{\boldsymbol{\xi}}$. We now introduce the partitioning

$$\mathbf{m} = [\mathbf{m}_1^{\mathrm{T}} \;\; \mathbf{m}_2^{\mathrm{T}} \;\; \mathbf{m}_3^{\mathrm{T}} \;\; \mathbf{k}_1^{\mathrm{T}} \;\; \mathbf{m}_4^{\mathrm{T}} \;\; \mathbf{k}_2^{\mathrm{T}} \;\; 1 \;\; m_5]^{\mathrm{T}}$$

where the element corresponding to the Jacobians with respect to t_d has been set to 1, as we are interested in detecting cases in which t_d is not identifiable.

The condition $\mathbf{D}_1(t)\,\mathbf{m} = \mathbf{0}$ is identical to the condition that was encountered in the proof of Lemma 6.1, and yields the first condition of Lemma 6.2. The second condition of Lemma 6.2 is derived from the equation $\mathbf{D}_2(t)\,\mathbf{m} = \mathbf{0}$. Computing the Jacobians of $\mathbf{c}_2$ with respect to $\bar{\boldsymbol{\xi}}$, and substituting in $\mathbf{D}_2(t)\,\mathbf{m} = \mathbf{0}$, we obtain

$$\begin{aligned}
&\mathbf{m}_1 + t\mathbf{m}_2 + \int_0^t \mathbf{R}_c(\tau)\, d\tau\, \mathbf{m}_3 + \dot{\mathbf{R}}_c(t)\,\mathbf{m}_4 \\
&+ s\int_0^t \mathbf{R}_c(\tau)\,{}_I^C\mathbf{R}\lfloor \mathbf{a}_m(\tau + t_d)\times\rfloor\, d\tau \mathbf{k}_2 \\
&+ s\int_0^t \mathbf{R}_c(\tau)\,{}_I^C\mathbf{R}\dot{\mathbf{a}}_m(\tau + t_d)\, d\tau \\
&+ \int_0^t \mathbf{R}_c(\tau)\,{}_I^C\mathbf{R}\mathbf{a}_m(\tau + t_d)\, d\tau m_5 = \mathbf{0}, \;\; \forall t > 0
\end{aligned}$$

Differentiating this expression with respect to t, and rearranging terms, yields

$$\dot{\mathbf{a}}_m(t + t_d) = \left(\lfloor \mathbf{k}_2\times\rfloor - \frac{m_5}{s}\mathbf{I}_3\right)\mathbf{a}_m(t + t_d) - \frac{1}{s}\boldsymbol{\gamma}(t), \;\; \forall t > 0$$

or, equivalently,

$$\dot{\mathbf{a}}_m(t) = \left(\lfloor \mathbf{k}_2\times\rfloor - \frac{m_5}{s}\mathbf{I}_3\right)\mathbf{a}_m(t) - \frac{1}{s}\boldsymbol{\gamma}(t - t_d), \;\; \forall t > 0 \tag{59}$$

where

$$\begin{aligned}
\boldsymbol{\gamma}(t - t_d) &= {}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{R}_c(t - t_d)^{\mathrm{T}}\,\mathbf{m}_2 + {}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_3 \\
&\quad + {}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{R}_c^{\mathrm{T}}(t - t_d)\,\ddot{\mathbf{R}}_c(t - t_d)\,\mathbf{m}_4 \\
&= {}_I^{I_o}\mathbf{R}(t)^{\mathrm{T}}\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_2 + {}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_3 \\
&\quad + {}_I^{I_o}\mathbf{R}(t)^{\mathrm{T}}\,{}_I^{I_o}\ddot{\mathbf{R}}(t)\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_4 \\
&= {}_I^{I_o}\mathbf{R}(t)^{\mathrm{T}}\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_2 + {}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_3 \\
&\quad + \left(\lfloor {}^I\boldsymbol{\omega}(t)\times\rfloor^2 + \lfloor {}^I\dot{\boldsymbol{\omega}}(t)\times\rfloor\right)\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_4
\end{aligned}$$

Substituting the last expression in (59), and defining $k_3 = m_5/s$, $\mathbf{k}_4 = (1/s)\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_3$, $\mathbf{k}_5 = (1/s)\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_4$, and $\mathbf{k}_6 = (1/s)\,{}_I^C\mathbf{R}^{\mathrm{T}}\mathbf{m}_2$, yields (56).