2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

3-D Motion Estimation and Online Temporal Calibration for Camera-IMU Systems

Mingyang Li and Anastasios I. Mourikis
Dept. of Electrical Engineering, University of California, Riverside
E-mail: mli@ee.ucr.edu, mourikis@ee.ucr.edu

***Abstract*— When measurements from multiple sensors are combined for real-time motion estimation, the time instant at which each measurement was recorded must be precisely known. In practice, however, the timestamps of each sensor's measurements are typically affected by a delay, which is different for each sensor. This gives rise to a temporal misalignment (i.e., a *time offset*) between the sensors' data streams. In this work, we propose an *online* approach for estimating the time offset between the data obtained from different sensors. Specifically, we focus on the problem of motion estimation using visual and inertial sensors in extended Kalman filter (EKF)-based methods. The key idea proposed here is to *explicitly* include the time offset between the camera and IMU in the EKF state vector, and estimate it online along with all other variables of interest (the IMU pose, the camera-to-IMU calibration, etc). Our proposed approach is general, and can be employed in several classes of estimation problems, such as motion estimation based on mapped features, EKF-based SLAM, or visual-inertial odometry. Our simulation and experimental results demonstrate that the proposed approach yields high-precision, consistent estimates, in scenarios involving both constant and time-varying offsets.**

I. Introduction

Autonomous vehicles moving in 3D, such as aerial vehicles or ground robots on uneven terrain, require accurate 3D pose estimates even in the absence of GPS. In this work, we focus on methods that provide such estimates by fusing the measurements from an inertial measurement unit (IMU) with those obtained from a vision sensor. To date, several different methods for this task (often termed *vision-aided inertial navigation*) have been proposed, tailored for different application scenarios. For instance, some techniques rely on observations of landmarks with known world coordinates [1, 2], others perform simultaneous localization and mapping in unknown environments [3, 4], while yet others focus on visual-inertial odometry with unknown features [5, 6].

In all cases, for the algorithms yield precise pose estimates, the timing of the sensor measurements must be precisely known. For this reason, a timestamp for each measurement is typically obtained, either on the computer used to record the data or from the sensor itself. However, these timestamps are often *temporally misaligned*, for a number of reasons. For example, if different clocks are used for timestamping (e.g., on different sensors), these clocks may suffer from clock skew. Moreover, due to the time needed for data transfer, sensor latency, and operating-system overhead, a delay – different for each sensor – exists between the actual sampling of a measurement and the generation of its timestamp. These effects give rise to an unknown *time offset*, t_d, between the timestamps of the camera and the IMU. This offset means that, if the two sensors record measurements simultaneously at a given instant, these measurements will be reported *as if* they occurred t_d seconds apart. If this effect is not compensated for, it will introduce *unmodelled errors* in the estimation process, and reduce its accuracy (see Section IV).

The existence of a time offset between sensors is a common occurrence in several applications, and therefore a significant amount of research has focused on it. However, the vast majority of work has concentrated on the problem of processing the sensor data when the time offset is *known* in advance, either precisely (e.g., [7]–[9] and references therein), or approximately [10, 11]. All these algorithms treat the time offset between the sensors as an *input*: they do not attempt to estimate this offset, or to improve a prior estimate using additional data. To date, relatively little work has focused on solving this problem in a principled manner. Presumably, developers of vision-aided inertial navigation algorithms often resort to ad-hoc methods, or to simple trial-and-error to identify the time offset on a case-by-case basis. An exception can be found in recent work by Kelly and Sukhatme [12], who propose a principled offline method for estimating the time offset between a camera and an IMU. Their approach relies on computing rotation estimates from each individual sensor, and temporally aligning the individual rotation trajectories via batch ICP-like registration in the space of rotations. Batch registration-based techniques for other sensors (e.g., laser scanner and odometry) have also appeared [13].

Both batch estimation methods and simpler ad-hoc solutions have two key disadvantages: First, being offline in nature, they cannot deal with time-varying offsets. These may arise due to changes in the computing environment (e.g., changing processor load), or due to clock drift, when multiple clocks are used for timestamping. Second, they do not provide a means for taking into account the uncertainty of the time-offset estimate during the subsequent pose estimation. To address these limitations, in this work we propose an *online* approach to estimating the time offset between the data streams of the camera and IMU. Specifically, we show that it is possible to *explicitly* include this offset in the state vector of an extended Kalman filter (EKF), and estimate it along with all other quantities of interest. In addition to being able to track time-varying offsets, the proposed EKF-based formulation also models the uncertainty of the time-offset

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

estimate in a natural way, via the EKF covariance matrix. Thus, the uncertainty is accounted for during pose estimation.

The proposed approach is general, and can be employed in different types of vision-aided inertial navigation problems[1]. Specifically, we show that the proposed EKF formulation can be used both when the positions of the observed features are known (e.g., when using fiducial points), and when they are unknown (e.g., in visual-inertial odometry). We present EKF estimators for both cases, which jointly estimate (i) the IMU state, (ii) the extrinsic calibration (position and orientation) between the camera and IMU frames, and (iii) the time offset between the sensors, in an online fashion. Note that, even though EKF-based estimators have appeared in the past for concurrent pose estimation and extrinsic calibration [3, 14, 15] they all assume the timing between the sensors to be known. Our simulation and experimental results demonstrate that the proposed approach yields high-precision, consistent estimates, in scenarios involving either known or unknown features, with both constant or time-varying offsets.

In what follows, we present the details of our work. We begin in Section II, by studying the simpler case of map-based pose estimation, in which the positions of the observed features in the world are known. This section presents the main idea of our work, which is the inclusion of the time offset in the EKF state vector. Next, in Section III we show how the same key idea can be applied in scenarios where the feature positions are not known in advance.

II. Map-based Pose Estimation

We first consider the case where a system comprising a camera and IMU is moving in an environment containing features with known 3D coordinates. Each sensor provides measurements at a constant frequency, which is known at least to a good approximation. However, there exists an unknown time offset, t_d, between the two sensors' reported timestamps. Specifically, we define t_d as the amount of time by which we should shift the camera timestamps, so that the camera and IMU data streams become temporally consistent. Note that t_d may have a positive or negative value: if the IMU has a longer latency than the camera, then t_d will be positive, while in the opposite case t_d will be negative. In what follows, we describe how the time offset can be estimated concurrently with all other quantities of interest.

A. State vector formulation

Our main objective is to estimate the 3D pose of the system with respect to the global coordinate frame, $\{G\}$. To this end, we track the motion of the IMU coordinate frame, $\{I\}$, with respect to $\{G\}$, using an EKF. To achieve precise estimation, in addition to the IMU state we include t_d in the EKF state vector. Moreover, to address situations in which the spatial configuration between the camera frame, $\{C\}$, and the IMU frame (i.e., the extrinsic calibration of the sensors) is not accurately known, we include the camera-to-IMU transformation in the filter state vector, and estimate it jointly with all other state variables. Therefore, the EKF's state vector is defined as[2]:

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

where $\mathbf{x}_I(t)$ is the IMU state at time t, the unit quaternion ${}^C_I\bar{\mathbf{q}}$ describes the rotation from the IMU to the camera frame, and ${}^C\mathbf{p}_I$ is the position of the IMU with respect to the camera. Note that, by convention, we use the "IMU time" to define the time reference. That is, $\mathbf{x}(t)$ is the system state at the time the IMU measurement with timestamp t was recorded.

Following standard practice, we define the IMU state as the 16×1 vector:

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

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

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

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

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

where for the position, velocity, and bias, as well as for the time offset t_d, the standard additive error definition has been used (e.g., ${}^G\tilde{\mathbf{v}}_I = {}^G\mathbf{v}_I - {}^G\hat{\mathbf{v}}_I$). On the other hand, for the orientation errors we use a minimal 3-dimensional representation, defined by the equations [6, 17]:

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

B. EKF propagation

The IMU measurements are used to propagate the IMU state estimates. Specifically, the IMU gyroscopes and accelerometers provide measurements of the rotational velocity, $\boldsymbol{\omega}_m$, and acceleration (more precisely, specific force), $\mathbf{a}_m$, respectively, described by the equations:

$$\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega} + \mathbf{b}_\mathbf{g} + \mathbf{n}_\mathbf{r} \tag{5}$$

$$\mathbf{a}_m = {}^I_G\mathbf{R}\left({}^G\mathbf{a} - {}^G\mathbf{g}\right) + \mathbf{b}_\mathbf{a} + \mathbf{n}_\mathbf{a} \tag{6}$$

where ${}^I\boldsymbol{\omega}$ is the IMU's rotational velocity, ${}^G\mathbf{g}$ is the gravitational acceleration, and $\mathbf{n}_\mathbf{r}$ and $\mathbf{n}_\mathbf{a}$ are zero-mean white

[1] In fact, the idea of explicitly including the time offset in the EKF's state vector is not specific to the case of pose estimation using cameras and IMUs, and can be employed in other localization problems as well.

[2] Notation: The preceding superscript for vectors (e.g., G in ${}^G\mathbf{a}$) denotes the frame of reference with respect to which quantities are expressed. ${}^A_B\mathbf{R}$ is the rotation matrix rotating vectors from frame $\{B\}$ to $\{A\}$, and ${}^A_B\bar{\mathbf{q}}$ is the corresponding unit quaternion [16]. $\otimes$ denotes quaternion multiplication, $\lfloor\mathbf{c}\times\rfloor$ is the skew symmetric matrix corresponding to vector $\mathbf{c}$, and $\mathbf{0}$ and $\mathbf{I}$ are zero and identity matrices respectively. Finally, $\hat{a}$ is the estimate of a variable a, and $\tilde{a} \doteq a - \hat{a}$ the error of the estimate.

Gaussian noise vectors. Using these measurements, we can write the dynamics of the state vector as:

$$
\begin{aligned}
{}^I_G\dot{\bar{\mathbf{q}}}(t) &= \frac{1}{2}\boldsymbol{\Omega}\Big(\boldsymbol{\omega}_m(t) - \mathbf{b}_{\mathbf{g}}(t) - \mathbf{n}_{\mathbf{r}}(t)\Big){}^I_G\bar{\mathbf{q}}(t) && (7)\\
{}^G\dot{\mathbf{v}}(t) &= {}^I_G\mathbf{R}(t)^T\left(\mathbf{a}_m(t) - \mathbf{b}_{\mathbf{a}}(t) - \mathbf{n}_{\mathbf{a}}(t)\right) + {}^G\mathbf{g} && (8)\\
{}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t) && (9)\\
\dot{\mathbf{b}}_{\mathbf{g}}(t) &= \mathbf{n}_{\mathbf{wg}}(t), \quad \dot{\mathbf{b}}_{\mathbf{a}}(t) = \mathbf{n}_{\mathbf{wa}}(t) && (10)\\
{}^C_I\dot{\bar{\mathbf{q}}}(t) &= \mathbf{0}, \qquad {}^C\dot{\mathbf{p}}_I(t) = \mathbf{0} && (11)\\
\dot{t}_d(t) &= 0 && (12)
\end{aligned}
$$

where $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is the quaternion multiplication matrix corresponding to the angular velocity vector $\boldsymbol{\omega}$ [16]. In the above, the first three lines describe the dynamics of the IMU motion, the fourth line describes the random walk processes that model the biases' slowly time-varying nature, (11) describes the fact that the camera-to-IMU transformation remains constant in time, while the last line expresses the fact that the time offset between the camera and 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)-(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 [17]. Specifically, for propagating the orientation from time instant t_k to t_{k+1}, we numerically integrate the differential equation:

$$
{}^I_G\dot{\hat{\bar{\mathbf{q}}}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_{\mathbf{g}}(t_k)\right){}^I_G\hat{\bar{\mathbf{q}}}(t), \qquad t \in [t_k, t_{k+1}]
$$

The velocity and position estimates are propagated by:

$$
\begin{aligned}
{}^G\hat{\mathbf{v}}_{k+1} &= {}^G\hat{\mathbf{v}}_k + {}^G_I\hat{\mathbf{R}}(t_k)\,\hat{\mathbf{s}}_k + {}^G\mathbf{g}\Delta t && (13)\\
{}^G\hat{\mathbf{p}}_{k+1} &= {}^G\hat{\mathbf{p}}_k + {}^G\hat{\mathbf{v}}_k\Delta t + {}^G_I\hat{\mathbf{R}}(t_k)\,\hat{\mathbf{y}}_k + \frac{1}{2}{}^G\mathbf{g}\Delta t^2 && (14)
\end{aligned}
$$

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

$$
\begin{aligned}
\hat{\mathbf{s}}_k &= \int_{t_k}^{t_{k+1}} {}^{I_k}_{I_\tau}\hat{\mathbf{R}}\Big(\mathbf{a}_m(\tau) - \hat{\mathbf{b}}_{\mathbf{a}}(t_k)\Big)d\tau && (15)\\
\hat{\mathbf{y}}_k &= \int_{t_k}^{t_{k+1}}\int_{t_k}^{s} {}^{I_k}_{I_\tau}\hat{\mathbf{R}}\Big(\mathbf{a}_m(\tau) - \hat{\mathbf{b}}_{\mathbf{a}}(t_k)\Big)d\tau ds && (16)
\end{aligned}
$$

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:

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

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

$$
\boldsymbol{\Phi}(t_{k+1}, t_k) = \begin{bmatrix} \boldsymbol{\Phi}_I(t_{k+1}, t_k) & \mathbf{0}_{15\times 7} \\ \mathbf{0}_{15\times 7}^T & \mathbf{I}_{7\times 7} \end{bmatrix} \qquad (17)
$$

with $\boldsymbol{\Phi}_I(t_{k+1}, t_k)$ being the 15 × 15 error-state transition matrix for the IMU state, derived in [6, 17].

C. EKF Updates

We now describe how the camera measurements are used 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 a random variable. Let us consider the observation of the i-th feature in the image timestamped at t. This is described by:

$$
\mathbf{z}_i(t) = \mathbf{h}\big({}^C\mathbf{p}_{f_i}(t{+}t_d)\big) + \mathbf{n}_i(t{+}t_d) \qquad (18)
$$

where $\mathbf{h}(\cdot)$ is the perspective camera model, $\mathbf{h}(\mathbf{f}) = [f_x/f_z \quad f_y/f_z]^T$, $\mathbf{n}_i$ is the measurement noise vector, modelled 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:

$$
{}^C\mathbf{p}_{f_i}(t{+}t_d) = {}^C_I\mathbf{R}\;{}^I_G\mathbf{R}(t{+}t_d)\big({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I(t{+}t_d)\big) + {}^C\mathbf{p}_I \qquad (19)
$$

In this equation ${}^G\mathbf{p}_{f_i}$ is the known position of the i-th feature in the global frame.

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 [18]:

$$
\mathbf{r}_i = \mathbf{z}_i(t) - \mathbf{h}\big(\widehat{{}^C\mathbf{p}_{f_i}(t{+}t_d)}\big) \qquad (20)
$$

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

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

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:

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

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:

$$
\begin{aligned}
\mathbf{H}_{\boldsymbol{\theta},i} &= \mathbf{J}_i\;{}^C_I\hat{\mathbf{R}}\;{}^I_G\hat{\mathbf{R}}(t{+}\hat{t}_d)\lfloor({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t{+}\hat{t}_d))\times\rfloor\\
\mathbf{H}_{\mathbf{p},i} &= -\mathbf{J}_i\;{}^C_I\hat{\mathbf{R}}\;{}^I_G\hat{\mathbf{R}}(t{+}\hat{t}_d)\\
\boldsymbol{\Pi}_{\boldsymbol{\phi},i} &= \mathbf{J}_i\;{}^C_I\hat{\mathbf{R}}\lfloor{}^I_G\hat{\mathbf{R}}(t{+}\hat{t}_d)({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t{+}\hat{t}_d))\times\rfloor\\
\boldsymbol{\Pi}_{\mathbf{p},i} &= \mathbf{J}_i\\
\boldsymbol{\Pi}_{t_d,i} &= -\mathbf{J}_i\;{}^C_I\hat{\mathbf{R}}\lfloor{}^I\hat{\boldsymbol{\omega}}(t{+}\hat{t}_d)\times\rfloor{}^I_G\hat{\mathbf{R}}(t{+}\hat{t}_d)\big({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I(t{+}\hat{t}_d)\big)\\
&\quad - \mathbf{J}_i\;{}^C_I\hat{\mathbf{R}}\;{}^I_G\hat{\mathbf{R}}(t{+}\hat{t}_d)\,{}^G\hat{\mathbf{v}}_I(t{+}\hat{t}_d) \qquad (21)
\end{aligned}
$$

In the above, $\mathbf{J}_i$ is the Jacobian of the perspective model:

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

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, $\mathbf{\Pi}_{t_d,i}$, requires the rotational velocity vector, which is available from the IMU measurements. 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:

$$\hat{\mathbf{x}}(t{+}\hat{t}_d) \leftarrow \hat{\mathbf{x}}(t{+}\hat{t}_d) + \mathbf{K}_i \mathbf{r}_i$$
$$\mathbf{P}(t{+}\hat{t}_d) \leftarrow \mathbf{P}(t{+}\hat{t}_d) - \mathbf{K}_i \mathbf{S}_i \mathbf{K}_i^T$$

where:

$$\mathbf{K}_i = \mathbf{P}(t{+}\hat{t}_d)\mathbf{H}_{\mathbf{x},i}(t{+}\hat{t}_d)^T \mathbf{S}_i^{-1}, \quad \text{with} \tag{22}$$
$$\mathbf{S}_i = \mathbf{H}_{\mathbf{x},i}(t{+}\hat{t}_d)\mathbf{P}(t{+}\hat{t}_d)\mathbf{H}_{\mathbf{x},i}(t{+}\hat{t}_d)^T + \sigma_{im}^2 \mathbf{I} \tag{23}$$

If more than one features are 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 (23)) the uncertainty in the time offset is explicitly modelled, 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.

Finally, we note that the proposed EKF-based approach to time-offset estimation requires minimal modifications: compared to the "standard" filter which only contains the IMU state and the camera-to-IMU extrinsic parameters, the proposed filter requires one additional state variable, and the computation of one additional block in the Jacobian matrix (see (21)). These small additions make it possible to seamlessly estimate the time offset online, along with all other variables of interest.

III. 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 applications. Broadly, these methods use the visual measurements in one of two ways [17, 19]: *feature-based* methods include feature positions in the state vector being estimated (as in the EKF-SLAM paradigm [3, 4]), and employ the feature observations directly for state updates. 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 [5, 6]. 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 methods.

A. Feature-based methods

In feature-based EKF algorithms (often referred to as EKF-SLAM algorithms) the state vector of the system is augmented with the positions of the features detected by the camera. Thus, the state vector contains the IMU state, the camera-to-IMU transformation (if its online estimation is needed), and N features:

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

where the feature parameterization, $\mathbf{f}_i$, $i = 1, \ldots N$ can have a number of different forms (e.g., XYZ position, inverse depth, homogeneous coordinates). To estimate the time offset between the sensors we can again include it in the state:

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

With this augmented state vector, the feature measurements, $\mathbf{z}_i$, can be directly employed for EKF updates. The only difference compared to the standard feature-based EKF algorithms is that an additional Jacobian block (the Jacobian with respect to t_d, shown in (21)) must be computed for each measurement. Apart from this modification, no further changes are needed, in order to implement the online estimation of t_d in EKF-based SLAM.

B. 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 the time instants M images were recorded. For instance, in [5, 6] the state vector is formulated as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^T & \mathbf{c}_1^T & \cdots & \mathbf{c}_M^T \end{bmatrix}^T \tag{24}$$

where $\mathbf{c}_j$ is the camera pose at the time the j-th image was recorded:

$$\mathbf{c}_j = \begin{bmatrix} {}^C_G\mathbf{q}_j^T & {}^G\mathbf{p}_{C_j}^T \end{bmatrix}^T \tag{25}$$

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 poses in the sliding window.

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:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^T & {}^C_I\bar{\mathbf{q}}^T & {}^C\mathbf{p}_I^T & t_d & \mathbf{c}_1^T & \cdots & \mathbf{c}_M^T \end{bmatrix}^T \tag{26}$$

To account for these additional parameters, only minimal modifications are needed in the EKF equations. More specifically, the only filter operation that needs to be changed

TABLE I: RMS errors in map-based simulations

RMS errors					
${}^{G}\mathbf{p}_I$	${}^{I}_{G}\bar{\mathbf{q}}$	${}^{G}\mathbf{v}_I$	${}^{C}\mathbf{p}_I$	${}^{C}_{I}\bar{\mathbf{q}}$	t_d
0.096 m	0.10°	0.021 m/sec	0.088 m	0.036°	1.519 msec

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 $t + t_d$:

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

The filter covariance matrix is also augmented, as:

$$\mathbf{P}(t+\hat{t}_d) \leftarrow \begin{bmatrix} \mathbf{P}(t+\hat{t}_d) & \mathbf{P}(t+\hat{t}_d)\mathbf{J}_{new}^T \\ \mathbf{J}_{new}\mathbf{P}(t+\hat{t}_d) & \mathbf{J}_{new}\mathbf{P}(t+\hat{t}_d)\mathbf{J}_{new}^T \end{bmatrix} \quad (27)$$

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:

$$\mathbf{J}_I = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ -\lfloor {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^{T\,I}\hat{\mathbf{p}}_C \times \rfloor & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times9} \end{bmatrix}$$

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

$$\mathbf{J}_{IC} = \begin{bmatrix} {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^T \end{bmatrix}$$

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

$$\mathbf{J}_t = \begin{bmatrix} {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^T\,{}^{I}\hat{\boldsymbol{\omega}}(t+\hat{t}_d) \\ {}^{I}_{G}\hat{\mathbf{R}}(t+\hat{t}_d)^T \lfloor {}^{I}\hat{\boldsymbol{\omega}}(t+\hat{t}_d)\times \rfloor {}^{I}\hat{\mathbf{p}}_C + {}^{G}\hat{\mathbf{v}}_I(t+\hat{t}_d) \end{bmatrix} \quad (28)$$

Compared to [5, 6], the above equations differ in that additional Jacobians are computed with respect to the camera-to-IMU extrinsic parameters, 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 in exactly the same way for EKF updates as in [5, 6], with no further alterations. Since the dependence of the camera poses on t_d has been modelled (via the Jacobian $\mathbf{J}_t$), when the measurements are used to update the camera pose estimates, t_d will also be updated, as normal in the EKF.

IV. Experiments

In this section we present the results of Monte-Carlo simulation tests and real-world experiments, which demonstrate the performance of the online time-offset estimation, both in mapped and unknown environments.

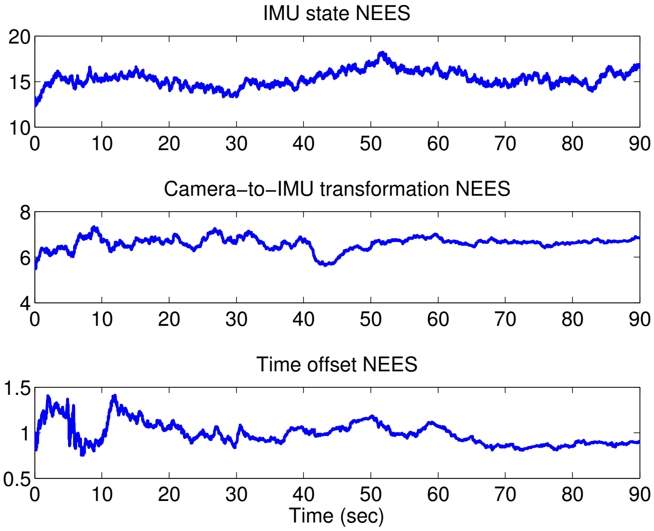


Fig. 1: Map-based simulations: average NEES values over 50 Monte-Carlo simulations.

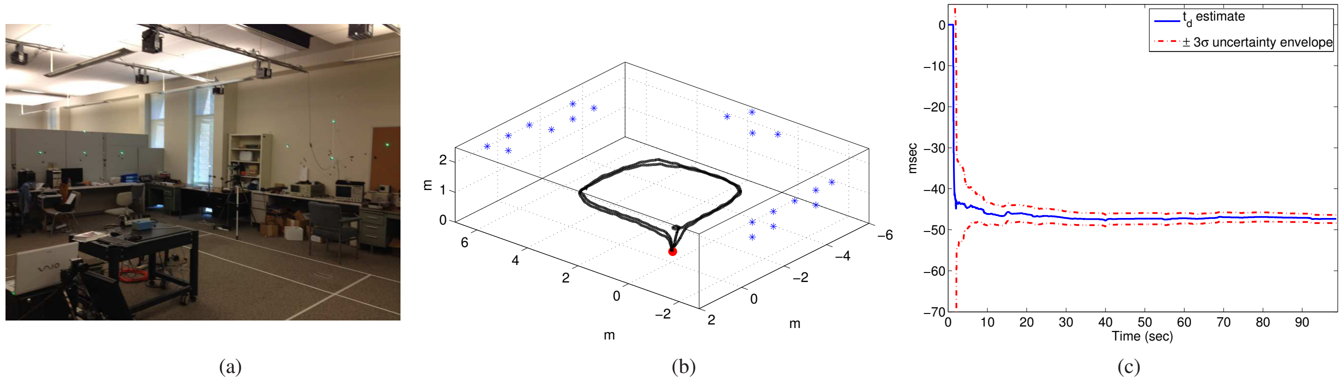
A. Map-based motion estimation

1) Simulations: We performed Monte-Carlo simulation tests, to examine the accuracy and consistency of the estimates computed by the algorithm described in Section II. For the simulations of map-based localization, in each simulated image six landmarks with known locations, with depths uniformly distributed between 5 and 20 meters, 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 IV-A.2. The IMU provided measurements at 100Hz, while the images were recorded at 10Hz.

To examine the statistical properties of our proposed algorithm, we carried out 50 Monte-Carlo trials. In each trial, the extrinsic parameters (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 $\delta\tilde{\boldsymbol{\phi}}$. In each trial, $\delta\mathbf{p}$ and $\delta\tilde{\boldsymbol{\phi}}$ were randomly drawn from zero-mean Gaussian distributions with standard deviations equal to $\sigma_p = 0.1$ m and $\sigma_\phi = 1.0^{\mathrm{o}}$ along each axis, respectively. In addition, t_d was randomly drawn from the Gaussian distribution $\mathcal{N}(0, \sigma_t^2)$, with $\sigma_t = 50$ msec, 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 I shows the RMS errors for the IMU position, orientation, and velocity, as well as for the camera-to-IMU extrinsic parameters and the time offset. The values shown are averages over all Monte-Carlo trials, and over the last 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, in Fig. 1 we plot the normalized estimation error squared (NEES) for the IMU state, the sensors' extrinsic calibration and the time offset, each averaged over all Monte-carlo trials. For a variable $\mathbf{a}$, the NEES at time step k of a given trial is computed as $\tilde{\mathbf{a}}_k^T \mathbf{P}_{\mathbf{a}_k}^{-1} \tilde{\mathbf{a}}_k$, where $\tilde{\mathbf{a}}_k$ is the



(a)

(b)

(c)

Fig. 2: Map-based estimation: real-world experiment. (a) The area where the motion took place. (b) The estimated trajectory. The red circle is the starting point of the trajectory, and the blue asterisks the LED lights. (c) The estimation result for t_d.

estimation error and $\mathbf{P}_{\mathbf{a}_k}$ is the covariance matrix reported by the filter. If the estimator is consistent, i.e., 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) [20]. Fig. 1 shows that the average NEES values for the three variables examined are close to their theoretical 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.

2) Real-world Experiment: In addition to the simulation tests, we carried out a real-world experiment to validate the proposed map-based EKF. The vision-inertial system consisted of a PointGrey Bumblebee2 stereo pair (only a single camera was used) and an Xsens MTI-G unit. The environment is shown in Fig. 2a. The blue LED lights, whose positions are accurately known, are used as the visual features. 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 known positions in the trajectory. The estimated trajectory is shown in Fig. 2b. For the known positions in the trajectory, the maximum estimation error was 4.6 cm, in the same order of magnitude as what we observed in the simulations.

In Fig. 2c 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. In addition to being practically significant, this also suggests that t_d is *observable*, a result that we will seek to prove in future work. We point out that the standard deviation of t_d over the last one minute of the experiment is only 0.40 msec, which shows the high precision attainable by the proposed online estimation method.

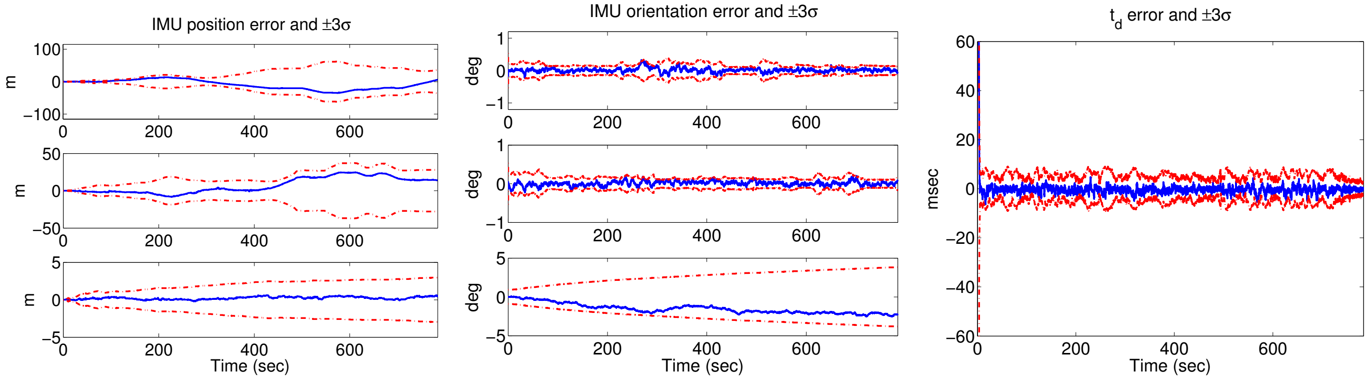
B. Motion estimation with unknown features

For the tests presented in this section, we implemented the online estimation of the time offset and camera-to-IMU extrinsic parameters in the modified MSCKF (multi-state constraint Kalman filter) algorithm presented in [6]. This algorithm performs visual-inertial odometry, and is a pose-based method (Section III-B).

1) Monte-Carlo simulations: To obtain realistic simulation environments, we generated the simulation data based on a real-world dataset. Specifically, the ground truth trajectory (position, velocity, orientation) for the simulation is generated by using the estimates computed by a GPS-INS system in a real-world dataset, which was about 13 minutes, 5.5 km long. Using these trajectories, we subsequently generated IMU measurements corrupted with noise and biases, as well as visual feature tracks with characteristics identical to those in the real-world data. For each trial the camera-to-IMU extrinsic parameters and the time offset were generated in a way identical to the map-based simulations, by perturbing known nominal values.

In the tests presented here, we compare the estimation performance in four cases. (i) camera-to-IMU calibration enabled, but t_d estimation disabled, (ii) t_d estimation enabled, but camera-to-IMU calibration disabled, (iii) both t_d and camera-to-IMU estimation enabled, and (iv) the case where t_d and the camera-to-IMU transformation are perfectly known and not estimated. In the first three cases (termed the “imprecise” ones), the precise values of the camera-to-IMU extrinsic parameters 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 II 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 NED (North-East-



(a) (b) (c)

Fig. 3: Visual-inertial odometry with unknown features and drifting time-offset t_d: estimation errors (blue lines) and associated $\pm 3\sigma$ envelopes (red dashed lines). (a) The IMU position errors in the North, East, Down directions, (b) The IMU orientation errors in roll, pitch, and 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.

TABLE II

Scenario	imprecise			precise	
Extrinsic calib.	on	off	on	N/A	
t_d estimation	off	on	on	N/A	
Pose RMSE	54.60	18.39	8.11	7.93	North (m)
	81.82	13.50	5.18	5.00	East (m)
	14.53	45.07	0.64	0.53	Down (m)
	0.39	0.18	0.06	0.06	roll (°)
	0.33	0.18	0.05	0.05	pitch (°)
	1.19	1.22	0.70	0.69	yaw (°)
IMU state NEES	85.4	2046	14.6	14.5	
Calib. RMSE	0.07	N/A	0.01	N/A	${}^{C}\mathbf{p}_I$ (m)
	0.31	N/A	0.05	N/A	${}^{C}_{I}\bar{\mathbf{q}}$ (°)
	N/A	0.28	0.25	N/A	t_d (msec)

Down) frame, and IMU orientation in roll-pitch-yaw. In these results, we first observe that, to be able to accurately estimate the IMU's motion, both the extrinsic calibration 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 degraded considerably (see the first two columns in Table II). Moreover, by comparing the third and fourth columns, we can see that the accuracy obtained by our online estimation approach is very close to that obtained when the the camera-to-IMU configuration and time offset are perfectly known. This is significant from a practical standpoint: it shows that there may not be a pressing need to perform tedious offline calibration or precise measurements of these parameters. By using the proposed online estimation approach, initialized with rough estimates, we can obtain pose estimates of quality almost indistinguishable to what we would get if an oracle provided us with the exact values of the parameters.

2) Time-varying t_d: For all the results presented up to now, a constant time offset was used. Next, we examine the case of a time-varying t_d. Instead of presenting Monte-Carlo simulation results (which are similar to those in Table II), it is interesting to show the results of a single representative trial. In this trial, the time offset varies linearly from 50 msec at the start, to 300 msec at the end, modelling a severe clock drift (250 msec in 13 minutes). Fig. 3 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.

3) Real-world Experiment: The visual-inertial odometry approach with concurrent estimation of the camera-to-IMU transformation and the time offset t_d was also tested in a real-world experiment. For this test, the camera-IMU system was mounted on the roof of a car driven for approximately 7.3 km. Feature extraction is performed via an optimized version of the Shi-Tomasi algorithm [21, 22] and matching is done by normalized cross-correlation. The estimated trajectory is plotted on a map of the area in Fig. 4, and compared to (i) ground truth obtained by a GPS-INS system, and (ii) the estimate computed without online estimation of the camera-to-IMU extrinsic parameters and t_d (for the extrinsic parameters manual measurements were used, and $t_d = 0$ was assumed in this case).

Similarly to what was observed in the previous cases, we see that the estimates obtained with online calibration are very precise (the error remains below 0.5% of the traveled distance). Moreover, these estimates are significantly better than the estimates obtained if online estimation is not used. In Fig. 5, we plot the estimate of t_d during the experiment. Similarly to Fig. 2c, we see that the estimate of t_d quickly converges close to its final value, and remains almost unchanged for the remainder of the trajectory. This shows that high-quality estimates for the time offset can be obtained quickly with the proposed approach, even in the absence of known landmark points.

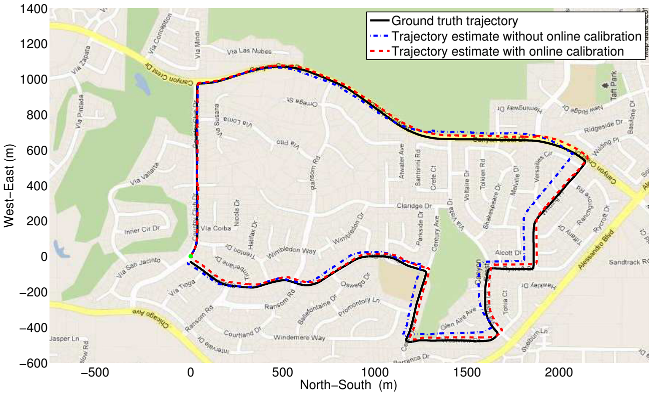


Fig. 4: Visual-inertial odometry experimental results. The trajectory estimate with the proposed approach (red dashed line), the estimate obtained without online calibration (blue dash-dotted line), and the ground truth (black solid line).

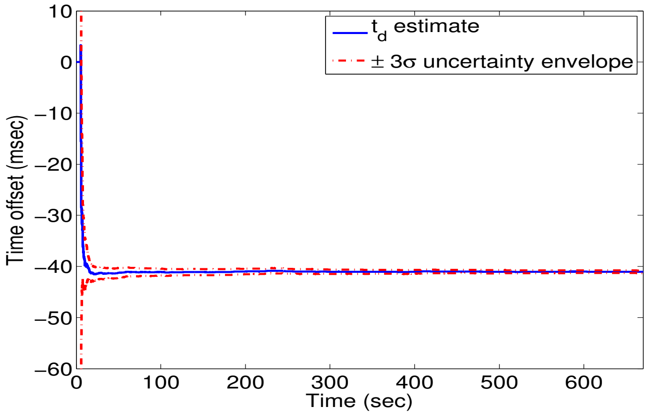


Fig. 5: The estimate for t_d and its uncertainty during the experiment.

V. Discussion

In this paper we have proposed an approach for the online estimation of 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. This makes it possible to track time-varying offsets, characterize the uncertainty in the estimate of t_d, and model the impact of this uncertainty on the pose estimation accuracy. Our simulation and experimental results indicate that the proposed approach leads to high-precision estimates for both the system motion, as well as for the temporal and spatial alignment between the camera and IMU. These results indicate that, at least in the trajectories used in our experiments, the time offset can be estimated using the sensor data (i.e., it is observable). In our future work, we plan to perform a detailed analysis to identify the conditions that guarantee observability.

References

[1] A. Wu, E. Johnson, and A. Proctor, "Vision-aided inertial navigation for flight control," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 9, pp. 348–360, Sep. 2005.

[2] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. Montgomery, "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, May 2007.

[3] E. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[4] P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos, "Inertial aiding of inverse depth SLAM using a monocular camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 2797–2802.

[5] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[6] M. Li and A. I. Mourikis, "Improving the accuracy of EKF-based visual-inertial odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation*, St Paul, MN, May 2012, pp. 828–835.

[7] S. Weiss, M. Achtelik, M. Chli, and R. Siegwart, "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, May 2012.

[8] K. Zhang, X. R. Li, and Y. Zhu, "Optimal update with out-of-sequence-measurements," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 1992–2005, 2005.

[9] M. Bak, T. Larsen, M. Norgaard, N. Andersen, N. K. Poulsen, and O. Ravn, "Location estimation using delayed measurements," in *Proceedings of the IEEE International Workshop on Advanced Motion Control*, Coimbra, Portugal, July 1998.

[10] S. J. Julier and J. K. Uhlmann, "Fusion of time delayed measurements with uncertain time delays," in *Proceedings of the American Control Conference*, Portland, OR, June 2005, pp. 4028 – 4033.

[11] M. Choi, J. Choi, J. Park, and W. K. Chung, "State estimation with delayed measurements considering uncertainty of time delay," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3987 –3992.

[12] J. Kelly and G. S. Sukhatme, "A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors," in *Proceedings of the International Symposium of Experimental Robotics*, New Delhi, India, December 2010.

[13] F. Tungadi and L. Kleeman, "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, December 2010.

[14] J. Kelly and G. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[15] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.

[16] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 6D pose estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Jan. 2005.

[17] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *International Journal of Robotics Research*, 2013, to appear.

[18] P. S. Maybeck, *Stochastic Models, Estimation and Control*, ser. Mathematics in Science and Engineering. London: Academic Press, 1982, vol. 141-2.

[19] B. Williams, N. Hudson, B. Tweddle, R. Brockers, and L. Matthies, "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, 2011, pp. 5655–5662.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[21] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994, pp. 593–600.

[22] M. Li and A. I. Mourikis, "Vision-aided inertial navigation for resource-constrained systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 1057–1063.