Sensor-Failure-Resilient Multi-IMU Visual-Inertial Navigation

Kevin Eckenhoff[†], Patrick Geneva[*], and Guoquan Huang[†]

***Abstract*— In this paper, we present a real-time multi-IMU visual-inertial navigation system (mi-VINS) that utilizes the information from *multiple* inertial measurement units (IMUs) and thus is resilient to IMU sensor failures. In particular, in the proposed mi-VINS formulation, one of the IMUs serves as the "base" of the system, while the rest act as auxiliary sensors aiding in state estimation. A key advantage of this architecture is the ability to seamlessly "promote" an auxiliary IMU as a new base, for example, upon detection of the base IMU failure, thus being *resilient* to the single point of sensor failure as seen in conventional VINS. Moreover, in order to properly fuse the information of multiple IMUs, both the spatial (relative pose) and temporal (time offset) calibration parameters between each sensor and the base IMU are estimated *online*. The proposed mi-VINS with online spatial and temporal calibration is validated in both simulations and real-world experiments, and is shown to be able to provide accurate localization and calibration even in scenarios with IMU sensor failures.**

I. Introduction

With recent advances in hardware design and manufacturing, low-cost light-weight MEMS inertial measurement units (IMUs) have become ubiquitous. which enables high-accuracy positioning for mobile devices [1] and micro aerial vehicles (MAVs) [2], holding huge implications in a wide range of practical applications from mobile augmented reality to autonomous driving, in part due to their ability to directly measure the dynamics of a moving object at high frequency. The measured angular velocity and local linear acceleration provided by these inertial sensors give them remarkable accuracy in predicting short-term dynamic motion, and thus are ideal for fusion with lower-rate exteroceptive sensors such as the cameras found in visual-inertial navigation systems (VINS) [3]. VINS technologies have emerged in recent years (e.g., see [4–8]), in part because of the complementary sensing capabilities of the sensors (IMU and camera) and their decreasing costs and sizes.

The vast majority of current VINS have primarily focused on a *single* IMU and camera (both monocular and stereo) [3, 7–10]. While such a sensor pair is considered as the minimal sensing capability for 3D motion estimation, clearly it is *not* resilient to sensor failure. In reality, sensors certainly may experience failures preventing the estimator from acquiring new measurements from the faulty sensor. If this sensor (such as IMUs in VINS) is required to fully constrain the estimation problem, its failure will result in the collapse of the entire system. Such failures can occur in practice due to sensor disconnection (due to impact), high temperatures, or sensitivity to vibrations [11]. To compensate for this issue, redundant sensors (i.e., hardware redundancy) are typically used [12]. Therefore, adding more IMUs into VINS appears to be a straightforward solution for improving the system resilience against sensor failures, in particular, given the low cost of IMUs. However, to the best of our knowledge, *few* VINS utilize *multiple* IMUs while performing real-time estimation.

In this work, we take up this challenge by developing a multi-IMU VINS (mi-VINS) algorithm resilient to IMU sensor failures. In particular, we propose an EKF-based estimation architecture that generalizes the multi-state constraint Kalman filter (MSCKF) from a single IMU setup [3] to a multi-IMU case, such that the proposed estimator can readily utilize all the information from multiple IMUs and can seamlessly recover from unit failures. Moreover, in order to properly fuse the information of multiple IMUs, the proposed mi-VINS performs online sensor calibration of both the spatial (relative pose) and temporal (time offset) calibration parameters between sensors.

Specifically, the main contributions of this paper include:

- We propose a tightly-coupled EKF-based estimation architecture that optimally fuses asynchronous measurements from multiple IMUs and a camera. In particular, we maintain and propagate each IMU's state estimate as well as the joint covariance, and by arbitrarily choosing one IMU as the base whose poses are stochastically cloned as in the standard MSCKF, we can perform EKF update using both visual measurements and rigid-body relative pose constraints between the IMUs.
- We perform *online* sensor calibration refinement of both the spatial *and* temporal calibration parameters between all sensors, allowing us to consistently fuse their asynchronous measurements without the need to perform a tedious offline calibration process, instead relying on only rough initial guesses.
- The proposed mi-VINS is resilient to IMU sensor failure. If the base IMU fails, an auxiliary IMU is "promoted" as a new base, offering uninterrupted localization solutions via estimate and covariance propagation.
- The proposed mi-VINS is validated in both simulations and real experiments, and is shown to provide high-precision navigation even in cases of sensor failures.

II. Related Work

To date, various algorithms are available for VINS problems, among which the MSCKF [3] appears to be one of

This work was partially supported by the University of Delaware College of Engineering, UD Cybersecurity Initiative, the NSF (IIS-1566129), the DTRA (HDTRA1-16-1-0039), and Google Daydream.

[†]The authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {keck,ghuang}@udel.edu

[*]The author is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. Email: pgeneva@udel.edu

the first real-time solutions. Their method uses quaternion dynamics [13] to process noisy inertial measurements of a *single* IMU in the EKF propagation, while camera measurements to point features are processed by projecting the measurement residuals onto the null space of the feature Jacobians, thereby creating measurements that rely only on the stochastic clones of the IMU/camera poses. The MSCKF remains popular primarily due to its efficiency, and has been extended in different directions [1, 4–6, 14–16]. On the other hand, batch optimization-based methods solve a nonlinear least-squares problem over the entire history (or a sliding window) of measurements and tend to achieve higher accuracy but at the cost of increased computational burden (e.g., [7, 17, 18]). Nevertheless, all these VINS algorithms remain vulnerable to single IMU failure.

While outside of VINS, fusing multiple IMUs has been widely studied [19], e.g., with the application to human motion tracking [20], these methods neither perform visual-inertial fusion nor online spatial/temporal calibration as in this work. Ma et al. [21] fused a tactical grade IMU, stereo camera, leg odometry, and GPS measurements in an EKF alongside a navigation-grade *gyroscope* for estimating the motion of a quadruped robot, but without calibration or the ability to use acceleration measurements from a second IMU.

When calibrating visual-inertial systems, substantial research efforts have been focused on *offline* processes that often require additional calibration aids (fiducial tags) [22–25]. For instance, Rehder et al. [22] used a continuous-time basis function representation [26] of the sensor trajectory to calibrate both the extrinsics and intrinsics of a multi-sensor system in a batch fashion. As this B-spline representation allows for the direct computation of expected local angular velocity and local linear acceleration, the difference between the expected and measured inertial readings served as errors in the batch optimization formulation.

A downside of offline calibration is that a time-intensive recalibration must be performed every time a sensor suite is even slightly reconfigured. Online calibration methods, by contrast, estimate the calibration parameters during every operation of the sensor suite, thereby making them more robust to, and easier to use, in such scenarios. Kim, Shin, and Kweon [27] reformulated IMU preintegration [18, 28, 29] by transforming the inertial readings from the IMU frame into a second frame. This allowed for spatial calibration with online initialization between an IMU and other sensors (including other IMUs), but did not include temporal calibration while also relying on computing angular accelerations from gyroscope measurements. Li and Mourikis [8] performed navigation with simultaneous calibration of both the spatial and temporal extrinsics between a single IMU-camera pair in a filtering framework for use on mobile devices, which was later extended to include the intrinsics of both the camera and the IMU [30], while considering a single IMU case. Qin and Shen [31] extended their prior work on batch-based monocular VINS [7] to include estimating the time offset between a camera and a single IMU by interpolating the location of features on the image plane. However, neither of these approaches are suitable to include online spatial/temporal calibration of *multiple* IMUs or to address the resilience to *sensor failures*.

III. Multi-IMU VINS

In this section, we present in detail the proposed multi-IMU VINS (mi-VINS) that fuses measurements from both multiple IMUs and a camera within the MSCKF framework.

A. State Vector

As compared to the standard MSCKF [3] that uses only a single IMU, we generalize it to incorporate an inertial state for *each* IMU on the sensor platform allowing for update and propagation of each. If $(N+1)$ IMUs are rigidly mounted on the sensor platform, their state vector is given by:

$$\mathbf{x}_I = \begin{bmatrix} \mathbf{x}_{I_0}^\top & \cdots & \mathbf{x}_{I_N}^\top \end{bmatrix}^\top \tag{1}$$

where $\mathbf{x}_{I_i}$ is the navigation state of the i-th IMU [13]:

$$\mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}_{G}\bar{q}^\top & \mathbf{b}_{i\omega}^\top & {}^{G}\mathbf{v}_{I_i}^\top & \mathbf{b}_{ia}^\top & {}^{G}\mathbf{p}_{I_i}^\top \end{bmatrix}^\top \tag{2}$$

In the above expression, ${}^{I_i}_{G}\bar{q} = [\mathbf{q}^\top \ q_4]^\top$ is the JPL unit quaternion [13] parameterizing the rotation ${}^{I_i}_{G}\mathbf{R}$ from the global frame $\{G\}$ to the i-th IMU local frame $\{I_i\}$, $\mathbf{b}_{i\omega}$ and $\mathbf{b}_{ia}$ are the gyroscope and accelerometer biases, and ${}^{G}\mathbf{v}_{I_i}$ and ${}^{G}\mathbf{p}_{I_i}$ are the velocity and position of the i-th IMU.

To perform EKF estimation, we define the corresponding error state of the i-th IMU as follows:

$$\delta\mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}\delta\boldsymbol{\theta}_G^\top & \delta\mathbf{b}_{i\omega}^\top & {}^{G}\delta\mathbf{v}_{I_i}^\top & \delta\mathbf{b}_{ia}^\top & {}^{G}\delta\mathbf{p}_{I_i}^\top \end{bmatrix}^\top \tag{3}$$

The relationship between the true value of the state, $\mathbf{x}_{I_i}$, the estimated value $\hat{\mathbf{x}}_{I_i}$, and the error state, $\delta\mathbf{x}_{I_i}$, can be written in terms of the generalized update operation [32]:

$$\mathbf{x}_{I_i} = \hat{\mathbf{x}}_{I_i} \boxplus \delta\mathbf{x}_{I_i} \tag{4}$$

For vector quantities, $\mathbf{v}$, this operation is simply addition, i.e., $\mathbf{v} = \hat{\mathbf{v}} + \delta\mathbf{v}$, while for quaternions, we have $\bar{q} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^\top & 1 \end{bmatrix}^\top \otimes \hat{\bar{q}}$, where $\otimes$ denotes quaternion multiplication [13]. Moreover, to allow for utilizing visual feature measurements, we select an *arbitrary* IMU to serve as the "base", denoted by $\{I_b\}$, which can be changed over the trajectory if needed. We will keep a historical window of stochastically cloned poses (positions and orientations) of this base IMU. Specifically, at time step k (which is corresponding to the k-th received image), we also maintain a sliding window of the base IMU clones at M past imaging times t_j $(j = k-M+1, \cdots, k)$, alongside with $\mathbf{x}_I$:

$$\mathbf{x}_{cl} = \begin{bmatrix} {}^{I_b(t_k)}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_b(t_k)}^\top & \cdots & {}^{I_b(t_{k-M+1})}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_b(t_{k-M+1})}^\top \end{bmatrix} \tag{5}$$

Lastly, as will be explained in further sections, we also estimate *online* the spatial and temporal calibration parameters, $\mathbf{x}_e$, between each sensor and the base IMU. In particular, we maintain in our calibration state the relative pose between the base IMU and camera, ${}^{b}\mathbf{x}_C = \begin{bmatrix} {}^{C}_{I_b}\bar{q}^\top & {}^{C}\mathbf{p}_{I_b}^\top \end{bmatrix}^\top$, as well as the pose between each auxiliary IMU and the base, ${}^{b}\mathbf{x}_i = \begin{bmatrix} {}^{I_i}_{I_b}\bar{q}^\top & {}^{I_b}\mathbf{p}_{I_i}^\top \end{bmatrix}^\top$. In addition to these spatial quantities, we also estimate the time offset between the camera and base IMU, ${}^{c}t_b$, as well as the time offsets between each auxiliary

IMU and the base, ${}^{i}t_b$. This results in our total state vector as [see (1) and (5)]:

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_{cl}^\top & \mathbf{x}_e^\top\end{bmatrix}^\top = \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_S^\top\end{bmatrix}^\top \tag{6}$$

$$\mathbf{x}_e = \begin{bmatrix}{}^{b}\mathbf{x}_C^\top & {}^{b}\mathbf{x}_0^\top & \dots & {}^{b}\mathbf{x}_N^\top & {}^{c}t_b & {}^{0}t_b & \dots & {}^{N}t_b\end{bmatrix}^\top \tag{7}$$

where $\mathbf{x}_S$ is the set of *static* quantities whose true values do not evolve over time.

B. mi-VINS Propagation

IMU (gyroscope and accelerometer) measurements, which are used to propagate state estimates and covariance, are given by:

$$\boldsymbol{\omega}_{im} = \boldsymbol{\omega}_i + \mathbf{b}_{i\omega} + \mathbf{n}_{i\omega} \tag{8}$$

$$\mathbf{a}_{im} = \mathbf{a}_i + {}^{I_i}_{G}\mathbf{R}\,{}^{G}\mathbf{g} + \mathbf{b}_{ia} + \mathbf{n}_{ia} \tag{9}$$

where $\boldsymbol{\omega}_i$ and $\mathbf{a}_i$ are the true angular velocity and local linear acceleration of the i-th IMU, $\mathbf{n}_{i\omega}$ and $\mathbf{n}_{ia}$ are continuous-time Gaussian white noises corrupting the measurements, and ${}^{G}\mathbf{g} \simeq [0\ 0\ 9.81]^\top$ is the gravity in the global frame.

At the current time step k (corresponding to the current imaging time t_k), we propagate the current state estimate forward to the next time step $k+1$ (corresponding to the next new image time t_{k+1}), by using all the IMU measurements available in the time window $[t_k, t_{k+1}]$, which are denoted by $\mathcal{I}_i$ for the i-th IMU ($i = 0, \cdots, N$), based on the IMU dynamics $\mathbf{f}(\cdot)$ [33]:[1]

$$\mathbf{x}_{I_i}(t_{k+1}) = \mathbf{f}\left(\mathbf{x}_{I_i}(t_k), \mathcal{I}_i, \mathbf{n}_{I_i}\right) \tag{10}$$

$$\Rightarrow \hat{\mathbf{x}}_{I_i}(t_{k+1}|t_k) = \mathbf{f}\left(\hat{\mathbf{x}}_{I_i}(t_k|t_k), \mathcal{I}_i, \mathbf{0}\right) \tag{11}$$

where $\mathbf{n}_{I_i}$ is the stacked vector of the IMU noises. The error covariance is propagated as follows (see [13]):

$$\mathbf{P}(t_{k+1}|t_k) = \boldsymbol{\Phi}_k \mathbf{P}(t_k|t_k) \boldsymbol{\Phi}_k^\top + \mathbf{Q}_k \tag{12}$$

$$\boldsymbol{\Phi}_k = \mathbf{Diag}\left(\boldsymbol{\Phi}_0(t_{k+1}, t_k), \dots, \boldsymbol{\Phi}_N(t_{k+1}, t_k), \mathbf{I}\right) \tag{13}$$

$$\mathbf{Q}_k = \mathbf{Diag}\left(\mathbf{Q}_0, \dots, \mathbf{Q}_N, \mathbf{0}\right)$$

where $\boldsymbol{\Phi}_i(t_{k+1}, t_k)$ is the linearized state-transition matrix for the error state of the i-th IMU across the time interval $[t_k,\ t_{k+1}]$ and $\mathbf{Q}_i$ is the corresponding noise covariance, while $\mathbf{Diag}(\cdot,\cdot,\cdot)$ places the argument matrix entries on the block diagonals of an otherwise zero matrix. Each of these matrices are computed *per IMU* using its measurements. Note that in the above, the identity matrix $\mathbf{I}$ in $\boldsymbol{\Phi}_k$ and the right-bottom zero matrix $\mathbf{0}$ in $\mathbf{Q}_k$ correspond to the static states [see (5) and (6)].

C. mi-VINS Update

To limit the navigation drift accumulated over propagation, we utilize both multi-IMU rigid constraints and camera visual measurements to perform efficient EKF updates.

1) Enforcing multi-IMU constraints: As all IMUs are rigidly connected, at any time t we have the following relative transformation between the base and non-base IMUs:[2]

$${}^{I_i}_{I_b}\bar{q} = {}^{I_i(t)}_{G}\bar{q} \otimes {}^{I_b(t)}_{G}\bar{q}^{-1}, \quad {}^{I_b}\mathbf{p}_{I_i} = {}^{I_b(t)}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{I_i(t)} - {}^{G}\mathbf{p}_{I_b(t)}\right)$$

where ${}^{I_i}_{I_b}\bar{q}$ and ${}^{I_b}\mathbf{p}_{I_i}$ are the *fixed* relative pose between the base IMU b and the i-th IMU. The residual associated with this constraint for each IMU can be written as:

$$\begin{cases} 2\text{vec}\left({}^{I_i(t)}_{G}\bar{q} \otimes {}^{I_b(t)}_{G}\bar{q}^{-1} \otimes {}^{I_i}_{I_b}\bar{q}^{-1}\right) & = \mathbf{0} \\ {}^{G}\mathbf{p}_{I_i}(t) - {}^{G}\mathbf{p}_{I_b}(t) - {}^{G}_{I_b(t)}\mathbf{R}\,{}^{I_b}\mathbf{p}_{I_i} & = \mathbf{0} \end{cases} \Rightarrow \mathbf{r}_{I_i}(\mathbf{x}) = \mathbf{0} \tag{14}$$

where $\text{vec}(\bar{q}) = \mathbf{q}$ returns the vector portion of the argument quaternion $\bar{q}$. We stack this constraint for each auxiliary IMU to form a system of residuals, $\mathbf{r}_I(\mathbf{x}) = \mathbf{0}$. Linearization of these residuals at the current state estimate yields:

$$\mathbf{r}_I(\hat{\mathbf{x}}) + \frac{\partial \mathbf{r}_I}{\partial \delta\mathbf{x}}\delta\mathbf{x} = \mathbf{0} \;\Rightarrow\; \mathbf{r}_I(\hat{\mathbf{x}}) = \mathbf{H}_I \delta\mathbf{x} \tag{15}$$

Note that this is a *hard* constraint that acts as a measurement with *zero* noise. In practice, such constraints may quickly degrade performance due to inaccuracies in the calibrated transforms between sensors, which motivates us to perform online calibration of these parameters. It should be noted that such relative-pose constraints have been used in previous multi-IMU systems [34], along with a constraint on the relationship between the IMUs' velocities. However, transferring velocities from one frame to another requires angular velocity measurements, which have already been used in the propagation step. As such, we choose to forgo this constraint to ensure consistency. In addition, as explained in Section IV, we utilize these relative-pose constraints to perform both spatial and temporal calibration of the sensors.

2) Visual feature measurements: Before diving into online sensor calibration, we describe how visual feature measurements are used for EKF update, by assuming we have performed stochastic cloning at the "true imaging time". Consider a 3D feature, ${}^{G}\mathbf{p}_f$, which is measured at the true image time t_m at time step m, and is given by:

$$\mathbf{z}_m = \boldsymbol{\Pi}\left({}^{C}\mathbf{p}_f\right) + \mathbf{n}_m = \begin{bmatrix}\frac{x}{z} \\ \frac{y}{z}\end{bmatrix} + \mathbf{n}_m$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^{C}\mathbf{p}_f = {}^{C}_{I_b}\mathbf{R}\,{}^{I_b(t_m)}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_f - {}^{G}\mathbf{p}_{I_b}(t_m)\right) + {}^{C}\mathbf{p}_{I_b}$$

where $\mathbf{n}_m$ is zero-mean Gaussian noise, and ${}^{C}_{I_b}\mathbf{R}$ and ${}^{C}\mathbf{p}_{I_b}$ represent the spatial calibration parameters between the base IMU and the camera. With the measurement residual being defined as $\mathbf{r}_m = \mathbf{z}_m - \boldsymbol{\Pi}\left({}^{C}\hat{\mathbf{p}}_f\right)$, and after computing the Jacobian and residual for each measurement corresponding to this feature that has been tracked over the sliding window, we formulate the following linearized residual system:

$$\mathbf{r}_C \approx \mathbf{H}_x \delta\mathbf{x} + \mathbf{H}_f\,{}^{G}\delta\mathbf{p}_f + \mathbf{n}_C \tag{16}$$

where $\mathbf{r}_C$, $\mathbf{H}_x$, $\mathbf{H}_f$, and $\mathbf{n}_C$ are the stacked residuals, measurement Jacobians with respect to the state and feature, and measurement noises. The key idea of MSCKF is to project (16) onto the null space of $\mathbf{H}_f$ using matrix $\mathbf{Q}_2$ whose columns span this space, arriving at a new measurement residual that does *not* depend on the feature, and thus does not require storage in our state vector (6):

$$\mathbf{Q}_2^\top \mathbf{r}_C \approx \mathbf{Q}_2^\top \mathbf{H}_x \delta\mathbf{x} + \mathbf{Q}_2^\top \mathbf{H}_f\,{}^{G}\delta\mathbf{p}_f + \mathbf{Q}_2^\top \mathbf{n}_C \tag{17}$$

$$\Rightarrow \mathbf{r}'_C = \mathbf{H}'_x \delta\mathbf{x} + \mathbf{n}'_C \tag{18}$$

At this point, we have derived both the multi-IMU measurement residuals $\mathbf{r}_I$ (15) and the camera measurement

[1] Throughout the paper, $\hat{\mathbf{x}}(t_\ell|t_k)$ or $\hat{\mathbf{x}}_{\ell|k}$ denotes the state estimate $\hat{\mathbf{x}}$ at time t_ℓ using all measurements up to time t_k.

[2] Note that while in this work we assume the spatial calibration parameters remain static, i.e., that the sensors are rigidly mounted, they can also be modelled as random walks when dealing with a more flexible mount. Such a treatment can also be extended to the temporal parameters.

residuals $\mathbf{r}'_C$ (18), which depend only on the considered states (6). With these linearized measurement residuals, we can perform the standard EKF update [35].

IV. Online Spatial/Temporal Sensor Calibration

Two common sources of error in practical VI-systems arise from temporal and spatial calibration inaccuracies. To combat these, in this section, we incorporate both spatial and temporal calibration between all IMUs as well as the camera into the proposed mi-VINS presented in the previous section.

A. IMU-IMU Spatial/Temporal Calibration

In the case of a multi-IMU system that consists of *asynchronous* independent inertial sensors with non-negligible time offsets relative to the base IMU clock, in addition to errors in the rigid transformation estimates, inaccurate time offsets can greatly impact the multi-IMU constraints (14) and thus the estimation, which motivates us to perform online calibration of these parameters.

Consider an event whose time as expressed in the base IMU's clock, ${}^b t$, is related to the same time in the i-th IMU's clock, ${}^i t$, by an *unknown*, static time offset ${}^i t_b$:[3]

$$ {}^b t = {}^i t + {}^i t_b \tag{19} $$

To determine this time offset ${}^i t_b$, we estimate it online as an additional random variable included in our state.

In order to correctly enforce the asynchronous multi-IMU relative pose constraint (14), each IMU must be expressed at the *same* time, and therefore this time offset must be compensated for. To this end, we propagate the i-th IMU's state up to the *estimate* of the current base IMU time as expressed in the i-th clock, ${}^i\hat{t} = {}^b t - {}^i\hat{t}_b$. Explicitly, the IMU state after propagation at time step $k+1$ can be written as [see (1)]:

$$ \mathbf{x}_I({}^b t_{k+1}) = \begin{bmatrix} \mathbf{x}_{I_0}({}^0\hat{t}_{k+1})^\top & \cdots & \mathbf{x}_{I_N}({}^N\hat{t}_{k+1})^\top \end{bmatrix}^\top $$

where $\mathbf{x}_{I_i}({}^i\hat{t}_{k+1})$ is the true value of the i-th IMU at the estimated time ${}^i\hat{t}_{k+1}$ in its own clock.

Using the first-order approximation for the motion of each IMU, the state of the i-th IMU at the time of the base IMU is a function of the state at the estimated time and the error in the time offset estimate, i.e.,

$$ \begin{aligned} {}^G\mathbf{p}_{I_i}({}^i t_{k+1}) &= {}^G\mathbf{p}_{I_i}({}^b t_{k+1} - {}^i\hat{t}_b - \delta{}^i t_b) = {}^G\mathbf{p}_{I_i}({}^i\hat{t}_{k+1} - \delta{}^i t_b) \\ &\approx {}^G\mathbf{p}_{I_i}({}^i\hat{t}_{k+1}) - {}^G\mathbf{v}_{I_i}({}^i\hat{t}_{k+1})\delta{}^i t_b \end{aligned} \tag{20} $$

$$ {}^{I_i({}^i t_{k+1})}_{G}\mathbf{R} \approx \text{Exp}\left(\boldsymbol{\omega}_i({}^i\hat{t}_{k+1})\delta{}^i t_b\right) {}^{I_i({}^i\hat{t}_{k+1})}_{G}\mathbf{R} \tag{21} $$

where $\text{Exp}(\cdot)$ is the matrix exponential mapping an axis-angle to a rotation matrix and $\boldsymbol{\omega}_i({}^i\hat{t}_{k+1})$ is the angular velocity of the i-th IMU. With that, we rewrite (14) in a residual form (for simplicity time index $k+1$ is dropped):

$$ \mathbf{r}_{\theta_i} = 2\text{vec}\left(\begin{bmatrix} -\frac{1}{2}\boldsymbol{\omega}_i({}^i\hat{t})\delta{}^i t_b \\ 1 \end{bmatrix} \otimes {}^{I_i({}^i\hat{t})}_{G}\bar{q} \otimes {}^{I_b({}^b t)}_{G}\bar{q}^{-1} \otimes {}^{I_i}_{I_b}\bar{q}^{-1} \right) \tag{22} $$

$$ \mathbf{r}_{p_i} = {}^G\mathbf{p}_{I_i}({}^i\hat{t}) - {}^G\mathbf{v}_{I_i}({}^i\hat{t})\delta{}^i t_b - {}^G\mathbf{p}_{I_b}({}^b t) - {}^{G}_{I_b({}^b t)}\mathbf{R}\,{}^{I_b}\mathbf{p}_{I_i} \tag{23} $$

The Jacobians of these constraints can be computed as:

$$ \frac{\partial \mathbf{r}_{\theta_i}}{\partial \delta{}^i t_b} = -\mathbf{A}\boldsymbol{\omega}_i, \quad \frac{\partial \mathbf{r}_{\theta_i}}{\partial {}^{I_i}\delta\boldsymbol{\theta}_{I_b}} = -\mathbf{A}^\top, \quad \frac{\partial \mathbf{r}_{\theta_i}}{\partial {}^{I_i({}^i\hat{t})}\delta\boldsymbol{\theta}_G} = \mathbf{A} $$

$$ \frac{\partial \mathbf{r}_{\theta_i}}{\partial {}^{I_b({}^b t)}\delta\boldsymbol{\theta}_G} = -\Big((q_{l,4}\mathbf{I}_3 - \lfloor \mathbf{q}_l \rfloor)(q_{r,4}\mathbf{I}_3 + \lfloor \mathbf{q}_r \rfloor) - \mathbf{q}_l\mathbf{q}_r^\top \Big) $$

$$ \frac{\partial \mathbf{r}_{p_i}}{\partial {}^G\delta\mathbf{p}_{I_i}({}^i\hat{t})} = \mathbf{I}_3, \quad \frac{\partial \mathbf{r}_{p_i}}{\partial {}^G\delta\mathbf{p}_{I_b}({}^b t)} = -\mathbf{I}_3, \quad \frac{\partial \mathbf{r}_{p_i}}{\partial \delta{}^i t_b} = -{}^G\hat{\mathbf{v}}_{I_i}({}^i\hat{t}) $$

$$ \frac{\partial \mathbf{r}_{p_i}}{\partial {}^{I_b({}^b t)}\delta\boldsymbol{\theta}_G} = {}^{G}_{I_b({}^b t)}\hat{\mathbf{R}}\lfloor {}^{I_b}\hat{\mathbf{p}}_{I_i} \rfloor, \quad \frac{\mathbf{r}_{p_i}}{\partial {}^{I_b}\delta\mathbf{p}_{I_i}} = -{}^{G}_{I_b({}^b t)}\hat{\mathbf{R}} $$

where we have used these definitions: $\mathbf{A} = (q_{res,4}\mathbf{I}_3 + \lfloor \mathbf{q}_{res} \rfloor)$, $\bar{q}_{res} = {}^{I_i({}^i\hat{t})}_{G}\hat{\bar{q}} \otimes {}^{I_b({}^b t)}_{G}\hat{\bar{q}}^{-1} \otimes {}^{I_i}_{I_b}\hat{\bar{q}}^{-1}$, $\bar{q}_l = {}^{I_i({}^i\hat{t})}_{G}\hat{\bar{q}} \otimes {}^{I_b({}^b t)}_{G}\hat{\bar{q}}^{-1}$ and $\bar{q}_r = {}^{I_i}_{I_b}\hat{\bar{q}}^{-1}$. In addition $\lfloor \cdot \rfloor$ denotes the skew-symmetric matrix.

We should point out that the value of $\boldsymbol{\omega}_i$ used in the linearization [see (21)] comes from the i-th IMU's gyro measurements, i.e., $\boldsymbol{\omega}_i = \boldsymbol{\omega}_{im}({}^i\hat{t}) - \hat{\mathbf{b}}_{i\omega}({}^i\hat{t})$. Errors in this estimate are multiplied by $\delta{}^i t_b$, and thus do not affect the measurement up to first order. It is also important to note that, after update with the multi-IMU relative-pose constraints, the state of the i-th non-base IMU will still be at the *prior* estimate at time ${}^i\hat{t}_{k+1|k}$ (i.e., the time we last propagated the IMU i to), while we will have an *updated* estimate for the time offset ${}^i\hat{t}_b$. To compensate for this, when we receive a new camera image at time ${}^b t_{k+2}$, propagation for each IMU is actually performed across the interval $[{}^i\hat{t}_{k+1|k}, {}^i\hat{t}_{k+2|k+1}]$. In summary, we propagate each of the IMU states and the joint covariance using measurements from each IMU. The states are propagated to the same estimated imaging time, after which we enforce the derived spatial-temporal multi-IMU relative-pose constraints (22)-(23) between the base and auxiliary IMUs through the EKF update.

B. IMU-Camera Spatial/Temporal Calibration

We also calibrate the time offset and rigid transform between the base IMU and the camera in analogy to [30]. In particular, we model the time offset as [see (19)]:

$$ {}^b t = {}^c t + {}^c t_b \tag{24} $$

When an image is received at the camera time ${}^c t_m$, we first propagate the base IMU up to the estimate of the image time expressed in the base IMU clock, i.e., ${}^b\hat{t}_m = {}^c t_m + {}^c\hat{t}_b$. Then, we propagate each auxiliary IMU up to the estimated time of the base IMU, i.e., ${}^i\hat{t}_m = {}^b\hat{t}_m - {}^i\hat{t}_b$. We note that while we have an estimate for the base IMU at the estimated image time, the visual measurements are actually a function of the base IMU at the *true* image time. To compensate, we perform stochastic cloning to create the state estimate of the pose of the base IMU at the image time ${}^b t_m$ [30]:

$$ \begin{aligned} {}^G\mathbf{p}_{I_b}({}^b t_m) &\approx {}^G\mathbf{p}_{I_b}({}^b\hat{t}_m) + {}^G\mathbf{v}_{I_b}({}^b\hat{t}_m)\delta{}^c t_b \\ {}^{I_b({}^b t_m)}_{G}\mathbf{R} &\approx \text{Exp}\left(-\boldsymbol{\omega}_b\delta{}^c t_b\right) {}^{I_b({}^b\hat{t}_m)}_{G}\mathbf{R} \end{aligned} \tag{25} $$

After cloning, we perform the EKF update with the multi-IMU relative-pose constraints as well as the MSCKF visual measurements. We should point out that the order of operations is important, because updating before cloning would change the estimate of ${}^b t_m$, thereby making (25) invalid. For clarity, the main steps of the proposed mi-VINS with online sensor calibration are outlined in Algorithm 1.

[3] Throughout the paper, the left superscript of a time refers to its clock while the base IMU's clock is chosen as the reference clock of the system.

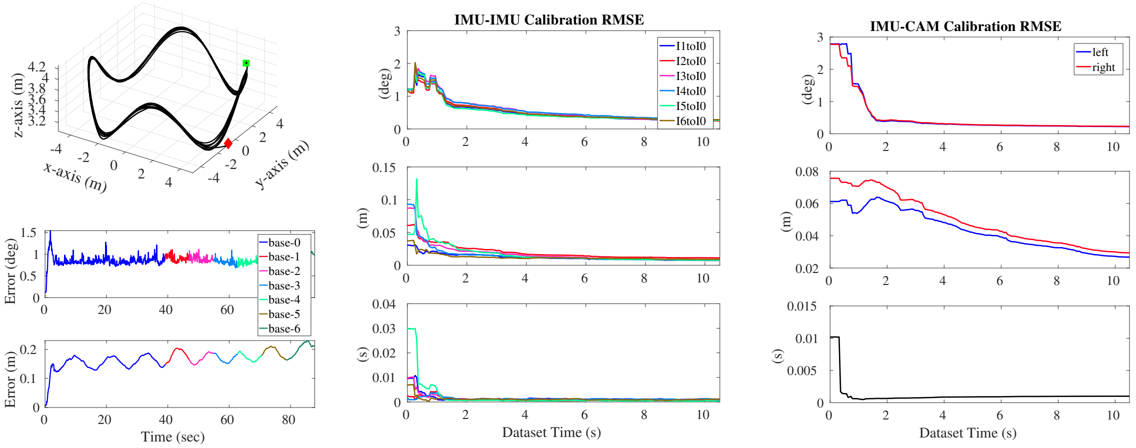

Fig. 1: Monte-Carlo simulation results: (top-left) 250 meter long 3D trajectory with the start and end locations denoted as green square and red diamond, respectively. (bottom-left) RMSE of pose estimates, color-coded segments based on which base IMU was active. (center) IMU-IMU calibration RMSE of the first 10 seconds of the dataset (as it converges after that). (right) IMU-CAM calibration RMSE for the first 10 seconds for the stereo pair transformation and time offset.

Algorithm 1 mi-VINS with online sensor calibration

```
1:  INPUT: New image available at time ${}^{c}t_m$
2:  if IMU failure detected then
3:      if Failed IMU is base then
4:          Switch to new IMU base.
5:      end if
6:      Marginalize failed IMU state.
7:  end if
8:  Propagate each IMU state estimates up to time ${}^{i}\hat{t}_m = {}^{c}t_m + {}^{c}\hat{t}_b - {}^{i}\hat{t}_b$ along with the joint covariance.
9:  Clone the base IMU pose at the imaging time ${}^{b}t_m$(25).
10: Perform feature tracking
11: Perform EKF update with the spatial-temporal multi-IMU relative-pose constraints (23) and MSCKF measurements (18).
```

V. Resilience to IMU Sensor Failures

In extreme environmental conditions, robustness to sensor failures is key to lifelong persistent localization. As compared to a failure of a stereo camera where the system can continue on monocular vision, failure of the IMU prevents VINS from performing state estimation. In this section, we describe how the proposed mi-VINS is resilient to IMU sensor failures and allows for up to N IMU sensors to fail before the system completely stops working.

In particular, we consider the most challenging scenario where the base IMU sensor fails, as when a non-base IMU sensor fails it is trivial to just marginalize its navigation state. During base failure, we "promote" an auxiliary sensor to be the new base (which is denoted as the n-th IMU). We then transform the quantities in our state to be expressed with respect to this new base. Specifically, each IMU clone refers to the base IMU sensor frame at the true imaging time, t_j, and we can write the following transformations for each:

$$ {}^{I_n(t_j)}_{G}\mathbf{R} = {}^{I_n}_{I_b}\mathbf{R}\,{}^{I_b(t_j)}_{G}\mathbf{R} \tag{26} $$

$$ {}^{G}\mathbf{p}_{I_n}(t_j) = {}^{G}\mathbf{p}_{I_b}(t_j) + {}^{I_b(t_j)}_{G}\mathbf{R}^{\top}\,{}^{I_b}\mathbf{p}_{I_n} \tag{27} $$

In the case of online spatial calibration, we propagate them to the new base IMU as follows:

$$ {}^{I_i}_{I_n}\mathbf{R} = {}^{I_i}_{I_b}\mathbf{R}\,{}^{I_n}_{I_b}\mathbf{R}^{\top},\ \ {}^{I_n}\mathbf{p}_{I_i} = {}^{I_n}_{I_b}\mathbf{R}\left({}^{I_b}\mathbf{p}_{I_i} - {}^{I_b}\mathbf{p}_{I_n}\right) \tag{28} $$

$$ {}^{C_k}_{I_n}\mathbf{R} = {}^{C_k}_{I_b}\mathbf{R}\,{}^{I_n}_{I_b}\mathbf{R}^{\top},\ \ {}^{C_k}\mathbf{p}_{I_n} = {}^{C_k}\mathbf{p}_{I_b} + {}^{C_k}_{I_b}\mathbf{R}\,{}^{I_b}\mathbf{p}_{I_n} \tag{29} $$

Lastly, we note that the relationship between any clock, the base clock, and the new base clock takes the form ${}^{n}t + {}^{n}t_b = {}^{i}t + {}^{i}t_b \Rightarrow {}^{i}t_n = {}^{i}t_b - {}^{n}t_b$. Using these constraints, we can modify the estimates such that the n-th IMU serves as the new base, through a proper mean and covariance propagation. This procedure can be triggered at any point, such as when base sensor failure is detected, allowing for *continuous, uninterrupted* estimation.

VI. Experimental Results

To validate the proposed mi-VINS we have performed both Monte-Carlo simulations and real-world tests, while focusing on evaluating the resilience to IMU sensor failures and the accuracy of online spatial and temporal calibration.

A. Monte-Carlo Simulations

In simulations, we consider a stereo visual-inertial (VI) system, while in our real-world tests shown later a monocular VI system is validated. Specifically, we simulated an Asctec Firefly UAV equipped with a stereo-VI sensor using the Gazebo simulator [36] traveling in a dynamic 3D trajectory (see Fig. 1). A series of 7 IMUs were placed 1.5 meters radially around the body frame of the UAV with random orientations. Ground-truth inertial measurements were collected at 400Hz and corrupted using the characteristics of an ADIS16448 IMU using the standard discrete-time simulation method of [37] (along with simulating biases), while image measurements were corrupted by one pixel noise at a rate of 20Hz. Constant (static) time offsets were subtracted from each IMU and camera readings introducing the time errors between measurements. The mi-VINS was initialized with perturbed ground-truth calibration of varying magnitudes, and 30 Monte-Carlo runs were performed, where each run represented a different realization of the measurement noises.

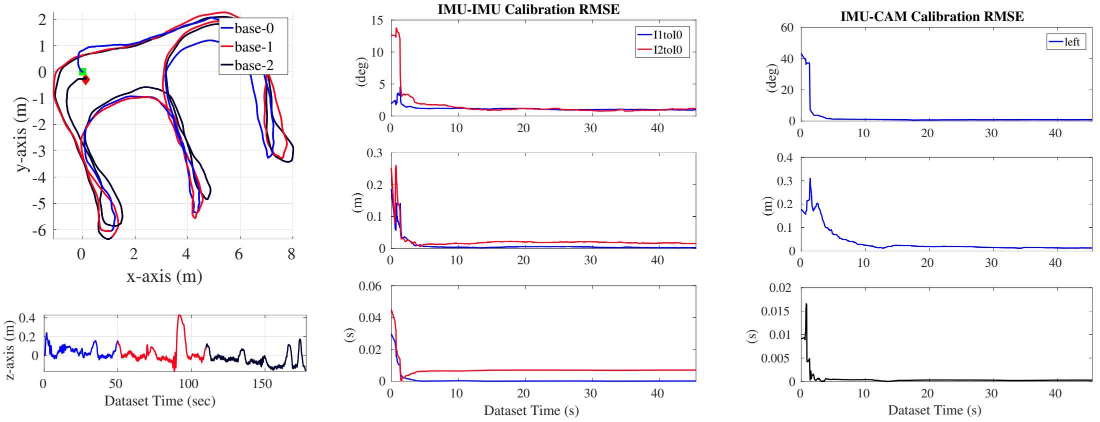

Fig. 2: Experimental results: (left) Top-down view and z-axis trajectory estimates (about 143-meter long) with different base IMUs. (center) IMU-IMU spacial and temporal calibration results and (right) IMU-camera calibration results.

The system was initialized from rest such that each IMU's starting velocity was zero, while the base IMU was initialized with the ground-truth pose. After this pose of the base IMU was determined, the IMU-IMU calibration parameters were used in a covariance propagation to initialize the pose of each auxiliary IMU. For this experiment, a sliding window size of 15 poses was used in the MSCKF.

After waiting 40 seconds to ensure convergence of calibration parameters, the base IMU was switched off every 6 seconds until only a single IMU sensor remained to mimic the sensor failures. Fig. 1 (bottom-left) shows the root mean squared errors (RMSE) of the trajectory estimates, where the color-coded different segments are obtained by using different base IMUs. For continuous trajectory estimates, we recorded the poses of IMU 0 at every timestep using the current estimates for the base IMU and the spatial calibration parameters between the base IMU and IMU 0. From the IMU-IMU calibration results shown Fig. 1 (center), we clearly see that the temporal and spatial calibrations quickly converge to the ground-truth. Finally, the IMU-CAM calibration shown in Fig. 1 (right) converges towards the ground-truth from poor initial guesses. These results show the ability to both perform accurate online localization with resilience to multiple sensor failures (in this case 6 sensors failed in total).

B. *Real-World Tests*

In our real-world tests, we assembled the multi-IMU VI system shown in Fig. 3, which consists of a Pointgrey Blackfly camera with a wide angle lens, MTI-100, MTI-g710, and VI-sensor [38] with an ADIS16448 IMU. We used the Blackfly camera as a monocular image feed, while the MTI-g710, MTI-100, and VI-sensor ADIS16448 IMU were used as the three inertial sensors. The MTI-g710 acted as the base IMU until 50 seconds into the dataset when switching to the MTI-100 sensor was triggered, followed by a switching to the ADIS16448 IMU 100 seconds in. It should be pointed out that this order of IMU failures is arbitrary as the proposed mi-VINS can handle the failure of any of the IMUs (base or non-base), while we here focus only on the most challenging case of *repeated* base IMU failures. KLT tracking [39] using the implementation from OpenCV [40] was performed on incoming image measurements, while outliers were rejected through 8-point RANSAC. When features reached the sliding window size (10 poses in this experiment) or were lost, they were used in the MSCKF update.

We evaluate our sensor calibration performance in comparison to the results obtained from the `Kalibr` calibration toolbox [37] which is an offline process and is *expected* to be close to the ground-truth. These parameters were manually perturbed to form very poor initial guesses for the system in order to demonstrate robustness. The results in Fig. 2 clearly show that the proposed mi-VINS achieves impressive localization and calibration performance. Two of the three IMUs "failed" during the trajectory but did not impact the estimation performance. As no ground-truth was available for this test, we returned to the starting location and computed the accumulated position error which is 0.35 meters, only about 0.24% of distance traveled. In addition the proposed system remained real-time, with an average processing time of 0.023 seconds (43 Hz) per image, compared to the 20 Hz rate of the camera.

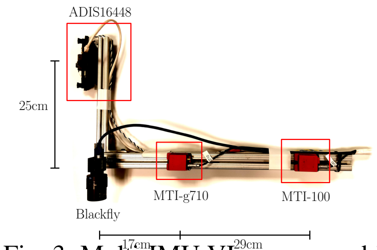

Fig. 3: Multi-IMU VI-sensor used in our real-world experiments.

VII. Conclusion and Future Works

In this paper we have developed a multi-IMU VINS (mi-VINS) with online sensor calibration which is resilient to sensor failures and is able to perform online calibration of both the spatial and temporal relationships between all involved sensors. The proposed mi-VINS has been validated in both simulated and real experiments, and shown to provide accurate estimates for both the trajectory and calibration parameters even in cases of sensor failure. In future work, we plan to perform online intrinsic calibration as well similar to [30] and also investigate automatic *initialization* of the IMU-IMU calibration parameters.

REFERENCES

[1] K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis. "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices". In: *2015 Robotics: Science and Systems Conference, RSS 2015*. MIT Press Journals. 2015.

[2] Y. Ling, T. Liu, and S. Shen. "Aggressive quadrotor flight using dense visual-inertial fusion". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1499–1506.

[3] 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, 2007, pp. 3565–3572.

[4] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. "Consistency Analysis and Improvement of Vision-aided Inertial Navigation". In: *IEEE Transactions on Robotics* PP.99 (2013), pp. 1–19. DOI: 10.1109/TRO.2013.2277549.

[5] M. Li and A. I. Mourikis. "High-Precision, Consistent EKF-based Visual-Inertial Odometry". In: *International Journal of Robotics Research* 32.6 (2013), pp. 690–711.

[6] G. Huang, M. Kaess, and J. Leonard. "Towards Consistent Visual-Inertial Navigation". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Hong Kong, China, 2014, pp. 4926–4933.

[7] T. Qin, P. Li, and S. Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[8] M. Li and A. I. Mourikis. "Online Temporal Calibration for Camera-IMU Systems: Theory and Algorithms". In: *International Journal of Robotics Research* 33.7 (June 2014), pp. 947–964.

[9] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis. "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 165–172.

[10] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual-inertial odometry using nonlinear optimization". In: *International Journal of Robotics Research* (Dec. 2014).

[11] R. C. Avram, X. Zhang, J. Campbell, and J. Muse. "IMU sensor fault diagnosis and estimation for quadrotor UAVs". In: *IFAC-PapersOnLine* 48.21 (2015), pp. 380–385.

[12] M. K. Jeerage. "Reliability analysis of fault-tolerant IMU architectures with redundant inertial sensors". In: *IEEE Aerospace and Electronic Systems Magazine* 5.7 (1990), pp. 23–28.

[13] N. Trawny and S. I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.

[14] C. Guo, D. Kottas, R. DuToit, A. Ahmed, R. Li, and S. Roumeliotis. "Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps". In: *Proc. of the Robotics: Science and Systems Conference*. Berkeley, CA, 2014.

[15] D. G. Kottas and S. I. Roumeliotis. "Efficient and consistent vision-aided inertial navigation using line observations". In: *Proc. of 2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany, 2013, pp. 1540–1547.

[16] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. "Camera-IMU-based localization: Observability analysis and consistency improvement". In: *International Journal of Robotics Research* 33 (2014), pp. 182–201.

[17] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual–inertial odometry using nonlinear optimization". In: *The International Journal of Robotics Research* 34.3 (2015), pp. 314–334.

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

[19] J. B. Bancroft and G. Lachapelle. "Data fusion algorithms for multiple inertial measurement units". In: *Sensors* 11.7 (2011), pp. 6771–6798.

[20] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker. "Survey of motion tracking methods based on inertial sensors: a focus on upper limb human motion". In: *Sensors* 17.6 (2017), p. 1257.

[21] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano. "Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation". In: *The International Journal of Robotics Research* 35.6 (2016), pp. 631–653.

[22] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart. "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 4304–4311.

[23] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka. "Optimization based IMU camera calibration". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 3297–3304.

[24] J. Nikolic, M. Burri, I. Gilitschenski, J. Nieto, and R. Siegwart. "Non-parametric extrinsic and intrinsic calibration of visual-inertial sensor systems". In: *IEEE Sensors Journal* 16.13 (2016), pp. 5433–5443.

[25] F. M. Mirzaei and S. I. Roumeliotis. "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation". In: *IEEE Transactions on Robotics* 24.5 (Oct. 2008), pp. 1143–1156. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2004486.

[26] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley. "Continuous-time batch trajectory estimation using temporal basis functions". In: *The International Journal of Robotics Research* 34.14 (2015), pp. 1688–1710.

[27] D. Kim, S. Shin, and I. S. Kweon. "On-Line Initialization and Extrinsic Calibration of an Inertial Navigation System With a Relative Preintegration Method on Manifold". In: *IEEE Transactions on Automation Science and Engineering* 15.3 (2018), pp. 1272–1285.

[28] T. Lupton and S. Sukkarieh. "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions". In: *IEEE Transactions on Robotics* 28.1 (Feb. 2012), pp. 61 –76.

[29] K. Eckenhoff, P. Geneva, and G. Huang. "High-Accuracy Preintegration for Visual-Inertial Navigation". In: *Proc. of International Workshop on the Algorithmic Foundations of Robotics*. San Francisco, CA, 2016.

[30] M. Li, H. Yu, X. Zheng, and A. I. Mourikis. "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 409–416.

[31] T. Qin and S. Shen. "Online temporal calibration for monocular visual-inertial systems". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3662–3669.

[32] C. Hertzberg, R. Wagner, U. Frese, and L. SchröDer. "Integrating Generic Sensor Fusion Algorithms with Sound State Representations Through Encapsulation of Manifolds". In: *Information Fusion* 14.1 (Jan. 2013), pp. 57–77. ISSN: 1566-2535.

[33] A. B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[34] J. B. Bancroft. "Multiple IMU integration for vehicular navigation". In: *Proceedings of ION GNSS*. Vol. 1. 2009, pp. 1–13.

[35] P. S. Maybeck. *Stochastic Models, Estimation, and Control*. Vol. 1. London: Academic Press, 1979.

[36] N. P. Koenig and A. Howard. "Design and use paradigms for Gazebo, an open-source multi-robot simulator." In: Citeseer.

[37] P. Furgale, J. Rehder, and R. Siegwart. "Unified temporal and spatial calibration for multi-sensor systems". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 1280–1286.

[38] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart. "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 431–437.

[39] S. Baker and I. Matthews. "Lucas-Kanade 20 Years On: A Unifying Framework". In: *International Journal of Computer Vision* 56 (2004), pp. 221–255.

[40] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).