Monocular Visual-Inertial Fusion with Online Initialization and Camera-IMU Calibration

Zhenfei Yang and Shaojie Shen

***Abstract*— There have been increasing demands in developing micro aerial vehicles with vision-based autonomy for search and rescue missions in complex environments. In particular, the monocular visual-inertial system (VINS), which consists of only an inertial measurement unit (IMU) and a camera, forms a great lightweight sensor suite due to its low weight and small footprint. In this work, we address two challenges for rapid deployment of monocular VINS - the initialization problem and the calibration problem. We propose a methodology that is able to initialize velocity, gravity, visual scale, and camera-IMU extrinsic calibration on-the-fly. Our approach does not require any prior knowledge about the mechanical configuration of the system. It is a significant step towards plug-and-play and highly customizable visual navigation for mobile robots. We show through online experiment that our method leads to accurate calibration of camera-IMU transformation with errors of $0.02\,$m in position and $2\,$degrees in rotation.**

I. Introduction

There have been increasing demands in developing high maneuverability robots, such as micro aerial vehicles (MAVs) with vision-based autonomy for search and rescue missions in confined environments. Such robots and sensor suites should be miniaturized, rapidly deployable, and requiring minimum maintenance even in hazardous environments. The monocular visual-inertial system, which consists of only a low cost MEMS IMU and a camera, has become a very attractive sensor choice due to its superior size, weight, and power (SWaP) characteristics. In fact, monocular VINS is the minimum sensor suite that allows both accurate state estimation and sufficient environment awareness.

However, the algorithmic challenges for processing information from monocular VINS are significantly more involved than stereo- [1], or RGB-D-based [2] configurations due to the lack of direct observation of visual scale. The performance of state-of-the-art nonlinear monocular VINS estimators [3]–[6] rely heavily on the accuracy of initial values (velocity, attitude, visual scale) and the calibration of camera-IMU transformation. In time-critical search and rescue mission, careful initialization (launch) of the robot platform or explicit calibration by professional users are often infeasible. In fact, it is desirable to simply plug sensors onto the MAV, throw it into the air, and have everything operational. This implies that the initialization and the camera-IMU extrinsic calibration procedure should be performed with no prior knowledge about either the dynamical motion or mechanical configuration of the system.

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. zyangag@connect.ust.hk, eeshaojie@ust.hk

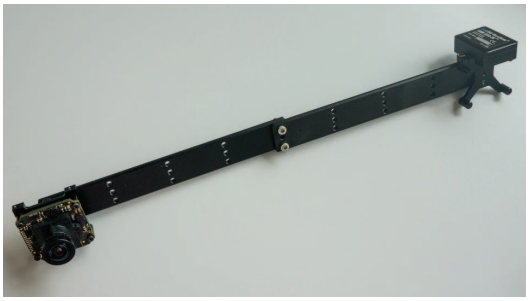


Fig. 1. Our monocular VINS setup with unknown camera-IMU extrinsic calibration. Note that there are multiple 90 degree offsets between the camera frame and the IMU frame. The rotation offsets are unknown to the estimator and have to be calibrated online. A video of the experiment can be found at http://www.ece.ust.hk/~eeshaojie/SSRR2015.mp4

Our earlier works [6]–[8] focused on initialization and tightly-coupled fusion of monocular VINS systems, but with the assumption of known camera-IMU transformation. In this work, we relax this assumption and propose a method for joint initialization and extrinsic calibration without any prior knowledge about the mechanical configuration of the system. We show through online experiment that our method leads to accurate calibration of camera-IMU transformation with errors of $0.02\,$m in position and $2\,$degrees in rotation. This is a substantial step towards minimum sensing for plug-and-play robotics system. We identify contributions of this work as fourfold:

- A probabilistic, optimization-based initialization procedure that is able to recover all essential navigation quantities (initial velocity and attitude, visual scale, and camera-IMU calibration) without any prior system knowledge or artificial calibration objects.
- A principled method to identify convergence and exit points for the initialization procedure.
- A tightly-coupled monocular visual-inertial fusion methodology for accurate state estimation with online calibration refinement.
- Online experiment in real-world environments.

The rest of this paper is organized as follows: In Sect. II, we discuss relevant literatures. We give an overview of the complete system pipeline in Sect. III. We detail our linear initialization and camera-IMU calibration procedure in Sect. IV. A tightly-coupled, nonlinear optimization-based monocular VINS estimator, which is built on top of our recent work [6, 7], is presented in Sect. V. We discuss implementation details and present experimental results in

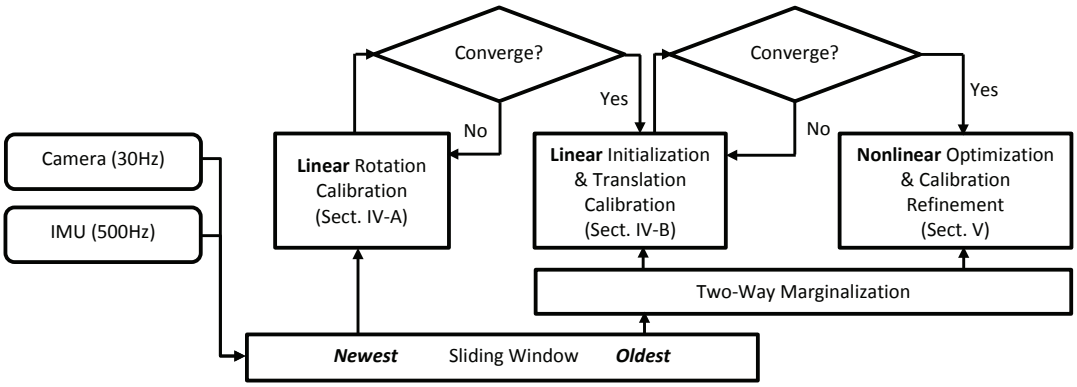


Fig. 2. A block diagram illustrating the full pipeline of the proposed approach.

Sect. VI. The paper is concluded with a discussion on possible future directions in Sect. VII.

II. Related Work

There is a rich body of scholarly work on VINS state estimation with either monocular [3, 4, 9], stereo [5], or RGB-D cameras [2]. We can categorize solutions to VINS as filtering-based [3, 4, 9]–[11], or graph optimization-/bundle adjustment-based [5, 6, 12]. Filtering-based approaches may require less computational resources due to the marginalization of past states, but the early fix of linearization points may lead to sub-optimal results. On the other hand, graph optimization-based approaches may improve performance via iterative re-linearization at the expenses of higher computational demands. In real-world applications, marginalization is usually employed for both filtering- and optimization-based approaches to achieve constant computational complexity.

Another way to categorize VINS solutions is to consider them as loosely-coupled [9] or tightly-coupled [3]–[5, 10]–[12]. Loosely-coupled approaches usually consist of a standalone vision-only state estimation module such as PTAM [13] or SVO [14], and a filtering-based IMU fusion module [15]. These approaches do not consider the information coupling between visual and IMU and thus unable to correct drifts in the vision-only estimator. Tightly coupled approaches perform systematic fusion of visual and IMU measurements and thus usually lead to better results [5]. In particular, for monocular VINS, tightly-coupled methods are able to implicitly incorporate the metric scale information from IMU into scene depth estimation, thus removing the need of explicit scale modeling.

However, all aforementioned VINS solutions rely on accurate initialization of system motions and accurate camera-IMU calibration. This is particularly critical for monocular VINS due to the lack of direct observation of visual scale. There is a wide body of work trying to deal with the velocity, attitude, and visual scale initialization problems for monocular VINS, with geometric approaches proposed in [16]–[18] and more principled probabilistic approaches proposed in [8, 19]. For camera-IMU calibration, [4, 15, 20] consider incorporating the camera-IMU transformation into the state vector for the nonlinear estimator. However, the convergence of the calibration parameters still depends on the accuracy of initial values, and the calibration performance is not systematically analyzed in either of these papers.

Our work is related to [21], where both aim to jointly initialize the motion of the system as well as camera-IMU calibration. However, [21] is a geometric method without consideration of sensor noise. In particular, the formulation of IMU measurements in [21] results in unbounded IMU error over time, which leads to downgraded performance as more IMU measurements are incorporated. On the other hand, our probabilistic formulation explicitly bounds the error for each measurement using a sliding window approach, thus able to fuse extensive amount of sensor measurements until good initial values are obtained. Also, [21] only shows results with simulated data, while we present extensive experimental results with real sensor data.

III. Overview

Our proposed monocular VINS estimator consists of three phases, as illustrated in Fig. 2. The first phase initializes the rotation between the camera and the IMU in a linear fashion (Sect. IV-A). The second phase handles on-the-fly initialization of velocity, attitude, visual scale, and camera-IMU translation with a probabilistic linear sliding window approach (Sect. IV-B). This phase is an extension of our earlier work [7, 8] by relaxing the known camera-IMU calibration assumption. Finally, the third phase that focuses on high accuracy nonlinear optimization for both state estimation and calibration refinement will be detailed in Sect. V. Note that the three phases run sequentially, continuously, with automatic switching. This suggests that all the user needs to do is moving the monocular VINS sensor suite freely with sufficient motion in natural environments. Our estimator is able to automatically identify convergence and switch to the next phase (Sect. IV-B.3 and Sect. IV-B.3).

We begin by defining notations. We consider $(\cdot)^w$ as the earth's inertial frame, $(\cdot)^b$ as the current IMU body frame. $(\cdot)^{c_k}$ as the camera body frame while taking the k^{th} image. We further note $(\cdot)^{b_k}$ as the IMU body frame while the

camera is taking the k^{th} image. Note that IMU usually runs at a higher rate than the camera, and that multiple IMU measurements may exist in the interval $[k, k+1]$. $\mathbf{p}_Y^X$, $\mathbf{v}_Y^X$, and $\mathbf{R}_Y^X$ are 3D position, velocity, and rotation of frame Y with respect to frame X. Specially, $\mathbf{p}_t^X$ represents the position of the IMU body frame at time t with respect to frame X. Similar conversion follows for other parameters. The camera-IMU transformation is an unknown constant that we denote as $\mathbf{p}_c^b$ and $\mathbf{R}_c^b$. Besides rotation matrices, we also use quaternions ($\mathbf{q} = [q_x, q_y, q_z, q_w]$) to represent rotation. The Hamilton notation is used for quaternions. $\mathbf{g}^w = [0, 0, g]^{\mathrm{T}}$ is the gravity vector in the world frame, and $\mathbf{g}^{b_k}$ is the earth's gravity vector expressed in the IMU body frame during the k^{th} image capture.

IV. Estimator Initialization and Camera-IMU Extrinsic Calibration

We now detail our online estimator initialization approach to recover all critical states, including velocity, attitude (gravity vector), depth of features, and camera-IMU extrinsic calibration. Our initialization procedure does not require any prior knowledge about the mechanical configuration of the sensor suite. It also does not require the estimator to be started from stationary, making it particularly useful for dynamical launching aerial robots in a search and rescue setting. The initialization and calibration process can be formulated as solving two sets of linear systems which will be discussed in Sect IV-A and Sect. IV-B respectively.

Given two time instants that corresponds to two images, we can write the IMU propagation model for position and velocity in the earth's inertial frame as follows:

$$\begin{aligned}\mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t + \iint_{t\in[k,k+1]} \left(\mathbf{R}_t^w \mathbf{a}_t^b - \mathbf{g}^w\right) dt^2 \\ \mathbf{v}_{b_{k+1}}^w &= \mathbf{v}_{b_k}^w + \int_{t\in[k,k+1]} \left(\mathbf{R}_t^w \mathbf{a}_t^b - \mathbf{g}^w\right) dt\end{aligned} \tag{1}$$

where $\mathbf{a}_t^b$ is the instantaneous linear acceleration in the IMU body frame, Δt is the time difference $[k, k+1]$ between two image captures. It can be seen that the rotation between the world frame and the body frame is required for IMU state propagation. This global rotation can only be determined with known initial attitude, which is hard to obtain in many applications. However, as introduced in [19], if the reference frame for IMU propagation is changed to the first state of the system (i.e. the first block of pose, velocity, and attitude that we are trying to estimate), (1) can be rewritten as:

$$\begin{aligned}\mathbf{p}_{b_{k+1}}^{b_0} &= \mathbf{p}_{b_k}^{b_0} + \mathbf{R}_{b_k}^{b_0} \mathbf{v}^{b_k} \Delta t - \mathbf{R}_{b_k}^{b_0} \mathbf{g}^{b_k} \Delta t^2/2 + \mathbf{R}_{b_k}^{b_0} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \mathbf{v}_{b_{k+1}}^{b_{k+1}} &= \mathbf{R}_{b_k}^{b_{k+1}} \mathbf{v}_{b_k}^{b_k} - \mathbf{R}_{b_k}^{b_{k+1}} \mathbf{g}^{b_k} \Delta t + \mathbf{R}_{b_k}^{b_{k+1}} \boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \mathbf{g}^{b_{k+1}} &= \mathbf{R}_{b_k}^{b_{k+1}} \mathbf{g}^{b_k}\end{aligned} \tag{2}$$

where

$$\begin{aligned}\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \iint_{t\in[k,k+1]} \mathbf{R}_t^{b_k} \mathbf{a}_t^b dt^2 \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \int_{t\in[k,k+1]} \mathbf{R}_t^{b_k} \mathbf{a}_t^b dt\end{aligned} \tag{3}$$

can be obtained solely with IMU measurements within $[k, k+1]$. $\mathbf{R}_{b_k}^{b_0}$ is the change in rotation since the first pose (or since the 0^{th} image), and $\mathbf{R}_{b_{k+1}}^{b_k}$ is the incremental rotation between two images. With this formulation, the dependency on global orientation is removed. Therefore, using the short term incremental rotation by integrating gyroscope measurements, and the camera-IMU rotation calibration obtained in Sect. IV-A, we are able to formulate the joint problem of monocular VINS initialization and camera-IMU translation calibration as a linear problem that can be solved without any prior knowledge of the system.

A. Calibration of Camera-IMU Rotation

The constant camera-IMU rotation offset can be obtained by aligning two rotation sequences from the IMU and the camera.

1) Linear Rotation Calibration: We assume that sufficient features can be tracked, and the incremental rotation between two images $\mathbf{R}_{c_{k+1}}^{c_k}$ can be estimated using the classic 5-point algorithm [22] with RANSAC-based outlier rejection. We further note that the corresponding rotation obtained by integrating gyroscope measurements represented in the IMU frame as $\mathbf{R}_{b_{k+1}}^{b_k}$. The following equation holds for any k:

$$\mathbf{R}_{b_{k+1}}^{b_k} \cdot \mathbf{R}_c^b = \mathbf{R}_c^b \cdot \mathbf{R}_{c_{k+1}}^{c_k} \tag{4}$$

With a quaternion representation for rotation, we can write (4) as:

$$\begin{aligned}&\mathbf{q}_{b_{k+1}}^{b_k} \otimes \mathbf{q}_c^b = \mathbf{q}_c^b \otimes \mathbf{q}_{c_{k+1}}^{c_k} \\ &\Rightarrow \left[\mathcal{Q}_1(\mathbf{q}_{b_{k+1}}^{b_k}) - \mathcal{Q}_2(\mathbf{q}_{c_{k+1}}^{c_k})\right] \cdot \mathbf{q}_c^b = \mathbf{Q}_{k+1}^k \cdot \mathbf{q}_c^b = \mathbf{0}\end{aligned} \tag{5}$$

where

$$\begin{aligned}\mathcal{Q}_1(\mathbf{q}) &= \begin{bmatrix} q_w \mathbf{I}_3 + \lfloor \mathbf{q}_{xyz} \times \rfloor & \mathbf{q}_{xyz} \\ -\mathbf{q}_{xyz} & q_w \end{bmatrix} \\ \mathcal{Q}_2(\mathbf{q}) &= \begin{bmatrix} q_w \mathbf{I}_3 - \lfloor \mathbf{q}_{xyz} \times \rfloor & \mathbf{q}_{xyz} \\ -\mathbf{q}_{xyz} & q_w \end{bmatrix}\end{aligned} \tag{6}$$

are matrix representations for left and right quaternion multiplication, and $\lfloor \mathbf{q}_{xyz} \times \rfloor$ is the skew-symmetric matrix from the first three elements $\mathbf{q}_{xyz}$ of a quaternion. $\otimes$ is the quaternion multiplication operator.

With multiple incremental rotations between pairs of consecutive images, we are able to construct the over-constrained linear system as:

$$\begin{bmatrix} w_1^0 \cdot \mathbf{Q}_1^0 \\ w_2^1 \cdot \mathbf{Q}_2^1 \\ \vdots \\ w_N^{N-1} \cdot \mathbf{Q}_N^{N-1} \end{bmatrix} \cdot \mathbf{q}_c^b = \mathbf{Q}_N \cdot \mathbf{q}_c^b = \mathbf{0} \tag{7}$$

where N is the index of the latest frame that keeps growing until the rotation calibration is completed. w_{k+1}^k is the robust

weight for better outlier handling. As the rotation calibration runs with more and more measurements, we are able to use the previously estimated camera-IMU rotation $\hat{\mathbf{R}}_c^b$ as the initial value to weight the residual in a similar fashion as the Huber norm [23]. The residual is defined as the angle norm in the angle-axis representation of the residual rotation matrix:

$$w_{k+1}^k = \text{acos}\left(\left(\text{tr}\left(\hat{\mathbf{R}}_c^{b^{-1}}\mathbf{R}_{b_{k+1}}^{b_k{}^{-1}}\hat{\mathbf{R}}_c^b\mathbf{R}_{c_{k+1}}^{c_k}\right)-1\right)/2\right) \quad (8)$$

If there are no sufficient features for estimating the camera rotation, w_{k+1}^k is set to zero. The solution to the above linear system can be found as the right unit singular vector corresponding to the smallest singular value of $\mathbf{Q}_N$.

2) Convergence and Exit Point: Successful calibration of the camera-IMU rotation $\mathbf{R}_c^b$ relies on sufficient rotation excitation. Under sufficient rotation, the null space of $\mathbf{Q}_N$ should be rank one. However, under degenerate motions in one or more axes, the null space of $\mathbf{Q}_N$ may be larger than one. Therefore, by checking whether the second smallest singular value of $\mathbf{Q}_N$, $\sigma_{\mathbf{R}}^{\text{min2}}$, is large enough, we have a good indicator of whether sufficient rotation excitation is achieved. We set a singular value threshold $\sigma_{\mathbf{R}}$. The camera-IMU rotation calibration process terminates if $\sigma_{\mathbf{R}}^{\text{min2}} > \sigma_{\mathbf{R}}$. A convergence plot can be found in Fig. 4.

B. Calibration of Camera-IMU Translation and On-the-Fly Initialization of Velocity, Attitude, and Feature Depth

Once the camera-IMU rotation is fixed, we can estimate the camera-IMU translation together with an initialization of velocity, attitude, feature depth, as well as the IMU poses with respect to the initial reference frame as in (2).

1) Linear Sliding Window Estimator: We use a tightly-coupled sliding window formulation for incorporating a large number of IMU and camera measurements with constant computational complexity. The initialization is done in the IMU frame, with the full state vector defined as (the transpose is ignored for the simplicity of presentation):

$$\begin{aligned}\mathcal{X} &= \left[\mathbf{x}_n,\, \mathbf{x}_{n+1},\, \cdots \mathbf{x}_{n+N},\, \mathbf{p}_c^b,\, \lambda_m,\, \lambda_{m+1},\, \cdots \lambda_{m+M}\right]\\ \mathbf{x}_k &= \left[\mathbf{p}_{b_k}^{b_0},\, \mathbf{v}_{b_k}^{b_k},\, \mathbf{g}^{b_k}\right]\end{aligned} \quad (9)$$

where $\mathbf{x}_k$ is the k^{th} IMU state. The gravity vector $\mathbf{g}^{b_k}$ determines the roll and pitch angles. N is the number of IMU states in the sliding window, M is the number of features that have sufficient parallax within the sliding window. n and m are starting indexes of the sliding window. λ_l is the depth of the l^{th} feature from its first observation. $\mathbf{p}_{b_0}^{b_0} = [0,\, 0,\, 0]$ is preset. Note that we reuse the sensor measurements that we used for camera-IMU rotation calibration in this linear initialization phase, but with $\mathbf{R}_c^b$ fixed as a constant. We also directly use the incremental ($\mathbf{R}_{b_{k+1}}^{b_k}$) and relative ($\mathbf{R}_{b_{k+1}}^{b_0}$) rotations obtained from short term integration of gyroscope measurements. As this linear initialization can usually be done in only a few seconds, using the IMU rotation directly will not cause significant drifts.

The linear initialization is done with maximum likelihood estimation by minimizing the sum of the Mahalanobis norm of all measurement errors from the IMU and the monocular camera within the sliding window:

$$\begin{aligned}\min_{\mathcal{X}}\Bigg\{&\left(\mathbf{b}_p - \mathbf{\Lambda}_p\mathcal{X}\right) + \sum_{k\in\mathcal{B}}\left\|\hat{\mathbf{z}}_{b_{k+1}}^{b_k} - \mathbf{H}_{b_{k+1}}^{b_k}\mathcal{X}\right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2\\ &+ \sum_{(l,j)\in\mathcal{C}}\left\|\hat{\mathbf{z}}_l^{c_j} - \mathbf{H}_l^{c_j}\mathcal{X}\right\|_{\mathbf{P}_l^{c_j}}^2\Bigg\}\end{aligned} \quad (10)$$

where $\mathcal{B}$ is the set of all IMU measurements, and $\mathcal{C}$ is the set of all observations between any features and any camera pose. Since incremental and relative rotations are known, (10) can be solved in a non-iterative linear fashion.

The IMU measurement $\hat{\mathbf{z}}_{b_{k+1}}^{b_k} = [\boldsymbol{\alpha}_{b_{k+1}}^{b_k}\, \boldsymbol{\beta}_{b_{k+1}}^{b_k}]^{\text{T}}$ is defined as (3). Detailed derivation of the measurement matrix $\mathbf{H}_{b_{k+1}}^{b_k}$ and the covariance $\mathbf{P}_{b_{k+1}}^{b_k}$ can be found in Chapter 6 of [7]. The camera measurement model $\{\hat{\mathbf{z}}_l^{c_j},\, \mathbf{H}_l^{c_j}\}$ for the observation of the l^{th} feature in the j^{th} image is defined as:

$$\begin{aligned}\hat{\mathbf{z}}_l^{c_j} &= \hat{\mathbf{0}} = \mathbf{H}_l^{c_j}\mathcal{X} + \mathbf{n}_l^{c_j}\\ &= \mathbf{M}\cdot\text{f}_c^{b^{-1}}\left(\mathbf{R}_{b_0}^{b_j}\left(\mathbf{R}_{b_i}^{b_0}\cdot\text{f}_c^b\left(\lambda_l\begin{bmatrix}u_l^{c_i}\\ v_l^{c_i}\\ 1\end{bmatrix}\right) + \mathbf{p}_{b_i}^{b_0} - \mathbf{p}_{b_j}^{b_0}\right)\right)\end{aligned} \quad (11)$$

where $\mathbf{M}$ is defined as:

$$\mathbf{M} = \begin{bmatrix}-1 & 0 & \hat{u}_l^{c_j}\\ 0 & -1 & \hat{v}_l^{c_j}\end{bmatrix} \quad (12)$$

and the function $\text{f}_c^b(.)$ that transform a 3D point $\mathbf{r}$ in the camera frame to the IMU frame is defined as:

$$\text{f}_c^b(\mathbf{r}) = \mathbf{R}_c^b\cdot\mathbf{r} + \mathbf{p}_c^b \quad (13)$$

and $[u_l^{c_i},\, v_l^{c_i}]$ is the noiseless first observation of the l^{th} feature that happened in the i^{th} image. $[\hat{u}_l^{c_j},\, \hat{v}_l^{c_j}]$ is the observation of the same feature in the j^{th} image. All rotation matrices $\mathbf{R}_Y^X$ are known quantities. $\mathbf{n}_l^{c_j}$ is the additive measurement noise. The camera measurement covariance has the form:

$$\mathbf{P}_l^{c_j} = {\lambda_l^{c_j}}^2\bar{\mathbf{P}}_l^{c_j} \quad (14)$$

where $\bar{\mathbf{P}}_l^{c_j}$ is the feature observation noise in the normalized image plane. Although we can only initialize the unknown $\lambda_l^{c_j}$ as the average scene depth, we found in practice that the solution is insensitive to the initial value of $\lambda_l^{c_j}$ as long as it is set to be *larger* than the actual depth.

The linear cost function (10) can be rearranged into the form

$$\left(\mathbf{\Lambda}_p + \mathbf{\Lambda}_{\mathcal{B}} + \mathbf{\Lambda}_{\mathcal{C}}\right)\mathcal{X} = \left(\mathbf{b}_p + \mathbf{b}_{\mathcal{B}} + \mathbf{b}_{\mathcal{C}}\right) \quad (15)$$

where $\{\mathbf{\Lambda}_{\mathcal{B}},\, \mathbf{b}_{\mathcal{B}}\}$ and $\{\mathbf{\Lambda}_{\mathcal{C}},\, \mathbf{b}_{\mathcal{C}}\}$ are information matrices and vectors for IMU and camera measurements respectively. Due to the known incremental and relative rotations, the cost is linear with respect to the states, the system in (15) has a unique solution even without the prior $\{\mathbf{b}_p,\, \mathbf{\Lambda}_p\}$. This suggests that our method is able to recover all quantities in the full state vector, including the camera-IMU translation, without any initial guess of the values.

2) Two-Way Marginalization: In order to bound the computational complexity of graph optimization-based methods, marginalization is usually used. It is well known that for monocular VINS, sufficient acceleration excitation is required for scale observability [10, 11]. To preserve scale observability, we need to keep IMU states that attach with large accelerations for the optimization (10), as such, a naive marginalization method that always removes old IMU states will not work. To this end, we employ the two-way marginalization scheme that is originally proposed in our earlier work [7, 8] to selectively remove recent or old IMU states based on a scene parallax test. This method ensures sufficient acceleration within the sliding window. The remaining frames in the sliding window have large pairwise parallax, thus exhibit similar property as keyframes that are widely used in vision-only approach [13]. However, the two-way marginalization is fundamentally different from keyframe-based methods, as information from non-keyframes is not dropped, but rather converted into priors $\{\mathbf{b}_p,\, \mathbf{\Lambda}_p\}$ for future optimization.

3) Convergence and Exit Point: The covariance matrix (inverse of the information matrix) $\left(\mathbf{\Lambda}_p + \mathbf{\Lambda}_\mathcal{B} + \mathbf{\Lambda}_\mathcal{C}\right)^{-1}$ naturally tells the uncertainty of the linear initialization estimator. The block that corresponds to the camera-IMU translation in the covariance matrix represents the uncertainty of the calibration parameters. We use the maximum singular value $\sigma_\mathbf{p}^{\text{max}}$ of the block as the convergence indicator, and exit the linear initialization process if $\sigma_\mathbf{p}^{\text{max}} < \sigma_\mathbf{p}$, where $\sigma_\mathbf{p}$ is a threshold. A convergence plot can be found in Fig. 4.

As computing the matrix inverse is much slower than solving the linear system (15) using Cholesky decomposition, we run the exit point checking at a much slower rate in another thread. This will slightly delay the exit time but will not harm overall performance. After this point, the whole initialization and calibration process is completed.

V. Tightly-Coupled Nonlinear Optimization with Calibration Refinement

After obtaining state initialization and camera-IMU calibration (Sect. IV), we proceed with a sliding window nonlinear estimator for high accuracy state estimation and calibration refinement. This is an extension of our earlier work [6, 7] by including camera-IMU calibration in the nonlinear optimization.

A. Formulation

The definition of the full state is similar to the linear case, with exceptions that the full 6-degree-of-freedom (DOF) camera-IMU transformation $\mathbf{x}_c^b$ is included in the state vector. The gravity vector is also replaced with the quaternion $\mathbf{q}_{b_j}^{b_i}$ for joint optimization of translation and rotation (the transpose is again ignored):

$$\begin{aligned}
\mathcal{X} &= \left[\mathbf{x}_n,\, \mathbf{x}_{n+1},\, \cdots \mathbf{x}_{n+N},\, \mathbf{x}_c^b,\, \lambda_0,\, \lambda_{m+1},\, \cdots \lambda_{m+M}\right] \\
\mathbf{x}_k &= \left[\mathbf{p}_{b_k}^{b_0},\, \mathbf{v}^{b_k},\, \mathbf{q}_{b_k}^{b_0}\right] \\
\mathbf{x}_c^b &= \left[\mathbf{p}_c^b,\, \mathbf{q}_c^b\right]
\end{aligned} \tag{16}$$

We minimize the sum of Mahalanobis norm of all measurement residuals to obtain a maximum a posteriori estimation:

$$\begin{aligned}
\min_{\mathcal{X}} \Bigg\{ &\left(\mathbf{b}_p - \mathbf{\Lambda}_p \mathcal{X}\right) + \sum_{k\in\mathcal{B}} \left\| r_\mathcal{B}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k},\, \mathcal{X}) \right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2 \\
&+ \sum_{(l,j)\in\mathcal{C}} \left\| r_\mathcal{C}(\hat{\mathbf{z}}_l^{c_j},\, \mathcal{X}) \right\|_{\mathbf{P}_l^{c_j}}^2 \Bigg\}
\end{aligned} \tag{17}$$

where $r_\mathcal{B}(\hat{\mathbf{z}}_{bk+1}^{b_k},\, \mathcal{X})$ and $r_\mathcal{C}(\hat{\mathbf{z}}_l^{c_j},\, \mathcal{X})$ are measurement residuals for IMU and camera, respectively, Corresponding measurement models are defined in Sect. V-B. The nonlinear system (17) is linearized and solved with the Gauss-Newton algorithm with Huber norm for robust outlier rejection. Two-way marginalization (Sect. IV-B.2) is again used for reducing the computational cost and preserving scale observability during degenerate motions.

B. Measurement Models

The IMU measurement model and the residual $r_\mathcal{B}(\hat{\mathbf{z}}_{bk+1}^{b_k},\, \mathcal{X})$ is computed the same as [6, 7]. We use the continuous dynamics of the IMU to derive the IMU measurement and error propagation model between two image captures. We are again able to write the IMU measurement as a standalone object without requiring the starting velocity and attitude thanks to the use of of frame transform (2).

The camera measurement model can be formulated similar to the linear initialization (Sect. IV-B) but with residual being the reprojection error due to the feature depth initialization presented in Sect. IV-B.

$$\begin{aligned}
r_\mathcal{C}(\hat{\mathbf{z}}_l^{c_j},\, \mathcal{X}) &= \begin{bmatrix} \frac{{}^f x_l^{c_j}}{{}^f z_l^{c_j}} - \hat{u}_l^j \\ \frac{{}^f y_l}{{}^f z_l^{c_j}} - \hat{v}_l^j \end{bmatrix} \\
\mathbf{f}_l^{c_j} &= \begin{bmatrix} {}^f x_l^{c_j} \\ {}^f y_l^{c_j} \\ {}^f z_l^{c_j} \end{bmatrix} \\
&= \mathbf{f}_c^{b^{-1}} \left(\mathbf{R}_{b_0}^{b_j} \left(\mathbf{R}_{b_i}^{b_0} \cdot \mathbf{f}_c^b \left(\lambda_l \begin{bmatrix} u_l^{c_i} \\ v_l^{c_i} \\ 1 \end{bmatrix} \right) + \mathbf{p}_{b_i}^{b_0} - \mathbf{p}_{b_j}^{b_0} \right) \right)
\end{aligned} \tag{18}$$

where $\mathbf{f}_c^b(.)$ is defined in (13).

VI. Experimental Results

A. Implementation Details

As shown in Fig. 1, our monocular VINS sensor suite consists of a mvBlueFOX-MLC200w grayscale HDR camera with standard lens that capture 752×480 images at 30 Hz, and a Microstrain 3DM-GX4 IMU. The mount for the sensor suite has significant translation between sensors. The sensors are also purposely mounted in different frames with approximately 90 degree rotation offsets in roll and yaw to test the performance of camera-IMU calibration.

Our algorithm runs real-time on a Lenovo ThinkPad T440s laptop. Two threads run in parallel in our implementation, the first thread performs detection of corner features and feature tracking [24] at 30 Hz. The second thread performs initialization, calibration, as well as nonlinear optimization at 10 Hz. During the linear translation initialization (Sect. IV-B), a third thread is launched for recovery of state covariance and detection of exit point. We maintain $N = 30$ IMU states (30 images) and $M = 200$ features in the sliding window. For each image, we detect a maximum of 100 new features with a minimum separation of 30 pixels. A tracked feature has to pass a parallax threshold of 30 pixels before it can be added into the optimization.

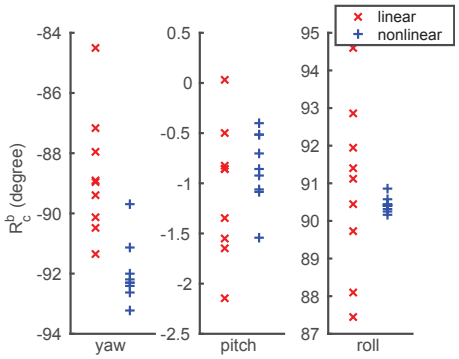
For the linear initialization (Sect. IV), we assume biases of the IMU can be removed by initial subtraction. Therefore, biases are not included in the state vector. Since the initialization phases normally takes only a few seconds, ignoring IMU biases will not lead to noticeable negative effects. For the nonlinear optimization, IMU biases are continuously estimated.

B. Camera-IMU Calibration Performance

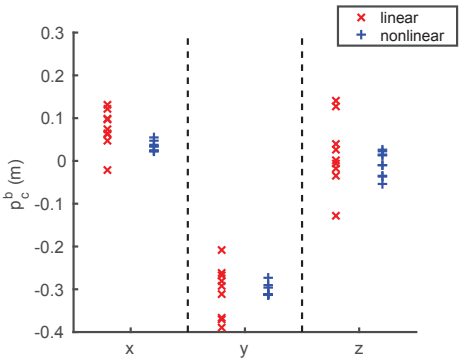
In this experiment, we evaluate the performance of our online camera-IMU calibration method. The camera and the IMU, as shown in Fig. 1, are rigidly mounted on a metal bar. The orientation offset between the camera and the IMU are approximately $[+90,\ 0,\ -90]$ degrees in roll, pitch, and yaw respectively. The translation between the two sensors are approximately $[0,\ -0.3,\ 0]$ meters in x, y, and z respectively. Note that although the sensors are rigidly mounted, we do not know the precise camera-IMU calibration. The performance of the proposed method is evaluated by the convergence of the method and repeatability across multiple trials.

During the experiment, the user moves the sensor suite freely in a typical lab environment with only natural visual features. We conducted 9 trials of experiments. The calibration results for both linear (Sect. IV) and nonlinear (Sect. V) methods are shown in Fig. 3). For both rotation and translation calibration, we can see that the linear method provides a reasonable initialization without any prior knowledge of the mechanical configuration of the system, while the nonlinear optimization further refines the calibration results. We achieve a final calibration accuracy of approximately 2 degrees in rotation and 0.02 m in translation.

Fig. 4 details all phases of the calibration process during one of the trials, with trajectory of the sensor suite shown in Fig. 5. The calibration starts with linear camera-IMU rotation calibration (Phase 1, Sect. IV-A), during which the camera-IMU rotation is recovered from scratch. Phase 1 exits when all three nonzero singular values of $\mathbf{Q}_N$ reaches a high level (Sect. IV-A.2). During Phase 1, the translation component is not estimated. In Phase 2 (Sect. IV-B), the velocity, attitude, and feature depth of the VINS, as well as the camera-IMU translation are estimated simultaneously using a linear sliding window estimator. Here we only show the camera-IMU translation calibration, and defer the discussion of the



(a) Camera-IMU Rotation



(b) Camera-IMU Translation

Fig. 3. Performance of camera-IMU calibration for both the linear initialization (Sect. IV) and the nonlinear refinement (Sect. V). The orientation offset between the camera and the IMU are approximately $[+90,\ 0,\ -90]$ degrees in roll, pitch, yaw, and $[0,\ -0.3,\ 0]$ meters in x, y, z respectively. Since we do not know the precise camera-IMU calibration, performance is evaluated by convergence of the method and repeatability across multiple trials. In all cases, the linear method provides a reasonable initialization without knowing the mechanical configuration of the sensor suite. The nonlinear method further refines the calibration to achieve a final accuracy of 2 degrees in rotation and 0.02 m in translation.

initialization of other quantities to Sect. VI-C. During this phase, the translation component is recovered, again with no prior knowledge about the mechanical configuration. The exit criteria is determined by the uncertainty of the calibration parameters (Sect. IV-B.3). Note that Phase 2 may last for an extensive period of time if there is insufficient motion excitation. However, our two-way sliding window marginalization scheme (Sect. IV-B.2) ensures a bounded complexity algorithm that is able to operate reliably until convergence of calibration parameters. This is the key advantage of our approach comparing to [21]. During Phase 2, the camera-IMU rotation is treated as a constant. Phase 3 (Sect. V) uses nonlinear optimization to jointly and continuously refine the camera-IMU rotation and translation, Since there is no exit criteria in Phase 3, we do not compute the calibration uncertainty to save computational resources.

C. Motion Estimation Performance

We now compare motion estimation performance of the overall monocular VINS estimator against a ground truth

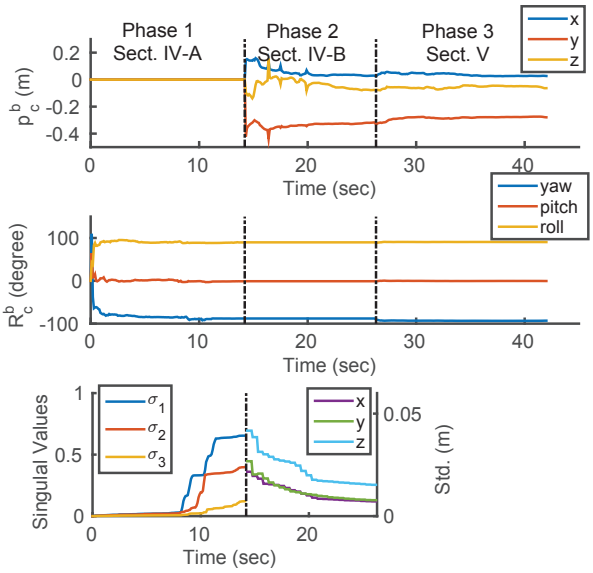


Fig. 4. Detailed illustration of the whole calibration process with system flow shown in Fig. 2 and trajectory shown in Fig. 5. Different phases are separated by dashed lines. In Phase 1, only the camera-IMU rotation is estimated, with the singular value-based exit criteria (Sect. IV-A.2) shown in the first segment in bottom figure. In Phase 2, the camera-IMU translation, as well as other VINS navigation quantities are recovered. The uncertainty-based exit criteria (Sect. IV-B.3) is shown in the second segment in the bottom figure. During Phase 2, the camera-IMU rotation remains constant. Phase 3 jointly and continuously optimize the full 6-DOF camera-IMU calibration.

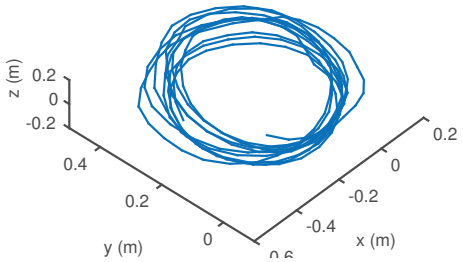
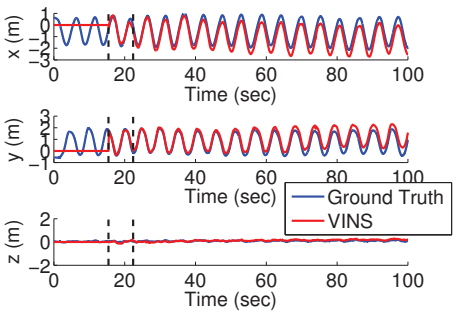


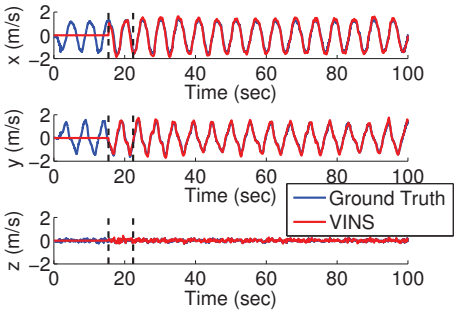
Fig. 5. Trajectory of the sensor suite for the trial shown in Fig. 4.

referencing system consists of 8 OptiTrack Flex13 cameras[1], as shown in Fig. 6. The two dashed lines in each plot indicate the switching between rotation calibration (Sect. IV-A), linear initialization (Sect. IV-B), and nonlinear optimization (Sect. V). The whole process started by moving the sensor suite freely in the space. During rotation calibration, position and velocity quantities are unavailable. After switching to the linear initialization at approximately 16 sec, the estimator recovers the nontrivial velocity on the fly without any initial guesses. After that, it can be seen that the onboard velocity estimates compare well with the ground truth with a standard deviation of $\{0.0278\,, 0.0117,\ 0.0026\}$ (unit in m/s) in x, y, and z axes respectively. As we do not know the exact starting position of the estimator, the position is aligned to the ground truth data with the initial zero pose from the estimator. Since the global position is unobservable, position drift will occurred. However, we can still visually verify that the scale

[1]http://www.optitrack.com/



(a) Position



(b) Velocity

Fig. 6. Comparison of estimator performance against ground truth. The two dashed lines in each plot indicate the switching between camera-IMU rotation calibration (Sect. IV-A), linear initialization (Sect. IV-B), and nonlinear optimization (Sect. V). The rotation calibration runs between 0-16 sec, after which the linear initialization starts and recovers the nontrivial velocity of the platform on-the-fly. It can be seen that the estimated velocity matches well with the ground truth data. As we do not know the exact starting position, the position is aligned to the ground truth data with its first estimate. Although there is unavoidable position drift, we can still visually verify that the scale estimation is correct, indicating the effectiveness of monocular visual-inertial fusion.

estimation is correct, which indicates the effectiveness of monocular visual-inertial fusion.

D. Performance in Large Scale Environments

In this experiment, we evaluate the performance of the overall system with challenging datasets in complex environments. Throughout the experiment, we encounter large rotation (Fig. 7(a)), motion blur (Fig. 7(b)), people walking and view obstruction (Fig. 7(c)), as well as mirrors (Fig. 7(d)).

Fig. 8 shows the position estimation from the overall monocular VINS estimator with challenging cases shown in Fig. 7. The total trajectory length is 247.36 meters and the final position drift is 3.28 meters. The error is 1.3 % of the total trajectory length. However, considering that during the experiment, the sensor suite reach angular velocity up to 120 degree/s, which causes significant motion blur, we can still claim that overall estimation accuracy is high.

VII. Conclusion and Future Work

In this paper, we propose a novel monocular VINS state estimator for real-time state estimation with unknown initialization and camera-IMU calibration. Specifically, our system initializes the velocity, attitude, visual scale, and

(a) Large rotation (b) Motion blur

(c) People walking (d) Mirror

Fig. 7. Onboard images during experiment in large scale environments.

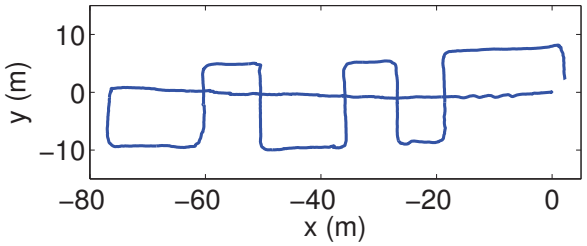


Fig. 8. Position estimation with our monocular VINS estimator in a complex indoor environment. The total trajectory length is 247.36 m and the final position drift is 3.28 m. A video of the experiment can be found at `http://www.ece.ust.hk/~eeshaojie/SSRR2015.mp4`.

camera-IMU calibration automatically while the system is performing free motion in natural environments. Our system is able to automatically identify convergence of the calibration parameters and switch between different system modules. After the initialization, a nonlinear optimization runs real-time recursively for high-accuracy state estimation. Online experimental results are presented to demonstrate the performance of our approach.

A limitation of monocular VINS is the need for motion excitation for scale observability. A multi-camera system may solve the problem but the extrinsic calibration process can be troublesome. In the future, we may extend our framework into a generalized multi-camera-inertial system for elimination of degenerate motions and online calibration of transformations between multiple cameras and the IMU.

REFERENCES

[1] A. Bachrach, S. Prentice, R. He, and N. Roy, "RANGE-robust autonomous navigation in gps-denied environments," *J. Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[2] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proc. of the Intl. Sym. of Robot. Research*, Flagstaff, AZ, Aug. 2011.

[3] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[4] M. Li and A. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Intl. J. Robot. Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[5] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," in *Proc. of Robot.: Sci. and Syst.*, Berlin, Germany, June 2013.

[6] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Seattle, WA, May 2014.

[7] S. Shen, "Autonomous navigation in complex indoor and outdoor environments with micro aerial vehicles," Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA, Aug. 2014.

[8] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Initialization-free monocular visual-inertial estimation with application to autonomous MAVs," in *Proc. of the Intl. Sym. on Exp. Robot.*, Marrakech, Morocco, 2014.

[9] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robot. Autom. Mag.*, vol. 21, no. 3, 2014.

[10] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Intl. J. Robot. Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[11] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Intl. J. Robot. Research*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[12] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. and Auton. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.

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

[14] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Hong Kong, China, May 2014.

[15] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Saint Paul, MN, May 2012, pp. 957–964.

[16] L. Kneip, S. Weiss, and R. Siegwart, "Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, San Francisco, CA, Sept. 2011, pp. 2235–2241.

[17] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International journal of computer vision*, vol. 106, no. 2, pp. 138–152, 2014.

[18] V. Lippiello and R. Mebarki, "Closed-form solution for absolute scale velocity estimation using visual and inertial data with a sliding least-squares estimation," in *Proc. of Mediterranean Conf. on Control and Automation*, Platanias-Chania, Crete, Greece, June 2013, pp. 1261–1266.

[19] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[20] L. Heng, G. H. Lee, and M. Pollefeys, "Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle," in *Proc. of Robot.: Sci. and Syst.*, Berkeley, CA, 2014.

[21] T. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Vilamoura, Algarve, Portugal, Oct. 2012, pp. 1064–1071.

[22] D. Nister, "An efficient solution to the five-point relative pose problem," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, Madison, WI, June 2003, pp. 195–202.

[23] P. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 73–101, 1964.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the Intl. Joint Conf. on Artificial Intelligence*, Vancouver, Canada, Aug. 1981, pp. 24–28.