Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs

Shaojie Shen, Yash Mulgaonkar, Nathan Michael and Vijay Kumar

Abstract The quest to build smaller, more agile micro aerial vehicles has led the research community to address cameras and Inertial Measurement Units (IMUs) as the primary sensors for state estimation and autonomy. In this paper we present a monocular visual-inertial system (VINS) for an autonomous quadrotor which relies only on an inexpensive off-the-shelf camera and IMU, and describe a robust state estimator which allows the robot to execute trajectories at 2 m/s with roll and pitch angles of 20 degrees, with accelerations over 4 m/s^2. The main innovations in the paper are an approach to estimate the vehicle motion without initialization and a method to determine scale and metric state information without encountering any degeneracy in real time.

1 Introduction

Micro-aerial vehicles (MAVs) are ideal platforms for missions in complex confined environments due to its small size and superior mobility. Recently, there have been great successes in deploying sensor-equipped autonomous micro-aerial vehicles (MAVs) in complex GPS-denied environments. In particular, monocular visual-inertial systems (VINS) that consists of a camera and a low cost IMU are very attractive to MAVs with limited payload budget due to their small footprint, low

S. Shen (✉) · Y. Mulgaonkar · V. Kumar
GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA
e-mail: eeshaojie@ust.hk

Y. Mulgaonkar
e-mail: yashm@seas.upenn.edu

V. Kumar
e-mail: kumar@seas.upenn.edu

N. Michael
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: nmichael@cmu.edu

M.A. Hsieh et al. (eds.), *Experimental Robotics*, Springer Tracts
in Advanced Robotics 109, DOI 10.1007/978-3-319-23778-7_15

Fig. 1 Our quadrotor experimental platform equipped with a Intel NUC computer, a MEMS IMU, and a camera. This quadrotor can be launched without initialization at a preferred state. It can autonomously maneuver through different trajectories, and recover from possible failure conditions on-the-fly

cost, and low maintenance. However, most existing monocular VINS approaches are inoperable without a good initial state estimate, nor are they able to recover/restart on the fly in case the estimator fails due to the lack of initial condition [1, 2]. Such autonomous MAVs are only capable of launching with known initial condition (e.g. stationary), and their operating envelop is limited in order to reduce the risk of estimator failure. This motivates the goal of this work as developing a dynamically-launchable and failure-recoverable MAV to autonomously fly through a wide variety of trajectories, including hovering, straight line, and aggressive maneuvers (Fig. 1).

Solutions to VINS has been proposed in a *filtering* setting [1–7] and in a *graph-based* optimization/bundle adjustment/smoothing setting [8–10]. Filtering approaches have the advantage of fast processing due to its continuous marginalization of past states, but their performance can be sub-optimal due to early fix of linearization points. Graph-based approaches benefit from iterative re-linearization of states but it requires more computation power. With proper marginalization, a constant complexity sliding window graph-based framework can be obtained [8]. A comparison between filtering and graph-based approaches is presented in [11]. The authors reported nearly identical results of two types of approaches. However, the platform for verification is only equipped with an optical flow sensor that is unable to perform long term feature tracking. This limits the power of graph-based approach, as a sufficiently connected graph is never constructed.

We can also categorize VINS solutions as *loosely coupled* [1] or *tightly coupled* [2–7, 9, 10]. Loosely coupled approaches usually utilize an independent vision processing module such as PTAM [12] for up-to-scale pose estimation, and integrating the vision pose with IMU using a filter for scale estimation. Tightly coupled approaches usually lead to better estimation results (up to linearization error) because they integrate camera measurement and noise models in a systematic manner.

However, all approaches mentioned above require good initializations due to the underlying linearized solver of the nonlinear VINS system.

Pioneering work on VINS *without* initialization is proposed in [13], where the authors proposed to perform the estimation in the body frame of the first pose in the sliding window. An IMU pre-integration technique is proposed to handle multi-rate sensor measurements. The authors show that the nonlinearity of the system mainly arises only from rotation drift.

Recent results suggests that by assuming the orientation is known or by estimating the rotation from short-term integrated gyroscope output, VINS may be solved in a linear *closed-form* [14–18]. It has been shown that both the initial gravity vector and the body frame velocity can be estimated linearly. These results have significant implications that a good initialization of the VINS problem may actually not be required. However, [14] is limited to use a fixed small number of IMU measurements, which makes it very sensitive to IMU noise. Approaches that utilize multiple IMU measurements in a sliding window [15–18] do not scale well to a large number of IMU measurements since they relie on double integration of accelerometer output over an extended period of time. Moreover, these closed-form approaches do not take the noise characteristic of the system into account, which lead to sub-optimal results.

Another issue of monocular VINS is the scale ambiguity due to degenerate motion. It is well known that in order to render the scale observable, accelerations in at least two axes are required [2, 4, 16]. However, for a MAV, degenerate motions such as hovering or constant velocity motions are unavoidable. The hover case is first addressed in [7] by proposing a last-in-first-out (LIFO) sliding window approach. However, [7] generates pessimistic covariance estimates due to the state-only measurement update scheme.

To address the problems of initialization, failure recovery, and degenerate motion altogether, we identify the contribution of this paper as threefold.

- We propose a linear sliding window formulation for monocular VINS that is able to estimate necessary navigation states (velocity and attitude) without any prior initial information (Sects. 2 and 4).
- We address the issues of degenerate motion and scale unobservability by proposing a two-way marginalization scheme (Sect. 3).
- We experimentally show that the proposed approach enables metric state estimation without initialization. We also show that our system is able to handle degenerate motion cases (Sect. 6).

2 Linear Sliding Window VINS Estimator

2.1 Notation

We consider (G) as the earth's inertial frame, (B) as the current IMU body frame, (B_k) as the camera frame while taking the kth image, Note that IMU usually runs at a higher rate than the camera, and that multiple IMU measurements may exist between

B_k and B_{k+1}. We assume that the camera and the IMU is pre-calibrated such that the camera optical axis is aligned with the z-axis of the IMU. $\mathbf{p}_Y^X$, $\mathbf{v}_Y^X$, and $\mathbf{R}_Y^X$ are 3D position , velocity, and rotation of frame X with respect to frame Y. $\mathbf{g}^G = [0,\ 0,\ g]^{\mathrm{T}}$ is the gravity vector in the world frame, and $\mathbf{g}^X$ is the earth's gravity vector expressed in frame X.

2.2 Formulation

Given two time instants (corresponds to two image frames), the IMU propagation model for position and velocity, expressed in the world frame, can be written as:

$$\begin{aligned}\mathbf{p}_{B_{k+1}}^G &= \mathbf{p}_{B_k}^G + \mathbf{v}_{B_k}^G \Delta t + \iint_{B_k}^{B_{k+1}} \left(\mathbf{R}_B^G \mathbf{a}^B - \mathbf{g}^G\right) dt^2 \\ \mathbf{v}_{B_{k+1}}^G &= \mathbf{v}_{B_k}^G + \int_{B_k}^{B_{k+1}} \left(\mathbf{R}_B^G \mathbf{a}^B - \mathbf{g}^G\right) dt \end{aligned} \tag{1}$$

where $\mathbf{a}^B$ is the accelerometer measurement in the body frame, Δt is the time difference between B_k and B_{k+1}. It can be seen that the rotation between the world frame and the body frame is required in order to propagate the states with IMU measurements. This rotation can only be determined if the initial attitude of the vehicle is known, which is not the case when the vehicle is dynamically launched or recovering from estimator failure. However, as suggested in [13], if the reference frame of the IMU propagation model is attached to the first pose of the VINS system (i.e. the first pose that we are trying to estimate), (1) can be rewritten as:

$$\begin{aligned}\mathbf{p}_{B_{k+1}}^{B_0} &= \mathbf{p}_{B_k}^{B_0} + \mathbf{v}_{B_k}^{B_0} \Delta t - \mathbf{g}^{B_0} \Delta t^2/2 + \mathbf{R}_{B_k}^{B_0} \alpha_{B_{k+1}}^{B_k} \\ \mathbf{v}_{B_{k+1}}^{B_0} &= \mathbf{v}_{B_k}^{B_0} - \mathbf{g}^{B_0} \Delta t + \mathbf{R}_{B_k}^{B_0} \beta_{B_{k+1}}^{B_k} \\ \alpha_{B_{k+1}}^{B_k} &= \iint_{B_k}^{B_{k+1}} \mathbf{R}_B^{B_k} \mathbf{a}^B dt^2 \\ \beta_{B_{k+1}}^{B_k} &= \int_{B_k}^{B_{k+1}} \mathbf{R}_B^{B_k} \mathbf{a}^B dt \end{aligned} \tag{2}$$

where $\mathbf{R}_{B_k}^{B_0}$ is the change in rotation since B_0, which can be obtained by combing the integral gyroscope measurements and relative epipolar constraints (Sect. 2.3). $\alpha_{B_{k+1}}^{B_k}$ and $\beta_{B_{k+1}}^{B_k}$ can be obtained solely with IMU measurements within Δt. We can see that the update equations for all the key quantities ($\mathbf{p}_{B_k}^{B_0}$, $\mathbf{v}_{B_k}^{B_0}$, $\mathbf{g}^{B_0}$) are now linear. It is thus expected that VINS system may be solved in a linear fashion, even without any knowledge of the initial condition.

2.3 Linear Rotation Estimation

The IMU propagation model in (2) can only be linear if good rotation estimates are provided. Although integrating gyroscope measurements will lead to reasonable rotation estimates, it still drifts over time. Therefore, we utilize additional epipolar constraints to eliminate rotation drift. We wish to estimate $R_{B_k}^{B_0}, k = 0, \ldots, N$, subject to following conditions:

$$\mathbf{R}_{B_0}^{B_0} = \mathbf{I}_3, \quad \mathbf{R}_{B_j}^{B_0} = \hat{\mathbf{R}}_{B_j}^{B_i} \mathbf{R}_{B_i}^{B_0} \tag{3}$$

where $\hat{\mathbf{R}}_{B_j}^{B_i}$ is a rotation that is obtained by either integrating gyroscope measurements between two consecutive images, or by finding the essential matrices between the current image and corresponding past images. For each incoming image, we try to compute the essential matrix between it and all other images within the sliding window.

As in [19], the above system can be solved linearly by relaxing orthonormality constraints of the rotations. Specifically, for a pair of rotation matrices $\mathbf{R}_{B_i}^{B_0}, \mathbf{R}_{B_j}^{B_0}$, and their relative constraint $\hat{\mathbf{R}}_{B_j}^{B_i}$, we have:

$$\left[\mathbf{I}_3, \; -\hat{\mathbf{R}}_{B_j}^{B_i}\right] \begin{bmatrix} \mathbf{r}_i^k \\ \mathbf{r}_j^k \end{bmatrix} = 0 \quad k = 1, 2, 3 \tag{4}$$

where $\mathbf{r}_i^k$ is the kth column of $\mathbf{R}_{B_i}^{B_0}$. The solution of the relaxed approximate rotation matrices can be found as the last three columns of the right singular matrix of the system (4). The true rotation matrices can then be obtained by enforcing unit singular values on the approximated rotation matrices: $\mathbf{R}_{B_i}^{B_0} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$, where $\bar{\mathbf{R}}_{B_i}^{B_0} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$ is an approximate rotation matrix. After this point, we assume that the rotation components within the sliding window is known and noise-free.

2.4 Linear Sliding Window Estimator

We apply a tightly-coupled, sliding window graph-based [8] formulation due to its constant computation complexity and its ability to incorporate constrains from multiple observations to refine its solution. The *full state* vector can be expressed as (the transpose is ignored for the simplicity of presentation):

$$\begin{aligned} \mathcal{X} &= \left[\mathbf{x}_{B_0}^{B_0}, \; \mathbf{x}_{B_1}^{B_0}, \; \ldots \mathbf{x}_{B_N}^{B_0}, \; \lambda_0, \; \lambda_1, \; \ldots \lambda_M\right] \\ \mathbf{x}_{B_k}^{B_0} &= \left[\mathbf{p}_{B_k}^{B_0}, \; \mathbf{v}_{B_k}^{B_k}, \; \mathbf{g}^{B_k}\right] \; \text{for k} = 1, \ldots, \text{N} \\ \mathbf{p}_{B_0}^{B_0} &= [0, \; 0, \; 0] \end{aligned} \tag{5}$$

where $\mathbf{x}_{B_k}^{B_0}$ is the kth *camera state*, N is the number of camera states in the sliding window, M is the number of all features that have been observed for at least twice within the sliding window. λ_l is the depth of the lth point feature from its first observation. We are able to use a one-dimensional representation for features due to the nature of the underlying image processing pipeline (Sect. 5.2). This saves significant amount of computation power. We keep the body frame velocity ($\mathbf{v}_{B_k}^{B_k}$) and gravity vector ($\mathbf{g}^{B_k}$) in the camera state. This helps reducing the impact of rotation error on the estimation results (Sect. 2.5).

Since the rotation is fixed as in Sect. 2.3, we can formulate the linear VINS by gathering all measurements from both the IMU and the monocular camera and solve for the maximum likelihood estimate by minimizing the sum of the Mahalanobis norm of all measurement errors:

$$\min_{\mathcal{X}} \left\{ \left(\mathbf{b}_p - \Lambda_p \mathcal{X}\right) + \sum_{k \in \mathcal{D}} \left\| \hat{\mathbf{z}}_{B_{k+1}}^{B_k} - \mathbf{H}_{B_{k+1}}^{B_k} \mathcal{X} \right\|_{\mathbf{P}_{B_{k+1}}^{B_k}}^2 + \sum_{(l,j) \in \mathcal{C}} \left\| \hat{\mathbf{z}}_l^{B_j} - \mathbf{H}_l^{B_j} \mathcal{X} \right\|_{\mathbf{P}_l^{B_j}}^2 \right\} \tag{6}$$

where the measurement triplets $\{\hat{\mathbf{z}}_{B_{k+1}}^{B_k}, \mathbf{H}_{B_{k+1}}^{B_k}, \mathbf{P}_{B_{k+1}}^{B_k}\}$ and $\{\hat{\mathbf{z}}_l^{B_j}, \mathbf{H}_l^{B_j}, \mathbf{P}_l^{B_j}\}$ are defined in Sects. 2.5 and 2.6 respectively. $\mathcal{D}$ is the set of all IMU measurements. $\mathcal{C}$ is the set of all observations between any features and any camera states within the sliding window. $\{\mathbf{b}_p, \Lambda_p\}$ is the *optional* prior for the system. This system can be solved by reorganizing in the following form:

$$\left(\Lambda_p + \Lambda_{imu} + \Lambda_{cam}\right) \mathcal{X} = \left(\mathbf{b}_p + \mathbf{b}_{imu} + \mathbf{b}_{cam}\right) \tag{7}$$

where $\{\Lambda_{imu}, \mathbf{b}_{imu}\}$ and $\{\Lambda_{cam}, \mathbf{b}_{cam}\}$ are information matrices and vectors for IMU and camera measurements respectively.

It should be noted that since the cost is linear with respect to the states, the system in (7) can have unique solution *without* the prior (initial condition):

$$\left(\Lambda_{imu} + \Lambda_{cam}\right) \mathcal{X} = \left(\mathbf{b}_{imu} + \mathbf{b}_{cam}\right) \tag{8}$$

This is the key to enable dynamic launching and failure recovery of MAVs. However, as will be shown in Sect. 3, there are degenerate motions for the monocular VINS setup, which will render the scale unobservable using only measurements within the sliding window. In such case, it is desirable to marginalize out states that are about to be removed from the window and convert them a prior to (implicitly) propagate the scale.

2.5 IMU Measurement Model

Given the locally drift-free rotation, we can rewrite (2) as a linear function of the state $\mathcal{X}$:

$$\begin{bmatrix} \alpha_{B_{k+1}}^{B_k} \\ \beta_{B_{k+1}}^{B_k} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{B_0}^{B_k}\left(\mathbf{p}_{B_{k+1}}^{B_0} - \mathbf{p}_{B_k}^{B_0}\right) - \mathbf{v}_{B_k}^{B_k}\Delta t + \mathbf{g}^{B_k}\frac{\Delta t^2}{2} \\ \mathbf{R}_{B_{k+1}}^{B_k}\mathbf{v}_{B_{k+1}}^{B_{k+1}} - \mathbf{v}_{B_k}^{B_k} + \mathbf{g}^{B_k}\Delta t \\ \mathbf{R}_{B_{k+1}}^{B_k}\mathbf{g}^{B_{k+1}} - \mathbf{g}^{B_k} \end{bmatrix} = \mathbf{H}_{B_{k+1}}^{B_k}\mathcal{X} \tag{9}$$

The last block line in (9) represents prediction of the gravity vector. We estimate the gravity vector for each pose in order to avoid the negative effects due to possible accumulated rotation error. All variables except the position component are independent of the accumulated rotation $\mathbf{R}_{B_0}^{B_k}$, making them insensitive to rotation error. The linear IMU measurement model has the form:

$$\mathbf{z}_{B_{k+1}}^{B_k} \sim \mathcal{N}\left(\mathbf{H}_{B_{k+1}}^{B_k}\mathcal{X}, \begin{bmatrix} \mathbf{P}_{B_{k+1}}^{B_k{}^{\alpha\beta}} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{B_{k+1}}^{B_k{}^{\mathbf{g}}} \end{bmatrix}\right) \tag{10}$$

Note that the terms $\alpha_{B_{k+1}}^{B_k}$ and $\beta_{B_{k+1}}^{B_k}$ and correlated since they both come from IMU measurements within Δt. Their joint covariance matrix $\mathbf{P}_{B_{k+1}}^{B_k{}^{\alpha\beta}}$ can be calculated using the pre-integration technique proposed in [13].

2.6 Camera Measurement Model

Let the lth feature be first detected in the ith frame. The observation of this feature in the jth normalized image plane $[u_l^{B_j}, v_l^{B_j}]^{\mathrm{T}}$ can be expressed as:

$$\lambda_l^{B_j}\begin{bmatrix} u_l^{B_j} \\ v_l^{B_j} \\ 1 \end{bmatrix} = \mathbf{R}_{B_0}^{B_j}\left(\mathbf{p}_{B_i}^{B_0} - \mathbf{p}_{B_j}^{B_0} + \lambda_l\mathbf{R}_{B_i}^{B_0}\begin{bmatrix} u_l^{B_i} \\ v_l^{B_i} \\ 1 \end{bmatrix}\right) \tag{11}$$

where $\lambda_l^{B_j}$ is the depth of the feature in the jth frame. We use tracking, instead of a descriptor-based method, as the tool for data association (Sect. 5.2). As such, the first observation *defines* the direction of a feature, and $[u_l^{B_i}, v_l^{B_i}]$ is noise-free. Note that (11) is now linear with respect to the state, but nonlinear to the image measurement since the depth is initially unknown. The unknown depth transforms into a unknown

weighting factor to the measurement covariance. Still, we can rewrite (11) as:

$$\mathbf{0} = \begin{bmatrix} -1 & 0 & u_l^{B_j} \\ 0 & -1 & v_l^{B_j} \end{bmatrix} \mathbf{R}_{B_0}^{B_j} \left(\mathbf{p}_{B_i}^{B_0} - \mathbf{p}_{B_j}^{B_0} + \lambda_l \mathbf{R}_{B_i}^{B_0} \begin{bmatrix} u_l^{B_i} \\ v_l^{B_i} \\ 1 \end{bmatrix} \right) = \mathbf{H}_l^{B_j} \mathcal{X} \tag{12}$$

and the camera measurement model has the form:

$$\mathbf{z}_l^{B_j} \sim \mathcal{N} \left(\mathbf{H}_l^{B_j} \mathcal{X},\ {\lambda_l^{B_j}}^2 \bar{\mathbf{P}}_l^{B_j} \right) \tag{13}$$

where $\bar{\mathbf{P}}_l^{B_j}$ is the feature observation noise in the normalized image plane. Note that although ${\lambda_l^{B_j}}^2$ is initially unknown, we can initialize it as the average depth of the scene. In practice, we found the solution very insensitive to the initial value of ${\lambda_l^{B_j}}^2$ as long as it is set to be *larger* than the actual depth. The reason for this is a possible direction for future research. Once the system is solved we can update the value of ${\lambda_i^{B_k}}^2$. The new value is used for subsequent optimization as long as both $\mathbf{p}_{B_k}^{B_0}$ and λ_l are still within the sliding window.

3 Handling Scale Ambiguity

It is well known that in order to render the scale of a monocular VINS observable, the IMU has to excite nonzero accelerations in at least two axes [2, 4, 16]. This is particularly critical for a initial-free sliding window estimator discussed in Sect. 2. In fact, when the vehicle is undergoing degenerate motions, such as constant velocity or hovering, and without any prior information, it can be verified that the position and velocity components of the solution of (8) in this situation can be scaled arbitrary without violating any constraints. Unfortunately, zero acceleration motion is unavoidable for a hover-capable MAV and it must be handled properly.

If the vehicle first undergoes generic motions with sufficient excitation in acceleration ($B_0, \ldots, B_n$), and then enters constant velocity motion ($B_{n+1}, \ldots, B_{N+n}$), the scale can only be observable if the camera states correspond to generic motion are included in the sliding window. This is unrealistic if available computation only allows N camera states in the sliding window. However, if we can provide an initial estimate of $\mathbf{x}_{B_{n+1}}^{B_0}$, we will be able to propagate (not observe) the scale from B_n to B_{n+1}. Naturally, this can be done by proper marginalization of $\mathbf{x}_{B_n}^{B_0}$ as it is removed from the window at the $(N+n)$th step.

For hovering, as proved in [7], if the vehicle first undergoes generic motions with sufficient acceleration excitation during $(B_0, \ldots, B_{N-1})$, and then enters a hover (B_N), the scale observability can be preserved by using a last-in-first-out (LIFO) sliding window scheme. [7] performs state-only measurement update during hovering, and covariance is updated only once as the vehicle exits hovering. This is due to the fact that features are not kept in the state, but instead marginalized out as covariance update is performed. However, this approach will lead to pessimistic covariance as observations obtained during hovering is not used to update the covariance.

3.1 Two-Way Marginalization

Based on the previous discussions, we propose a novel two-way marginalization scheme to handle both constant velocity and hovering cases. The pseudo code is shown in Algorithm 1. Consider the full state vector $\mathcal{X} = [\mathbf{x}_{B_0}^{B_0}, \ldots \mathbf{x}_{B_{N-1}}^{B_0} \,|\, \lambda_{\mathcal{L}}]$, where $\lambda_{\mathcal{L}}$ is the set of all features that have at least two observations within the sliding window. We add the next camera state $(\mathbf{x}_{B_N}^{B_0})$ to the sliding window if any of the following two criteria are satisfied:

1. The time between two images Δt is larger than δ.
2. After eliminating the relative rotation, the average parallax of all common features between the most recent two images is larger than ε.

The first condition ensures that the error in the integrated IMU measurement (Sect. 2.5) between two camera states is bounded, while second condition ensures that the new camera state is added when translation motion of the vehicle with respect to the scene is significant. We require that all newly added features $\lambda_{\mathcal{L}^+}$ to have at least two observations to ensure successful triangulation (Line 1). The system is then solved with all available measurements within the sliding window plus any available prior (Line 2).

We keep a variable $s = \textit{float/fix}$ to indicate whether we should marginalize out the second newest camera state $(\mathbf{x}_{B_{N-1}}^{B_0})$ or the oldest one $(\mathbf{x}_{B_0}^{B_0})$. To marginalize a chosen camera state $\mathbf{x}_{B_k}^{B_0}$, we first remove the camera state and all features $\lambda_{\mathcal{I}^-}$ that are first observed by it (Lines 5 and 13). We then construct a new prior based on all measurements related to the removed states (Lines 4 and 12):

$$\Lambda_p^+ = \Lambda_p + \sum_{k \in \mathcal{D}^-} \mathbf{H}_{B_{k+1}}^{B_k\,T} \mathbf{P}_{B_{k+1}}^{B_k\,-1} \mathbf{H}_{B_{k+1}}^{B_k} + \sum_{(l,j) \in \mathcal{C}^-} \mathbf{H}_l^{B_j\,T} \mathbf{P}_l^{B_j\,-1} \mathbf{H}_l^{B_j} \tag{14}$$

where $\mathcal{D}^-$ and $\mathcal{C}^-$ are sets of removed IMU and camera measurements respectively. The marginalization can the be carried out via Schur Complement [8].

The value of s is reevaluated after the marginalization based solely on the parallax between two most recent remaining images (Lines 6–9 and 14–17). Intuitively, our approach will keep removing the recent camera states if the vehicle has small or no motion. Keeping older camera states in this case will preserve acceleration information that is necessary to recover the scale, while still preserve all information provided by the marginalized states. On the other hand, if the vehicle is under fast constant speed motion, older camera states will be removed and converted into priors for the subsequent estimates. We do note that the scale in this case is subject to drifting. However, without global loop closure, marginalization is the best that can be done to propagate the scale information forward, while still maintaining constant computation complexity. Figure 2 demonstrates different working scenarios of the proposed marginalization approach.

We also note that due to the marginalization, the information matrix in (7) will eventually become dense. However, since our formulation is linear, no iterative method is required. In practice, we find that even a dense matrix solver is able to give real-time performance with tens of camera states and hundreds of features.

Algorithm 1 Two-Way Marginalization

Require:

$$\mathcal{X} \leftarrow \left[\mathbf{x}_{B_0}^{B_0}, \ldots \mathbf{x}_{B_{N-1}}^{B_0} \mid \lambda_{\mathcal{L}}\right]$$
$$s \leftarrow \textit{float}\ \textbf{or}\ \textit{fix}$$
$$\{\Lambda_p,\ \mathbf{b}_p\} \leftarrow \text{Prior Information}$$

Ensure: $\Delta t > \delta$ **or** $\text{Parallex}(\mathbf{x}_{B_{N-1}}^{B_0}, \mathbf{x}_{B_N}^{B_0}) > \varepsilon$

1: $\mathcal{X} \leftarrow \mathcal{X} \cup \left[\mathbf{x}_{B_N}^{B_0} \mid \lambda_{\mathcal{L}^+}\right]$
2: Solve $\mathcal{X}$ using (6) and (7), optionally with $\{\Lambda_p,\ \mathbf{b}_p\}$
3: **if** $s = \textit{float}$ **then**
4: $\{\Lambda_p,\ \mathbf{b}_p,\ \lambda_{\mathcal{L}^-}\} \leftarrow \text{Marginalization}(\mathbf{x}_{B_{N-1}}^{B_0})$
5: $\mathcal{X} \leftarrow \mathcal{X} \setminus [\mathbf{x}_{B_{N-1}}^{B_0} \mid \lambda_{\mathcal{L}^-}]$
6: **if** $\text{Parallex}(\mathbf{x}_{B_{N-2}}^{B_0}, \mathbf{x}_{B_N}^{B_0}) < \varepsilon$ **then**
7: $s \leftarrow \textit{float}$
8: **else**
9: $s \leftarrow \textit{fix}$
10: **end if**
11: **else**
12: $\{\Lambda_p,\ \mathbf{b}_p,\ \lambda_{\mathcal{L}^-}\} \leftarrow \text{Marginalization}(\mathbf{x}_{B_0}^{B_0})$
13: $\mathcal{X} \leftarrow \mathcal{X} \setminus [\mathbf{x}_{B_0}^{B_0} \mid \lambda_{\mathcal{L}^-}]$
14: **if** $\text{Parallex}(\mathbf{x}_{B_{N-1}}^{B_0}, \mathbf{x}_{B_N}^{B_0}) < \varepsilon$ **then**
15: $s \leftarrow \textit{float}$
16: **else**
17: $s \leftarrow \textit{fix}$
18: **end if**
19: **end if**
20: **return** $\{\mathcal{X},\ \Lambda_p,\ \mathbf{b}_p,\ s\}$

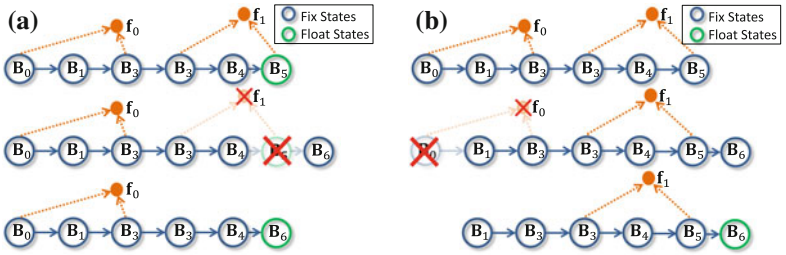

Fig. 2 **a** Structure of the full state before, during, and after marginalizing a recent camera state ($\mathbf{B}_5$) after a newer camera state $\mathbf{B}_6$ is added. Similar marginalization process of the oldest camera state is shown in (**b**)

4 Initialization and Failure Recovery

Our linear VINS formulation (Sect. 2) naturally allows on-the-fly initialization and failure recovery. In fact, these two tasks are identical, as both involve solving the linear system (7) with no priors after sufficient images are collected.

After initialization, in order to allow concurrently running the two-way marginalization algorithm for handling scale ambiguity (Sect. 3.1), we maintain two subsystems, which corresponds to two arrays of image/IMU measurements and camera/feature states. For the first subsystem, which is used for two-way marginalization, the newest two camera states may be separated arbitrarily far in time if the vehicle is hovering. On the other hand, the second subsystem refreshes independent of vehicle motion. It always maintains a queue structure where the oldest camera state and its corresponding measurements are removed as new image comes in. This way, we make sure that the IMU measurement in the queue always have bounded error.

When the system is in normal operation, only the first subsystem is solved and the second subsystem only collects data. A failure is indicated by insufficient features in the environment. When failure occurs, the first array, as well as all prior information, are discarded. Instead, we repeatedly try to find a valid solution using measurements within the second subsystem. Once a valid solution is found, we put all measurements back to the first array and resume normal operation. The block diagram of this failure recovery mechanism is shown in Fig. 3.

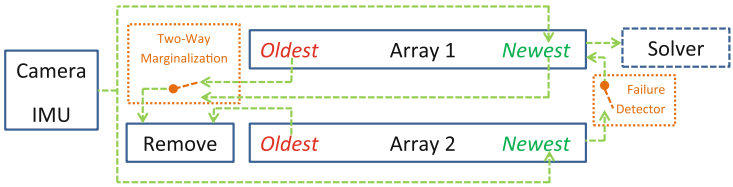

Fig. 3 Block diagram of the proposed failure recovery approach. Note the switching mechanism for two-way marginalization and the failure detector.

5 Implementation Details

5.1 Experimental Platform

The experimental platform shown in Fig. 1 is based on the Pelican quadrotor from Ascending Technologies, GmbH.[1] This platform is natively equipped with an AutoPilot board consisting of an IMU and a user-programmable ARM7 microcontroller. The main computation unit onboard is an Intel NUC with a 1.8 GHz Core i3 processor with 8 GB of RAM and a 120 GB SSD. The only addition to onboard sensing is a mvBlueFOX-MLC200w grayscale HDR camera with standard lens that capture 752×480 images at 25 Hz. The total mass of the platform is 1.39 kg, which leads to a thrust to weight ratio of approximately two. The entire algorithm is developed in C++ using ROS[2] as the interfacing robotics middleware.

5.2 Real-Time Implementation

Although the onboard camera captures images at 25 Hz, it is both computationally infeasible and unnecessary to perform the optimization (Sect. 2) at such a high rate. The system starts with one camera state in the sliding window, and a fixed number of corner features detected in that camera image. We utilize the KLT tracker to track features in the high-rate image sequence until the next camera state is added to the sliding window (Sect. 3.1). At this point, we apply RANSAC with epipolar constraints for outlier rejection, and then add extra features if computation permits. The pre-integrated IMU measurement (Sect. 2.5) is also computed as new camera state is added. All tracked features are used for rotation estimation (Sect. 2.3), however, only a subset of features with strong corner responses are used for position estimation (Sect. 2.4). We marginalize out additional features if the total number of features goes beyond a threshold.

We utilize multi-thread implementation to achieve real-time operation. Three threads run concurrently. The first thread is the image processing front end that we just described. The second thread is the main VINS optimizer (Sect. 2) and the marginalization module (Sect. 3.1). Finally, due to computation constraint, our VINS system runs at 10 Hz with approximate processing latency of 30 ms. This is not sufficient for autonomous control of MAVs. We therefore implement a third thread to propagate the latest VINS solution forward using the high-rate IMU measurements. The output of this thread is used directly as the feedback for the trajectory tracking controller. A breakdown of computation time of each components in our system is shown in Table 1. It suggests that our algorithm is able to run stably onboard.

[1] Ascending Technologies, GmbH, http://www.asctec.de/.

[2] Robot Operating System, http://www.ros.org/.

Table 1 Computation breakdown of major modules in our system for 30 camera states and 200 features

Module	Time (ms)	Rate (Hz)	Thread
Feature tracking	5	25	1
Add new camera state and features	14	10	1
Rotation estimation	2	10	2
Linear sliding window estimator	20	10	2
Marginalization	17	10	2
IMU forward propagation	1	100	3

6 Experimental Results

The experiments are conducted in a cluttered lab space with Vicon[3] motion capture system for ground truth comparison purpose. We highlight that in all experiments, the MAV has no prior knowledge of the environment, not even its own initial speed and attitude. With our linear formulation, the VINS estimator is initialized as the MAV takes off. This on-the-fly initialization enables rapid deployment of monocular visual-inertial systems.

6.1 Trajectory Tracking with Onboard State Estimation

We first test the performance of using the output from our VINS estimator for feedback control of an autonomous MAV. In this experiment, the MAV is set to autonomously fly through a figure eight pattern at different speeds. The time parameterized trajectory is generated using the minimum jerk cost function [20], which proved to be beneficial for vision-based approaches by reducing angular velocities.

In Fig. 4a, c and e, the MAV flies at average speed of 1 m/s with maximum acceleration of 1 m/s^2. It is well known that the position and the yaw angle of the platform in the world frame is unobservable without global reference sensors such as GPS. Indeed, we can see a drift in x, y, z at $\{-0.0584,\ 0.1191,\ 0.1229\}$ meters, and yaw at 1.563 degrees. On the other hand, the body frame velocity of the MAV, as well as the attitude which can be derived from the estimated gravity vector, remains observable throughout the flight. The standard deviation in three dimensional velocity compares close to the ground truth as $\{0.0734,\ 0.0410,\ 0.0229\}$, while the deviation in pitch and roll angles are $\{0.2487,\ 0.2477\}$, units in m/s and degrees, respectively. In Fig. 4b, d and f, the MAV is commanded a much higher speed of 2 m/s with maximum

[3] Vicon, http://www.vicon.com/.

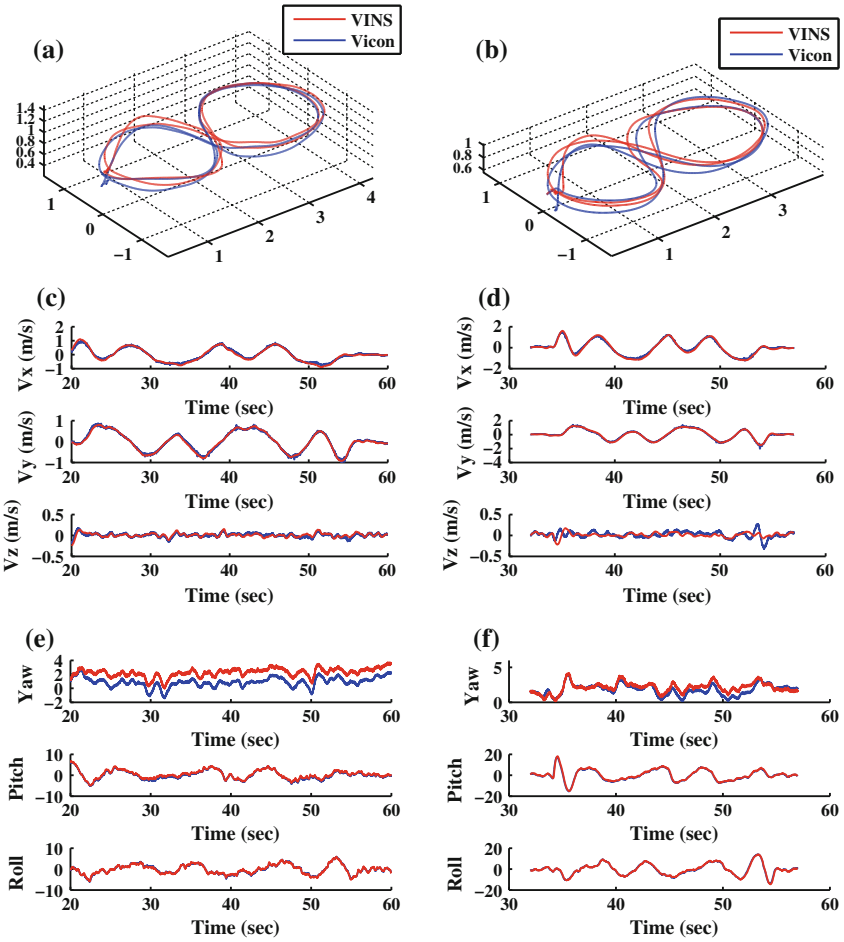

Fig. 4 The vehicle is set to autonomously fly through a figure eight pattern at the speed/acceleration of $\{1\ \text{m/s},\ 1\ \text{m/s}^2\}$ and $\{2\ \text{m/s},\ 4\ \text{m/s}^2\}$ respectively. Videos of the experiments are available at http://mrsl.grasp.upenn.edu/shaojie/ISER2014.mp4. **a** Slow velocity. **b** Fast velocity. **c** Slow orientation. **d** Fast orientation

acceleration of $4\ \text{m/s}^2$. The drift in position is now $\{0.0319,\ 0.0357,\ 0.2329\}$ meters, while the yaw drift remains small as 0.409 degrees. The error statistics for velocity and attitude are $\{0.0921,\ 0.0678,\ 0.0731\}$ and $\{0.4744,\ 0.4563\}$ in meters and degrees respectively.

It can be seen that although the speed, acceleration, as well as attitude of the MAV increases significantly for the faster trajectory, the estimation quality largely remains the same. This highlights the robustness of our system for handling fast maneuvers.

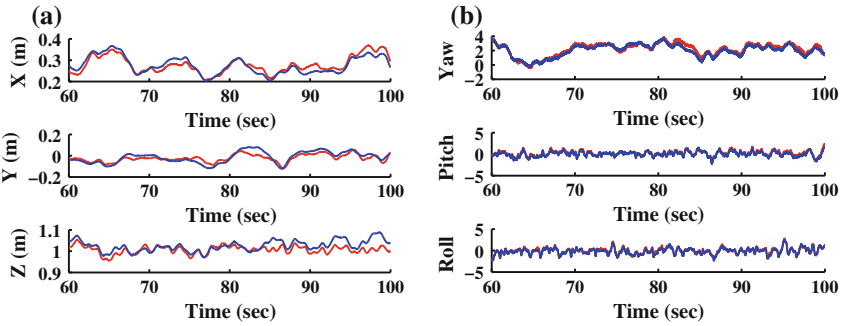

Fig. 5 Hover performance of the MAV using onboard state estimates for feedback control comparing with ground truth. Note that there is no drift in all directions. **a** Position, **b** Orientation

6.2 Hover Performance

One major issue of using a monocular VINS sensor suite for autonomous MAVs is the lack of direct measurement of metric scale. To this end, we use this experiment to highlight the effectiveness of our two-way marginalization (Sect. 3.1) scheme. The MAV takes off from the floor without any initial knowledge of its states. It is commanded to hover after the on-the-fly initialization is completed. As shown in Fig. 5, since the two-way marginalization always removes newer camera states while hovering, previous feature observations that have sufficient parallax, as well as IMU measurements that have sufficient accelerations are kept. As such, drift-free estimates in full 6 degree-of-freedom is obtained. During hovering, the onboard position estimate compares well with the ground truth with standard deviation of $\{0.018,\ 0.028,\ 0.019\}$ meters. The hover performance with such onboard estimates has the standard deviation of $\{0.041,\ 0.053,\ 0.025\}$.

7 Conclusion and Future Work

In this paper, we presented what we believe to be the first autonomous quadrotor capable of fast navigation (2 m/s traversal of 0.9 m arcs with 0.5 g acceleration) that relies only on a single off-the-shelf camera and an IMU and does not require careful initialization. The main technical challenges we overcome stem from the fact that the state is not observable in general, and specifically, the scale is ambiguous for constant velocity or hover motions. The main practical challenge has to do with robustness to sudden changes in the number and quality of observed features and the dependence on robot/camera motions. Both of these challenges are addressed using a linear sliding window formulation for estimation and a two-way marginalization scheme that enable the estimation of the necessary navigation states without any prior initial information while being robust to degeneracies in the motion. We show

results on our 1.4 kg quadrotor and show that the errors in an unstructured indoor environment are less than 5 cm using ground truth measurements from a motion capture system.

In the future, we would like to investigate into control and planning methodologies for generating motions that excite sufficient, but just enough accelerations for scale observability. We would also like to develop perception and mapping modules to enable autonomous obstacle avoidance in cluttered environments.

Acknowledgments We gratefully acknowledge support from ARL Micro Autonomous Systems and Technology Collaborative Technology Alliance Grant No. W911NF-08-2-0004.

References

1. Weiss, S., Achtelik, M.W., Lynen, S., Chli, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 957–964. Saint Paul, MN, May 2012
2. Jones, E.S., Soatto, S.: Visual-inertial navigation, mapping and localization: a scalable real-time causal approach. Intl. J. Robot. Res. **30**(4), 407–430 (2011)
3. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3565–3572. Roma, Italy, Apr 2007
4. Kelly, J., Sukhatme, G.S.: Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration. Intl. J. Robot. Res. **30**(1), 56–79 (2011). January
5. Kottas, D.G., Hesch, J.A., Bowman, S.L., Roumeliotis, S.I.: On the consistency of vision-aided inertial navigation. In: Proceedings of the International Symposium on Experimental Robotics, Quebec, Canada, June 2012
6. Li, M., Mourikis, A.: High-precision, consistent ekf-based visual-inertial odometry. Int. J. Robot. Res **32**(6), 690–711 (2013)
7. Kottas, D., Wu, K., Roumeliotis, S.: Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, Nov 2013 (To appear)
8. Sibley, G., Matthies, L., Sukhatme, G.: Sliding window filter with application to planetary landing. J. Field Robot. **27**(5), 587–608 (2010)
9. Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., Siegwart, R.: Keyframe-based visual-inertial SLAM using nonlinear optimization. In: Proceedings of Robotics: Science and Systems, p. 0. Berlin, Germany June 2013
10. Indelman, V., Melim, A., Dellaert, F.: Incremental light bundle adjustment for robotics navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1952–1959. Tokyo, Japan, Nov 2013
11. Lange, S., Sunderhauf, N., Protzel, P.: Incremental smoothing vs. filtering for sensor fusion on an indoor UAV. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1773–1778. Karlsruhe, Germany, May 2013
12. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07), Nara, Japan, Nov 2007
13. Lupton, T., Sukkarieh, S.: Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Trans. Robot. **28**(1), 61–76 (2012)

14. Kneip, L., Weiss, S., Siegwart, R.: Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2235–2241. San Francisco, CA, Sept 2011
15. Dong-Si, T., Mourikis, A.I.: Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1064–1071. Vilamoura, Algarve, Portugal, Oct 2012
16. Martinelli, A.: Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. IEEE Trans. Robot. **28**(1), 44–60 (2012)
17. Martinelli, A.: Closed-form solution of visual-inertial structure from motion. IEEE Trans. Robot. (2013)
18. Lippiello, V., Mebarki, R.: Closed-form solution for absolute scale velocity estimation using visual and inertial data with a sliding least-squares estimation. In: Proceedings of Mediterranean Conference on Control and Automation, pp. 1261–1266. Platanias-Chania, Crete, Greece, June 2013
19. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Proceedings of the IEEE International Conference on Pattern Recognition, pp. 1–8. Minneapolis, MN (2007)
20. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In: Proceedings of Robotics: Science and Systems. Berlin, Germany (2013)