2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Detecting and Dealing with Hovering Maneuvers in Vision-aided Inertial Navigation Systems

Dimitrios G. Kottas[†], Kejian J. Wu[‡], and Stergios I. Roumeliotis[†]

Abstract— **In this paper, we study the problem of hovering (i.e., absence of translational motion) detection and compensation in Vision-aided Inertial Navigation Systems (VINS). We examine the system's unobservable directions for two common hovering conditions (with and without rotational motion) and propose a robust motion-classification algorithm, based on both visual and inertial measurements. By leveraging our observability analysis and the proposed motion classifier, we modify existing state-of-the-art filtering algorithms, so as to ensure that the number of the system's unobservable directions is minimized. Finally, we validate experimentally the proposed modified sliding window filter, by demonstrating its robustness on a quadrotor with rapid transitions between hovering and forward motions, within an indoor environment.**

I. Introduction and Related Work

Current approaches to 3D localization rely on inertial measurements units (IMUs) that provide rotational velocity and linear acceleration measurements. Low-cost, commercial-grade IMUs, however, suffer from the presence of noise and bias in the inertial measurements, which when integrated even over a short period of time, can result in unreliable estimates. When available, GPS measurements can be employed for aiding an inertial navigation system (INS). Many robotic applications, however, require operation in GPS-denied areas (e.g., indoors or within urban canyons). A Vision-aided INS (VINS) employs camera observations of tracked features over multiple time steps for imposing geometric constraints between the motion of the vehicle and the structure of the observed scene. Such geometric constraints, provide corrections to the pose (position and orientation) estimates of an INS, and can significantly improve the localization accuracy within GPS-denied areas. As a result, recent advances in VINS have led to successful applications to ground [1], [2], aerial [3], [4], and space exploration [5] vehicles.

Existing approaches to VINS rely either on filtering or bundle-adjustment (BA)-based optimization methods. BA methods, originally developed for problems in photogrammetry [6] and computer vision [7], [8], perform batch optimization, without marginalization, over all the variables, including the entire robot trajectory and every detected landmark using all available measurements. In order to reduce the BA's computational complexity, different approximate methods have been developed that either optimize over a subset of measurements and variables or solve the BA problem intermittently. An example of the first category of relaxed solutions to the vision-only BA formulation is the Parallel Tracking and Mapping (PTAM) algorithm, originally developed by Klein and Murray [9] for augmented reality applications within confined spaces. PTAM manages to bound the increasing complexity of BA-based methods as new poses and features are added to the state vector, by optimizing over a fixed number of camera poses (*key-frames*) and mapped features. Such a framework is efficient and robust under the assumption that the camera observes the same scene over long periods of time [10]. As the scene changes, new key-frames and mapped features may be added, while past poses and features, as well as all their associated measurements are ignored. Although observing the same scene is a common scenario for augmented reality applications, it is rather restrictive for robotic vehicles, where exploration of large areas is often required. As a result, PTAM, when modified to be fused with an INS in a loosely coupled manner for the purpose of micro aerial vehicle (MAV) localization, needs special consideration for failure detection during rapid changes of the observed scene [11].

Among the methods that incrementally solve the BA problem, iSAM2 has been applied to a GPS-aided INS, employing visual measurements [12]. In order to reduce the prohibitively high computational complexity of solving the full BA problem, the authors employ factorization-updating methods which allow reusing the information matrix available from previous steps. Computationally demanding procedures, however, such as relinearization followed by batch factorization, are only performed when a variable significantly deviates from the current estimate. Nevertheless, due to the accumulation of fill-ins between periodic batch steps, the iSAM2's efficiency degrades when many variables are affected at every relinearization step [13].

Recursive filtering approaches to VINS can be classified into two main categories. The first one comprises non-trivial extensions of EKF-based SLAM algorithms [14], appropriately modified for VINS [2], where the estimator's state vector includes both the pose of the vehicle and a map of the environment. EKF-SLAM can deal with both cases of hovering and exploration. Its high computational complexity (quadratic in the number of mapped features), however, limits its applicability to small-size areas. In contrast, sliding window filtering approaches, avoid the inclusion of a map of the environment by maintaining a sliding window

[†]D. G. Kottas, and S. I. Roumeliotis are with the Department of Computer Science and Engineering, Univ. of Minnesota, Minneapolis, MN 55455, USA. Emails: {dkottas,stergios}@cs.umn.edu

[‡]K. J. Wu is with the Department of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, MN 55455, USA. Email: kejian@cs.umn.edu

This work was supported by the University of Minnesota (UMN) through the Digital Technology Center (DTC) and AFOSR (FA9550-10-1-0567).

of past camera poses. Among these methods, the Multi-State Constrained Kalman Filter (MSC-KF) [1] exploits all available geometric information provided by the camera measurements, while keeping its computational complexity *linear* in the number of features observed over the filter's window. Although the MSC-KF has been successfully applied to various applications (e.g., [1], [5]), and has been demonstrated to operate in real time [15], [16], it is not suitable for scenarios that include hovering over the same scene, since it requires sufficient baseline between the camera poses within the sliding window.

At this point, we define two distinct cases of motion. By *hovering* we describe the case of zero translation, while by *generic motion* we refer to motion profiles that excite sufficient degrees of freedom, so that the number of unobservable directions of the VINS reaches its minimum [17], [15].

Recent work on VINS, addresses the case of hovering by utilizing hybrid filter estimators that include both a sliding window of camera poses, as well as a fixed number of mapped landmarks [3], [18], or by separately building a map of the environment [19]. Although such methods bound the processing cost of SLAM (the number of mapped landmarks in the state vector is kept small), their performance during a hovering scenario hinges upon the criterion employed for selecting which features to be included in the state vector.

The present paper's contributions address the above limitations by appropriately modifying the sliding window over which the MSC-KF operates, so as to perform robustly both under hovering and generic motion conditions, without the need of building a map of the environment. Specifically:

- We analyze the observability properties of a VINS when hovering, with and without rotational motions, and show that it has 5 and 7 unobservable degrees of freedom (dof), respectively. This is in contrast to the case of a VINS under generic motions where the number of unobservable dof is 4 [15].
- We prove that for a sliding window-based estimator, such as the MSC-KF, whose state vector comprises camera poses corresponding to both motion profiles (i.e., generic motions and hovering), the number of unobservable dof remains 4.
- We propose a method for classifying the vehicle's motion into hovering versus non-hovering, by utilizing visual information from the feature tracks.
- We leverage the results of our observability analysis, as well as the proposed motion-classification algorithm, for deciding which frames to be added/dropped from the MSC-KF, while keeping the filter's computational complexity linear in the number of observed features.
- Finally, we demonstrate the robustness of the proposed approach by testing it on a MAV rapidly transitioning between hovering and generic motions.

The rest of the paper is organized as follows: In Sect. II, we describe the system and measurement models used by the MSC-KF. Subsequently, (Sect. III) we present the observability analysis of a VINS under hovering, which we leverage for appropriately modifying the MSC-KF. The proposed method is validated experimentally in Sect. IV. Finally, we provide our concluding remarks and outline our future research directions in Sect. V.

II. Background

In what follows, we first present the system model used for state and covariance propagation based on inertial measurements (Sect. II-A), and then describe the measurement model for performing tightly-coupled visual-inertial odometry through the MSC-KF framework.

A. IMU State Model

The 16×1 IMU state vector is:

$$\mathbf{x}_R = \begin{bmatrix} {}^I\bar{q}_G^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T \end{bmatrix}^T. \tag{1}$$

The first component of the IMU state is ${}^I\bar{q}_G(t)$ which is the unit quaternion representing the orientation of the *global frame* $\{G\}$ in the IMU frame, $\{I\}$, at time t. The frame $\{I\}$ is attached to the IMU, while $\{G\}$ is a local-vertical reference frame whose origin coincides with the initial IMU position. The IMU state also includes the position, ${}^G\mathbf{p}_I(t)$, and velocity, ${}^G\mathbf{v}_I(t)$, of $\{I\}$ in $\{G\}$, while $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ denote the gyroscope and accelerometer biases, respectively.

The system model describing the time evolution of the state is (see [20]):

$$\begin{gathered} {}^I\dot{\bar{q}}_G(t) = \tfrac{1}{2}\Omega({}^I\omega(t)){}^I\bar{q}_G(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \\ \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \end{gathered} \tag{2}$$

where ${}^I\omega$ and ${}^G\mathbf{a}$ are the rotational velocity and linear acceleration, $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}(t)$ are the white-noise processes driving the IMU biases, and

$$\Omega(\omega) \triangleq \begin{bmatrix} -\lfloor\omega\times\rfloor & \omega \\ -\omega^T & 0 \end{bmatrix}, \quad \lfloor\omega\times\rfloor \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The gyroscope and accelerometer measurements are:

$$\omega_m(t) = {}^I\omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{3}$$

$$\mathbf{a}_m(t) = \mathbf{C}({}^I\bar{q}_G(t))\,({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{4}$$

where $\mathbf{C}(\bar{q})$ is the rotation matrix corresponding to the quaternion $\bar{q}$, ${}^G\mathbf{g}$ is the gravitational acceleration expressed in $\{G\}$, and $\mathbf{n}_g(t)$, $\mathbf{n}_a(t)$ are white-noise processes contaminating the corresponding measurements. Linearizing at the current estimates and applying the expectation operator on both sides of (2), we obtain the IMU state propagation model:

$$\begin{gathered} {}^I\dot{\hat{\bar{q}}}_G(t) = \frac{1}{2}\Omega({}^I\hat{\omega}(t)){}^I\hat{\bar{q}}_G(t), \;\; {}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t) \\ {}^G\dot{\hat{\mathbf{v}}}_I(t) = \mathbf{C}^T({}^I\hat{\bar{q}}_G(t))\,\hat{\mathbf{a}}(t) + {}^G\mathbf{g}, \;\; \dot{\hat{\mathbf{b}}}_g(t) = \mathbf{0}_{3\times 1}\;, \;\; \dot{\hat{\mathbf{b}}}_a(t) = \mathbf{0}_{3\times 1} \end{gathered} \tag{5}$$

where $\hat{\mathbf{a}}(t) \triangleq \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$, and ${}^I\hat{\omega}(t) \triangleq \omega_m(t) - \hat{\mathbf{b}}_g(t)$. By defining the 15×1 error-state vector as:[1]

$$\tilde{\mathbf{x}}_R = \begin{bmatrix} {}^I\delta\theta_G^T & \tilde{\mathbf{b}}_g^T & {}^G\tilde{\mathbf{v}}_I^T & \tilde{\mathbf{b}}_a^T & {}^G\tilde{\mathbf{p}}_I^T \end{bmatrix}^T, \tag{6}$$

[1]For the IMU position, velocity, and biases, we use a standard additive error model (i.e., $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the error in the estimate $\hat{\mathbf{x}}$ of a random variable $\mathbf{x}$). To ensure minimal representation for the covariance, we employ a multiplicative attitude error model where the error between the quaternion $\bar{q}$ and its estimate $\hat{\bar{q}}$ is the 3×1 angle-error vector, $\delta\boldsymbol{\theta}$, implicitly defined by the *error quaternion* $\delta\bar{q} = \bar{q}\otimes\hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^T & 1 \end{bmatrix}^T$, where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide.

the continuous-time IMU error-state equation becomes:

$$\dot{\tilde{\mathbf{x}}}_R(t) = \mathbf{F}_R(t)\tilde{\mathbf{x}}_R(t) + \mathbf{G}_R(t)\mathbf{n}(t) \tag{7}$$

where $\mathbf{F}_R(t)$ is the error-state transition matrix and $\mathbf{G}_R(t)$ is the noise input matrix, with

$$\mathbf{F}_R(t) = \begin{bmatrix} -\lfloor \hat{\boldsymbol{\omega}}(t)\times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^I\hat{\bar{q}}_G(t))\lfloor \hat{\mathbf{a}}(t)\times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I\hat{\bar{q}}_G(t)) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$$\mathbf{G}_R(t) = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I\hat{\bar{q}}_G(t)) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

and $\mathbf{n}(t) \triangleq \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T \end{bmatrix}^T$ is the system noise with autocorrelation $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \mathbf{Q}_R\delta(t-\tau)$, where $\delta(.)$ is the Dirac delta; $\mathbf{Q}_R$ depends on the IMU noise characteristics and is computed offline.

The state transition matrix from time t_1 to t_k, $\Phi_{k,1}$, is computed in analytical form [21] as the solution to the matrix differential equation $\dot{\Phi}_{k,1} = \mathbf{F}_R(t_k)\Phi_{k,1}$, $\Phi_{1,1} = \mathbf{I}_{15}$:

$$\Phi_{k,1} = \begin{bmatrix} \Phi_{k,1}^{(1,1)} & \Phi_{k,1}^{(1,2)} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{k,1}^{(3,1)} & \Phi_{k,1}^{(3,2)} & \mathbf{I}_3 & \Phi_{k,1}^{(3,4)} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \Phi_{k,1}^{(5,1)} & \Phi_{k,1}^{(5,2)} & (t_k - t_1)\mathbf{I}_3 & \Phi_{k,1}^{(5,4)} & \mathbf{I}_3 \end{bmatrix}. \tag{8}$$

Finally, the discrete-time system noise covariance matrix is computed as: $\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \Phi_{k,\tau}\mathbf{G}_R(\tau)\mathbf{Q}_R\mathbf{G}_R^T(\tau)\Phi_{k,\tau}^T \mathrm{d}\tau$.

B. MSC-KF Propagation Model

As the sensor platform moves in the environment, the camera observes point features, which are tracked across images. Generally, in a VINS [22], these measurements are exploited to concurrently estimate the motion of the sensing platform and, optionally, the structure of the environment. The MSC-KF [1] is a VINS that performs tightly-coupled visual-inertial odometry over a sliding window of N poses, while maintaining linear complexity in the number of observed features. The key advantage of the MSC-KF is that it utilizes all constraints for each feature observed by the camera over N poses, without requiring to build a map or estimate the features as part of the state vector. Each time the camera records an image, a stochastic clone [23], of the sensor pose is created. This enables the utilization of delayed image measurements; in particular, it allows all observations of a given feature $\mathbf{f}_i$ to be processed during a single update step (when the first pose that observed the feature is about to be marginalized). Hence, at a given time-step k, the filter tracks the 16×1 evolving state, $\mathbf{x}_{R_k}$ [see (1)], as well as the cloned sensor poses $\{\mathbf{x}_C = \begin{bmatrix} {}^{I_{k-N+i}}\bar{q}_G^T & {}^G\mathbf{p}_{I_{k-N+i}}^T \end{bmatrix}^T\}$, $i = 0,\ldots,N-1$ corresponding to the last N images. That is:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{x}_C^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{x}_{C_{k-1}}^T & \cdots & \mathbf{x}_{C_{k-N}}^T \end{bmatrix}^T \tag{9}$$

Correspondingly, the covariance consists of the 15×15 block of the evolving state, $\mathbf{P}_{RR}$, the $6N\times 6N$ block corresponding to the cloned robot poses, $\mathbf{P}_{CC}$, and their cross-correlations, $\mathbf{P}_{RC}$. Hence, the covariance of the augmented state vector has the following structure:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{RR} & \mathbf{P}_{RC} \\ \mathbf{P}_{RC}^T & \mathbf{P}_{CC} \end{bmatrix} \tag{10}$$

During propagation, the current state estimate evolves forward in time by integrating (5), while the cloned poses are static. The covariance propagation of the entire state is given by:

$$\mathbf{P}_{RR} \leftarrow \Phi_{k+1,k}\mathbf{P}_{RR}\Phi_{k+1,k}^T + \mathbf{Q}_k \tag{11}$$

$$\mathbf{P}_{RC} \leftarrow \Phi_{k+1,k}\mathbf{P}_{RC} \tag{12}$$

$$\mathbf{P}_{CC} \leftarrow \mathbf{P}_{CC} \tag{13}$$

C. MSC-KF Update Model

In this section, we describe the processing of a single feature $\mathbf{f}_i$, which was first observed by the oldest clone corresponding to time-step $k-N$, and then reobserved over the cloned camera poses corresponding to time-steps $k-N,\ldots,k-1$.[2] We employ the pinhole camera model to describe the perspective projection of the 3D point $\mathbf{f}_i$ on the image plane and model the measurement $\mathbf{z}_\ell^i$ at time step ℓ as:

$$\mathbf{z}_\ell^i = \frac{1}{z_\ell^i}\begin{bmatrix} x_\ell^i \\ y_\ell^i \end{bmatrix} + \eta_\ell^i, \quad \begin{bmatrix} x_\ell^i \\ y_\ell^i \\ z_\ell^i \end{bmatrix} = {}^{I_\ell}\mathbf{f}_i = \mathbf{C}\left({}^{I_\ell}\bar{q}_G\right)\left({}^G\mathbf{f}_i - {}^G\mathbf{p}_{I_\ell}\right) \tag{14}$$

where the noise η_ℓ^i follows a Gaussian distribution with zero mean and covariance $\mathbb{E}[\eta_\ell^i\eta_\ell^{iT}] = \sigma_\eta^2\mathbf{I}_2$. Note also that, without loss of generality, we express the image measurement in normalized pixel coordinates, and consider the camera frame to be coincident with the IMU frame[3]. By differentiating the nonlinear measurement model (14) with respect to the augmented state (9), we obtain the linearized measurement Jacobian:

$$\tilde{\mathbf{z}}_\ell^i = \mathbf{H}_{c,\ell}^i\begin{bmatrix} \mathbf{H}_{\theta_G,\ell}^i & \mathbf{H}_{p_I,\ell}^i \end{bmatrix}\tilde{\mathbf{x}}_{C_\ell} + \mathbf{H}_{c,\ell}^i\mathbf{H}_{f,\ell}\tilde{\mathbf{f}}_i + \eta_\ell^i \tag{15}$$

$$= \mathbf{H}_{c,\ell}^i\begin{bmatrix} \mathbf{0}_{3\times 15} & \mathbf{0}_{3\times 6} & \cdots & \underbrace{\begin{bmatrix} \mathbf{H}_{\theta_G,\ell}^i & \mathbf{H}_{p_I,\ell}^i \end{bmatrix}}_{\ell\text{-th clone position}} & \cdots & \mathbf{0}_{3\times 6} \end{bmatrix}\tilde{\mathbf{x}}_k + \mathbf{H}_{c,\ell}^i\mathbf{H}_{f,\ell}\tilde{\mathbf{f}}_i + \eta_\ell^i \tag{16}$$

$$= \mathbf{H}_{x,\ell}^i\tilde{\mathbf{x}}_k + \mathbf{H}_{c,\ell}^i\mathbf{H}_{f,\ell}\tilde{\mathbf{f}}_i + \eta_\ell^i \tag{17}$$

where

$$\mathbf{H}_{c,\ell}^i = \frac{1}{\hat{z}_\ell^i}\begin{bmatrix} 1 & 0 & \frac{-\hat{x}_\ell^i}{\hat{z}_\ell^i} \\ 0 & 1 & \frac{-\hat{y}_\ell^i}{\hat{z}_\ell^i} \end{bmatrix}, \ \mathbf{H}_{\theta_G,\ell}^i = \lfloor {}^{I_\ell}\hat{\mathbf{f}}_i\times \rfloor \tag{18}$$

$$\mathbf{H}_{p_I,\ell} = -\mathbf{C}\left({}^{I_\ell}\hat{\bar{q}}_G\right), \quad \mathbf{H}_{f,\ell} = \mathbf{C}\left({}^{I_\ell}\hat{\bar{q}}_G\right).$$

[2] The interested reader is referred to [1] on how the same methodology can be applied efficiently to multiple features.

[3] We perform both intrinsic camera and extrinsic IMU-camera calibration off-line [24], [25].

After collecting all measurements of feature $\mathbf{f}_i$ across time-steps $k-N,...,k-1$, we arrive at:

$$\underbrace{\begin{bmatrix}\tilde{\mathbf{z}}^i_{k-1}\\ \vdots \\ \tilde{\mathbf{z}}^i_{k-N}\end{bmatrix}}_{\tilde{\mathbf{z}}^i} = \underbrace{\begin{bmatrix}\mathbf{H}^i_{x,k-1}\\ \vdots \\ \mathbf{H}^i_{x,k-N}\end{bmatrix}}_{\mathbf{H}^i_x}\tilde{\mathbf{x}}_k + \underbrace{\begin{bmatrix}\mathbf{H}^i_{c,k-1}\mathbf{H}_{f,k-1}\\ \vdots \\ \mathbf{H}^i_{c,k-N}\mathbf{H}_{f,k-N}\end{bmatrix}}_{\mathbf{H}^i_f}\tilde{\mathbf{f}}_i + \boldsymbol{\eta}^i_k \quad (19)$$

So as to avoid including $\mathbf{f}_i$ into the state vector, the feature is marginalized by projecting (19) onto the left nullspace $\mathbf{W}$ of $\mathbf{H}^i_f$. This yields

$$\mathbf{W}^T\tilde{\mathbf{z}}^i = \mathbf{W}^T\mathbf{H}^i_x\tilde{\mathbf{x}}_k + \mathbf{W}^T\boldsymbol{\eta}^i_k \quad (20)$$

which we employ to update the state and covariance estimates using the standard EKF update equations. After the update, we marginalize out the oldest cloned pose, by removing $\mathbf{x}_{C_{k-N}}$ from $\mathbf{x}_k$, and dropping the corresponding rows and columns of $\mathbf{P}$.

III. Problem Description and Solution

As described in the previous section, the MSC-KF algorithm processes visual observations over a sliding window of camera poses. Thus, at every time-step, we need to decide which camera pose to include/remove from the sliding window, so that its size remains constant. A natural choice, especially during exploration tasks, would be the first-in-first-out (FIFO) scheme, i.e., to remove the oldest camera pose and replace it with the newest (current) one. This scheme performs robustly when the platform is undergoing generic motions [5]. In the case of hovering, however, (i.e., when the platform stays at the same position for a period of time) FIFO-based MSC-KF would fail. To address this issue, we propose a last-in-first-out (LIFO) image management approach for the MSC-KF where we replace the image last included in the sliding window with the one currently provided by the camera. The motivation for switching from a FIFO to a LIFO strategy is that we want to ensure that there is always sufficient baseline between camera poses included in the sliding window. The exact impact of this selection (i.e., FIFO vs. LIFO) on the observability properties of the system is discussed in the following sections (Sect. III-A and Sect. III-B), while the criterion for switching between FIFO and LIFO is presented in Sect. III-C.

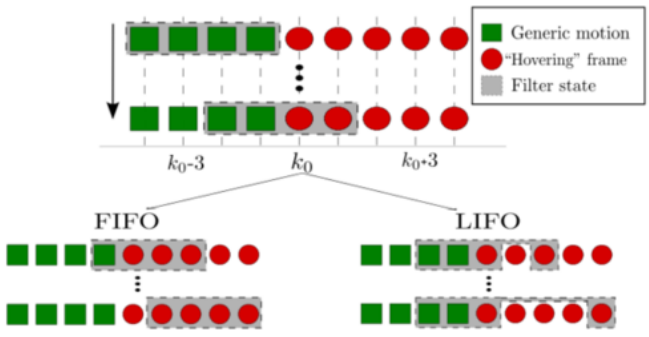
A. FIFO: Exploration Mode

1) FIFO Scheme:

Assume that at time-step k, the sliding window of camera poses corresponds to the states [see (9)]

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}^T_{R_k} & \mathbf{x}^T_C\end{bmatrix}^T = \begin{bmatrix}\mathbf{x}^T_{R_k} & \mathbf{x}^T_{C_{k-1}} & \cdots & \mathbf{x}^T_{C_{k-N}}\end{bmatrix}^T \quad (21)$$

Then, the camera observations from time steps $k-N, k-N+1,...,k-1$, are processed for updating the state and covariance using the measurement model of (20). For the next time-step, the FIFO scheme first drops the state $\mathbf{x}_{C_{k-N}}$ which corresponds to the oldest camera pose inside this window. Then, the newest state $\mathbf{x}_{R_k}$ is cloned into $\mathbf{x}_{C_k}$,



(a)

Fig. 1: FIFO vs. LIFO: The evolution of the sliding window of filter states (shaded area) during hovering (poses in red circles). The bottom figures show that while eventually all the states in the FIFO scheme correspond to hovering camera poses, the states in LIFO always contain generic-motion camera poses.

followed by propagation. So at time-step $k+1$ the window of states becomes

$$\mathbf{x}_{k+1} = \begin{bmatrix}\mathbf{x}^T_{R_{k+1}} & \mathbf{x}^T_C\end{bmatrix}^T = \begin{bmatrix}\mathbf{x}^T_{R_{k+1}} & \mathbf{x}^T_{C_k} & \cdots & \mathbf{x}^T_{C_{k-N+1}}\end{bmatrix}^T \quad (22)$$

after which an update is performed and the same FIFO procedure is repeated (see Fig. 1).

In short, the FIFO scheme *slides* the window of camera poses forward in time, which is the most commonly used image management scheme employed by sliding window filters.

2) FIFO During Generic Motions and Hovering - Observability Analysis:

In what follows, we evaluate the performance of the FIFO MSC-KF by studying the observability properties of the corresponding VINS model. As shown in [17], [15], when *generic motions* of the camera poses are involved, the VINS model has four unobservable directions: three corresponding to global translations, and one to rotations about the gravity vector.

Hereafter, we study the case when the platform hovers, meaning that there is little or no change in the positions of the camera poses, while the camera may rotate or stand still. Assume that hovering starts at time step k_0 and the size of the sliding window is N. Then, in this case, because of the FIFO scheme, the camera poses in the sliding window of states $\mathbf{x}_k$, with $k \geq k_0+N$, will all correspond to hovering states. So for any time step $k \geq k_0+N$, the FIFO MSC-KF model is equivalent to a VINS model with only hovering motions, i.e., no translation between consecutive camera poses.

At this point, we state the first main result of our observability analysis

Theorem 1.1. *The linearized VINS model, for the case when multiple features (≥ 3) are observed by a sensor platform performing no translational motion, but with generic rotational motions, has five unobservable directions: three for global translations, one for rotations around the gravity vector (yaw), and one for scale.*

Theorem 1.2. *The linearized VINS model, for the case when multiple features (≥ 3) are observed by a sensor platform performing no translational or rotational motion, has seven unobservable directions: three for global translations, three for rotations (roll, pitch, and yaw), and one for scale.*

Proof: see Appendix part A and B.

Thus, for the FIFO MSC-KF, when the platform hovers, more unobservable directions appear besides the inevitable four ones. This will lead to degradation of the system performance: when the scale is unobservable, we cannot estimate the scale of the motion or the scene. Furthermore, when the roll and pitch angles are also unobservable, we cannot measure gravity in the local (sensor) frame ${}^{I}\mathbf{g}$, which in turn makes it impossible to extract the true body accelerations ${}^{G}\mathbf{a}$ from the accelerometer readings [see (4)].

B. LIFO: Hovering Mode

In what follows, we introduce an alternative LIFO-based scheme that will extract the same information during hovering as in the case of generic motions, i.e., the corresponding VINS model has only four unobservable directions.

1) LIFO Scheme:

Assume that the sensor platform starts hovering at time-step k_0, with generic motions before k_0, and the size of the sliding window is N. We employ the FIFO MSC-KF for the generic motion time interval, i.e., $k \leq k_0$. Thus, at the time-step k_0+1, we have the following sliding window of the states from FIFO [see (22)]

$$\mathbf{x}_{k_0+1} = \begin{bmatrix} \mathbf{x}_{R_{k_0+1}}^T & \mathbf{x}_C^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{R_{k_0+1}}^T & \mathbf{x}_{C_{k_0}}^T & \cdots & \mathbf{x}_{C_{k_0-N+1}}^T \end{bmatrix}^T \tag{23}$$

Once we detect that the platform is in hovering mode between time-step k_0 and k_0+1 (the criterion is described in Sect. III-C), we switch to the LIFO scheme: instead of dropping the oldest camera pose $\mathbf{x}_{C_{k_0-N+1}}$ as in FIFO, we drop the newest pose $\mathbf{x}_{R_{k_0+1}}$, and replace it with the state corresponding to the next time-step $\mathbf{x}_{R_{k_0+2}}$. This procedure corresponds to the MSC-KF performing propagation only, without any state dropping or cloning. At this point, the sliding window of states becomes

$$\mathbf{x}_{k_0+2} = \begin{bmatrix} \mathbf{x}_{R_{k_0+2}}^T & \mathbf{x}_C^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{R_{k_0+2}}^T & \mathbf{x}_{C_{k_0}}^T & \cdots & \mathbf{x}_{C_{k_0-N+1}}^T \end{bmatrix}^T \tag{24}$$

and a filter update is performed using the camera observations corresponding to the poses in the window. The same process is repeated for as long as the platform continues to hover (see Fig. 1).

Once the sensor platform leaves hovering (i.e., it starts to perform generic motions again), the MSC-KF switches back to FIFO mode.

2) LIFO During Hovering - Observability Analysis:

In what follows, we show that the LIFO-based MSC-KF, designed for dealing with hovering conditions, acquires sufficient information, i.e., the unobservable directions of the corresponding VINS model are the same as in the case of generic motions.

When following the LIFO scheme [see (24)], the sliding window of states for any particular time-step k during hovering is

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{x}_C^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{x}_{C_{k_0}}^T & \cdots & \mathbf{x}_{C_{k_0-N+1}}^T \end{bmatrix}^T \tag{25}$$

where only the latest camera pose is being replaced. There are two parts in this state vector: the right part (time-steps $k_0-N+1,...,k_0-1$), corresponds to camera poses that underwent generic motions, while the left part (time-step k_0 and k), corresponds to hovering poses. Thus, the LIFO MSC-KF for hovering is equivalent to a VINS model where the sensor platform initially performs generic motions and then switches to hovering.

At this point, we state the second main result of our observability analysis

Theorem 2. *The linearized VINS model, for the case when multiple features (≥ 3) are observed by a sensor platform performing generic motions for at least 4 time-steps, and then starting to hover either with or without rotational motions, has only four unobservable directions: three for global translations, and one for rotations around the gravity vector (yaw).*

Proof: see Appendix part C.

Thus, the unobservable directions of the LIFO MSC-KF are exactly the same as the ones of VINS undergoing generic motions, which validates our choice of the LIFO scheme for dealing with hovering conditions.

3) LIFO-based MSC-KF Update:

Assume that the sensor platform hovers from time-step k_0 to time-step k_0+N_H. Ideally we would like to solve a bundle adjustment problem, where the state consists of both generic-motion poses $\mathbf{x}_{C_{k_0-j}}$, $j=0,...,N-1$, and hovering poses $\mathbf{x}_{R_{k_0+s}}$, $s=1,...,N_H$, with all the feature measurements observed during the time interval k_0-N+1 to k_0+N_H. However, if the hovering period lasts for a long time (i.e., N_H is large), it is computationally challenging to solve this optimization problem in a batch form. Moreover, it also suffers from numerical instability due to the lack of sufficient baseline between the hovering poses [8].

Alternatively, the MSC-KF provides us with a framework to solve this optimization problem incrementally [26], in a recursive manner. Specifically, at each time-step k during hovering, we perform a *state-only* MSC-KF update using the measurement model of (20). In contrast, since while hovering we retain the same poses $\mathbf{x}_{C_{k_0-j}}$, $j=0,...,N-1$, in the state vector [see (25)], the covariance should only be updated *once* using all measurements, otherwise the filter will become inconsistent. In our LIFO-based MSC-KF, this covariance update takes place at time step k_0+N_H, right before existing the hovering period.

C. Hovering Detection

The method we employ for detecting whether the camera motion between two consecutive image frames includes a translational component, plays a crucial role for switching in a timely manner between the FIFO and LIFO schemes. We achieve this by appropriately modifying our existing

tightly-coupled visual-inertial framework, for robust feature tracking. Specifically, let $\mathbf{b}_k^i$ denote the unit-norm bearing measurement to a feature (i.e., $\mathbf{b}_k^i = \frac{\mathbf{z}_k^i}{||\mathbf{z}_k^i||_2}$) at time step k. Between two consecutive camera poses, k and $k+1$, all feature observations that correspond to inliers satisfy the epipolar constraint [27]:

$$\mathbf{b}_{k+1}^{iT} \lfloor {}^{I_{k+1}}\mathbf{p}_{I_k} \times \rfloor \mathbf{C}\left({}^{I_{k+1}}\hat{\bar{q}}_{I_k}\right) \mathbf{b}_k^i = 0. \tag{26}$$

where we use the filter's state estimates to evaluate $\mathbf{C}\left({}^{I_{k+1}}\hat{\bar{q}}_{I_k}\right)$.

When there is sufficient baseline between the camera poses, we employ the 2-pt RANSAC [28] to estimate the unit vector of translation ${}^{I_{k+1}}\mathbf{p}_{I_k}$ in (26). In contrast, for zero-translational motions, $\mathbf{b}_{k+1}^i$ is (approximately) parallel to $\mathbf{C}\left({}^{I_{k+1}}\hat{\bar{q}}_{I_k}\right)\mathbf{b}_k^i$ and (26) becomes ill-conditioned. In that case, we employ a 0-pt RANSAC framework, where we classify point correspondences as inliers or outliers by directly using a model provided by the state estimates:

$$||\mathbf{b}_{k+1}^i - \mathbf{C}\left({}^{I_{k+1}}\hat{\bar{q}}_{I_k}\right)\mathbf{b}_k^i||_2 = 0. \tag{27}$$

In particular, we compute

$$d_k = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{b}_{k+1}^i - \mathbf{C}\left({}^{I_{k+1}}\hat{\bar{q}}_{I_k}\right)\mathbf{b}_k^i||_2 \tag{28}$$

and threshold d_k, so as to decide whether the vehicle excited sufficient translational motion, between time-steps k and $k+1$, by setting the boolean variable:

$$\xi_k = 1 \text{ if } d_k < \varepsilon, \text{ else } \xi_k = 0. \tag{29}$$

So as to ensure smooth transitions from hovering to non-hovering decisions, we expect multiple consecutive decisions to be the same, before classifying the robot's motion.

IV. Experimental Results and Implementation Details

We validated the robustness of the proposed approach using a MAV. Our experimental platform, consists of a low-cost quadrotor, the Parrot AR.DRONE, equipped with a low-weight ($\leq$ 100 gr) sensing platform [see Fig. 2 (a)].

Specifically, the sensing modalities comprise a Point Grey Chameleon monochrome camera[4] with resolution 640×480 pixels and an InterSense NavChip IMU[5]. IMU signals were sampled at a frequency of 100 Hz while camera images were acquired at 7.5 Hz using an ARM CPU.

Images and inertial measurements were streamed through the wireless module of the quadrotor and processed in real-time on a ground station computer. A sliding window of 12 images was employed, with MSC-KF updates occurring at 3.75 Hz, while the filter was providing state estimates at the frequency of the IMU (100 Hz) in real-time.

Features were extracted from the first image (last image) of the FIFO (LIFO) sliding window of the images, using the Shi-Tomasi corner detector [29]. While in FIFO (LIFO) mode, features were tracked forward (backward) in the sliding window, using the KLT tracking algorithm [30].

For the purpose of validating the robustness of the proposed algorithm, the quadrotor was commanded to perform rapid transitions from hovering to forward motion and then again to hovering. As it is demonstrated in Fig. 2 (b) and the accompanying video, the proposed localization framework that employs the FIFO/LIFO MSC-KF schemes (depending on the decision of the motion classifier), was able to robustly detect the transitions of the vehicle's motion profile, and successfully track its pose. In contrast, the regular FIFO-only sliding window MSC-KF, using the same window size of 12 images, failed to track the vehicle's pose, during significant periods of hovering and diverged, as predicted by our observability analysis.

V. Conclusion and Future Work

In this paper, we introduced a robust motion classifier for detecting transitions between hovering and generic motions. Additionally, we studied the observability properties of a vision-aided inertial navigation system (VINS) undergoing two different types of motion: (i) hovering-only (i.e., zero-translation) maneuvers, and (ii) motion with sufficient baseline followed by hovering maneuvers. Moreover, we leveraged the results of our observability analysis to introduce a LIFO-FIFO switching strategy for selecting the images processed by a sliding-window filter under different operating conditions. Finally, we demonstrated the robustness of the proposed strategy for dealing with singular motion configurations using a quadrotor rapidly transitioning from hovering to forward motion within an indoor environment. As part of our future work, we plan to further modify existing VINS frameworks so as to incorporate kinematic constraints depending on the vehicle's motion profile.

Appendix

A. Proof of Theorem 1.1

For a VINS model we have the following state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_R^T & {}^G\mathbf{f}_1^T & \cdots & {}^G\mathbf{f}_M^T \end{bmatrix}^T \tag{30}$$

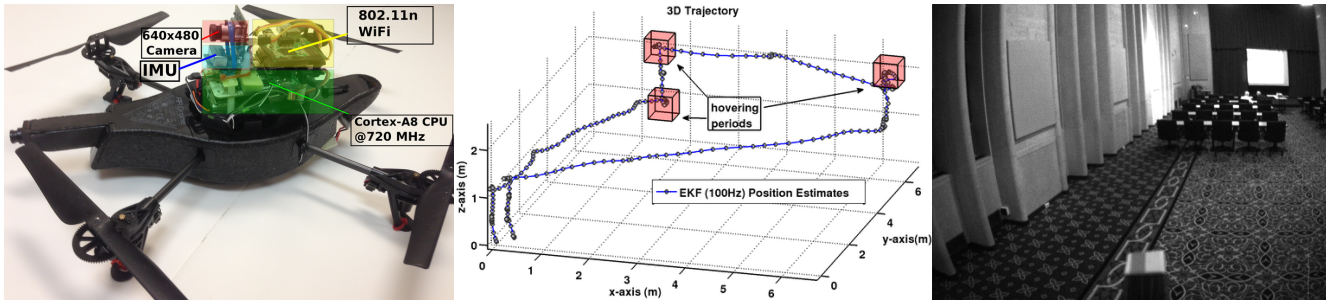
where $\mathbf{x}_R$ is defined in (1), ${}^G\mathbf{f}_i$ is the position of feature[6] i, $i = 1, ..., M$, in the global frame, and M is the number of features with $M \geq 3$.

The observability matrix $\mathbf{M}$ [31] of the VINS model has as its k-th block row $\mathbf{M}_k = \mathbf{H}_k \Phi_{k,1}$, for $k \geq 1$, where $\Phi_{k,1}$ is the state transition matrix from time-step 1 to k [see (8)], and $\mathbf{H}_k$ is the measurement Jacobian of the feature observation model at time-step k [see (14)]. From [21], we have the analytical expression for $\mathbf{M}_k$:

[4] http://www.ptgrey.com

[5] http://www.intersense.com

[6] Note that for the purpose of this observability analysis, we include the feature positions in the state vector. The same analysis holds for the MSC-KF as well since the latter performs sliding window SLAM with marginalization with respect to the feature positions.



(a) (b) (c)

Fig. 2: Experiment: A Parrot AR.DRONE rapidly transitioning from hovering to forward motion maneuvers. (a) Close-view of the quadrotor testbed along with its onboard sensors. (b) 3D view of the overall estimated trajectory with the "hovering" periods annotated. (c) On-board view from the experimental dataset.

$$\begin{bmatrix} \Gamma_1^1 & \begin{bmatrix}\Gamma_2^1 & \Gamma_3^1\end{bmatrix} & -\delta t_k \mathbf{I}_3 & \Gamma_4 & -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \Gamma_1^2 & \begin{bmatrix}\Gamma_2^2 & \Gamma_3^2\end{bmatrix} & -\delta t_k \mathbf{I}_3 & \Gamma_4 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \cdots & \mathbf{0}_3 \\ \vdots & & & & \vdots & & & \ddots & \\ \Gamma_1^M & \begin{bmatrix}\Gamma_2^M & \Gamma_3^M\end{bmatrix} & -\delta t_k \mathbf{I}_3 & \Gamma_4 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{I}_3 \end{bmatrix} \tag{31}$$

where

$$\Gamma_1^i = \mathbf{H}_{c,k}^i \mathbf{C}({}^{I_k}\bar{q}_G) \tag{32}$$

$$\Gamma_2^i = \lfloor {}^G\mathbf{f}_i - {}^G\mathbf{p}_{I_1} - {}^G\mathbf{v}_{I_1}\delta t_k + \frac{1}{2}{}^G\mathbf{g}\,\delta t_k^2 \times \rfloor \mathbf{C}({}^{I_1}\bar{q}_G)^T \tag{33}$$

$$\Gamma_3^i = \lfloor {}^G\mathbf{f}_i - {}^G\mathbf{p}_{I_k} \times \rfloor \mathbf{C}^T({}^{I_k}\bar{q}_G)\Phi_{k,1}^{(1,2)} - \Phi_{k,1}^{(5,2)} \tag{34}$$

$$\Gamma_4 = -\Phi_{k,1}^{(5,4)}. \tag{35}$$

and $i = 1,2,...,M$, is the feature index.

In the case of hovering with generic rotations, since there is no translation and the velocity is zero, we set ${}^G\mathbf{p}_{I_k} = {}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_{I_k} = \mathbf{0}$, for all k. Then, we compute the right nullspace of $\mathbf{M}$, which is:

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}({}^{I_1}\bar{q}_G){}^G\mathbf{g} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{p}_I \times \rfloor {}^G\mathbf{g} & {}^G\mathbf{p}_I \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f}_1 \times \rfloor {}^G\mathbf{g} & {}^G\mathbf{f}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f}_M \times \rfloor {}^G\mathbf{g} & {}^G\mathbf{f}_M \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{t,1} & \mathbf{N}_{r,1} & \mathbf{N}_{s,1} \end{bmatrix} \tag{36}$$

Thus, we have 5 unobservable directions for this model: 3 for global translations ($\mathbf{N}_{t,1}$), 1 for rotations about gravity ($\mathbf{N}_{r,1}$), and 1 for scale ($\mathbf{N}_{s,1}$).

B. Proof of Theorem 1.2

When no rotation is present, compared with the case in part A, we additionally have ${}^{I_k}\bar{q}_G = {}^I\bar{q}_G$. This brings some further simplifications to the elements of $\mathbf{M}$, and its right nullspace becomes:

$$\mathbf{N}_2 = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & -\lfloor \mathbf{C}({}^I\bar{q}_G){}^G\mathbf{g} \times \rfloor & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{p}_I \times \rfloor \mathbf{C}^T({}^I\bar{q}_G) & {}^G\mathbf{p}_I \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f}_1 \times \rfloor \mathbf{C}^T({}^I\bar{q}_G) & {}^G\mathbf{f}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f}_M \times \rfloor \mathbf{C}^T({}^I\bar{q}_G) & {}^G\mathbf{f}_M \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{t,2} & \mathbf{N}_{r,2} & \mathbf{N}_{s,2} \end{bmatrix} \tag{37}$$

Thus, we have 7 unobservable directions for this model: 3 for global translations ($\mathbf{N}_{t,2}$), 3 for rotations ($\mathbf{N}_{r,2}$), and 1 for scale ($\mathbf{N}_{s,2}$).

C. Proof of Theorem 2

We employ the batch least squares (BLS) formulation for processing all IMU and camera measurements up to time-step k and form the state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{R_1}^T & \cdots & \mathbf{x}_{R_\ell}^T & \mathbf{x}_{R_{\ell+1}}^T & \cdots & \mathbf{x}_{R_k}^T & {}^G\mathbf{f}_1^T & \cdots & {}^G\mathbf{f}_M^T \end{bmatrix}^T \tag{38}$$

where $\mathbf{x}_{R_1},...,\mathbf{x}_{R_\ell}$ correspond to generic motions with $\ell \geq 4$, and $\mathbf{x}_{R_{\ell+1}},...,\mathbf{x}_{R_k}$ are of any motion (generic or hovering). Then, the unobservable directions of the linearized VINS model span the right nullspace of the information matrix of this BLS under marginalization [32], or equivalently, span the nullspace of the corresponding Jacobian matrix. The Jacobian $\mathbf{A}_k$ has the following sparse structure

$$\left[\begin{array}{ccccc|cc} \Phi_{2,1} & -\mathbf{I}_{15} & & & & & \\ & \Phi_{3,2} & -\mathbf{I}_{15} & & & & \\ & & \ddots & & & & \\ & & & \Phi_{k,k-1} & -\mathbf{I}_{15} & & \\ \hline \mathbf{H}_{x,1}^1 & & & & & \mathbf{H}_{f,1}^1 & \\ \vdots & & & & & & \ddots \\ \mathbf{H}_{x,1}^M & & & & & & \mathbf{H}_{f,1}^M \\ & \ddots & & & & \vdots & \\ & & & \mathbf{H}_{x,k}^1 & & \mathbf{H}_{f,k}^1 & \\ & & & \vdots & & & \ddots \\ & & & \mathbf{H}_{x,k}^M & & & \mathbf{H}_{f,k}^M \end{array}\right] \tag{39}$$

where $\Phi_{k,k-1}$ is the state transition matrix from time-step $k-1$ to k, and $\mathbf{H}_{x,k}^i$, $\mathbf{H}_{f,k}^i$ are the measurement Jacobians for the i-th feature observation at time-step k with respect to $\mathbf{x}_{R_k}$ and ${}^G\mathbf{f}_i$, respectively.

Now we use mathematical induction to show that the right nullspace of $\mathbf{A}_k$ is of dimension 4, for any $k \geq \ell$.

1) Initial step: When $k = \ell$, all states $\mathbf{x}_R$ correspond to generic motions. As shown in [17], [15], a VINS undergoing generic motions has only 4 unobservable directions: 3 for global translations and 1 for rotations about gravity.

2) Induction step: Assume that the nullspace $\mathbf{N}_{k-1}$ of $\mathbf{A}_{k-1}$ is of dimension 4. The Jacobian $\mathbf{A}_k$ takes the form:

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{A}_{k-1} & \mathbf{0} \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \tag{40}$$

where $\mathbf{B}$, $\mathbf{C}$ consist of different measurement Jacobian matrices. So to find the nullspace $\mathbf{N}_k$ of $\mathbf{A}_k$, we need to solve:

$$\mathbf{A}_k\mathbf{N}_k = \mathbf{0} \;\Leftrightarrow\; \mathbf{A}_{k-1}\mathbf{N}_k^1 = \mathbf{0} \;\text{ and }\; \mathbf{B}\mathbf{N}_k^1 + \mathbf{C}\mathbf{N}_k^2 = \mathbf{0} \tag{41}$$

where $\mathbf{N}_k = \begin{bmatrix} \mathbf{N}_k^{1\,T} & \mathbf{N}_k^{2\,T} \end{bmatrix}^T$. From (41), we have $\mathbf{N}_k^1 = \mathbf{N}_{k-1}$, and it can be shown that $\mathbf{N}_k^2$ is uniquely determined for each solution of $\mathbf{N}_k^1$. Hence, there are a total number of 4 independent directions for $\mathbf{N}_k$, and it is easy to check that these directions are the same as those of the VINS when undergoing generic motions.

ACKNOWLEDGEMENTS

The authors would like to thank Ahmed Ahmed, Elliot Branson, and Luis Carlos Carrillo for their help developing the software and hardware infrastructure of the quadrotor used in our experiments.

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

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

[3] B. Williams, N. Hudson, B. Tweddle, R. Brockers, and L. Matthies, "Feature and pose constrained visual aided inertial navigation for computationally constrained aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 431–438.

[4] 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 environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, May 14–18, 2012, pp. 957–964.

[5] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 264–280, Apr. 2009.

[6] D. C. Brown, "A solution to the general problem of multiple station analytical stereotriangulation," Patrick Air Force Base, Florida, RCA-MTP Data Reduction Technical Report, Tech. Rep. 43, 1958.

[7] R. Szeliski and S. B. Kang, "Recovering 3d shape and motion from image streams using nonlinear least squares," in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, New York City, NY, Jun. 15–17, 1993, pp. 752–753.

[8] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2000, pp. 298–372.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. 6th IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, Nara, Japan, Nov. 13 - 16, 2007, pp. 225–234.

[10] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, Alaska, May 3–8, 2010, pp. 2657–2664.

[11] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 4531–4537.

[12] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *Proc. Intl. Conf. on Information Fusion, FUSION*, Singapore, Jul. 9–12, 2012, pp. 2154–2161.

[13] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[14] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous robot vehicles*, vol. 1, pp. 167–193, 1990.

[15] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Towards consistent vision-aided inertial navigation," in *Proc. 10th Int. Workshop Alg. Found. Robot.*, Cambridge, MA, Jun. 13–15, 2012.

[16] D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis, "On the consistency of vision-aided inertial navigation," in *Proc. Int. Symp. Exper. Robot.*, Quebec City, Canada, Jun. 17–21, 2012.

[17] A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 44–60, Feb. 2012.

[18] M. Li and A. I. Mourikis, "Vision-aided inertial navigation for resource-constrained systems," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, Oct. 7–12, 2012, pp. 1057–1063.

[19] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft mavs in complex environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 6–10, 2013, pp. 1750–1756.

[20] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep. 2005-002, Mar. 2005. [Online]. Available: http://www-users.cs.umn.edu/~trawny/Publications/Quaternions_3D.pdf

[21] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Observability-constrained vision-aided inertial navigation," University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep. 2012-001, Feb. 2012. [Online]. Available: http://www-users.cs.umn.edu/~dkottas/pdfs/vins_tr_winter_2012.pdf

[22] A. I. Mourikis and S. I. Roumeliotis, "A dual-layer estimator architecture for long-term localization," in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Anchorage, AK, Jun. 24–26, 2008, pp. 1–8.

[23] S. I. Roumeliotis and J. W. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, D.C., May 11–15, 2002, pp. 1788–1795.

[24] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2006. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calibdoc/

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

[26] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Analytical Mechanics Associates, Inc., 1970.

[27] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[28] L. Kneip, M. Chli, and R. Siegwart, "Robust real-time visual odometry with a single camera and an imu," in *Proc. British Machine Vision Conf.*, Dundee, UK, Aug. 29–Sep. 2, 2011, pp. 16.1–16.11.

[29] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Washington, DC, Jun. 27–Jul. 2, 1994, pp. 593–600.

[30] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. on Artificial Intelligence*, Vancouver, B.C., Canada, Aug. 24–28, 1981, pp. 674–679.

[31] P. S. Maybeck, *Stochastic models, estimation, and control*. New York, NY: Academic Press, 1979, vol. I.

[32] G. Huang, "Improving the consistency of nonlinear estimators: Analysis, algorithms, and applications," Ph.D. dissertation, University of Minnesota, 2012.