

Efficient and Consistent Vision-aided Inertial Navigation using Line Observations

Dimitrios G. Kottas and Stergios I. Roumeliotis[†]

Abstract—This paper addresses the problem of estimating the state of a vehicle moving in 3D based on inertial measurements and visual observations of lines. In particular, we investigate the observability properties of the corresponding vision-aided inertial navigation system (VINS) and prove that it has five (four) unobservable degrees of freedom when one (two or more) line(s) is (are) detected. Additionally, we leverage this result to improve the *consistency* of the extended Kalman filter (EKF) estimator introduced for efficiently processing line observations over a sliding time-window at cost only *linear* in the number of line features. Finally, we validate the proposed algorithm experimentally using a miniature-size camera and a micro-electromechanical systems (MEMS)-quality inertial measurement unit (IMU).

I. INTRODUCTION AND RELATED WORK

The miniaturization, reduced cost, and increased accuracy of cameras and inertial measurement units (IMUs) makes them ideal sensors for determining the 3D position and attitude of vehicles (e.g., automotive [1], aerial [2], spacecraft [3], etc.) navigating in GPS-denied areas. In particular, fast and highly dynamic motions can be precisely estimated over short periods of time by fusing rotational velocity and linear acceleration measurements provided by the IMU’s gyroscopes and accelerometers, respectively. On the other hand, errors caused due to the integration of the bias and noise in the inertial measurements can be significantly reduced by processing observations to point features detected in camera images in what is known as a vision-aided inertial navigation system (V-INS). Recent advances in VINS, have addressed several issues, such as studying its observability [4], [5], reducing its computational requirements [1], [6], dealing with delayed and faulty observations [7], [8], increasing the accuracy and robustness of feature initialization [4], [9], and improving the estimator’s consistency [10], [11], [12].

Despite the significant progress in studying and fielding VINS, most approaches have focused on processing visual observations of *point features*. Although points are the simplest form of geometric primitives and can be found in any environment, tracking them can be especially challenging when considering large changes in the viewing direction and/or the lighting conditions. In contrast, edges and in particular *straight lines*, which are omnipresent in structured environments (e.g., indoors, urban areas, construction sites, etc.), can be reliably extracted and tracked under a wide

range of conditions [13], [14]. Additionally, robust edge descriptors [15] have been developed for gradient edges corresponding to the occluding boundaries of a scene (e.g., wall corners, stairwell edges, etc.), areas where point-tracking methods often fail to provide reliable matches.

Furthermore, the problem of motion estimation based on line observations¹ has been well-studied [16], [20]. In particular, given observations of 13 lines across three views, the motion of the camera, up to scale, can be determined in closed form [21], [22], while the impact of noise can be reduced by processing line observations in batch [23], [24] or filter form [25], [14]. Resolving the scale ambiguity, however, and dealing with highly dynamic motions requires fusing line observations with inertial measurements.

Employing measurements to line features for improving the accuracy of VINS has received limited attention to date. In one of the earliest works [26], it was shown that all 6 d.o.f. of a bias-free VINS are observable when measuring lines of *known* position and direction, and a Luenberger observer was proposed for fusing them. Later works, *partially* overcome the restricting assumption of an *a priori* known map of lines and consider measurements to lines of known direction. Specifically, in [27] measurements to a line of known direction are used to extract a vanishing point for determining the attitude of a static, bias-free, IMU-camera pair. Recently, observations to lines of *known direction* (parallel to the gravity) were proposed for improving the roll and pitch estimates, in point feature-based VINS [6].

To the best of our knowledge, the problem of vision-aided inertial navigation using observations, over multiple time steps, of previously *unknown* 3D lines, has not been investigated despite the potential gains in estimation accuracy and robustness.

The work described in this paper, addresses this limitation through the following three main contributions:

- We study the observability of a VINS that observes a line and prove that it has five unobservable degrees of freedom (dof): one corresponding to rotations about the gravity vector, three concerning the global translation of the IMU-camera and the line, and one corresponding to motions of the camera parallel to the line. Furthermore, we show that this last direction becomes observable when more than one non-parallel lines are detected.

[†]The authors are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, Emails: {dkottas|stergios} at cs.umn.edu. This work was supported by the University of Minnesota through the Digital Technology Center (DTC) and AFOSR MURI FA9550-10-1-0567

¹In this work, we make no assumptions about the direction or location of lines. Methods for computing the attitude and/or position of a camera using observations of *known* lines, are discussed in [16], [17], [18], [19] and references therein.

- We introduce an extended Kalman filter (EKF)-based algorithm whose consistency is improved by ensuring that no information is acquired along the unobservable directions of its linearized system model. Moreover, by concurrently processing line measurements across a sliding window of camera poses [i.e., by performing visual-inertial odometry (VIO) instead of simultaneous localization and mapping (SLAM)], the proposed estimator's computational complexity is only *linear* (instead of quadratic) in the number of line features processed.
- We confirm the key findings of the observability analysis and demonstrate the performance gains of the proposed VIO algorithm experimentally.

The remainder of this paper is structured as follows. In Sect. II, we present the system and measurement model based on inertial and line measurements. In Sect. III, we study the observability properties of VINS based on line observations. The key findings of this analysis are leveraged in Sect. IV to improve the consistency of the EKF-based estimator. Sect. V, presents experiments that confirm the observability analysis and demonstrate the performance improvement when using lines within VIO. Finally, Sect. VI summarizes the presented work and provides an outline of future research directions.

II. VINS STATE AND MEASUREMENT MODELS

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 processing straight-line observations (Sect. II-B).

A. IMU State and Covariance Propagation Model

The 16×1 IMU state vector is:

$$\mathbf{x}_R = [{}^l\hat{q}_G^T \quad \mathbf{b}_g^T \quad {}^G\mathbf{v}_I^T \quad \mathbf{b}_a^T \quad {}^G\mathbf{p}_I^T]^T \quad (1)$$

The first component of the IMU state is ${}^l\hat{q}_G(t)$ 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 [28]):

$$\begin{aligned} {}^l\dot{\hat{q}}_G(t) &= \frac{1}{2}\Omega({}^l\hat{\omega}(t)){}^l\hat{q}_G(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \\ \dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \end{aligned} \quad (2)$$

where ${}^l\hat{\omega}$ 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

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

The gyroscope and accelerometer measurements are:

$$\omega_m(t) = {}^l\hat{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \quad (3)$$

$$\mathbf{a}_m(t) = \mathbf{C}({}^l\hat{q}_G(t))({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \quad (4)$$

where $\mathbf{C}(\hat{q})$ is the rotation matrix corresponding to the quaternion \hat{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:

$${}^l\dot{\hat{q}}_G(t) = \frac{1}{2}\Omega({}^l\hat{\omega}(t)){}^l\hat{q}_G(t), \quad {}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t) \quad (5)$$

$${}^G\hat{\mathbf{v}}_I(t) = \mathbf{C}^T({}^l\hat{q}_G(t))\hat{\mathbf{a}}(t) + {}^G\mathbf{g}, \quad \dot{\hat{\mathbf{b}}}_g(t) = \mathbf{0}_{3 \times 1}, \quad \dot{\hat{\mathbf{b}}}_a(t) = \mathbf{0}_{3 \times 1}$$

where $\hat{\mathbf{a}}(t) \triangleq \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$, and ${}^l\hat{\omega}(t) \triangleq \omega_m(t) - \hat{\mathbf{b}}_g(t)$.

By defining the 15×1 error-state vector as:²

$$\tilde{\mathbf{x}}_R = [{}^l\delta\theta_G^T \quad \tilde{\mathbf{b}}_g^T \quad {}^G\tilde{\mathbf{v}}_I^T \quad \tilde{\mathbf{b}}_a^T \quad {}^G\tilde{\mathbf{p}}_I^T]^T, \quad (6)$$

the continuous-time IMU error-state equation becomes:

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

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

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

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

and $\mathbf{n}(t) \triangleq [\mathbf{n}_g^T \quad \mathbf{n}_{wg}^T \quad \mathbf{n}_a^T \quad \mathbf{n}_{wa}^T]^T$ is the system noise with autocorrelation $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \mathbf{Q}_R\delta(t-\tau)$, where $\delta(\cdot)$ 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 [29] as the solution to the matrix differential equation $\dot{\Phi}_{k,1} = \mathbf{F}_R(t_k)\Phi_{k,1}$, $\Phi_{1,1} = \mathbf{I}_{15}$:

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

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

B. Measurement Model for Straight Lines

1) *Minimal (4 dof) Representation of Straight Lines in 3D*: Consider the line \mathbf{L} in Fig. 1 and the coordinate frame

²For 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 \hat{q} and its estimate $\hat{\hat{q}}$ is the 3×1 angle-error vector, $\delta\theta$, implicitly defined by the *error quaternion* $\delta\hat{q} = \hat{q} \otimes \hat{\hat{q}}^{-1} \simeq [\frac{1}{2}\delta\theta^T \quad 1]^T$, where $\delta\hat{q}$ describes the small rotation that causes the true and estimated attitude to coincide.

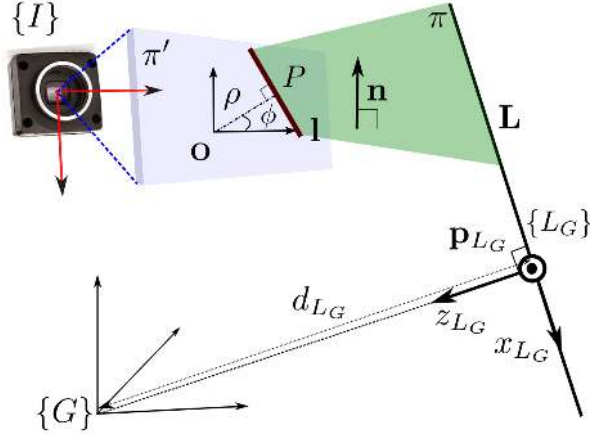


Fig. 1: Geometric depiction of the line parameterization (${}^G\bar{q}_{LG}$, d_{LG}) employed in our analysis.

$\{L_G\}$ whose origin \mathbf{p}_{LG} is the point on the line at minimum distance, d_{LG} , from $\{G\}$, its x -axis is aligned with the line's direction, \mathbf{L} , and its z -axis points to the origin of $\{G\}$. Then, the line \mathbf{L} with respect to $\{G\}$ can be represented by the parameter vector:

$$\mathbf{x}_L = [{}^G\bar{q}_{LG}^T \quad d_{LG}]^T \quad (9)$$

while its corresponding error vector is:

$$\tilde{\mathbf{x}}_L = [\delta\theta_{LG}^T \quad \tilde{d}_{LG}]^T. \quad (10)$$

For simplicity, we consider the IMU frame of reference $\{I\}$ to coincide with the camera frame of reference³ and define $\mathbf{C}_L = \mathbf{C}({}^G\bar{q}_{LG})$ and $d_L = d_{LG}$. The optical center of the camera $\{I\}$, together with the 3D line \mathbf{L} , define a plane π in space. The image sensor observes the 2D line \mathbf{I} , i.e., the intersection of π with the image plane π' . The line detection algorithm, returns line \mathbf{I} parameterized in polar form by (ϕ, ρ) , which represent the orientation and magnitude of the line's normal vector OP in the 2D image plane (see Fig. 1). A point \mathbf{p} with homogeneous image coordinates $\mathbf{p}^T = [u \quad v \quad 1]$, lies on the line \mathbf{I} if it satisfies the equality:

$$\mathbf{p}^T [\cos\phi \quad \sin\phi \quad -\rho] = 0. \quad (11)$$

Let O denote the principal point of the image plane, I the optical center of the camera, and $\mathbf{u} = [\sin\phi \quad -\cos\phi \quad 0]^T$ be a (free) unit vector along the line on the image plane. From Fig. 1, the vectors \mathbf{u} and $IP = IO + OP = [\rho\cos\phi \quad \rho\sin\phi \quad 1]^T$ define the plane π . The vector \mathbf{n} perpendicular to the plane π , is:

$$\mathbf{n} = IP \times \mathbf{u} = [\cos\phi \quad \sin\phi \quad -\rho]^T. \quad (12)$$

2) *Geometric Constraints*: We now derive two geometric constraints relating the measurements of the lines on the image plane with the robot's attitude and position, in the absence of noise. At time step t_k , the sensor's frame of reference $\{I\}$ is parameterized by ${}^{I_k}\bar{q}_G$ and ${}^G\mathbf{p}_{I_k}$, and it

observes the line \mathbf{L} , through its normal vector $\mathbf{n}_k \triangleq {}^{I_k}\mathbf{n}$. The direction of line \mathbf{L} expressed in frame $\{I\}$, lies on plane π , and hence satisfies the constraint:

$$\mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) \mathbf{C}_L \mathbf{e}_1 = 0 \quad (13)$$

where $\mathbf{e}_1 = [1 \quad 0 \quad 0]^T$. Similarly, for the point ${}^G\mathbf{p}_{LG} = -\mathbf{C}_L d_L [0 \quad 0 \quad 1]^T = -\mathbf{C}_L d_L \mathbf{e}_3$ expressed in the $\{I\}$, we have:

$$\begin{aligned} \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) ({}^G\mathbf{p}_{LG} - {}^G\mathbf{p}_{I_k}) &= 0 \implies \\ \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) (-\mathbf{C}_L d_L \mathbf{e}_3 - {}^G\mathbf{p}_{I_k}) &= 0. \end{aligned} \quad (14)$$

Stacking the two constraints, (13) and (14), we arrive at:

$$\mathbf{h}(\mathbf{n}_k, \mathbf{x}_{R_k}, \mathbf{x}_L)_{2 \times 1} = \begin{bmatrix} \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) \mathbf{C}_L \mathbf{e}_1 \\ \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) (-\mathbf{C}_L d_L \mathbf{e}_3 - {}^G\mathbf{p}_{I_k}) \end{bmatrix} = \mathbf{0}_{2 \times 1} \quad (15)$$

where \mathbf{x}_{R_k} is the vector \mathbf{x}_R at time step t_k . In the next section, we describe the measurement model induced by these geometric constraints in the presence of camera sensor noise.

3) *Measurement Model*: In practice, the camera measures

$$\mathbf{z}_k = [\phi \quad \rho]^T + \xi_k \quad (16)$$

where ξ_k follows a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}_{2 \times 1}, \mathbf{R}_{\phi\rho})$ and models the noise, induced by the camera sensor and the line extraction algorithm. The effect of ξ_k on $\mathbf{h}(\mathbf{n}_k, \mathbf{x}_{R_k}, \mathbf{x}_L)$, denoted by \mathbf{w}_k can be approximated through linearization as:

$$\mathbf{w}_k = \mathbf{A}_{2 \times 3} \mathbf{B}_{3 \times 2} \xi_k, \quad \mathbf{A} = \nabla_{\mathbf{n}_k} \mathbf{h}, \quad \mathbf{B} = [\nabla_{\phi} \mathbf{n}_k \quad \nabla_{\rho} \mathbf{n}_k].$$

Hence, \mathbf{w}_k can be approximated by a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}_{2 \times 1}, \mathbf{Z}_k)$ with $\mathbf{Z}_k = \mathbf{A} \mathbf{B} \mathbf{R}_{\phi\rho} \mathbf{B}^T \mathbf{A}^T$.

Using this noise parameterization, we arrive at the following measurement model, that couples the measurement of line \mathbf{L} at time step t_k , \mathbf{n}_k , with the robot's state vector, \mathbf{x}_{R_k} , and the line parameters \mathbf{x}_L :

$$\mathbf{z}_k = \mathbf{h}(\mathbf{n}_k, \mathbf{x}_{R_k}, \mathbf{x}_L) + \mathbf{w}_k. \quad (17)$$

We now, linearize (17), with respect to the error state $\tilde{\mathbf{x}}_{R_k}$ and the line parameters error $\tilde{\mathbf{x}}_L$, which yields:

$$\tilde{\mathbf{z}}_k = \nabla_{\mathbf{x}_{R_k}} \mathbf{h}|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_{R_k} + \nabla_{\mathbf{x}_L} \mathbf{h}|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_L + \mathbf{w}_k \quad (18)$$

$$= \mathbf{H}_{R_k}|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_{R_k} + \mathbf{H}_L|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_L + \mathbf{w}_k \quad (19)$$

with the corresponding Jacobians given by:

$$\begin{aligned} \mathbf{H}_{R_k}|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} &= \begin{bmatrix} \mathbf{n}_k^T [\mathbf{C}({}^{I_k}\bar{q}_G) \mathbf{C}_L \mathbf{e}_1 \times] & \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 3} \\ \mathbf{n}_k^T [\mathbf{C}({}^{I_k}\bar{q}_G) (-\mathbf{C}_L d_L \mathbf{e}_3 - {}^G\mathbf{p}_{I_k}) \times] & \mathbf{0}_{1 \times 9} & -\mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) \end{bmatrix} \\ \mathbf{H}_L|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} &= \begin{bmatrix} \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) [\mathbf{C}_L \mathbf{e}_1 \times] & 0 \\ \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) [-\mathbf{C}_L d_L \mathbf{e}_3 \times] & \mathbf{n}_k^T \mathbf{C}({}^{I_k}\bar{q}_G) (-\mathbf{C}_L \mathbf{e}_3) \end{bmatrix}. \end{aligned}$$

Which, can be written in a compact form as:

$$\begin{aligned} \mathbf{H}_{R_k}|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} &= \begin{bmatrix} \mathbf{H}_{q1k} & \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 3} \\ \mathbf{H}_{q2k} & \mathbf{0}_{1 \times 9} & \mathbf{H}_{p2k} \end{bmatrix} \\ \mathbf{H}_L|_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} &= \begin{bmatrix} \mathbf{H}_{l1k} & 0 \\ \mathbf{H}_{l2k} & \mathbf{H}_{d2k} \end{bmatrix} \end{aligned} \quad (20)$$

where $\mathbf{x}_{R_k}^*$ and \mathbf{x}_L^* denote the estimates at which the Jacobians are computed.

³In practice, we perform IMU-camera extrinsic calibration following the approach of [30].

III. OBSERVABILITY ANALYSIS

In this section, we study the observability properties of a VINS, that measures a single line \mathbf{L} over m time steps, denoted by t_1, \dots, t_m . The system state consists of the vector \mathbf{x}_R , that includes the IMU pose and linear velocity together with the time-varying IMU biases (see Sec. II-A), as well as the vector \mathbf{x}_L that describes the line parameters (see Sec. II-B.1) with respect to frame $\{G\}$. The time evolution of the linearized system error state between time steps t_k and t_{k+1} is described by:

$$\begin{bmatrix} \tilde{\mathbf{x}}_{R_{k+1}} \\ \tilde{\mathbf{x}}_L \end{bmatrix} = \begin{bmatrix} \Phi_{k+1,k} |_{\mathbf{x}_{R_{k+1}}^*, \mathbf{x}_{R_k}^*} & \mathbf{0}_{15 \times 4} \\ \mathbf{0}_{4 \times 15} & \mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{R_k} \\ \tilde{\mathbf{x}}_L \end{bmatrix} \quad (21)$$

where $\Phi_{k+1,k} |_{\mathbf{x}_{R_{k+1}}^*, \mathbf{x}_{R_k}^*}$ is the system Jacobian described in Sec. II-A, evaluated at the points $\mathbf{x}_{R_k}^*$, $\mathbf{x}_{R_{k+1}}^*$. Note that the line coordinates' error does not change in time since we observe a static scene. From eqs. (18)-(20), the linearized measurement model is:

$$\tilde{z}_k = \mathbf{H}_{R_k} |_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_{R_k} + \mathbf{H}_L |_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \tilde{\mathbf{x}}_L. \quad (22)$$

In the study of the system's observability properties, we set $\mathbf{n}(t) = 0$ and $\mathbf{w}_k = 0$ in (7) and (18), respectively. Therefore, (21) and (22) represent the system's error dynamics in the absence of noise. The k -th block row of the observability matrix \mathbf{M} , defined over a time period t_1, \dots, t_k , is given by:

$$\mathbf{M}^{(k,:)}(\mathbf{x}_{R_k}^*, \mathbf{x}_L^*) = \begin{bmatrix} \mathbf{H}_{R_k} |_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \Phi_{k,1} |_{\mathbf{x}_{R_k}^*, \mathbf{x}_{R_1}^*} & \mathbf{H}_L |_{\mathbf{x}_{R_k}^*, \mathbf{x}_L^*} \end{bmatrix}.$$

Any vector belonging to the right nullspace of \mathbf{M} , does not affect our measurements and hence it corresponds to an unobservable direction for any consistent estimator.

A. True Linearized System

Hereafter, we investigate the directions that span the right nullspace of the observability matrix \mathbf{M} under an "ideal" linearization around the true state (i.e., $\mathbf{x}_{R_k}^* = \bar{\mathbf{x}}_{R_k}$, $\mathbf{x}_L^* = \bar{\mathbf{x}}_L$, $\mathbf{n}_k = \bar{\mathbf{n}}_k$), so as to determine the analytical form of the system's unobservable directions. For simplicity, let us evaluate four rows of \mathbf{M} , corresponding to two measurements, at time steps t_1 and t_k , respectively. The two rows, of the observability matrix, for t_1 are:

$$\mathbf{M}(\bar{\mathbf{x}}_{R_1}, \bar{\mathbf{x}}_L)^{(1,:)} = \begin{bmatrix} \mathbf{H}_{R_1} |_{\bar{\mathbf{x}}_{R_1}, \bar{\mathbf{x}}_L} & \mathbf{H}_L |_{\bar{\mathbf{x}}_{R_1}, \bar{\mathbf{x}}_L} \end{bmatrix} \quad (23)$$

while for t_k , are given by:

$$\mathbf{M}(\bar{\mathbf{x}}_{R_k}, \bar{\mathbf{x}}_L)^{(k,:)} = \begin{bmatrix} \mathbf{M}^{(k,1)} & \mathbf{M}^{(k,2)} & \mathbf{M}^{(k,3)} & \mathbf{M}^{(k,4)} & \mathbf{M}^{(k,5)} & \mathbf{M}^{(k,6;7)} \end{bmatrix}$$

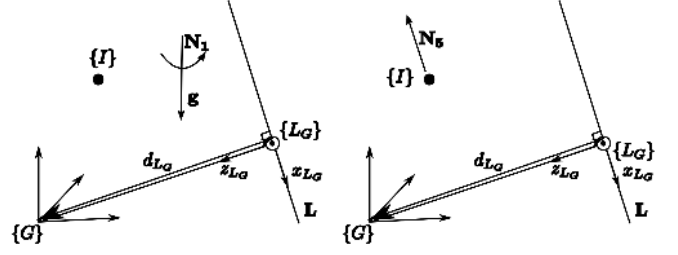


Fig. 3: Unobservable directions \mathbf{N}_1 and \mathbf{N}_5 , corresponding to a rotation around the gravity vector and changes of the sensor platform's velocity, along the direction of the line.

where:

$$\mathbf{M}^{(k,1)} = \begin{bmatrix} \bar{\mathbf{n}}_k^T [\mathbf{C}^{(k)} \bar{q}_G] \mathbf{C}_L \mathbf{e}_1 \times \Phi_{k,1}^{(1,1)} \\ \bar{\mathbf{n}}_k^T [\mathbf{C}^{(k)} \bar{q}_G] (-\mathbf{C}_L d_L \mathbf{e}_3 - {}^G \mathbf{p}_{1k}) \times \Phi_{k,1}^{(1,1)} - \bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G \Phi_{k,1}^{(5,1)} \end{bmatrix}$$

$$\Phi_{k,1}^{(1,1)} = \mathbf{C}^{(k)} \bar{q}_1, \quad \mathbf{M}^{(k,3)} = \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ -\bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G (t_k - t_1) \end{bmatrix}$$

$$\mathbf{M}^{(k,5)} = \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ -\bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G (t_k - t_1) \end{bmatrix}$$

$$\mathbf{M}^{(k,6;7)} = \begin{bmatrix} \bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G [\mathbf{C}_L \mathbf{e}_1 \times] & 0 \\ \bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G [-\mathbf{C}_L d_L \mathbf{e}_3 \times] & \bar{\mathbf{n}}_k^T \mathbf{C}^{(k)} \bar{q}_G (-\mathbf{C}_L \mathbf{e}_3) \end{bmatrix}$$

$$\Phi_{k,1}^{(5,1)} = [{}^G \mathbf{p}_{1k} + {}^G \mathbf{v}_{1k} (t_k - t_1) - \frac{1}{2} {}^G \mathbf{g} (t_k - t_1)^2 - {}^G \mathbf{p}_{1k} \times] \mathbf{C}^{(k)} \bar{q}_1.$$

For brevity, we omit the expressions for the time-varying block matrices $\mathbf{M}^{(k,2)}$, $\mathbf{M}^{(k,4)}$, since they don't affect the following analysis.

It can be verified, that *at least* the following *five* directions lie in the unobservable subspace of the system, meaning that $\text{rank}(\mathbf{M}) \leq 14$ (since the dimension of \mathbf{M} is $2m \times 19$).

$$\begin{aligned} \mathbf{N}_1 &= \begin{bmatrix} \mathbf{C}^{(1)} \bar{q}_G \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -[{}^G \mathbf{v}_{11} \times] \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -[{}^G \mathbf{p}_{11} \times] \mathbf{g} \\ -\mathbf{g} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{1q} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{N}_{1v} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{N}_{1p} \\ \mathbf{N}_{11} \\ 0 \end{bmatrix} \\ \mathbf{N}_2 &= \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_L \mathbf{e}_1 \\ \mathbf{0}_{3 \times 1} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{N}_{2p} \\ \mathbf{0}_{3 \times 1} \\ 0 \end{bmatrix}, \quad \mathbf{N}_3 = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ -\mathbf{C}_L \mathbf{e}_3 \\ \mathbf{0}_{3 \times 1} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{N}_{3p} \\ \mathbf{0}_{3 \times 1} \\ 1 \end{bmatrix} \\ \mathbf{N}_4 &= \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_L d_L \mathbf{e}_2 \\ -\mathbf{C}_L \mathbf{e}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{N}_{4p} \\ \mathbf{N}_{41} \\ 0 \end{bmatrix}, \quad \mathbf{N}_5 = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{C}_L \mathbf{e}_1 \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ 0 \end{bmatrix}. \end{aligned} \quad (24)$$

Direction \mathbf{N}_1 corresponds to the rotation of the sensor platform and the line around the gravity vector. $\mathbf{N}_{2..4}$ span

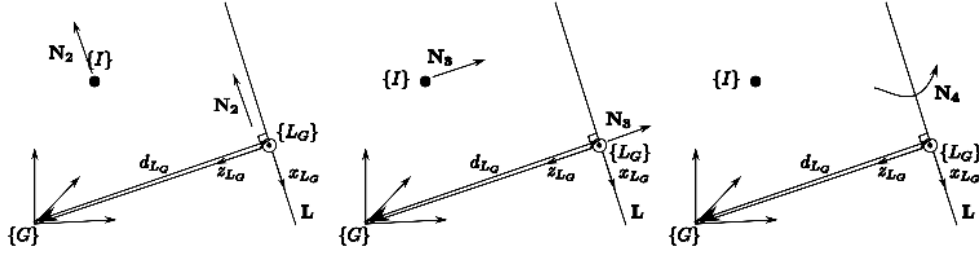


Fig. 2: Unobservable directions \mathbf{N}_2 , \mathbf{N}_3 , and \mathbf{N}_4 . The combinations of these directions represent any translation of the sensor platform together with the line.

the space of all possible translations of the sensor platform together with the line. The fifth direction \mathbf{N}_5 corresponds to a change of the sensors' velocity, in the direction of line \mathbf{L} (see Figs. 2 and 3).

Consider now, the joint observability matrix for the case of measurements of two non-parallel lines \mathbf{L}_1 , \mathbf{L}_2 , which is $\mathbf{M}' = \begin{bmatrix} \mathbf{M}_{\mathbf{L}_1} \\ \mathbf{M}_{\mathbf{L}_2} \end{bmatrix}$. As we prove in [31], neither $\mathbf{N}_{5\mathbf{L}_1}$ nor $\mathbf{N}_{5\mathbf{L}_2}$ lies in the nullspace of \mathbf{M}' , since \mathbf{L}_1 and \mathbf{L}_2 are non-parallel.

B. Linearized System in Practice

We now examine the observability matrix corresponding to the linearized system, when the linearization points are not ideal (i.e., they do not correspond to the true quantities $\bar{\mathbf{n}}_k$, $\bar{\mathbf{x}}_{\mathbf{R}_k}$, $\bar{\mathbf{x}}_{\mathbf{L}}$). Interestingly, when we linearize around the current state estimate, the directions \mathbf{N}_1 , \mathbf{N}_2 , and \mathbf{N}_5 erroneously become observable. This is easy to verify, for example, for \mathbf{N}_2 . In the absence of noise, and with linearization performed around the true states, the vector $\mathbf{C}_{\mathbf{L}}\mathbf{e}_1$ is always perpendicular to $\mathbf{C}^{(k}\hat{q}_G)^T\bar{\mathbf{n}}_k$, hence it always lies in the right nullspace of the ideal observability matrix \mathbf{M} . In practice, however, no vector $\mathbf{C}_{\mathbf{L}}\mathbf{e}_1$ exists that is perpendicular to every element of the set of vectors $\{\mathbf{C}^{(k}\hat{q}_G)^T\bar{\mathbf{n}}_k\}$, due to (i) the errors in the different estimates of ${}^k\hat{q}_G$ at different time steps, and the fact that (ii) we do not measure the true $\bar{\mathbf{n}}_k$, but a perturbed version of it (see (16)). Moreover, if for two different time steps, corresponding to two different block rows of the observability matrix \mathbf{M} , we linearize around different estimates of the line parameters $\hat{\mathbf{x}}_{\mathbf{L}}$, the directions \mathbf{N}_3 , \mathbf{N}_4 also become observable. This leads to the conclusion that any filter, applied to this problem, which employs linearization around the current state estimate, violates the observability properties of the underlying system and results in injection of spurious information. We conjecture that this causes the EKF estimator to become inconsistent (i.e., being over-confident for the accuracy of its estimates) and propose a formal, yet simple, approach for addressing this issue in the following section.

IV. APPLICATION TO VISION-AIDED INERTIAL ODOMETRY: DESCRIPTION OF THE ALGORITHM

We employ the results of the observability analysis to improve the consistency of the MSC-KF algorithm [1] when modified for processing line observations. The main

advantage of the MSC-KF is that it processes all geometric constraints induced by camera measurements over a finite window of image frames, with linear computational complexity in the number of observed features. This is accomplished by not including a map of the environment in the state vector, but rather using all provided information for localization purposes.

A. State Vector

At a given time step k , the filter tracks the 16×1 evolving state, $\hat{\mathbf{x}}_{\mathbf{R}}$ (see (1)). In order to process measurements over a time window of the last M images, we employ stochastic cloning [32] and keep in our state the cloned sensor poses $\{\hat{\mathbf{x}}_{\mathbf{C}} = [{}^{k-M+i}\bar{q}_G^T \quad {}^G\mathbf{p}_{l_{k-M+i}}^T]^T\}$, $i = 0 \dots (M-1)$. Correspondingly, the covariance consists of the 15×15 block of the evolving state, $\mathbf{P}_{\mathbf{RR}}$, the $6M \times 6M$ block corresponding to the cloned robot poses, $\mathbf{P}_{\mathbf{CC}}$, and their cross-correlations, $\mathbf{P}_{\mathbf{RC}}$. Hence, the covariance of the augmented state vector has the following structure:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathbf{RR}} & \mathbf{P}_{\mathbf{RC}} \\ \mathbf{P}_{\mathbf{RC}}^T & \mathbf{P}_{\mathbf{CC}} \end{bmatrix} \quad (25)$$

Moreover, in order to enforce the correct observability properties to the linearized system, we maintain a copy of the evolving part of the nullspace directions, which corresponds to rotations around gravity, \mathbf{N}_1 . Hence, we construct the set $\mathbf{S}_{\mathbf{N}} = \{\mathbf{N}_{\mathbf{I}(k-M+i)}\}_{i=0 \dots (M-1)}$, as part of the state cloning and propagation procedure.

B. State Cloning

Upon the acquisition of a new image, we clone the portions of the current state estimate and covariance, corresponding to the robot pose:

$$\mathbf{x}_{\mathbf{C}} \leftarrow \begin{bmatrix} \mathbf{x}_{\mathbf{C}} \\ {}^l\bar{q}_G \\ {}^G\mathbf{p}_l \end{bmatrix}, \quad \mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}(:, [1:3, 13:15]) \\ \mathbf{P}(:, [1:3, 13:15])^T & \mathbf{P}([1:3, 13:15], [1:3, 13:15]) \end{bmatrix}$$

and compute the nullspace direction \mathbf{N}_1 , evaluated at the current state estimate, (i.e., $\mathbf{N}_{\mathbf{I}(k-M)}$ for $t = t_{k-M}$) using eq. (24), and add it to the set $\mathbf{S}_{\mathbf{N}}$:

$$\mathbf{S}_{\mathbf{N}} \leftarrow \{\mathbf{S}_{\mathbf{N}}, \mathbf{N}_{\mathbf{I}(k-M)}\}. \quad (26)$$

C. State, Covariance and Nullspace Propagation

Between time steps t_{k-M} and t_{k-M+1} , we propagate the evolving state estimate to $\hat{\mathbf{x}}_{\mathbf{R}_{k-M+1}}$ (see Sec. II-A), using the IMU measurements. We evaluate now the new nullspace direction, $\mathbf{N}_{\mathbf{I}(k-M+1)}$, by substituting the new state estimate $\hat{\mathbf{x}}_{\mathbf{R}_{k-M+1}}$ in (24). Using the analytical expressions [29] for the state transition matrix, we evaluate $\hat{\Phi}_{k-M+1,k-M}$. We now seek a modified $\Phi_{k-M+1,k-M}^*$, that adheres to the correct observability properties of the system [11]. Since, only the first 15 rows of $\mathbf{N}_{\mathbf{I}}$ evolve through time, we set $\mathbf{u} = \mathbf{N}_{\mathbf{I}}^{(1:15,:)}_{(k-M)}$, $\mathbf{w} = \mathbf{N}_{\mathbf{I}}^{(1:15,:)}_{(k-M+1)}$, $\mathbf{A}^* = \Phi_{k-M+1,k-M}^*$, $\mathbf{A} = \Phi_{k-M+1,k-M}$ and solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A}^*} \|\mathbf{A}^* - \mathbf{A}\|_{\mathcal{F}}^2 & \quad (27) \\ \text{s.t. } \mathbf{A}^* \mathbf{u} = \mathbf{w} & \end{aligned}$$

where $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius matrix norm and the optimal solution is $\mathbf{A}^* = \mathbf{A} - (\mathbf{A}\mathbf{u} - \mathbf{w})(\mathbf{u}^T \mathbf{u})^{-1} \mathbf{u}^T$. Finally, we employ the modified transition matrix, for covariance propagation.

$$\mathbf{P}_{\mathbf{RR}} \leftarrow \Phi_{k-M+1,k-M}^* \mathbf{P}_{\mathbf{RR}} \Phi_{k-M+1,k-M}^{*T} + \mathbf{Q}_{\mathbf{k}} \quad (28)$$

$$\mathbf{P}_{\mathbf{RC}} \leftarrow \Phi_{k-M+1,k-M}^* \mathbf{P}_{\mathbf{RC}} \quad (29)$$

D. Processing Line Measurements

We now describe the update step for processing N tracks of straight lines that initiate from image $k-M+1$ and reach at most, image k . From (18), the linearized measurement model for line j , $j = 1 \dots N$, acquired at time step i , $i = (k-M+1) \dots k$, is:

$$\tilde{\mathbf{z}}_i^j = \mathbf{H}_{\mathbf{C}_i}^j \tilde{\mathbf{x}}_{\mathbf{C}_i} + \mathbf{H}_{\mathbf{L}_i}^j \tilde{\mathbf{x}}_{\mathbf{L}_j} + \mathbf{w}_i^j \quad (30)$$

since, our measurements relate only to the cloned robot poses, $\mathbf{H}_{\mathbf{C}_i}^j = \begin{bmatrix} \mathbf{H}_{\mathbf{R}_i}^{j([1:2],[1:3])} & \mathbf{H}_{\mathbf{R}_i}^{j([1:2],[13:15])} \end{bmatrix}$ (see (20)). For the evaluation of the Jacobians we need the parameters of the line \mathbf{L}_j , an estimate of which we compute through triangulation. So as to reduce the computational complexity, we approximate the covariance matrices of all \mathbf{z}_i^j , that we are about to process by $\max\{\lambda_{\max}(\{\mathbf{Z}_i^j\}_{i=(k-M+1)\dots k, j=1\dots N})\} \mathbf{I}_2 = \sigma^2 \mathbf{I}_2$. The process of retrieving σ^2 has complexity at most $O(NM)$.

1) Observability Constrained Measurement Jacobians:

The measurement Jacobians that adhere to the correct observability properties should satisfy:

$$\begin{bmatrix} \mathbf{H}_{\mathbf{C}_i}^j & \mathbf{H}_{\mathbf{L}_i}^j \end{bmatrix} \Theta \mathbf{N}_{\mathbf{I}i} = 0, \quad \begin{bmatrix} \mathbf{H}_{\mathbf{C}_i}^j & \mathbf{H}_{\mathbf{L}_i}^j \end{bmatrix} \Theta \mathbf{N}_{\mathbf{I}2\dots 5}^j = 0 \quad (31)$$

where $\Theta = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 7} \\ \mathbf{0}_{7 \times 3} & \mathbf{0}_{7 \times 9} & \mathbf{I}_7 \end{bmatrix}$, $\mathbf{N}_{\mathbf{I}i}$ is the i -th element of the set $\mathbf{S}_{\mathbf{N}}$, and by $\mathbf{N}_{\mathbf{I}2\dots 5}^j$ we denote any of the directions $\mathbf{N}_2 \dots \mathbf{N}_5$ evaluated at the parameters of line j . To acquire the modified $\mathbf{H}_{\mathbf{C}_i}^j$ and $\mathbf{H}_{\mathbf{L}_i}^j$, we re-arrange (31), to bring it in the form of the optimization problem (27), and arrive at the

following expressions (see also (20)):

$$\begin{aligned} \mathbf{H}_{\mathbf{p}2i}^{j*} &= \mathbf{H}_{\mathbf{p}2i}^j - \mathbf{H}_{\mathbf{p}2i}^{j*} \mathbf{N}_{\mathbf{2p}}^j (\mathbf{N}_{\mathbf{2p}}^{jT} \mathbf{N}_{\mathbf{2p}}^j)^{-1} \mathbf{N}_{\mathbf{2p}}^{jT} \\ \mathbf{H}_{\mathbf{d}2i}^{j*} &= -\mathbf{H}_{\mathbf{p}2i}^{j*} \mathbf{N}_{\mathbf{3p}}^j \\ \mathbf{H}_{\mathbf{l}1i}^{j*} &= \mathbf{H}_{\mathbf{l}1i}^j - \mathbf{H}_{\mathbf{l}1i}^j \mathbf{N}_{\mathbf{4l}}^j (\mathbf{N}_{\mathbf{4l}}^{jT} \mathbf{N}_{\mathbf{4l}}^j)^{-1} \mathbf{N}_{\mathbf{4l}}^{jT} \\ \mathbf{H}_{\mathbf{l}2i}^{j*} &= \mathbf{H}_{\mathbf{l}2i}^j - (\mathbf{H}_{\mathbf{l}2i}^j \mathbf{N}_{\mathbf{4l}}^j + \mathbf{H}_{\mathbf{p}2i}^j \mathbf{N}_{\mathbf{4p}}^j) (\mathbf{N}_{\mathbf{4l}}^{jT} \mathbf{N}_{\mathbf{4l}}^j)^{-1} \mathbf{N}_{\mathbf{4l}}^{jT} \\ \mathbf{H}_{\mathbf{q}1i}^{j*} &= \mathbf{H}_{\mathbf{q}1i}^j - (\mathbf{H}_{\mathbf{q}1i}^j \mathbf{N}_{\mathbf{1qi}} + \mathbf{H}_{\mathbf{l}1i}^{j*} \mathbf{N}_{\mathbf{l}1i}) (\mathbf{N}_{\mathbf{1qi}}^T \mathbf{N}_{\mathbf{1qi}})^{-1} \mathbf{N}_{\mathbf{1qi}}^T \\ \mathbf{H}_{\mathbf{q}2i}^{j*} &= \mathbf{H}_{\mathbf{q}2i}^j - (\mathbf{H}_{\mathbf{q}2i}^j \alpha - \beta) (\alpha^T \alpha)^{-1} \alpha^T \\ \alpha &= \mathbf{N}_{\mathbf{1qi}}, \quad \beta = -\mathbf{H}_{\mathbf{p}2i}^{j*} \mathbf{N}_{\mathbf{1pi}} + \mathbf{H}_{\mathbf{l}2i}^{j*} \mathbf{g} \end{aligned}$$

2) *Linearized Constraint among all Camera Poses:* By stacking the measurements of line j over the time window $i = (k-M+1) \dots k$, we arrive at:

$$\tilde{\mathbf{z}}^j = \mathbf{H}_{\mathbf{C}}^{j*} \tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{H}_{\mathbf{L}}^{j*} \tilde{\mathbf{x}}_{\mathbf{L}_j} + \mathbf{w}^j \quad (32)$$

The matrix $\mathbf{H}_{\mathbf{L}}^{j*}$, has dimensions $2M \times 4$, and for $M \geq 3$ it is full column rank. Hence, its left nullspace \mathbf{U}_j , is of dimensions $2M-4$. By premultiplying (32) by \mathbf{U}_j^T , we arrive at a measurement model, independent of the line parameters' error:

$$\mathbf{U}_j^T \tilde{\mathbf{z}}^j = \mathbf{U}_j^T \mathbf{H}_{\mathbf{C}}^{j*} \tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{U}_j^T \mathbf{w}^j \implies \tilde{\mathbf{z}}'^j = \mathbf{H}_{\mathbf{C}}'^j \tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}'^j. \quad (33)$$

This key step in MSC-KF [1], defines a linearized constraint, *independent of the feature parameters*, among all the camera poses, from which the line j was observed. By employing this methodology on the line-based VINS framework, we exploit all the geometric information induced by a line, without the requirement of augmenting our state vector, with its parameters. Furthermore the computation of $\tilde{\mathbf{z}}'^j$, of dimensions $2M-4 \times 1$, and $\mathbf{H}_{\mathbf{C}}'^j$ can be performed efficiently using Givens rotations. Notice also, that the resulting noise term $\mathbf{U}_j^T \mathbf{w}^j$ has covariance $\sigma^2 \mathbf{I}_{2M-4}$.

3) *Linear Complexity EKF Update:* By collecting all $\tilde{\mathbf{z}}'^j$, over all observed lines, $j = 1 \dots N$, we acquire:

$$\tilde{\mathbf{z}}'_{N(2M-4) \times 1} = \mathbf{H}'_{\mathbf{C}} \tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}' \quad (34)$$

where the matrix $\mathbf{H}'_{\mathbf{C}}$ has dimensions $N(2M-4) \times 6M$ and is, in general, tall, since the number of line observations typically exceeds the number of robot-pose states. As it is described in [1] and [3] for the case of point features, we factorize $\mathbf{H}'_{\mathbf{C}}$ as:

$$\mathbf{H}'_{\mathbf{C}} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_{\text{upper}} \\ \mathbf{0} \end{bmatrix} \quad (35)$$

Where $\mathbf{R}_{\text{upper}}$ can have at most $6(M-1) - 1$ non-zero rows. By performing Givens rotations on $\tilde{\mathbf{z}}'_{N(2M-4) \times 1}$ and $\mathbf{H}'_{\mathbf{C}}$ (see [33], pg. 227), we form, our final measurement model:

$$\tilde{\mathbf{r}} = \mathbf{R}_{\text{upper}} \tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}'' \quad (36)$$

Note that the process of projecting on $[\mathbf{Q}_1 \quad \mathbf{Q}_2]^T$, has computational cost at most, $O(NM^3)$ [33], while the resulting residual $\tilde{\mathbf{r}}$ has dimension smaller or equal to $6(M-1) - 1$, and \mathbf{w}'' follows $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{6(M-1)-1})$. After constructing (36) we perform a regular EKF update using the new measurement Jacobian matrix $\mathbf{R}_{\text{upper}}$.

V. EXPERIMENTAL RESULTS

Our experimental testbed consists of a Point Grey monochrome monocular camera⁴ with resolution 640×480 pixels and an InterSense NavChip IMU⁵. IMU signals were sampled at a frequency of 100 Hz while camera images were acquired at 7.5 Hz.

The trajectory (Fig. 5) has total length $\sim 22\text{m}$ and covers one loop in an indoor office area, dominated by lines, after which the testbed was returned to its starting location, so as to provide a quantitative characterization of the achieved accuracy.

Point and line features were tracked over a sliding window of 10 images. Edges were extracted using the Canny edge detector [13] and straight line features are detected using OpenCV’s probabilistic Hough transform [34]. For the purpose of tracking lines between images, we employ the methodology described in [35].

We compared the performance of the following Visual-Inertial Odometry algorithms:

- The Multi-state constraint Kalman filter (MSC-KF) of [1] using only observations of points (referred to as Std-VINS w/o Lines in Table I).
- The MSC-KF modified to also process tracked straight line edges (referred to as Std-VINS w/ Lines in Table I), where visual measurements to lines features over multiple images, are processed as described earlier.
- The proposed OC-MSC-KF, where observations of point and line features are used (referred to as OC-VINS w/ Lines in Table I). The main difference with the OC-MSC-KF of [11] is that measurements to both point and line features are processed, while the Jacobians are appropriately modified (as explained earlier) to ensure that no spurious information is injected during line-measurement updates.

Our, experimental results validate our conjecture that the Std-VINS, erroneously injects information along the unobservable directions (see Fig. 4), which makes it overly confident. In contrast, the OC-VINS adheres to the correct observability properties of the system, which is evident when considering its yaw and position uncertainty (see Fig. 4).

Furthermore, the reduced final position errors (see Table I) when processing both line and point features validate the potential gains to VINS, when employing line observations, over multiple time steps.

Visual-Inertial Odometry Framework	Final Position Error (cm)
Std-VINS w/o Lines	20
Std-VINS w/ Lines	19
OC-VINS w/ Lines	18

TABLE I: Final position error comparison

VI. CONCLUSION AND FUTURE WORK

In this paper, we studied the observability properties of a vision-aided inertial navigation system (VINS) that

⁴<http://www.ptgrey.com>

⁵<http://www.intersense.com>

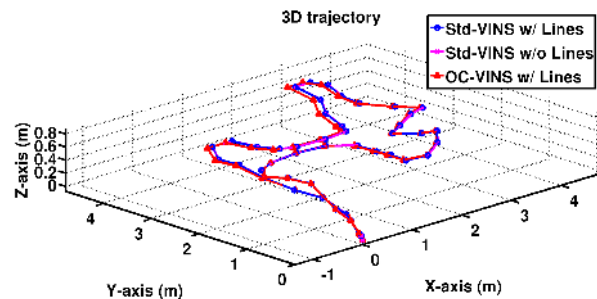


Fig. 5: Top: The 3D trajectory of the IMU-camera pair during the experiment. The Std-VINS erroneously treats rotations around gravity as observable, which results to a rotated trajectory. Bottom: Representative camera image from the experimental dataset with the extracted straight line edges superimposed (best viewed in color).

employs measurements of straight lines over multiple time steps. We proved that for the case of a single line, the system has five unobservable directions and presented their analytical form, as well as a discussion on their physical interpretation. Additionally, we introduced an EKF-based algorithm that fuses efficiently line-feature image tracks with inertial measurements, and improved its consistency by ensuring that no information is injected along the system’s unobservable directions. Furthermore, by performing visual-inertial odometry (VIO), instead of SLAM, the proposed algorithm has computational complexity only linear in the number of lines. Finally, we confirmed the key findings of the line-based VINS observability analysis and demonstrated the performance of the proposed algorithm experimentally using a MEMS-quality IMU and a miniature-size camera. As part of our future work, we plan to extend our approach to also include information about lines corresponding to known directions and study the impact on the filter’s consistency and accuracy.

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] 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. of the IEEE Int. Conf. on*

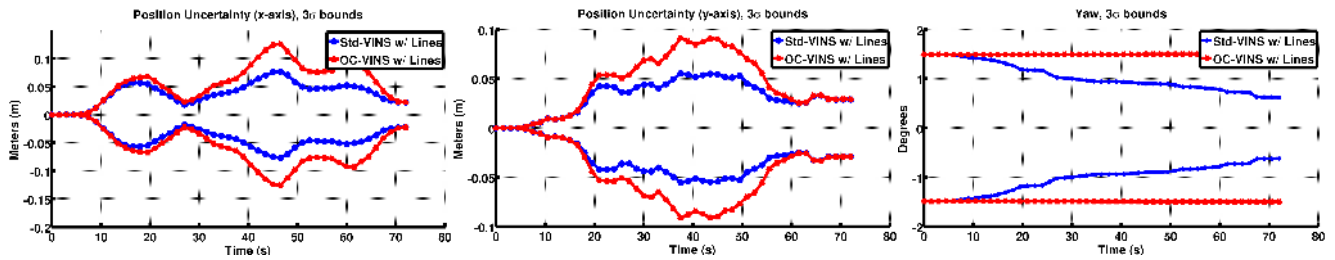


Fig. 4: Comparison of the x -axis, y -axis and yaw uncertainties for the Std-VINS w/ Lines and the OC-VINS w/ Lines. Note that the yaw uncertainty for the Std-VINS erroneously decreases even though no absolute orientation is available to the filter. This also causes an unjustifiably large decrease in the uncertainty along the x and y coordinates.

- Robotics and Automation*, St. Paul, MN, May 14–18, 2012, pp. 957–964.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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. of the IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 431–438.
- [7] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, “Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV,” in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, May 14–18, 2012, pp. 31–38.
- [8] 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.
- [9] T. Lupton and S. Sukkarieh, “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions,” *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.
- [10] M. Li and A. I. Mourikis, “Improving the accuracy of EKF-based visual-inertial odometry,” in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, May 14–18, 2012, pp. 828–835.
- [11] 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.
- [12] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Towards consistent vision-aided inertial navigation,” in *10th Int. Workshop Alg. Found. Robot.*, Cambridge, MA, Jun. 13–15, 2012.
- [13] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
- [14] P. Smith, I. Reid, and A. Davison, “Real-time monocular slam with straight lines,” in *British Machine Vision Conference*, vol. 1, Edinburgh, UK, Sep. 4–7 2006, pp. 17–26.
- [15] J. Meltzer and S. Soatto, “Edge descriptors for robust wide-baseline correspondence,” in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Anchorage, AK, Jun. 23–28, 2008, pp. 1–8.
- [16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [17] H. H. Chen, “Pose determination from line-to-plane correspondences: existence condition and closed-form solutions,” in *Proc. 3rd Int. Conf. Comp. Vision*, Osaka, Japan, Dec. 4–7, 1990, pp. 374–378.
- [18] F. M. Mirzaei and S. I. Roumeliotis, “Optimal estimation of vanishing points in a manhattan world,” in *IEEE Int. Conf. Comp. Vision*, Barcelona, Spain, Nov. 6–13, 2011, pp. 2454–2461.
- [19] —, “Globally optimal pose estimation from line correspondences,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 5581–5588.
- [20] J. Weng, T. Huang, and N. Ahuja, “Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 318–336, Mar. 1992.
- [21] M. E. Spetsakis and J. Aloimonos, “Structure from motion using line correspondences,” *Int. Journal of Computer Vision*, vol. 4, no. 3, pp. 171–183, Jun. 1990.
- [22] Y. Liu and T. S. Huang, “Estimation of rigid body motion using straight line correspondences,” *Computer Vision, Graphics, and Image Processing*, vol. 43, no. 1, pp. 37–52, Jul. 1988.
- [23] A. Bartoli and P. Sturm, “Structure from motion using lines: representation, triangulation and bundle adjustment,” *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 416–441, Dec. 2005.
- [24] C. J. Taylor and D. J. Kriegman, “Structure and motion from line segments in multiple images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1021–1032, Nov. 1995.
- [25] N. Ayache and O. D. Faugeras, “Maintaining representations of the environment of a mobile robot,” *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 804–819, Dec. 1989.
- [26] H. Rehbinder and B. Ghosh, “Pose estimation using line-based dynamic vision and inertial sensors,” *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 186–199, Feb. 2003.
- [27] J. Lobo and J. Dias, “Vision and inertial sensor cooperation using gravity as a vertical reference,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1597–1608, Dec. 2003.
- [28] 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
- [29] 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
- [30] 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.
- [31] D. G. Kottas and S. I. Roumeliotis, “Efficient and consistent vision-aided inertial navigation using line observations,” University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep. 2012-003, Sep. 2012. [Online]. Available: http://www-users.cs.umn.edu/~dkottas/pdfs/lines_vins_tr_winter_2012.pdf
- [32] 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.
- [33] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins Univ. Press, 1996.
- [34] J. Matas, C. Galambos, and J. Kittler, “Robust detection of lines using the progressive probabilistic hough transform,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, Apr. 2000.
- [35] C. Schmid and A. Zisserman, “Automatic line matching across views,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Jun. 17–19, 1997, pp. 666–671.