Efficient and Consistent Vision-aided Inertial Navigation using Line Observations

Dimitrios G. Kottas[1] and Stergios I. Roumeliotis[1]

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 (IMU) 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 observations[1] 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. To the best of our knowledge, with the exception of [6] where lines of *known direction* (parallel to the gravity) are used for improving the roll and pitch estimates, the problem of vision-aided inertial navigation using line observations 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 the VINS using observations of 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 along the line's direction. Furthermore, we show that this last direction becomes observable when more than one non-parallel lines are detected.
- 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.

[1] 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).

[1]In 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.

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, respectively. In Sect. III, we study the observability properties of the VINS based on line observations. The key findings of this analysis are leveraged in Sect. IV to improve the consistency of the EKF-based estimation algorithm. 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:

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

where ${}^I\bar{q}_G(t)$, ${}^G\mathbf{p}_I(t)$, and ${}^G\mathbf{v}_I(t)$ are the orientation, position, and velocity of the IMU frame $\{I\}$ with respect to the global frame $\{G\}$, and $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ denote the gyroscope and accelerometer biases, repsectively.

The system model describing the time evolution of the state is (see [26]):

$${}^I\dot{\bar{q}}_G(t) = \tfrac{1}{2}\Omega({}^I\omega(t)){}^I\bar{q}_G(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \tag{2}$$

$$\dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \tag{3}$$

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:

$$\omega_m(t) = {}^I\omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{4}$$

$$\mathbf{a}_m(t) = \mathbf{C}({}^I\bar{q}_G(t))\,({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t). \tag{5}$$

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 $\mathbf{n}_g(t)$ and $\mathbf{n}_a(t)$ are white-noise processes contaminating the corresponding measurements.

Linearizing at the current estimates and applying the expectation operator on both sides of (2)-(3), we obtain the IMU state propagation model:

$${}^I\dot{\hat{\bar{q}}}_G(t) = \frac{1}{2}\Omega({}^I\hat{\omega}(t)){}^I\hat{\bar{q}}_G(t), \; {}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t) \tag{6}$$

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

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:[2]

$$\widetilde{\mathbf{x}_R} = \begin{bmatrix} {}^I\delta\theta_G^T & \widetilde{\mathbf{b}}_g^T & {}^G\widetilde{\mathbf{v}}_I^T & \widetilde{\mathbf{b}}_a^T & {}^G\widetilde{\mathbf{p}}_I^T \end{bmatrix}^T, \tag{7}$$

the continuous-time IMU error-state equation becomes:

$$\dot{\widetilde{\mathbf{x}_R}}(t) = \mathbf{F}_R(t)\widetilde{\mathbf{x}_R}(t) + \mathbf{G}_R(t)\mathbf{n}(t) \tag{8}$$

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

$$\mathbf{F}_R = \begin{bmatrix} -\lfloor \hat{\omega}(t) \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^I\hat{\bar{q}}_G(t))\lfloor \hat{\mathbf{a}}(t) \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I\hat{\bar{q}}_G(t)) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{9}$$

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

and $\mathbf{n} \triangleq \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T \end{bmatrix}^T$ 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, and $\mathbf{Q}_R$ depends on the IMU noise characteristics and is computed offline.

The discrete-time state transition matrix from time t_1 to t, $\Phi(t,t_1)$, is computed in analytical form [27] as the solution to the matrix differential equation $\dot{\Phi}(t,t_1) = \mathbf{F}_R(t)\Phi(t,t_1)$. As we show in [27], the structure of $\Phi(t_{k+1},t_k) = \Phi(t_{k+1},t_1)\Phi(t_k,t_1)^{-1}$ with $\Phi_1 := \Phi(t_1,t_1) = \mathbf{I}_{15}$, is given by:

$$\Phi_{k+1} = \Phi(t_{k+1},t_k) = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \delta t\mathbf{I}_3 & \Phi_{54} & \mathbf{I}_3 \end{bmatrix}. \tag{10}$$

Finally, the discrete-time system noise covariance matrix is computed as:

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1},\tau)\mathbf{G}_R\mathbf{Q}_R\mathbf{G}_R^T\Phi^T(t_{k+1},\tau)\mathrm{d}\tau.$$

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 $\{L_G\}$ whose origin is the point on the line at minimum distance, d_{L_G}, 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:

$$\mathbf{x}_{\mathbf{L}} = \begin{bmatrix} {}^G\bar{q}_{L_G}^T & d_{L_G} \end{bmatrix}^T \tag{11}$$

[2]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, $\delta\theta$, implicitly defined by the *error quaternion* $\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\theta^T & 1 \end{bmatrix}^T$, where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide.

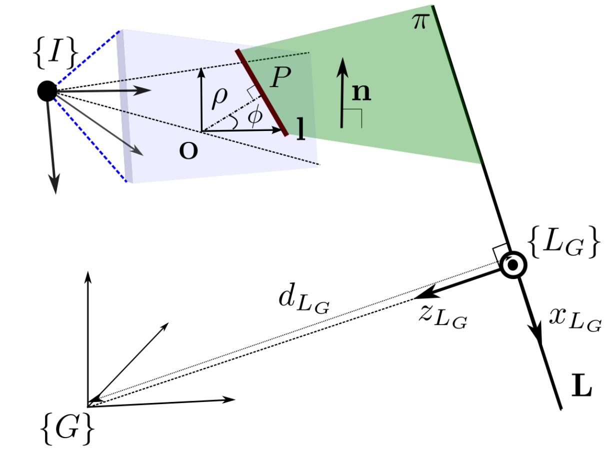

Fig. 1: Geometrical depiction of the line parameterization $({}^{G}\bar{q}_{L_G}^{T},\, d_{L_G})$ employed in our analysis.

while its corresponding error vector is:

$$\tilde{\mathbf{x}}_{\mathbf{L}} = \begin{bmatrix} {}^{G}\delta\theta_{L_G}^{T} & \tilde{d}_{L_G} \end{bmatrix}^{T} \tag{12}$$

For simplicity, we consider the IMU frame of reference $\{I\}$ to coincide with the camera frame of reference[3] and define $\mathbf{C_L} = \mathbf{C}({}^{G}\bar{q}_{L_G})$ and $d_L = d_{L_G}$. The optical center of the camera $\{I\}$, together with the 3D line $\mathbf{L}$, defines a plane π in space. The image sensor observes the 2D line $\mathbf{l}$, i.e. the intersection of the plane π and the image plane. The line detection algorithm, returns a line $\mathbf{l}$ 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 $\mathbf{p}^T = \begin{bmatrix} u & v & 1 \end{bmatrix}$, lies on the line if it satisfies the equality:

$$\mathbf{p}^T \begin{bmatrix} \cos\phi & \sin\phi & -\rho \end{bmatrix} = 0 \tag{13}$$

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

$$\mathbf{n} = IP \times \mathbf{u} = \begin{bmatrix} \cos\phi & \sin\phi & -\rho \end{bmatrix}^T \tag{14}$$

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 ${}^{I_{t_k}}\bar{q}_G$ and ${}^{G}\mathbf{p}_{I_{t_k}}$, and it observes the line $\mathbf{L}$, through its normal vector $\mathbf{n_{t_k}}$. The direction of line $\mathbf{L}$ expressed in the $\{I\}$ frame lies on the plane π , and hence satisfies the constraint:

$$\mathbf{n_{t_k}}^T \mathbf{C}({}^{I_{t_k}}\bar{q}_G)\mathbf{C_L}\mathbf{e_1} = 0 \tag{15}$$

[3] In practice, we perform IMU-camera extrinsic calibration following the approach of [28].

where $\mathbf{e_1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. Similarly, for the point ${}^{\mathbf{G}}\mathbf{p_{L_G}} = -\mathbf{C_L} d_L \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T = -\mathbf{C_L} d_L \mathbf{e_3}$ expressed in the $\{I\}$, we have:

$$\begin{aligned} \mathbf{n_{t_k}}^T \mathbf{C}({}^{I_{t_k}}\bar{q}_G)({}^{G}\mathbf{p}_{L_G} - {}^{G}\mathbf{p}_{I_{t_k}}) &= \\ \mathbf{n_{t_k}}^T \mathbf{C}({}^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L} d_L \mathbf{e_3} - {}^{G}\mathbf{p}_{I_{t_k}}) &= 0 \end{aligned} \tag{16}$$

Stacking the two constraints together, we arrive at:

$$\mathbf{h}(\mathbf{n_{t_k}}, \mathbf{x_{R_{t_k}}}, \mathbf{x_L})_{2\times 1} = \begin{bmatrix} \mathbf{n_{t_k}}^T \mathbf{C}({}^{I_{t_k}}\bar{q}_G)\mathbf{C_L}\mathbf{e_1} \\ \mathbf{n_{t_k}}^T \mathbf{C}({}^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L} d_L \mathbf{e_3} - {}^{\mathbf{G}}\mathbf{p_{I_{t_k}}}) \end{bmatrix} = \mathbf{0_{2\times 1}} \tag{17}$$

where $\mathbf{x_{R_{t_k}}}$ 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

$$\mathbf{z_{t_k}} = \begin{bmatrix} \phi & \rho \end{bmatrix}^T + \xi_{\mathbf{t_k}} \tag{18}$$

where $\xi_{\mathbf{t_k}}$ 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 $\xi_{\mathbf{t_k}}$ on $\mathbf{h}(\mathbf{n_{t_k}}, \mathbf{x_{R_{t_k}}}, \mathbf{x_L})$, denoted by $\mathbf{w_k}$ can be approximated through linearization as:

$$\mathbf{w_{k_{2\times 1}}} = \mathbf{A_{2\times 3}}\mathbf{B_{3\times 2}}\xi_{\mathbf{t_k}} \tag{19}$$

where $\mathbf{A} = \frac{\partial h}{\partial n_{t_k}}$ and $\mathbf{B} = \begin{bmatrix} \frac{\partial n_{t_k}}{\partial \phi} & \frac{\partial n_{t_k}}{\partial \rho} \end{bmatrix}$. Hence, $\mathbf{w_k}$ can be approximated by a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0_{2\times 1}}, \mathbf{Z_{t_k}})$ with $\mathbf{Z_{t_k}} = \mathbf{A}\mathbf{B}\mathbf{R}_{\phi\rho}\mathbf{B}^T\mathbf{A}^T$.

Using this noise parameterization, we arrive at the following measurement model, that couples the measurement of line $\mathbf{L}$ at time step t_k, $\mathbf{n_{t_k}}$, with the robot's state vector, $\mathbf{x_{R_{t_k}}}$, and the line parameters $\mathbf{x_L}$:

$$\mathbf{z}_{L,t_k} = \mathbf{h}(\mathbf{n_{t_k}}, \mathbf{x_{R_{t_k}}}, \mathbf{x_L}) + \mathbf{w_k} \tag{20}$$

We now, linearize (20), with respect to the error state $\tilde{\mathbf{x}}_{\mathbf{R}}$ and the line parameters error $\tilde{\mathbf{x}}_{\mathbf{L}}$, which yields:

$$\mathbf{z}_{L,t_k} - \hat{\mathbf{z}}_{L,t_k} = \nabla\mathbf{h}_{\mathbf{x_R}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}}\tilde{\mathbf{x}}_{\mathbf{R}} + \nabla\mathbf{h}_{\mathbf{x_L}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}}\tilde{\mathbf{x}}_{\mathbf{L}} + \mathbf{w_k} = \tag{21}$$

$$\mathbf{H_{R_k}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}}\tilde{\mathbf{x}}_{\mathbf{R}} + \mathbf{H_L}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}}\tilde{\mathbf{x}}_{\mathbf{L}} + \mathbf{w_k} \tag{22}$$

with the corresponding Jacobians given by:

$$\mathbf{H_{R_k}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}} = \begin{bmatrix} \mathbf{n}_{t_k}^T\lfloor\mathbf{C}({}^{I_{t_k}}\bar{q}_G)\mathbf{C_L}\mathbf{e_1}\times\rfloor & \mathbf{0}_{1\times 9} & \mathbf{0}_{1\times 3} \\ \mathbf{n}_{t_k}^T\lfloor\mathbf{C}({}^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L} d_L \mathbf{e_3} - {}^{G}p_{I_k})\times\rfloor & \mathbf{0}_{1\times 9} & -\mathbf{n}_{t_k}^T\mathbf{C}({}^{I_{t_k}}\bar{q}_G) \end{bmatrix} \tag{23}$$

$$\mathbf{H_{Lk}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}} = \begin{bmatrix} \mathbf{n}_{t_k}^T\mathbf{C}({}^{I_{t_k}}\bar{q}_G)\lfloor\mathbf{C_L}\mathbf{e_1}\times\rfloor & 0 \\ \mathbf{n}_{t_k}^T\mathbf{C}({}^{I_{t_k}}\bar{q}_G)\lfloor-\mathbf{C_L} d_L \mathbf{e_3}\times\rfloor & \mathbf{n}_{t_k}^T\mathbf{C}({}^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L}\mathbf{e_3}) \end{bmatrix}. \tag{24}$$

This, can be written in a compact form as:

$$\mathbf{H_{R_k}}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}} = \begin{bmatrix} \mathbf{H_{q_k 1}} & \mathbf{0}_{1\times 9} & \mathbf{0}_{1\times 3} \\ \mathbf{H_{q_k 2}} & \mathbf{0}_{1\times 9} & \mathbf{H_{p_k 2}} \end{bmatrix},\ \mathbf{H_L}|_{\mathbf{x}_{\mathbf{R}}^{\star},\mathbf{x}_{\mathbf{L}}^{\star}} = \begin{bmatrix} \mathbf{H_{l_k 1}} & \mathbf{0}_{1\times 3} \\ \mathbf{H_{l_k 2}} & \mathbf{H_{d_k 2}} \end{bmatrix}. \tag{25}$$

where $\mathbf{x}_{\mathbf{R}}^{\star}$ and $\mathbf{x}_{\mathbf{L}}^{\star}$ denote the estimates at which the Jacobians are computed.

III. Observability Analysis

In this section, we study the observability properties of a VINS system, that measures the same line $\mathbf{L}$ over m time steps, denoted by $t_1, \ldots t_m$. 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 the frame $\{G\}$. The time evolution of the linearized system state between time steps k and $k+1$ is described, by:

$$\begin{bmatrix} \tilde{\mathbf{x}}_{\mathbf{R}_{\mathbf{t}_{k+1}}} \\ \tilde{\mathbf{x}}_{\mathbf{L}} \end{bmatrix} = \begin{bmatrix} \Phi_{k+1}|_{\mathbf{x}^\star_{\mathbf{R_k}}} & \mathbf{0}_{15,4} \\ \mathbf{0}_{4,15} & \mathbf{I}_{4,4} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{\mathbf{R}_{\mathbf{t_k}}} \\ \tilde{\mathbf{x}}_{\mathbf{L}} \end{bmatrix} \tag{26}$$

where $\Phi_{k+1}|_{\mathbf{x}^\star_{\mathbf{R}}}$ is the system jacobian described in Sec. II-A, evaluated at the point $\mathbf{x}^\star_{\mathbf{R}}$. Note that the line coordinates' error does not change in time since we observe a static scene. Similarly, the linearized measurement model, is:

$$\tilde{\mathbf{z}}_{\mathbf{k}} = \mathbf{H}_{\mathbf{R_k}}|_{\mathbf{x}^\star_{\mathbf{R_k}}, \mathbf{x}^\star_{\mathbf{L}}} \tilde{\mathbf{x}}_{\mathbf{R}} + \mathbf{H}_{\mathbf{L}}|_{\mathbf{x}^\star_{\mathbf{R_k}}, \mathbf{x}^\star_{\mathbf{L}}} \tilde{\mathbf{x}}_{\mathbf{L}} \tag{27}$$

Since we study the system's observability properties, we set $\mathbf{n(t)} = 0$ and $\mathbf{w_k} = 0$ in (8) and (21), respectively. Therefore, (26) and (27) represent the system in the absence of noise. The $k-th$ row of the observability matrix $\mathbf{M}$, defined over the time period $t_1, \ldots t_k$, is given by:

$$\mathbf{M}_{\mathbf{k},:} = \begin{bmatrix} \mathbf{H}_{\mathbf{R_k}}|_{\mathbf{x}^\star_{\mathbf{R_k}}, \mathbf{x}^\star_{\mathbf{L}}} \Phi_k|_{\mathbf{x}^\star_{\mathbf{R}}} & \mathbf{H}_{\mathbf{L}}|_{\mathbf{x}^\star_{\mathbf{R_k}}, \mathbf{x}^\star_{\mathbf{L}}} \end{bmatrix} \tag{28}$$

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

A. True Linearized System

Hereafter, we investigate the directions that span the left nullspace of the observability matrix $\mathbf{M}$ under an "ideal" linearization around teh true $\mathbf{x}^\star_{\mathbf{R}}$ and $\mathbf{x}^\star_{\mathbf{L}}$, so as to derive an analytical form for the unobservable directions of the system. For simplicity, let us evaluate four rows of $\mathbf{M}$, corresponding to two measurements, at time steps t_1 and t_k, respectively. The block rows, of the observability matrix, for t_1 are:

$$\mathbf{M}(\mathbf{x}^\star_{\mathbf{R_1}}, \mathbf{x}^\star_{\mathbf{L}})_{1,:} = \begin{bmatrix} \mathbf{H}_{\mathbf{R_1}}|_{\mathbf{x}^\star_{\mathbf{R_1}}, \mathbf{x}^\star_{\mathbf{L}}} & \mathbf{H}_{\mathbf{L}}|_{\mathbf{x}^\star_{\mathbf{R_1}}, \mathbf{x}^\star_{\mathbf{L}}} \end{bmatrix} \tag{29}$$

while for t_k, are given by:

$$\mathbf{M}(\mathbf{x}^\star_{\mathbf{R_k}}, \mathbf{x}^\star_{\mathbf{L}})_{k,:} = \begin{bmatrix} \mathbf{M_{k,1}} & \mathbf{M_{k,2}} & \mathbf{M_{k,3}} & \mathbf{M_{k,4}} & \mathbf{M_{k,5}} & \mathbf{M_{k,6:7}} \end{bmatrix} \tag{30}$$

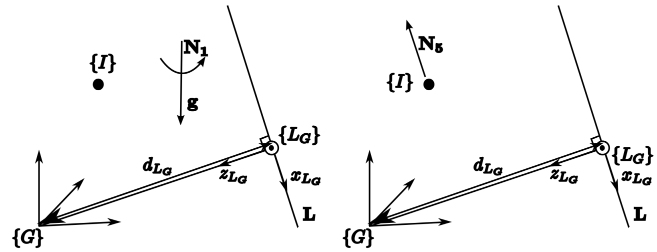

Fig. 3: Unobservable directions $\mathbf{N_1}$ and $\mathbf{N_5}$, corresponding to a rotation around the gravity vector and changes in the velocity of the sensor platform, parallel to the line direction.

where:

$$\mathbf{M_{k,1}} = \begin{bmatrix} \mathbf{n}_{t_k}^T \lfloor \mathbf{C}(^{I_{t_k}}\bar{q}_G)\mathbf{C_L e_1} \times \rfloor \Phi_{k11} \\ \mathbf{n}_{t_k}^T \lfloor \mathbf{C}(^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L} d_L \mathbf{e_3} - {}^{\mathbf{G}}\mathbf{p}_{\mathbf{I_k}}) \times \rfloor \Phi_{k11} - \mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)\Phi_{k51} \end{bmatrix} \tag{31}$$

$$\Phi_{k11} = \mathbf{C}(^{I_{t_k}}\bar{q}_{I_{t_1}}), \ \mathbf{M_{k,3}} = \begin{bmatrix} \mathbf{0_{1\times 3}} \\ -\mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)(t_k - t_1) \end{bmatrix} \tag{32}$$

$$\mathbf{M_{k,5}} = \begin{bmatrix} \mathbf{0_{1\times 3}} \\ -\mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)(t_k - t_1) \end{bmatrix} \tag{33}$$

$$\mathbf{M_{k,6:7}} = \mathbf{H_L} = \begin{bmatrix} \mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)\lfloor \mathbf{C_L e_1} \times \rfloor & 0 \\ \mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)\lfloor -\mathbf{C_L} d_L \mathbf{e_3} \times \rfloor & \mathbf{n}_{t_k}^T \mathbf{C}(^{I_{t_k}}\bar{q}_G)(-\mathbf{C_L e_3}) \end{bmatrix} \tag{34}$$

$$\Phi_{k51} = \lfloor {}^G\mathbf{p}_{I_{t_1}} + {}^G\mathbf{v}_{I_{t_1}}(t_k - t_1) - \frac{1}{2}{}^G\mathbf{g}((t_k - t_1))^2 - {}^G\mathbf{p}_{I_{t_k}} \times \rfloor \mathbf{C}(^G\bar{q}_{I_{t_1}}) \tag{35}$$

It can be verified, that *at least* the following *five* directions, lie in the unobservable subspace of the system. Meaning that $rank(M) \leq 14$.

$$\mathbf{N_1} = \begin{bmatrix} \mathbf{C}(^{I_{t_1}}\bar{q}_G)\mathbf{g} \\ \mathbf{0_{3\times 1}} \\ -\lfloor {}^G\mathbf{v}_{I_{t_1}} \times \rfloor \mathbf{g} \\ \mathbf{0_{3\times 1}} \\ -\lfloor {}^G\mathbf{p}_{I_{t_1}} \times \rfloor \mathbf{g} \\ -\mathbf{g} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{N_{1q}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{N_{1v}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{N_{1p}} \\ \mathbf{N_{1l}} \\ 0 \end{bmatrix}, \mathbf{N_2} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{C_L e_1} \\ \mathbf{0_{3\times 1}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{N_{2p}} \\ \mathbf{0_{3\times 1}} \\ 0 \end{bmatrix}, \tag{36}$$

$$\mathbf{N_3} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ -\mathbf{C_L e_3} \\ \mathbf{0_{3\times 1}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{N_{3p}} \\ \mathbf{0_{3\times 1}} \\ 1 \end{bmatrix}, \mathbf{N_4} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{C_L} d_L \mathbf{e_2} \\ -\mathbf{C_L e_1} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{N_{4p}} \\ \mathbf{N_{4l}} \\ 0 \end{bmatrix}, \mathbf{N_5} = \begin{bmatrix} \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{C_L e_1} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ \mathbf{0_{3\times 1}} \\ 0 \end{bmatrix} \tag{37}$$

Direction $\mathbf{N_1}$ corresponds to the rotation of the sensor platform and the line around the gravity vector. $\mathbf{N_{2\ldots 4}}$ span the space of all possible translations of the sensor platform and the line together with respect to the global frame of reference. The fifth direction $\mathbf{N_5}$ corresponds to a change of the velocity of the sensor platform, parallel to the line $\mathbf{L}$.

Consider now, the joint observability matrix, for the case of observing two non-parallel lines $\mathbf{L_1}$, $\mathbf{L_2}$, which is $\mathbf{M'} = \begin{bmatrix} \mathbf{M_{L_1}} \\ \mathbf{M_{L_2}} \end{bmatrix}$, where $\mathbf{M_{L_1}}$ and $\mathbf{M_{l_2}}$ are properly padded with zeros since we are considered two lines. As shown in [29], it can be easily verified that neither $\mathbf{N_{5L_1}}$, nor $\mathbf{N_{5L_2}}$ lie in the nullspace of $\mathbf{M'}$, since $\mathbf{L_1}$ and $\mathbf{L_2}$ are non-parallel.

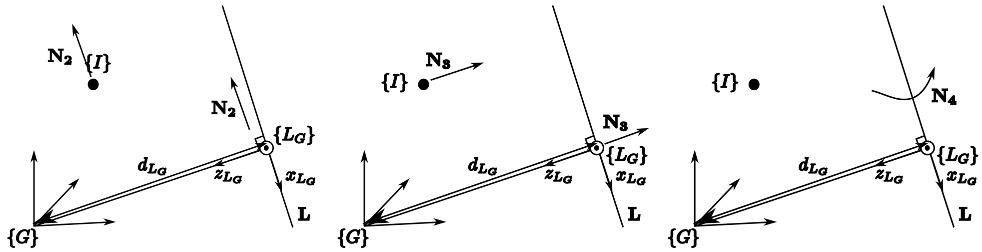

Fig. 2: Unobservable directions $\mathbf{N_2}$, $\mathbf{N_3}$ and $\mathbf{N_4}$. The combinations of these directions represent any translation of the sensor platform together with the line.

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 values of $\mathbf{n_{t_k}}$, $\mathbf{x_{R_{t_k}}}$, $\mathbf{x_L}$). 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 $\mathbf{C_L e_1}$ is always perpendicular to $\mathbf{C}({}^{I_{t_k}}\bar{q}_G)^T\mathbf{n}_{t_k}$, hence it always lies in the right nullspace of the ideal observability matrix $\mathbf{M}$. In the presence of noise, however, no vector is always perpendicular to every element of the set of vectors $\{\mathbf{C}({}^{I_{t_k}}\hat{\bar{q}}_G)^T\mathbf{n}_{t_k}\}$, due to the noise in the different estimates of ${}^{I_{t_k}}\hat{\bar{q}}_G$ at different time steps, and the fact that we never measure the true $\mathbf{n}_{t_k}$, but a perturbed version of it (i.e. (18)). The same applies for the rest of the directions. 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 coordinates $\mathbf{x_L}$, 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 to injection of spurious information. We conjecture that this can cause the EKF estimator to become inconsistent (i.e. being overconfident 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 computational complexity linear in the number of observed features. This is accomplished through avoiding to include a map of the environment in the state vector, but rather using all provided information for localization purposes.

A. Structure of the state vector and variables of interest

At a given time step k, the filter tracks the 16×1 evolving state, $\mathbf{x_R}$ (see. (1)). For the purpose of using measurements over a time window of the last M images, we employ stochastic cloning [30] and keep in our state the cloned sensor poses $\{\mathbf{x}_C = \begin{bmatrix} {}^{I_{k-M+i}}\bar{q}_G^T & {}^G\mathbf{p}_{I_{k-M+i}}^T \end{bmatrix}^T\}$, $i=0\dots(M-1)$. Correspondingly, the covariance consists of the 15×15 block of the evolving state, $\mathbf{P_{RR}}$, the $6M\times 6M$ block corresponding to the cloned robot poses, $\mathbf{P_{CC}}$ and their cross-correlations. Hence, the covariance of the augmented state vector has the following structure:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P_{RR}} & \mathbf{P_{RC}} \\ \mathbf{P_{RC}}^T & \mathbf{P_{CC}} \end{bmatrix} \tag{38}$$

Moreover, and in order to enforce the correct observability properties to the system, we maintain a copy of the evolving part of the nullspace directions, which corresponds to rotations around gravity, $\mathbf{N_1}$. Hence, we always keep the current nullspace direction $\mathbf{N_1}$ and the set $\mathbf{S_N} = \{\mathbf{N_{1(k-M+i)}}\}_{i=0\dots(M-1)}$, whose construction is described in the next paragraph, as part of the state cloning and propagation procedures.

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:

$$\mathbf{x}_C \leftarrow \begin{bmatrix} \mathbf{x}_C \\ {}^I\bar{q}_G \\ {}^G\mathbf{p}_I \end{bmatrix},\ \mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}_{:,[\mathbf{1:3},\mathbf{13:15}]} \\ \mathbf{P}^{\mathbf{T}}_{:,[\mathbf{1:3},\mathbf{13:15}]} & \mathbf{P}_{[\mathbf{1:3},\mathbf{13:15}],[\mathbf{1:3},\mathbf{13:15}]} \end{bmatrix} \tag{39}$$

If this is the first time, that we record an image, we initialize the nullspace direction $\mathbf{N_1}$, by direct evaluation of the corresponding expression (36) at our current state estimate. Afterwards, we add to the set $\mathbf{S_N}$ the current direction $\mathbf{N_1}$.

$$\mathbf{S_N} \leftarrow \{\mathbf{S_N},\ \mathbf{N_1}\} \tag{40}$$

C. State, Covariance and Nullspace Propagation

The moment we receive a new IMU measurement, we propagate the evolving state $\mathbf{x_R}$ (see Sec. II-A). We evaluate now the new nullspace direction, $\mathbf{N}_1'$, by substituting the propagated state vector $\mathbf{x_R}$ in equation (36). Using the analytical expressions [27] for the state transition matrix, we evaluate Φ, at the propagated $\mathbf{x_R}$. We now seek a modified

$\Phi^{\star}$, that adheres to the correct observability properties of the system [11]. Since, only the first 15 rows of $\mathbf{N_1}$ change, we denote $\mathbf{u} = \mathbf{N_1}_{[1:15]}$ and $\mathbf{w} = \mathbf{N_1'}_{[1:15]}$. and solve the following optimization problem:

$$\min_{\Phi^*} ||\Phi^* - \Phi||^2_{\mathscr{F}}, \quad \text{s.t. } \Phi^*\mathbf{u} = \mathbf{w} \tag{41}$$

where $||\cdot||_{\mathscr{F}}$ denotes the Frobenius matrix norm and the optimal solution is $\Phi^{\star} = \Phi - (\Phi\mathbf{u} - \mathbf{w})(\mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T$. We update the current nullspace direction $\mathbf{N_1} \leftarrow \mathbf{N_1'}$ and use the modified transition matrix, for covariance propagation.

$$\mathbf{P_{RR}} \leftarrow \Phi^{\star}\mathbf{P_{RR}}\Phi^{\star T} + \mathbf{Q_k} \tag{42}$$

$$\mathbf{P_{RC}} \leftarrow \Phi^{\star}\mathbf{P_{RC}} \tag{43}$$

D. Processing of 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. The linearized measurement model for line $j, j = 0\ldots(N-1)$, acquired at time step $i, (i = (k-M+1)\ldots k)$, is (21):

$$\tilde{\mathbf{z}}^{\mathbf{i}}_{\mathbf{j}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{C_i}}\tilde{\mathbf{x}}_{\mathbf{C_i}} + \mathbf{H}^{\mathbf{j}}_{\mathbf{L_i}}\tilde{\mathbf{x}}_{\mathbf{L_j}} + \mathbf{w_{i,j}} \tag{44}$$

since, our measurements relate only to the cloned robot poses, $\mathbf{H}^{\mathbf{j}}_{\mathbf{C_i}} = \begin{bmatrix}\mathbf{H}^{\mathbf{j}}_{\mathbf{Ri}[1:2],[1:3]} & \mathbf{H}^{\mathbf{j}}_{\mathbf{Ri}[1:2],[13:15]}\end{bmatrix}$. For the evaluation of the Jacobians we need the parameters of the line $\mathbf{L}$, which we compute through triangulation. So as to reduce the computational complexity, we approximate all $\mathbf{Z}^{\mathbf{j}}_{\mathbf{i}}$, for all the measurements that we are about to process by $\mathbf{R'}^{j}_{i} = \mathbf{I_2}\max(\lambda_{\max}(\{\mathbf{Z}^{\mathbf{j}}_{\mathbf{i}}\}_{i=(k-M+1)\ldots k,\ j=0\ldots(N-1)})) = \mathbf{I_2}\sigma'^2$. 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:

$$\begin{bmatrix}\mathbf{H}^{\mathbf{j}}_{\mathbf{C_i}} & \mathbf{H}^{\mathbf{j}}_{\mathbf{L_i}}\end{bmatrix}\Theta\ \mathbf{N_{1i}} = 0, \ \begin{bmatrix}\mathbf{H}^{\mathbf{j}}_{\mathbf{C_i}} & \mathbf{H}^{\mathbf{j}}_{\mathbf{L_i}}\end{bmatrix}\Theta\ \mathbf{N}^{\mathbf{j}}_{\mathbf{2\ldots5}} = 0 \tag{45}$$

where $\Theta = \begin{bmatrix}\mathbf{I_3} & \mathbf{0_{3\times9}} & \mathbf{0_{3\times7}}\\ \mathbf{0_{7\times3}} & \mathbf{0_{7\times9}} & \mathbf{I_7}\end{bmatrix}$, $\mathbf{N_{1i}}$ is the $i-th$ element of the set $\mathbf{S_N}$, and by $\mathbf{N}^{\mathbf{j}}_{\mathbf{2\ldots5}}$ we denote any of the directions $\mathbf{N_2}\ldots\mathbf{N_5}$ evaluated at the parameters of line j. To acquire the modified $\mathbf{H}^{\mathbf{j}\star}_{\mathbf{C\,i}}$ and $\mathbf{H}^{\mathbf{j}\star}_{\mathbf{L\,i}}$, we re-arrange (45), to bring it in the form:

$$\min_{\mathbf{A}^{\star}} ||\mathbf{A}^{\star} - \mathbf{A}||^2_{\mathscr{F}} \quad , \quad \text{s.t. } \mathbf{A}^{\star}\mathbf{u} = \mathbf{w} \tag{46}$$

and arrive at the following expressions:

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{p_i\,2}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{p_i\,2}} - \mathbf{H}^{\mathbf{j}}_{\mathbf{p_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{2_p}}(\mathbf{N}^{\mathbf{j}}_{\mathbf{2_p}}{}^{T}\mathbf{N}^{\mathbf{j}}_{\mathbf{2_p}})^{-1}\mathbf{N}^{\mathbf{j}}_{\mathbf{2_p}}{}^{T} \tag{47}$$

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{d_i\,2}} = -\mathbf{H}^{\mathbf{j}\star}_{\mathbf{p_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{3_p}} \tag{48}$$

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{l_i\,1}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{l_i\,1}} - \mathbf{H}^{\mathbf{j}}_{\mathbf{l_i\,1}}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}}(\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}}{}^{T}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}})^{-1}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}}{}^{T} \tag{49}$$

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{l_i\,2}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{l_i\,2}} - (\mathbf{H}^{\mathbf{j}}_{\mathbf{l_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}} + \mathbf{H}^{\mathbf{j}\star}_{\mathbf{p_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_p}})(\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}}{}^{T}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}})^{-1}\mathbf{N}^{\mathbf{j}}_{\mathbf{4_l}}{}^{T} \tag{50}$$

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{q_i\,1}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{q_i\,1}} - (\mathbf{H}^{\mathbf{j}}_{\mathbf{q_i\,1}}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_q}} + \mathbf{H}^{\mathbf{j}\star}_{\mathbf{l_i\,1}}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_l}})(\mathbf{N}^{\mathbf{j}}_{\mathbf{1_q}}{}^{T}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_q}})^{-1}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_q}}{}^{T} \tag{51}$$

$$\mathbf{H}^{\mathbf{j}\star}_{\mathbf{q_i\,2}} = \mathbf{H}^{\mathbf{j}}_{\mathbf{q_i\,2}} - (\mathbf{H}^{\mathbf{j}}_{\mathbf{q_i\,2}}\mathbf{u} - \mathbf{w})(\mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T \tag{52}$$

$$\mathbf{u} = \mathbf{N}^{\mathbf{j}}_{\mathbf{1_q}}, \quad \mathbf{w} = -\mathbf{H}^{\mathbf{j}\star}_{\mathbf{p_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_p}} - \mathbf{H}^{\mathbf{j}\star}_{\mathbf{l_i\,2}}\mathbf{N}^{\mathbf{j}}_{\mathbf{1_l}} \tag{53}$$

2) Linearized constraint among all camera poses: By stacking the measurements of line j over the time window $i = (k-M+1)\ldots k$, we arrive at:

$$\tilde{\mathbf{z}}_{\mathbf{j}} = \mathbf{H}^{\mathbf{j}\star}_{\mathbf{C}}\tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{H}^{\mathbf{j}\star}_{\mathbf{L}}\tilde{\mathbf{x}}_{\mathbf{L_j}} + \mathbf{w_j} \tag{54}$$

The matrix $\mathbf{H}^{\mathbf{j}\star}_{\mathbf{L}}$, has dimensions $2M\times4$, and for $M \geq 3$ it is full column rank. Hence, its left nullspace $\mathbf{U_j}$, is of dimensions $2M-4$. By premultiplying (54) by $\mathbf{U_j^T}$, we arrive at a measurement model, independent on the line parameters error.

$$\mathbf{U_j^T}\tilde{\mathbf{z}}_{\mathbf{j}} = \mathbf{U_j^T}\mathbf{H}^{\mathbf{j}\star}_{\mathbf{C}}\tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{U_j^T}\mathbf{w_j} \tag{55}$$

$$\implies \tilde{\mathbf{z}}'_{\mathbf{j}} = \mathbf{H}^{\mathbf{j}'}_{\mathbf{C}}\tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}'_{\mathbf{j}} \tag{56}$$

This key step in MSC-KF [1], defines a linearized constraint, *independent of the feature parameters*, among all the camera poses, that the line j was observed. By employing this methodology on the line-based VINS framework, we exploit all the geometric information induced by line $\mathbf{L}$, without the requirement of augmenting our state vector, with its parameters. Furthermore the computation of $\tilde{\mathbf{z}}'_{\mathbf{j}}$, of dimensions $2M-4\times1$, and $\mathbf{H}^{\mathbf{j}'}_{\mathbf{C}}$ can be performed in $O(M^2)$ operations 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, $j = 1\ldots N$, we acquire:

$$\tilde{\mathbf{z}}'_{N(2M-4)\times1} = \mathbf{H}'_{\mathbf{C}}\tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}'_{\mathbf{j}} \tag{57}$$

where the matrix $\mathbf{H}'_{\mathbf{C}}$ has size $N(2M-4)\times6M$ and hence is in general tall, since the number of observed lines (generally) exceeds the number of robot poses. As it is described in [3] and [1] for the case of point features, we factorize $\mathbf{H}'_{\mathbf{C}}$ as:

$$\mathbf{H}'_{\mathbf{C}} = \begin{bmatrix}\mathbf{Q_1} & \mathbf{Q_2}\end{bmatrix}\begin{bmatrix}\mathbf{R}_{\text{upper}}\\ \mathbf{0}\end{bmatrix} \tag{58}$$

Where $\mathbf{R}_{\text{upper}}$ can have at most $6(M-1)-1$ non-zero rows. By performing Givens rotations on $\tilde{\mathbf{z}}'_{N(2M-4)\times1}$ and $\mathbf{H}'_{\mathbf{C}}$ (see [31], pg.227), we can form, our final measurement model:

$$\tilde{\mathbf{r}} = \mathbf{R}_{\text{upper}}\tilde{\mathbf{x}}_{\mathbf{C}} + \mathbf{w}'' \tag{59}$$

The final prosess of projecting on $\begin{bmatrix}\mathbf{Q_1} & \mathbf{Q_2}\end{bmatrix}^T$, has computational cost at most, $O(NM^2)$ [31], while the final residual $\tilde{r}$ has dimension smaller or equal to $6(M-1)-1$, and $\mathbf{w}''$ follows $\mathscr{N}(\mathbf{0}, \sigma'^2\mathbf{I_{6(M-1)-1}})$. After constructing (59) we perform a regular EKF update using the new measurement Jacobian matrix $\mathbf{R}_{\text{upper}}$.

We compute the Kalman gain, $\mathbf{K} = \mathbf{P}\mathbf{R}_{\text{upper}}{}^{T}\left(\mathbf{R}_{\text{upper}}\mathbf{P}\mathbf{R}_{\text{upper}}{}^{T} + \sigma'^2\mathbf{I_{6(M-1)-1}}\right)^{-1}$, and perform state and covariance update, following the standard EKF equations [32].

Finally, we marginalize the oldest cloned pose, by simply removing $\mathbf{x}_{\mathbf{C}(k-M+1)}$ from $\mathbf{x_C}$, $\mathbf{N_{1(k-M+1)}}$ from the set $\mathbf{S_N}$, and marginalizing the corresponding rows and columns of $\mathbf{P}$.

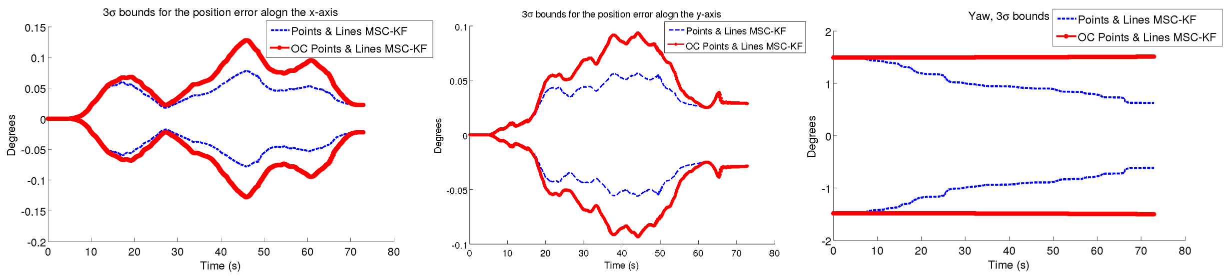

Fig. 4: Comparison of the x-axis, y-axis and yaw uncertainties for the MSC-KF and the OC-MSC-KF. Note that the yaw uncertainty for the MSC-KF erroneously decreases even though no absolute orientation is available to the filter. This also causes an unjustifiably large decrease in the uncertainty along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

V. Experimental Results

We validated the proposed line-based OC-MSC-KF on real data. Our hardware testbed consists of a Point Grey monochrome monocular camera with resolution 640x480 pixels and an InterSense NavChip IMU. Gradient edge detection is performed using the Canny Edge detector [13] and straight lines are extracted using OpenCV's probabilistic Hough transform [33]. For the purpose of tracking lines between images, we employ the methodology described in [34]. The trajectory (Fig. 5) has total length approximately 22∼m and covers one loop in an indoor office area, after which the testbed was returned to its starting location, so as to provide a quantitative characterization of the achieved accuracy. Tracking of points and lines is performed over a sliding window of 10 images.

We compared two filters, denoted by MSC-KF and OC-MSC-KF, both of which were processing line and point measurements concurrently. For the OC-MSC-KF, we process lines as described in Sec. IV-D as well as point feature measurements following [11], so to adhere to the correct observability properties of the system. Our, experimental results validate our conjecture that the regular MSC-KF, erroneously injects information along the unobservable directions (see Fig. 4), which makes it overly confident. In contrast, the OC-MSC-KF adheres to the correct observability properties of the system, which is evident when considering its yaw and position uncertainty (see Fig. 4). Finally, Table I, shows the final position error for the filters considered when processing points only or points and lines.

Filter	Final Position Error(cm)
Regular (points only) MSC-KF	20
Regular (points and lines) MSC-KF	19
OC (points only) MSC-KF	18
OC (points and lines) MSC-KF	18

TABLE I: Final position error comparison

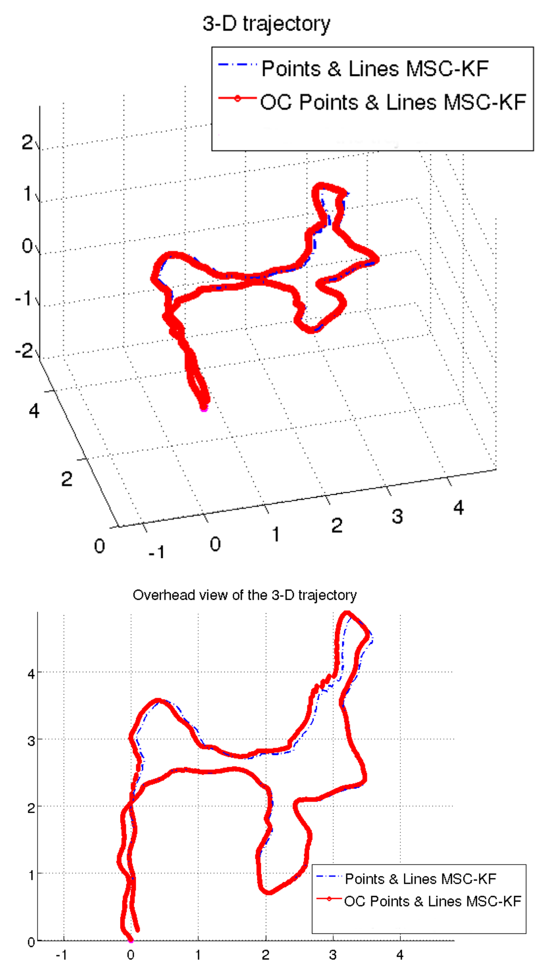

Fig. 5: The 3d trajectory (up) used for the experimental validation. The standard MSC-KF erroneously treats rotations around gravity as observable, which results to a rotated trajectory, as seen in the overhead view (down).

VI. Conclusion

In this paper, we studied the observability properties of the VINS when it employs measurements to straight lines over multiple time steps. We proved that for the case of a single line, the system has five unobservable degrees.

Additionally, we introduced an EKF-based algorithm, that fuses visual observations of lines with inertial measurements, and improved its consistency by ensuring that no information is injected along unobservable directions. Furthermore, by performing visual-inertial odometry (VIO) instead of SLAM, the proposed algorithm achieves complexity only linear in the number of measurements. 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 Robotics and Automation*, St. Paul, MN, May 14–18, 2012.

[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. Analy. 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, 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, May9–13, 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, 1992.

[21] M. E. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences," *Int. Journal Computer Vision*, vol. 4, no. 3, pp. 171–183, 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, 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. Patt. Analy. Machine Intell.*, 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. Robot. Autom.*, vol. 5, no. 6, pp. 804–819, Dec. 1989.

[26] 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.

[27] 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.

[28] 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.

[29] 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/

[30] 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.

[31] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins Univ. Press, 1996.

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

[33] 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, 2000.

[34] C. Schmid and A. Zisserman, "Automatic line matching across views," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 1997, pp. 666–671.