

# Observability Analysis of Aided INS With Heterogeneous Features of Points, Lines, and Planes

Yulin Yang<sup>✉</sup>, Student Member, IEEE, and Guoquan Huang<sup>✉</sup>, Member, IEEE

**Abstract**—In this article, we perform a thorough observability analysis for linearized inertial navigation systems (INS) aided by exteroceptive range and/or bearing sensors (such as cameras, LiDAR, and sonars) with different geometric features (points, lines, planes, or their combinations). In particular, by reviewing common representations of geometric features, we introduce two sets of unified feature representations, i.e., the quaternion and closest point (CP) parameterizations. While the observability of vision-aided INS (VINS) with point features has been extensively studied in the literature, we analytically show that the general aided INS with point features preserves the same observability property, i.e., four unobservable directions, corresponding to the global yaw and the global translation of the sensor platform. We further prove that there are at least five (or seven) unobservable directions for the linearized aided INS with a single line (plane) feature, and, for the first time, analytically derive the unobservable subspace for the case of multiple lines or planes. Building upon this analysis for homogeneous features, we examine the observability of the same system but with combinations of heterogeneous features, and show that, in general, the system preserves at least four unobservable directions, while if global measurements are available, as expected, the unobservable subspace will have lower dimensions. We validate our analysis in Monte-Carlo simulations using both EKF-based visual-inertial SLAM and visual-inertial odometry (VIO) with different geometric features.

**Index Terms**—Extended Kalman filter, inertial navigation system, observability analysis, SLAM, visual-inertial odometry.

## I. INTRODUCTION

INERTIAL navigation systems (INS) have been widely used for providing six-degrees-of-freedom (DoF) pose estimation when navigating in three-dimensional (3-D) space [1]. However, due to the noises and biases that corrupt the inertial measurement unit (IMU) readings, simple integration of the local angular velocity and linear acceleration measurements can cause large drifts in a short period of time, in particular, when using cheap MEMS IMUs. To mitigate this issue, additional sensors (e.g., optical camera [2]–[6], event camera [7], RGBD camera [8], [9],

Manuscript received March 17, 2019; accepted June 5, 2019. Date of publication August 14, 2019; date of current version December 3, 2019. This article was recommended for publication by Associate Editor M. Chli and Editor F. Chaumette upon evaluation of the reviewers comments. This work was supported in part by the University of Delaware College of Engineering, in part by the NSF under Grant IIS-1566129, in part by DTRA under Grant HDTRA1-16-1-0039, and in part by the Google. (*Corresponding author: Yulin Yang.*)

The authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: yuyang@udel.edu; ghuang@udel.edu).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2019.2927835

LiDAR [10], [11], and underwater sonar [12], [13]) are often used, i.e., aided INS. Among possible exteroceptive sensors, optical cameras—which are of low cost and energy-efficient while providing rich environmental information—are ideal aiding sources for INS and, thus, vision-aided INS (i.e., VINS) has recently prevailed, in particular, when navigating in GPS-denied environments (e.g., indoors) [4], [5], [14]–[18]. While many different VINS algorithms were developed in the last decade, the extended Kalman filter (EKF)-based methods are still among the most popular ones, such as multi-state constraint Kalman filter (MSCKF) [2], observability-constrained (OC)-EKF [14], [19], optimal-state constraint (OSC)-EKF [20], and right invariant error (RI)-EKF [21].

System observability plays an important role in state estimation [22]. Understanding system observability provides a deep insight about the system's geometric properties [14], [23], [24] and determines the minimal measurement modalities or state parameters needed to initialize an estimator. It can be used to identify degenerate motions [25], [26] that cause additional unobservable directions and should be avoided or alerted if possible in practice. Moreover, the observability-based methodologies used in OC-EKF [19] and OC-VINS [14] that enforce the system observability properties can be adopted to improve estimation consistency. Last but not least, analytical measurement Jacobians for aided INS estimators can be verified through the observability analysis process. For these reasons, significant research efforts have been devoted to the observability analysis of VINS. For example, it has been proved in [27] that biases, velocity, and roll and pitch angles in VINS are observable; in [14], [28], the null space of observability matrix (unobservable subspace) of linearized VINS was analytically derived; and in [8], [15], the Lie-derivative-based nonlinear observability analysis was presented. However, since most of the current VINS algorithms (e.g., [4], [5], [14]–[17]) are developed based on point features, the observability analysis is performed primarily using point measurements. Few has yet studied the observability properties of the aided INS with heterogeneous geometric features (e.g., points, lines and planes) which can be extracted from range and/or bearing sensor measurements.

In this article, building upon our recent conference publications [25], [29], we perform a thorough observability analysis for the linearized aided INS using points, lines, planes, and their different combinations. In particular, we conduct a brief overview of the most commonly used representations of point, line, and plane features and introduce two sets of unified parameterization: the quaternion and closest point (CP) forms, which will

greatly facilitate our ensuing analysis. Moreover, we perform an in-depth study to identify several degenerate motions, which is of practical significance, as these motions may negatively impact the system observability by causing more unobservable directions and thus exacerbate canonical VINS estimators (e.g., see [24], [26]). The insights obtained from the observability analysis are leveraged when developing our EKF-based VINS algorithms [including visual-inertial odometry (VIO) and visual-inertial SLAM (VI-SLAM)] using heterogeneous geometric features, which are evaluated in simulations to validate our analysis. Specifically, the main contributions of this article include:

- 1) We introduce unified quaternion and CP representations for points, lines, and planes. In particular, the CP form of line is proposed for the first time, which is a minimal parameterization with a four-dimensional (4-D) Euclidean vector to represent a line and its error state.
- 2) In the case of point features, we generalize the VINS observability analysis to encompass any type of aiding sensors (such as 3-D LiDAR, two-dimensional (2-D) imaging sonar, and stereo cameras) and analytically show that the same observability properties remain (i.e., four unobservable directions). In the case of line (or plane) features, we show that there exist at least five (or seven) unobservable directions for linearized aided INS with a single line (or plane) feature, which is also generalized to the cases with multiple line (or plane) features in the state vector. In the case of different feature combinations, we show that in general, there are at least four unobservable directions.
- 3) We study in-depth the effects of global measurements on the system observability, and show that they, as expected, will greatly improve the observability. On the other hand, by employing spherical coordinates for the point feature, we identify several degenerate motions that cause the aided INS to have more unobservable directions.
- 4) To validate our observability analysis of linearized aided INS, we develop our own MSCKF-based VIO and MSCKF hybrid VI-SLAM using heterogeneous geometric features (i.e., points, lines, planes, and their combinations) and perform extensive Monte-Carlo simulations by comparing the standard and the benchmark (ideal) filters.

## II. RELATED WORK

Aided INS is a classical research topic with a significant body of literature [30] and has recently been re-emerging in part due to the advancement of sensing and computing technologies. In this section, we briefly review the related literature by focusing on the vision-aided scenarios.

### A. Aided INS With Points, Lines, and Planes

As mentioned earlier, VINS is among the most popular localization methods, in particular for resource-constrained sensor platforms [such as mobile devices and micro-aerial vehicles (MAVs)] navigating in GPS-denied environments (e.g., see [4], [31]–[33]). While most current VINS algorithms focus on using point features (e.g., [4], [14]–[16]), line and plane features should not be blindly discarded in structured environments [6],

[9], [26], [34]–[40], in part because (i) they are ubiquitous and compact in many urban or indoor environments (e.g., doors, walls, and stairs), (ii) they can be detected and tracked over a relatively long time period, and (iii) they are more robust in textureless environments compared to point features.

In the case of utilizing line features, Kottas *et al.* [34], [41] represented the line with a quaternion and a (distance) scalar. With this parameterization, they studied the observability properties for linearized VINS with point and line features. They also leveraged the structural line constraints (e.g., parallel lines) to improve their estimator. Yu *et al.* [35] proposed a minimal four-parameter representation of line features for VIO using rolling-shutter cameras, while Zheng *et al.* [6] used two endpoints (of a line segment) to represent a line and designed point/line VIO based on the MSCKF. Recently, He *et al.* [36] employed the Plücker representation for line parameterization with the minimal orthonormal error states [42], and developed a tightly coupled keyframe-based monocular inertial SLAM system.

In the case of exploiting plane features, Guo *et al.* [38] analyzed the observability of VINS using both point and plane features, while assuming the plane orientation was *a priori* known. The authors showed that VINS with only plane bearing measurements have 12 unobservable directions as compared to four if both point and plane measurements are present. Hesch *et al.* [39] developed a 2-D LiDAR-aided INS algorithm that jointly estimates the perpendicular structural planes associated with buildings, along with the IMU states. In our previous work [9], a tightly coupled aided INS with point and plane features was designed and point-on-plane constraints were also incorporated to improve the estimator accuracy.

From the abovementioned work, one particular challenge of estimating these geometric features is to find proper parameterization. However, different geometric feature representations are presented in literature. In this article, we summarize the commonly used feature representations and propose two unified forms: the quaternion and CP. Both proposed forms can be incorporated into existing estimators due to their minimal error states. We analytically derive the Jacobians with these unified parameterizations and use these Jacobians for observability analysis. In addition, in most of the abovementioned works, only aided INS with point/line or point/plane features were developed. In this article, we build an estimator that can handle all geometric features of points, lines, and planes.

### B. VINS Observability Analysis

As system observability is important for consistent estimation [23], we have been the first to design observability-constrained consistent estimators for robot localization and mapping problems in our prior work [17], [19], [43]–[48]. Since then, significant research efforts have been devoted to the observability analysis of VINS. In particular, in [49], [50], the system's indistinguishable trajectories were examined from the observability perspective. By employing the concept of continuous symmetries as in [51], Martinelli [27] analytically derived the closed-form solution of VINS and identified that IMU biases, 3-D velocity, and global roll and pitch angles are observable. He also examined system observability with degenerate

motions [24], minimum available sensors [52], and unknown inputs [53], [54]. Recently, he provided analytic solutions for cooperative VIO [55] with observability analysis. Based on the Lie derivatives and observability matrix rank test [56], Hesch *et al.* [15] analytically showed that the monocular VINS has four unobservable directions, i.e., the global yaw and the global position of the exteroceptive sensor. Guo *et al.* [38] extended this method to the RGBD-camera-aided INS and showed that the system preserves the same unobservable directions if both point and plane measurements are available. Using a similar idea, the observability of IMU-camera (monocular, RGBD) calibration was analytically studied in [8], [57], [58], and the extrinsic calibration between the IMU and camera was shown to be observable, given generic motions. Additionally, in [59], [60], the system with a downward-looking camera measuring point features from horizontal planes was shown to have the sensor's global  $z$  position observable.

More importantly, as in practice, VINS estimators are built upon the linearized system, it necessitates to perform observability analysis for the linearized VINS whose observability properties can be exploited when designing an estimator. For instance, Li *et al.* [16], [61] performed observability analysis for the linearized VINS (without considering biases) and adopted the idea of first-estimates Jacobian [44] to improve filter consistency. Analogously, in [14], [28], [62], the authors conducted observability analysis for the linearized VINS with full states (including IMU biases) and analytically derived the system unobservable directions by finding the right null space of the observability matrix [14]. Based on this analysis, the observability-constrained (OC)-VINS algorithm was developed. In addition, Wu *et al.* [26] identified two degenerate motions (pure translation and constant acceleration) which could cause more unobservable directions for monocular camera-aided INS. Based on similar methodology, in our recent work [63], we incorporated both spatial and temporal calibrations into aided INS and proved that all these calibration parameters are observable under random sensor motion. We also identified four degenerate motions that cause certain calibration parameters to be unobservable.

Because of its practical significance in state estimation, the observability analysis of the linearized aided INS with heterogeneous features is the primary focus of this work, in which we have also studied the effects of global measurements and degenerate motions on the system observability. In particular, unlike [14], [28], [62], in our observability analysis, we consider generic aiding sensors and relax the assumption that the aiding sensor frame coincides with the IMU frame; and different with the degenerate motion analysis in [24] and [26], we show that the rigid transformation between the aiding sensor and IMU does affect the degenerate motions. As compared to the observability analysis of aided INS with global measurements in [64], which is based on Lie derivatives and only considers global roll and pitch measurements, we considered linearized aided INS with global position and orientation measurements. Weiss [64] verified the unobservable directions numerically (in Mathematica), instead of computing them analytically as in this article.

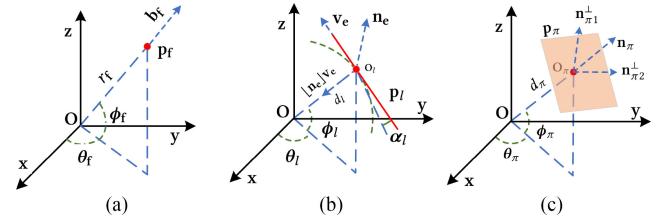


Fig. 1. Geometrical parameters for (a) point  $p_f$ , (b) line  $p_l$ , and (c) plane  $p_\pi$ .

### III. POINT, LINE, AND PLANE REPRESENTATIONS

As proper representations of geometrical features are important for state estimation, based on an extensive review of the most commonly used representations of points, lines and planes—which are summarized in Table I and illustrated in Fig. 1—we introduce two sets of *unified* representations, i.e., the quaternion and CP parameterizations.

#### A. Point Representations

Model 1 in Table I represents the homogeneous coordinate,  $f_i, i \in \{1, \dots, 4\}$ , for a point feature  $p_f$ , which is the most general form. Model 2 represents a point with a unit bearing vector  $b_f$  and a range scalar  $r_f$  measuring the distance from point to the origin  $O$  [see Fig. 1(a)]. Since the 3D unit vector  $b_f$  can be represented by two angles  $\theta_f$  and  $\phi_f$ :  $b_f = [\cos \theta_f \cos \phi_f \quad \sin \theta_f \cos \phi_f \quad \sin \phi_f]^\top$ , we can easily derive Model 3, which is similar to spherical coordinates. If we use the inverse of range scalar  $\lambda_f = 1/d_f$ , we get Model 4, which essentially is equivalent to inverse depth representation [65]. Recently, Maley and Huang [66] introduced a unit quaternion representation (Model 5) for points, which wraps  $b_f$  and  $r_f$  into a unit quaternion. Model 5 leverages the quaternion error state, which is minimal for point state estimation. For example, a point in quaternion form can be written as<sup>1</sup>

$$\bar{q}_f = \begin{bmatrix} q_f \\ q_f \end{bmatrix} = \frac{1}{\sqrt{1+r_f^2}} \begin{bmatrix} b_f \\ r_f \end{bmatrix} = \delta \bar{q}_f \otimes \hat{\bar{q}}_f \simeq \begin{bmatrix} \frac{1}{2}\delta\theta_f \\ 1 \end{bmatrix} \otimes \hat{\bar{q}}_f \quad (1)$$

where  $\delta\theta_f$  is the error state for point in unit quaternion form. Let  $p_f$  be the CP from the point to the origin, which is the most conventional parameterization for point feature (i.e., Model 6). It can be computed by multiplying the bearing vector  $b_f$  with the range scalar  $r_f$  as

$$p_f = r_f b_f = \hat{p}_f + \tilde{p}_f \quad (2)$$

where  $\tilde{p}_f$  is the error state in the CP form.

<sup>1</sup>Throughout this article,  $\hat{x}$  is used to denote the estimate of a random variable  $x$ , while  $\tilde{x} = x - \hat{x}$  is the error in this estimate. The quaternion error state  $\delta\theta$  is defined as  $\delta\bar{q} = [\frac{1}{2}\delta\theta^\top \ 1]^\top = \bar{q} \otimes \hat{\bar{q}}^{-1}$ , where  $\otimes$  denotes JPL quaternion multiplication [67].  $O_{m \times n}$  and  $\mathbf{0}_n$  denote  $m \times n$  and  $n \times n$  matrices of zeros, respectively, and  $I_n$  is the identity matrix.

TABLE I  
SUMMARY OF POINT, LINE, AND PLANE REPRESENTATIONS

Model #	Point	Error states	Line	Error states	Plane	Error states
1: General Form	$f_1, f_2, f_3, f_4$	not minimal	$\mathbf{n}_l, \mathbf{v}_l$	$\delta\theta_l, \delta\phi_l$	$\pi_1, \pi_2, \pi_3, \pi_4$	not minimal
2: Geometric Form	$\mathbf{b}_f, r_f$	not minimal	$\mathbf{n}_e = \frac{\mathbf{n}_l}{\ \mathbf{n}_l\ }$ $\mathbf{v}_e = \frac{\mathbf{v}_l}{\ \mathbf{v}_l\ }$ $d_l = \frac{\ \mathbf{n}_l\ }{\ \mathbf{v}_l\ }$	not minimal	$\mathbf{n}_\pi, d_\pi$	not minimal
3: Spherical Form	$\theta_f, \phi_f, r_f$	$\tilde{\theta}_f, \tilde{\phi}_f, \tilde{r}_f$	$\theta_l, \phi_l, \alpha_l, d_l$	$\tilde{\theta}_l, \tilde{\phi}_l, \tilde{\alpha}_l, \tilde{d}_l$	$\theta_\pi, \phi_\pi, d_\pi$	$\tilde{\theta}_\pi, \tilde{\phi}_\pi, \tilde{d}_\pi$
4: Inverse Depth	$\theta_f, \phi_f, \lambda_f = \frac{1}{r_f}$	$\tilde{\theta}_f, \tilde{\phi}_f, \tilde{\lambda}_f$	$\theta_l, \phi_l, \alpha_l, \lambda_l = \frac{1}{d_l}$	$\tilde{\theta}_l, \tilde{\phi}_l, \tilde{\alpha}_l, \tilde{\lambda}_l$	$\theta_\pi, \phi_\pi, \lambda_\pi = \frac{1}{d_\pi}$	$\tilde{\theta}_\pi, \tilde{\phi}_\pi, \tilde{\lambda}_\pi$
5: Quaternion	$\bar{q}_f = \frac{1}{\sqrt{1+r_f^2}} \begin{bmatrix} \mathbf{b}_f \\ r_f \end{bmatrix}$	$\delta\theta_f$	$\bar{q}_l, d_l$	$\delta\theta_l, \tilde{d}_l$	$\bar{q}_\pi = \frac{1}{\sqrt{1+d_\pi^2}} \begin{bmatrix} \mathbf{n}_\pi \\ d_\pi \end{bmatrix}$	$\delta\theta_\pi$
6: Closest Point	$\mathbf{p}_f = r_f \mathbf{b}_f$	$\mathbf{p}_f = \hat{\mathbf{p}}_f + \check{\mathbf{p}}_f$	$\mathbf{p}_l = d_l \bar{q}_l$	$\mathbf{p}_l = \hat{\mathbf{p}}_l + \check{\mathbf{p}}_l$	$\mathbf{p}_\pi = d_\pi \mathbf{n}_\pi$	$\mathbf{p}_\pi = \hat{\mathbf{p}}_\pi + \check{\mathbf{p}}_\pi$

### B. Line Representations

Given two 3-D points  $\mathbf{p}_{f1}$  and  $\mathbf{p}_{f2}$  in a line  $\mathbf{x}_l$ , we can obtain its Plücker coordinates (see Model 1 of lines in Table I) as in [42], [68]

$$\begin{bmatrix} \mathbf{n}_l \\ \mathbf{v}_l \end{bmatrix} = \begin{bmatrix} [\mathbf{p}_{f1} \mid \mathbf{p}_{f2}] \\ \mathbf{p}_{f2} - \mathbf{p}_{f1} \end{bmatrix} \quad (3)$$

where  $[\cdot]$  is the cross-product operation (or a skew symmetric matrix),  $\mathbf{n}_l$  represents the normal direction of the plane constructed by the two points and the origin, and  $\mathbf{v}_l$  represents the line direction. In Model 1, the distance from the origin to the line can be computed as  $d_l = \frac{\|\mathbf{n}_l\|}{\|\mathbf{v}_l\|}$ . Bartoli *et al.* [42] introduced the minimal orthonormal error state ( $\delta\theta_l$  and  $\delta\phi_l$ ; see [25] for detailed explanations) of Model 1 when involving lines in structure from motion. We can also represent the line with Model 2, which contains all the geometric elements of a line, including a unit normal direction  $\mathbf{n}_e = \frac{\mathbf{n}_l}{\|\mathbf{n}_l\|}$ , a unit line direction  $\mathbf{v}_e = \frac{\mathbf{v}_l}{\|\mathbf{v}_l\|}$ , and the distance scalar  $d_l$  [see Fig. 1(b)]. Alternatively, a line can be parameterized by three angles  $\theta_l, \phi_l, \alpha_l$  and a distance  $d_l$  (Model 3; see our companion technical report [69] for the transformation between Models 2 and 3). In analogy to the case of point features, we can use the inverse depth ( $\lambda_l = 1/d_l$ ) representation (Model 4) for line features. Interestingly, Kottas *et al.* [34] used a unit quaternion  $\bar{q}_l$  and a distance scalar  $d_l$  to represent a line (Model 5), where the quaternion describes the line direction

$$\mathbf{R}(\bar{q}_l) = [\mathbf{n}_e \ \mathbf{v}_e \ [\mathbf{n}_e \mid \mathbf{v}_e]] \quad (4)$$

$$\bar{q}_l = \delta\bar{q}_l \otimes \hat{\bar{q}}_l \simeq \begin{bmatrix} \frac{1}{2}\delta\theta_l \\ \frac{1}{2}\delta\phi_l \\ 1 \end{bmatrix} \otimes \hat{\bar{q}}_l \quad (5)$$

where  $\delta\theta_l$  represents the error state of the line quaternion. The 4-D minimal error states of the line include the quaternion error angle and the distance scalar error  $[\delta\theta_l^\top \ \tilde{d}_l]^\top$ .

More importantly, if we multiply the unit quaternion  $\bar{q}_l$  with the distance scalar  $d_l$ , we obtain a 4-D vector, which can be considered as the “closest point” for a line in the 4-D vector space (Model 6):

$$\mathbf{p}_l = d_l \bar{q}_l = d_l [\mathbf{q}_l^\top \ q_l]^\top = \hat{\mathbf{p}}_l + \check{\mathbf{p}}_l \quad (6)$$

where  $\check{\mathbf{p}}_l$  is the 4-D error state for the CP of a line. To the best of our knowledge, this minimal CP parameterization for 3-D lines (6) is proposed for the first time and shown to have good numerical stability (see Section VIII). The transformation between the quaternion and CP error states can be found in Appendix A.

### C. Plane Representations

Similar to point features, the homogeneous coordinates  $(\pi_i, i \in \{1, \dots, 4\})$  are the most general form of planes (Model 1) [70]. The Hesse form (Model 2) uses the normal direction  $\mathbf{n}_\pi$  and the distance scalar  $d_\pi$  to represent a plane. As  $\mathbf{n}_\pi$  can be represented by two angles  $\theta_\pi$  and  $\phi_\pi$  [see Fig. 1(c)],  $\mathbf{n}_\pi = [\cos \theta_\pi \cos \phi_\pi \ \sin \theta_\pi \cos \phi_\pi \ \sin \phi_\pi]^\top$ , the spherical coordinates (Model 3) can be used to represent the plane with two angles ( $\theta_\pi$  and  $\phi_\pi$ ) and the distance scalar  $d_\pi$ . If using the inverse depth  $\lambda_\pi = 1/d_\pi$ , we have the inverse depth representation for planes (Model 4). Recently, Kaess [40] proposed to use a unit quaternion to represent a plane by stacking the unit normal direction and the distance scalar into a quaternion (Model 5)

$$\bar{q}_\pi = \begin{bmatrix} \mathbf{q}_\pi \\ q_\pi \end{bmatrix} = \frac{1}{\sqrt{1+d_\pi^2}} \begin{bmatrix} \mathbf{n}_\pi \\ d_\pi \end{bmatrix} = \delta\bar{q}_\pi \otimes \hat{\bar{q}}_\pi \simeq \begin{bmatrix} \frac{1}{2}\delta\theta_\pi \\ 1 \end{bmatrix} \otimes \hat{\bar{q}}_\pi \quad (7)$$

where  $\delta\theta_\pi$  is the minimal error state for the quaternion plane representation. In Model 6, the CP from the plane to the origin is used to represent the plane [11], [71], which has the minimal Euclidean error state  $\check{\mathbf{p}}_\pi$

$$\mathbf{p}_\pi = d_\pi \mathbf{n}_\pi = \hat{\mathbf{p}}_\pi + \check{\mathbf{p}}_\pi. \quad (8)$$

### D. Remarks

It is clear that parameterizations based on Models 1 and 2 may cause numerical issues (e.g., singular information matrices) if they are directly used in least-squares optimization. While Models 3 and 4 are minimal representations, these models might suffer from singularities when the elevation angle  $\phi$  approximates  $\pm\frac{\pi}{2}$ , similar to gimbal lock for Euler angles.

Interestingly, all point [66], line [34], and plane [40] features can be parameterized by the *unified* representation of quaternion (Model 5), which exploits the minimal error states of quaternion

for better numerical stability during state estimation. However, the observability properties of quaternion representation for point and plane are missing in the literature, although it has been studied in the case of line features [34]. Therefore, we perform an extensive observability analysis with the unified quaternion representation for points, lines, and planes, where we have analytically derived the measurement Jacobians (see our companion technical report [69] for detailed derivations), showing that the same observability properties of aided INS are preserved.

More importantly, the CP form (i.e., Model 6) provides another *unified* parameterization for different geometric features. The CP model for point is simply its 3-D position in Euclidean space. While the CP representation for plane feature was introduced for LiDAR-aided INS in our prior work [11], in this work, we propose a novel 4-D CP model for line features with the minimal 4-D error states in 4-D Euclidean space. Note that when using the CP parameterization, the error propagation for different geometric features can be easily defined in the Euclidean vector space, and thus cost functions with intuitive geometric interpretation can be formulated.

We should point out that in order to make our presentation concise and easy to follow, we will use the quaternion representation of line and the CP of plane for observability analysis, while the detailed analysis for unified quaternion and CP parameterization is documented in our companion technical report [69].

#### IV. AIDED INS WITH DIFFERENT FEATURES

In this section, we describe the system and measurement models for aided INS with different geometric features, providing the basis for our ensuing observability analysis.

The state vector of the aided INS contains the current IMU state  $\mathbf{x}_I$  and the feature state  ${}^G\mathbf{x}_f$ :

$$\begin{aligned} \mathbf{x} &= [\mathbf{x}_I^\top \ {}^G\mathbf{x}_f^\top]^\top \\ &= [{}^I_G\bar{q}^\top \ \mathbf{b}_g^\top \ {}^G\mathbf{v}_I^\top \ \mathbf{b}_a^\top \ {}^G\mathbf{p}_I^\top \ {}^G\mathbf{x}_f^\top]^\top. \end{aligned} \quad (9)$$

In the above expressions,  ${}^I_G\bar{q}$  is a unit JPL quaternion [67] that represents the rotation from the global frame  $\{G\}$  to the current IMU frame  $\{I\}$ , whose corresponding rotation matrix is  ${}^I_G\mathbf{R}(\bar{q})$ .  $\mathbf{b}_g$  and  $\mathbf{b}_a$  represent the gyroscope and accelerometer biases, respectively, while  ${}^G\mathbf{v}_I$  and  ${}^G\mathbf{p}_I$  denote the current IMU velocity and position in the global frame.  ${}^G\mathbf{x}_f$  denotes the generic features (expressed in global frame), which might include points, lines, planes, or their combinations.

##### A. System Dynamic Model

The system dynamic model is given by (see [67])

$$\begin{aligned} {}^I_G\dot{\bar{q}}(t) &= \frac{1}{2}\boldsymbol{\Omega}({}^I\boldsymbol{\omega}(t)){}^I_G\bar{q}(t) \\ {}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \\ \dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad {}^G\dot{\mathbf{x}}_f(t) = \mathbf{0}_{m_f \times 1} \end{aligned} \quad (10)$$

where  $\boldsymbol{\omega}$  and  $\mathbf{a}$  are the angular velocity and linear acceleration, respectively.  $\mathbf{n}_{wg}$  and  $\mathbf{n}_{wa}$  are the zero-mean Gaussian noises driving the gyroscope and accelerometer biases.  $m_f$  is the dimension of  ${}^G\mathbf{x}_f$ , and  $\boldsymbol{\Omega}(\boldsymbol{\omega}) \triangleq \begin{bmatrix} -[\omega] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix}$ . The continuous-time linearized error-state equation is given by

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}(t) &\simeq \begin{bmatrix} \mathbf{F}_c(t) & \mathbf{0}_{15 \times m_f} \\ \mathbf{0}_{m_f \times 15} & \mathbf{0}_{m_f} \end{bmatrix} \tilde{\mathbf{x}}(t) + \begin{bmatrix} \mathbf{G}_c(t) \\ \mathbf{0}_{m_f \times 12} \end{bmatrix} \mathbf{n}(t) \\ &=: \mathbf{F}(t)\tilde{\mathbf{x}}(t) + \mathbf{G}(t)\mathbf{n}(t) \end{aligned} \quad (11)$$

where  $\mathbf{F}_c(t)$  and  $\mathbf{G}_c(t)$  are the continuous-time error-state transition matrix and noise Jacobian matrix for IMU state, respectively.  $\mathbf{n}(t) = [\mathbf{n}_g^\top \ \mathbf{n}_{wg}^\top \ \mathbf{n}_a^\top \ \mathbf{n}_{wa}^\top]^\top$  is modeled as zero-mean Gaussian noise with autocorrelation  $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^\top(\tau)] = \mathbf{Q}_c\delta(t-\tau)$ . Note that  $\mathbf{n}_g(t)$  and  $\mathbf{n}_a(t)$  are the Gaussian noises contaminating the angular velocity and linear acceleration measurements, respectively. The discrete-time state transition matrix  $\Phi_{(k+1,1)}$  from time  $t_k$  to  $t_{k+1}$  can be derived recursively from  $\dot{\Phi}_{(k+1,k)} = \mathbf{F}(t_k)\Phi_{(k+1,k)}$  with the identity matrix as the initial condition [14]:

$$\Phi_{(k+1,1)} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{m_f \times 3} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{m_f \times 3} \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 & \mathbf{0}_{m_f \times 3} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{m_f \times 3} \\ \Phi_{51} & \Phi_{52} & \Phi_{53} & \Phi_{54} & \mathbf{I}_3 & \mathbf{0}_{m_f \times 3} \\ \mathbf{0}_{3 \times m_f} & \mathbf{I}_{m_f} \end{bmatrix} \quad (12)$$

where  $\Phi_{ij}$  is the  $(i,j)$  block of this matrix. In particular, for pure translation, the expression of  $\Phi_{54}$  can be analytically given by (see [14], [26]):

$$\Phi_{54} = -{}^G_I\mathbf{R} \int_{t_1}^{t_k} \int_{t_1}^{t_s} \mathbf{d}\tau \mathbf{d}t_s = -\frac{1}{2} {}^G_I\mathbf{R} \delta t_k^2 \quad (13)$$

where  $\delta t_k = t_k - t_1$  is the time elapsed from the beginning. With the state transition matrix (12), we can also analytically or numerically compute the discrete-time noise covariance

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \Phi_{(kH,\tau)} \mathbf{G}(\tau) \mathbf{Q}_c \mathbf{G}^\top(\tau) \Phi_{(kH,\tau)} \mathbf{d}\tau. \quad (14)$$

##### B. Point Measurements

Point measurements from different exteroceptive sensors (such as monocular/stereo camera, acoustic sonar, and LiDAR) in the aided INS can be generally modeled as range and/or bearing observations which are functions of the relative position of the point feature expressed in the sensor frame  ${}^C\mathbf{p}_f = [{}^C\mathbf{x}_f \ {}^C\mathbf{y}_f \ {}^C\mathbf{z}_f]^\top$  (see [25])

$$\mathbf{z}_p = \underbrace{\begin{bmatrix} \lambda_r & \mathbf{0}_{1 \times 2} \\ \mathbf{0}_{2 \times 1} & \lambda_b \mathbf{I}_2 \end{bmatrix}}_{\Lambda} \begin{bmatrix} z^{(r)} \\ \mathbf{z}^{(b)} \end{bmatrix} = \mathbf{\Lambda} \begin{bmatrix} \sqrt{{}^C\mathbf{p}_f^\top {}^C\mathbf{p}_f} + n^{(r)} \\ \mathbf{h}_b({}^C\mathbf{p}_f, \mathbf{n}^{(b)}) \end{bmatrix} \quad (15)$$

$${}^C\mathbf{p}_f = {}^C_I\mathbf{R}_G^I \mathbf{R}({}^G\mathbf{p}_f - {}^G\mathbf{p}_I) + {}^C\mathbf{p}_I \quad (16)$$

where  $z^{(r)}$  and  $\mathbf{z}^{(b)}$  represent generic range and bearing measurements, respectively.  $\mathbf{h}_b(\cdot)$  is a generic bearing measurement function whose actual form depends on the particular sensor used. For example, such bearing model in the case of monocular camera is provided in Appendix B, and more comprehensive cases can be found in our companion technical report [69].  $\Lambda$  denotes the measurement selection matrix with scalars  $\lambda_r$  and  $\lambda_b$ ; for example, if  $\lambda_b = 1$  and  $\lambda_r = 1$ , then  $\mathbf{z}_p$  contains both range and bearing measurements. In (15),  $n^{(r)}$  and  $\mathbf{n}^{(b)}$  are zero-mean Gaussian noises<sup>2</sup> (inferred from sensor raw data) for the range and bearing measurements. We then linearize these measurements about the current state estimate

$$\begin{aligned}\tilde{\mathbf{z}}_p &\simeq \Lambda (\mathbf{H}_I \tilde{\mathbf{x}} + \mathbf{H}_n \mathbf{n}) =: \Lambda \begin{bmatrix} \mathbf{H}_r \mathbf{H}_f \tilde{\mathbf{x}} + n^{(r)} \\ \mathbf{H}_b \mathbf{H}_f \tilde{\mathbf{x}} + \mathbf{H}_n \mathbf{n}^{(b)} \end{bmatrix} \\ &=: \underbrace{\Lambda \begin{bmatrix} \mathbf{H}_r \\ \mathbf{H}_b \end{bmatrix}}_{\mathbf{H}_{proj}} \mathbf{H}_f \tilde{\mathbf{x}} + \Lambda \begin{bmatrix} n^{(r)} \\ \mathbf{H}_n \mathbf{n}^{(b)} \end{bmatrix}\end{aligned}\quad (17)$$

where we have

$$\mathbf{H}_r = \frac{\partial \tilde{\mathbf{z}}^{(r)}}{\partial {}^C \tilde{\mathbf{p}}_f}, \mathbf{H}_b = \frac{\partial \tilde{\mathbf{z}}^{(b)}}{\partial {}^C \tilde{\mathbf{p}}_f}, \mathbf{H}_f = \frac{\partial {}^C \tilde{\mathbf{p}}_f}{\partial \tilde{\mathbf{x}}}, \mathbf{H}_n = \frac{\partial \tilde{\mathbf{z}}^{(b)}}{\partial \mathbf{n}^{(b)}}.$$

Clearly, depending on the selection matrix  $\Lambda$ ,  $\mathbf{H}_{proj}$  may be the range-only measurement Jacobian  $\mathbf{H}_r$  ( $\lambda_r = 1$  and  $\lambda_b = 0$ ), bearing-only measurements Jacobian  $\mathbf{H}_b$  ( $\lambda_r = 0$  and  $\lambda_b = 1$ ), or both. Interested readers can refer to [69] for detailed derivations of these Jacobians when using different aiding sensors.

### C. Line Measurements

The visual line measurement is given by the distances from two endpoints  $\mathbf{x}_s := [u_1, v_1, 1]^\top$  and  $\mathbf{x}_e := [u_2, v_2, 1]^\top$  of the detected line segment to the line in the image denoted by  $\mathbf{l}'$  which is the projection of the 3-D line  ${}^C \mathbf{L}$  in Plücker coordinates (also see our prior work [68])

$$\mathbf{z}_l = \begin{bmatrix} e_1 & e_2 \\ \ln & \ln \end{bmatrix}^\top, {}^C \mathbf{L} = \begin{bmatrix} {}^C d_l {}^C \mathbf{n}_e \\ {}^C \mathbf{v}_e \end{bmatrix}\quad (18)$$

$$\mathbf{l}' = [\mathbf{K} \ 0_3]^T {}^C \mathbf{L} = [l_1 \ l_2 \ l_3]^\top\quad (19)$$

$$\mathbf{K} = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_1 & 0 \\ -f_2 c_1 & -f_1 c_2 & f_1 f_2 \end{bmatrix}\quad (20)$$

$${}^C \mathbf{L} = \begin{bmatrix} {}^C \mathbf{R} & [{}^C \mathbf{p}_I] {}^C \mathbf{R} \\ 0_3 & {}^C \mathbf{R} \end{bmatrix} \begin{bmatrix} {}^I \mathbf{R} & -{}^I \mathbf{G} \mathbf{R} | {}^G \mathbf{p}_I \\ 0_3 & {}^I \mathbf{R} \end{bmatrix} {}^G \mathbf{L}\quad (21)$$

where  $e_1 = \mathbf{x}_s^\top \mathbf{l}'$ ,  $e_2 = \mathbf{x}_e^\top \mathbf{l}'$ ,  $l_n = \sqrt{l_1^2 + l_2^2}$ ,  $\mathbf{K}$  is the projection matrix for line (not point) features (see [69]), with  $f_1$ ,  $f_2$ ,  $c_1$ , and  $c_2$  as the camera intrinsic parameters. The relationship (21) is derived based on the geometry of the two points on the

<sup>2</sup>Throughout the article,  $\mathbf{n}$  with superscript (e.g.,  $\mathbf{n}^{(\pi)}$ ) denotes the noise that follows a zero-mean Gaussian distribution, while  $\mathbf{n}$  with subscript (e.g.,  $\mathbf{n}_\pi$ ) represents the normal direction of a plane.

line,  ${}^G \mathbf{p}_{fi} = {}^G \mathbf{p}_C + {}^G \mathbf{R} {}^C \mathbf{p}_{fi}$  ( $i = 1, 2$ ) [see (3)]. Moreover, the measurement Jacobian can be computed as follows [69]:

$$\mathbf{H}_I = \frac{\partial \tilde{\mathbf{z}}_l}{\partial \mathbf{l}'} \frac{\partial \mathbf{l}'}{\partial \tilde{\mathbf{x}}} =: \mathbf{H}_l \mathbf{H}_f\quad (22)$$

$$\mathbf{H}_l = \frac{1}{l_n} \begin{bmatrix} u_1 - l_1 e_1 / l_n^2 & v_1 - l_2 e_1 / l_n^2 & 1 \\ u_2 - l_1 e_2 / l_n^2 & v_2 - l_2 e_2 / l_n^2 & 1 \end{bmatrix}.\quad (23)$$

### D. Plane Measurements

Plane features (e.g., from point clouds) can be written as

$$\mathbf{z}_\pi = {}^C d_\pi {}^C \mathbf{n}_\pi + \mathbf{n}^{(\pi)} =: {}^C \mathbf{p}_\pi + \mathbf{n}^{(\pi)}\quad (24)$$

where  ${}^C \mathbf{p}_\pi$  represents the plane in the sensor's local frame and  $\mathbf{n}^{(\pi)}$  represents the plane measurement noise. To compute the corresponding measurement Jacobians, the plane parameters in the global frame can be transformed to the local frame as

$$\begin{bmatrix} {}^C \mathbf{n}_\pi \\ {}^C d_\pi \end{bmatrix} = \begin{bmatrix} {}^C \mathbf{R} & 0_{3 \times 1} \\ {}^C \mathbf{p}_I^\top {}^C \mathbf{R} & 1 \end{bmatrix} \begin{bmatrix} {}^I \mathbf{R} & 0_{3 \times 1} \\ -{}^G \mathbf{p}_I^\top & 1 \end{bmatrix} \begin{bmatrix} {}^G \mathbf{n}_\pi \\ {}^G d_\pi \end{bmatrix}.\quad (25)$$

Therefore, the measurement Jacobians are given by [69]

$$\mathbf{H}_I = \frac{\partial \tilde{\mathbf{z}}}{\partial {}^C \tilde{\mathbf{p}}_\pi} \frac{\partial {}^C \tilde{\mathbf{p}}_\pi}{\partial \tilde{\mathbf{x}}} =: \mathbf{H}_\pi \mathbf{H}_f\quad (26)$$

$$\mathbf{H}_\pi = [{}^C \hat{d}_\pi \mathbf{I}_3 \ {}^C \hat{\mathbf{n}}_\pi].\quad (27)$$

### E. Observability Analysis

The key reasons for observability analysis include the following: (i) it provides a deep insight about the system's geometric properties [14], [23], [24] and determines the minimum measurement modalities or state parameters needed to initialize the estimator, (ii) it can be used to identify degenerate motions [25], [26] which cause additional unobservable directions and should be avoided in real applications whenever possible, and (iii) the observability-constrained (OC)-based methodology as in OC-EKF [23] and OC-VINS [14], that enforce the correct observability properties, can be adopted to improve consistency.

Observability analysis for the linearized aided INS can be performed in a similar way as in [14], [19]. In particular, the observability matrix  $\mathbf{M}(\mathbf{x})$  can be constructed as

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{H}_{I_1} \Phi_{(1,1)} \\ \mathbf{H}_{I_2} \Phi_{(2,1)} \\ \vdots \\ \mathbf{H}_{I_k} \Phi_{(k,1)} \end{bmatrix}\quad (28)$$

where  $\mathbf{H}_{I_k}$  is the measurement Jacobian at time step  $k$ .  $\Phi_{(k,1)}$  can be computed based on (12). The unobservable directions span the right null space of this observability matrix, that is,  $\mathbf{M}(\mathbf{x}) \mathbf{N} = \mathbf{0}$ , where  $\mathbf{N}$  represents the right null space.

## V. OBSERVABILITY ANALYSIS OF AIDED INS WITH DIFFERENT FEATURES

In this section, we first perform observability analysis for the linearized systems of aided INS with one single *homogeneous* geometric feature (point, line, or plane) and then extend to the case of *heterogeneous* features. Note that for concise presentation of the article, we include the generalization to multiple point, line, and plane features instead in the companion technical report [69].

### A. Aided INS With One Point Feature

We first consider the aided INS with one point feature and the state vector can be written as

$$\mathbf{x} = [\mathbf{x}_I^\top \ {}^G\mathbf{p}_f^\top]^\top. \quad (29)$$

We conduct the observability analysis in a similar way as in [14], [19]. In particular, as the unobservable directions of this aided INS span the right null space of  $\mathbf{M}(\mathbf{x})$  (28), we compute the measurement Jacobians  $\mathbf{H}_{I_k}^{(p)}$  based on (15) as follows [see (17)]:

$$\begin{aligned} \mathbf{H}_{I_k}^{(p)} &= \Lambda \left[ \begin{array}{c} \mathbf{H}_{r,k} \\ \mathbf{H}_{b,k} \end{array} \right] \underbrace{{}^C\hat{\mathbf{R}} \left[ \begin{array}{ccc} {}^I\hat{\mathbf{p}}_f & \mathbf{0}_{3 \times 9} & -{}^I\hat{\mathbf{R}} {}^G\hat{\mathbf{R}} \end{array} \right]}_{\mathbf{H}_{f,k}} \\ &= \mathbf{H}_{proj,k} {}^C\hat{\mathbf{R}} {}^I\hat{\mathbf{R}} \left[ \begin{array}{cccccc} \mathbf{H}_{p1} & \mathbf{0}_{3 \times 9} & -\mathbf{I}_3 & \mathbf{I}_3 \end{array} \right] \end{aligned} \quad (30)$$

where we have used (16) and (17) as well as the following matrix:

$$\mathbf{H}_{p1} = \left[ \begin{array}{c} ({ }^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}) \end{array} \right] {}^I\hat{\mathbf{R}}^\top. \quad (31)$$

Specifically, for each block row of  $\mathbf{M}(\mathbf{x})$  [see (28)], we have

$$\begin{aligned} \mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} &= \mathbf{H}_{proj,k} {}^C\hat{\mathbf{R}} {}^I\hat{\mathbf{R}}^\top \\ &\quad \times \left[ \begin{array}{cccccc} \boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_3 & \boldsymbol{\Gamma}_4 & -\mathbf{I}_3 & \mathbf{I}_3 \end{array} \right] \end{aligned} \quad (32)$$

where

$$\begin{aligned} \boldsymbol{\Gamma}_1 &= \left[ \begin{array}{c} {}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_1} - {}^G\hat{\mathbf{v}}_{I_1} \delta t_k - \frac{1}{2} {}^G\mathbf{g} (\delta t_k)^2 \end{array} \right] {}^I\hat{\mathbf{R}} \\ \boldsymbol{\Gamma}_2 &= \left[ \begin{array}{c} ({ }^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_{I_k}) \end{array} \right] {}^I\hat{\mathbf{R}}^\top \Phi_{12} - \Phi_{52} \\ \boldsymbol{\Gamma}_3 &= -\mathbf{I}_3 \delta t_k, \quad \boldsymbol{\Gamma}_4 = -\Phi_{54} \end{aligned} \quad (33)$$

where  $g = \|{}^G\mathbf{g}\|$  and  ${}^G\mathbf{g} = [0, 0, -g]^\top$ . Note that for the analysis purpose, we assume that when computing different Jacobians, the linearization points for the same state variables remain the same [19]. By inspection, it is not difficult to see that the null space of  $\mathbf{M}(\mathbf{x})$  in this case is given by

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_g & \mathbf{0}_{12 \times 3} \\ -[{}^G\hat{\mathbf{p}}_{I_1}] {}^G\mathbf{g} & \mathbf{I}_3 \\ -[{}^G\hat{\mathbf{p}}_f] {}^G\mathbf{g} & \mathbf{I}_3 \end{bmatrix} =: [\mathbf{N}_r \ \mathbf{N}_p] \quad (34)$$

where  $\mathbf{N}_g$  is defined by

$$\mathbf{N}_g = \left[ \begin{array}{cc} \left( {}^I\hat{\mathbf{R}} {}^G\mathbf{g} \right)^\top \mathbf{0}_{1 \times 3} & -\left( [{}^G\hat{\mathbf{v}}_{I_1}] {}^G\mathbf{g} \right)^\top \mathbf{0}_{1 \times 3} \end{array} \right]^\top. \quad (35)$$

It is interesting to notice that in (34),  $\mathbf{N}_p$  corresponds to the sensor's global translation, while  $\mathbf{N}_r$  relates to the global rotation around the gravity direction. We thus see that the system has at least four unobservable directions ( $\mathbf{N}_p$  and  $\mathbf{N}_r$ ). Moreover, in analogy to [8], [15], [19], we have further performed the nonlinear observability analysis based on Lie derivatives [56] for the continuous-time nonlinear aided INS, which is summarized as follows:

*Lemma 1:* The continuous-time nonlinear aided INS with point features (detected from generic range and/or bearing measurements) has four unobservable directions. ■

*Proof:* See [69].

### B. Aided INS With One Line Feature

When navigating in structured environments, line features are ubiquitous and should be exploited for aided INS to improve system performances. In the following, we perform observability analysis for the aided INS with one line feature to provide insights for building consistent estimators. For conciseness of presentation, the quaternion form for line (Model 5 for line in Section III-B) is used and the state vector containing one line feature can be written as

$$\mathbf{x} = [\mathbf{x}_I^\top \ {}^G\bar{q}_l^\top \ {}^Gd_l]^\top. \quad (36)$$

With the line measurements (18), the measurement Jacobian is computed by (see (22) and [69])

$$\mathbf{H}_{I_k}^{(l)} = \mathbf{H}_{l,k} \underbrace{\frac{\partial \tilde{\mathbf{l}}'}{\partial {}^C\tilde{\mathbf{L}}} \frac{\partial {}^C\tilde{\mathbf{L}}}{\partial {}^I\tilde{\mathbf{L}}} \left[ \begin{array}{c} \frac{\partial {}^I_k \tilde{\mathbf{L}}}{\partial \tilde{\mathbf{x}}_I} \\ \frac{\partial {}^I_k \tilde{\mathbf{L}}}{\partial [\delta \theta_l^\top \ {}^G\tilde{d}_l]^\top} \end{array} \right]}_{\mathbf{H}_{f,k}}. \quad (37)$$

With this, the block row of the observability matrix  $\mathbf{M}(\mathbf{x})$  (28) at time step  $k$  can be written as

$$\begin{aligned} \mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} &= \mathbf{H}_{l,k} \frac{\partial \tilde{\mathbf{l}}'}{\partial {}^C\tilde{\mathbf{L}}} \frac{\partial {}^C\tilde{\mathbf{L}}}{\partial {}^I\tilde{\mathbf{L}}} \left[ \begin{array}{cc} {}^I_k \hat{\mathbf{R}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{array} \right] \\ &\quad \times \left[ \begin{array}{cccccc} \boldsymbol{\Gamma}_{l11} & \boldsymbol{\Gamma}_{l12} & \boldsymbol{\Gamma}_{l13} & \boldsymbol{\Gamma}_{l14} & \boldsymbol{\Gamma}_{l15} & \boldsymbol{\Gamma}_{l16} & \boldsymbol{\Gamma}_{l17} \\ \boldsymbol{\Gamma}_{l21} & \boldsymbol{\Gamma}_{l22} & \boldsymbol{\Gamma}_{l23} & \boldsymbol{\Gamma}_{l24} & \boldsymbol{\Gamma}_{l25} & \boldsymbol{\Gamma}_{l26} & \boldsymbol{\Gamma}_{l27} \end{array} \right] \end{aligned} \quad (38)$$

where  $\boldsymbol{\Gamma}_{lij}, i \in \{1, 2\}, j \in \{1, \dots, 7\}$  can be found in our companion technical report [69]. Therefore, we have the following result.

*Lemma 2:* The aided INS with a single line feature has at least five unobservable directions denoted by  $\mathbf{N}_l$

$$\mathbf{N}_l = \begin{bmatrix} \mathbf{N}_g & \mathbf{0}_{12 \times 3} & \mathbf{N}_v \\ -[{}^G\hat{\mathbf{p}}_{I_1}] {}^G\mathbf{g} & {}^G\hat{\mathbf{R}}_l & \mathbf{0}_{3 \times 1} \\ -{}^G\mathbf{g} & \frac{1}{G\hat{d}_l} {}^G\hat{\mathbf{v}}_e \mathbf{e}_1^\top & \mathbf{0}_{3 \times 1} \\ 0 & -\mathbf{e}_3^\top & 0 \end{bmatrix} =: [\mathbf{N}_{l1} \ \mathbf{N}_{l2:5}] \quad (39)$$

where  $\mathbf{N}_v$  and  ${}^G\hat{\mathbf{R}}_l$  are defined by

$$\mathbf{N}_v = [\mathbf{0}_{1 \times 3} \ \mathbf{0}_{1 \times 3} \ {}^G\hat{\mathbf{v}}_e^\top \ \mathbf{0}_{1 \times 3}]^\top \quad (40)$$

$${}^G\hat{\mathbf{R}}_l = [{}^G\hat{\mathbf{n}}_e \ {}^G\hat{\mathbf{v}}_e \ [{}^G\hat{\mathbf{n}}_e]{}^G\hat{\mathbf{v}}_e]. \quad (41)$$

*Proof:* See [69].  $\blacksquare$

It is not difficult to see that  $\mathbf{N}_{l1}$  relates to the sensor rotation around the gravitational direction,  $\mathbf{N}_{l2:4}$  associates with the sensor's global translation, and  $\mathbf{N}_{l5}$  corresponds to the sensor's motion along the line direction. Note also that the above analysis is based on the projective line measurement model (18). Additionally, in our companion report [69], we have also considered the direct line measurement model, for example, by extracting lines from point clouds, and show that the same unobservable subspace  $\mathbf{N}_l$  (39) holds.

We want to point out again that Lemma 2 holds even with multiple parallel lines in the state vector; if there are nonparallel lines, the unobservable direction  $\mathbf{N}_{l5}$  would not hold and there would remain four unobservable directions with  $\mathbf{N}_{l1:4}$  for the system.

### C. Aided INS With One Plane Feature

Now we perform observability analysis of the aided INS with plane features which are also important geometric features commonly seen in structured environments. In particular, to simplify the presentation, our analysis is based on the CP parameterization of one plane feature in the state vector. The state vector is given by

$$\mathbf{x} = [\mathbf{x}_I^\top \ {}^G\mathbf{p}_\pi^\top]^\top. \quad (42)$$

Based on the plane measurements (24), we can compute the Jacobians with respect to  $\tilde{\mathbf{x}}$  (see [69]) as

$$\mathbf{H}_{I_k}^{(\pi)} = \mathbf{H}_{\pi,k} \underbrace{\frac{\partial [{}^C\tilde{\mathbf{n}}_\pi^\top \ {}^C\tilde{d}_\pi]^\top}{\partial [{}^I\tilde{\mathbf{n}}_\pi^\top \ {}^I\tilde{d}_\pi]^\top}}_{\mathbf{H}_{f,k}} \left[ \begin{array}{c} \frac{\partial [{}^I\tilde{\mathbf{n}}_\pi]^\top}{\partial \tilde{\mathbf{x}}_I} \\ \frac{\partial [{}^I\tilde{d}_\pi]^\top}{\partial {}^G\tilde{\mathbf{p}}_\pi} \end{array} \right]. \quad (43)$$

The block row of the observability matrix is computed by

$$\begin{aligned} \mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} &= \mathbf{H}_{\pi,k} \frac{\partial [{}^C\tilde{\mathbf{n}}_\pi^\top \ {}^C\tilde{d}_\pi]^\top}{\partial [{}^I\tilde{\mathbf{n}}_\pi^\top \ {}^I\tilde{d}_\pi]^\top} \left[ \begin{array}{cc} {}^I_k\hat{\mathbf{R}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{array} \right] \\ &\quad \times \left[ \begin{array}{cccccc} \Gamma_{\pi 11} & \Gamma_{\pi 12} & \Gamma_{\pi 13} & \Gamma_{\pi 14} & \Gamma_{\pi 15} & \Gamma_{\pi 16} \\ \Gamma_{\pi 21} & \Gamma_{\pi 22} & \Gamma_{\pi 23} & \Gamma_{\pi 24} & \Gamma_{\pi 25} & \Gamma_{\pi 26} \end{array} \right] \end{aligned} \quad (44)$$

where  $\Gamma_{\pi ij}, i \in \{1, 2\}, j \in \{1, \dots, 6\}$  can be found in our companion technical report [69]. With that, we have the following result.

*Lemma 3:* The aided INS with a single plane feature has at least seven unobservable directions

$$\begin{aligned} \mathbf{N}_\pi &= \left[ \begin{array}{ccc} \mathbf{N}_g & \mathbf{0}_{12 \times 3} & \mathbf{N}_{123} \\ -[{}^G\hat{\mathbf{p}}_{I_1}]{}^G\mathbf{g} & {}^G\hat{\mathbf{R}}_\pi & \mathbf{0}_3 \\ -[{}^G\hat{\mathbf{p}}_\pi]{}^G\mathbf{g} & {}^G\hat{\mathbf{n}}_\pi \mathbf{e}_3^\top & \mathbf{0}_3 \end{array} \right] \\ &=: [\mathbf{N}_{\pi 1} \ \mathbf{N}_{\pi 2:4} \ \mathbf{N}_{\pi 5:7}]. \end{aligned} \quad (45)$$

In the above expression, given  ${}^G\hat{\mathbf{n}}_1^\perp$  and  ${}^G\hat{\mathbf{n}}_2^\perp$  that are the unit vectors orthonormal to each other and perpendicular to  ${}^G\hat{\mathbf{n}}_\pi$ , we have defined  $\mathbf{N}_{123}$  and the plane orientation  ${}^G\hat{\mathbf{R}}_\pi$  as follows:

$$\mathbf{N}_{123} = \begin{bmatrix} \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & {}^I_1\hat{\mathbf{R}} {}^G\hat{\mathbf{n}}_\pi \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ {}^G\hat{\mathbf{n}}_1^\perp & {}^G\hat{\mathbf{n}}_2^\perp & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (46)$$

$${}^G\hat{\mathbf{R}}_\pi = [{}^G\hat{\mathbf{n}}_1^\perp \ {}^G\hat{\mathbf{n}}_2^\perp \ {}^G\hat{\mathbf{n}}_\pi]. \quad (47)$$

*Proof:* See [69].  $\blacksquare$

Note that as compared to [38] where it was shown that the VINS with bearing measurements to planes has 12 unobservable directions, we analytically show that, given the direct plane measurements (24), the aided INS with a single plane feature has at least seven unobservable directions: (i)  $\mathbf{N}_{\pi 1}$  that relates to the rotation around the gravity, (ii)  $\mathbf{N}_{\pi 2:4}$  that associates with the position of the sensor platform, (iii)  $\mathbf{N}_{\pi 5:6}$  that corresponds to the 2-D translation on a plane parallel to the feature plane, and (iv)  $\mathbf{N}_{\pi 7}$  that corresponds to the 2-D rotation around the plane normal  ${}^G\hat{\mathbf{n}}_\pi$ .

Note also that Lemma 3 can be easily extended to the case of multiple planes, in which we have the following findings:

- 1) If all planes are parallel to each other, the unobservable directions  $\mathbf{N}_{\pi 5:7}$  will still hold.
- 2) If all the planes have parallel intersecting lines, the dimension of the unobservable subspace becomes 5 and the extra unobservable direction is related to the sensor motion along these parallel lines. This situation is similar to the case of a single line feature in the state vector.
- 3) If planes have nonparallel intersection lines, the system will still have four unobservable directions as  $\mathbf{N}_{\pi 1:4}$ .

### D. Aided INS With Heterogeneous Features

We now examine the observability properties for the case of one feature of each type included in the state vector

$${}^G\mathbf{x}_f = [{}^G\mathbf{p}_f^\top \ {}^G\bar{q}_l^\top \ {}^Gd_l \ {}^G\mathbf{p}_\pi^\top]^\top. \quad (48)$$

The measurement model becomes [see (15), (18) and (24)]

$$\mathbf{z}_{pl\pi} = [\mathbf{z}_p^\top \ \mathbf{z}_l^\top \ \mathbf{z}_\pi^\top]^\top. \quad (49)$$

The measurement Jacobians can be computed based on (30), (22), and (44). Then, the  $k$ -th block row of the observability

matrix  $\mathbf{M}(\mathbf{x})$  is

$$\mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} = \begin{bmatrix} \mathbf{H}_P & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_{\Pi} \end{bmatrix}$$

$$\times \begin{bmatrix} \Gamma_1 & \Gamma_2 & \Gamma_3 & \Gamma_4 & -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_3 \\ \Gamma_{l11} & \Gamma_{l12} & \Gamma_{l13} & \Gamma_{l14} & \Gamma_{l15} & \mathbf{0}_3 & \Gamma_{l16} & \Gamma_{l17} & \mathbf{0}_3 \\ \Gamma_{l21} & \Gamma_{l22} & \Gamma_{l23} & \Gamma_{l24} & \Gamma_{l25} & \mathbf{0}_3 & \Gamma_{l26} & \Gamma_{l27} & \mathbf{0}_3 \\ \Gamma_{\pi11} & \Gamma_{\pi12} & \Gamma_{\pi13} & \Gamma_{\pi14} & \Gamma_{\pi15} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \Gamma_{\pi16} \\ \Gamma_{\pi21} & \Gamma_{\pi22} & \Gamma_{\pi23} & \Gamma_{\pi24} & \Gamma_{\pi25} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & \Gamma_{\pi26} \end{bmatrix} \quad (50)$$

$$\mathbf{H}_P = \mathbf{H}_{proj,k} {}^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} \quad (51)$$

$$\mathbf{H}_L = \mathbf{H}_{l,k} \frac{\partial \tilde{\mathbf{l}}'}{\partial {}^C \tilde{\mathbf{L}}} \frac{\partial {}^C \tilde{\mathbf{L}}}{\partial {}^I \tilde{\mathbf{L}}} \begin{bmatrix} {}^{I_k} \hat{\mathbf{R}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad (52)$$

$$\mathbf{H}_{\Pi} = \mathbf{H}_{\pi,k} \frac{\partial [{}^C \tilde{\mathbf{n}}_{\pi}^T \ {}^C \tilde{d}_{\pi}]^T}{\partial [{}^I \tilde{\mathbf{n}}_{\pi}^T \ {}^I \tilde{d}_{\pi}]^T} \begin{bmatrix} {}^{I_k} \hat{\mathbf{R}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (53)$$

The observability properties of this aided INS are given by Lemma 4.

*Lemma 4:* The aided INS with one point, one line, and one plane feature in the state vector has four unobservable directions

$$\mathbf{N}_{pl\pi} = \begin{bmatrix} \mathbf{N}_g & \mathbf{0}_{12 \times 3} \\ -[{}^G \hat{\mathbf{p}}_{I_1}]^G \mathbf{g} & \mathbf{I}_3 \\ -[{}^G \hat{\mathbf{p}}_{\mathbf{f}}]^G \mathbf{g} & \mathbf{I}_3 \\ -{}^G \mathbf{g} & \frac{1}{G \hat{d}_l} {}^G \hat{\mathbf{v}}_{\mathbf{e}} \mathbf{e}_1^T {}^G \hat{\mathbf{R}}_l^T \\ 0 & -\mathbf{e}_3^T {}^G \hat{\mathbf{R}}_l^T \\ -[{}^G \hat{\mathbf{p}}_{\pi}]^G \mathbf{g} & {}^G \hat{\mathbf{n}}_{\pi} \mathbf{e}_3^T {}^G \hat{\mathbf{R}}_{\pi}^T \end{bmatrix}$$

$$=: [\mathbf{N}_{pl\pi 1} \ \mathbf{N}_{pl\pi 2:4}]. \quad (54)$$

*Proof:* See Appendix C.  $\blacksquare$

## VI. OBSERVABILITY ANALYSIS FOR AIDED INS WITH GLOBAL MEASUREMENTS

Aided INS may also have access to (partially) global measurements provided by, for example, GPS receivers, sun/star sensors, barometers, and compasses. Intuitively, such measurements would alter the system observability properties, even if only partial (not full 6-DoF pose) information is available. In this section, we systematically examine the impacts of such measurements on the system observability.

### A. Global Orientation Measurements

We consider the case where the aided INS has access to global orientation measurements; for example, provided by a sun sensor, or a magnetic compass, or by detecting a plane with

known orientation [38], [59]:  $\mathbf{z}^{(n)} = {}^C \mathbf{N}_n = {}_I^C \mathbf{R}_G^I \mathbf{R}^G \mathbf{N}_n$ . In this case, the Jacobian and the block row of the observability matrix can be computed as

$$\mathbf{H}_{I_k} \Phi_{(k,1)} = \begin{bmatrix} \mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} \\ \mathbf{H}_{I_k}^{(n)} \Phi_{(k,1)} \end{bmatrix} \quad (55)$$

where  $\mathbf{H}_{I_k}^{(n)}$  is the orientation measurement Jacobian with respect to (48), yielding

$$\mathbf{H}_{I_k}^{(n)} \Phi_{(k,1)} = {}_I^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} \left[ [{}^G \mathbf{N}_n]_I^G \hat{\mathbf{R}} \ \Gamma_5 \ \mathbf{0}_{3 \times 19} \right] \quad (56)$$

where  $\Gamma_5 = [{}^G \mathbf{N}_n]_G^I \hat{\mathbf{R}}^T \Phi_{12}$ . If  ${}^G \mathbf{N}_n$  is not parallel to  ${}^G \mathbf{g}$ , i.e.,  $[{}^G \mathbf{N}]^G \mathbf{g} \neq \mathbf{0}$ , the rotation around the gravity direction becomes observable, and the unobservable directions are

$$\mathbf{N}_n = \begin{bmatrix} \mathbf{0}_{12 \times 3} \\ \mathbf{I}_3 \\ \mathbf{I}_3 \\ \frac{1}{G \hat{d}_l} {}^G \hat{\mathbf{v}}_{\mathbf{e}} \mathbf{e}_1^T {}^G \hat{\mathbf{R}}_l^T \\ -\mathbf{e}_3^T {}^G \hat{\mathbf{R}}_l^T \\ {}^G \hat{\mathbf{n}}_{\pi} \mathbf{e}_3^T {}^G \hat{\mathbf{R}}_{\pi}^T \end{bmatrix}. \quad (57)$$

### B. Global Position Measurements

In addition to the point, line, and plane measurements, we consider the case where global position measurements are also available from, for example, a GPS receiver or a barometer. In the following, we use such a global measurement individually along the  $x$ -,  $y$ -, and  $z$ -axis.

If several measurements on the global  $x$  direction are presented, the sensor's translation along  $x$  direction (both the  $x$  position and velocity) will be known, and therefore, there will be no ambiguity about the global yaw. The additional global  $x$ -axis measurement can be given by  $z^{(x)} = \mathbf{e}_1^T {}^G \mathbf{p}_I$ . The measurement Jacobians and the block row of observability matrix can be computed as [see (50)]

$$\mathbf{H}_{I_k} \Phi_{(k,1)} = \begin{bmatrix} \mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} \\ \mathbf{H}_{I_k}^{(x)} \Phi_{(k,1)} \end{bmatrix} \quad (58)$$

where  $\mathbf{H}_{I_k}^{(x)}$  is the global  $x$  measurement Jacobian, yielding

$$\mathbf{H}_{I_k}^{(x)} \Phi_{(k,1)} = [\mathbf{0}_{1 \times 12} \ \mathbf{e}_1^T \ \mathbf{0}_{1 \times 3} \ \mathbf{0}_{1 \times 4} \ \mathbf{0}_{1 \times 3}]. \quad (59)$$

We can show that the unobservable subspace becomes

$$\mathbf{N}_x = \begin{bmatrix} \mathbf{0}_{12 \times 2} \\ \mathbf{A}_x \\ \mathbf{A}_x \\ \frac{1}{G \hat{d}_l} {}^G \hat{\mathbf{v}}_{\mathbf{e}} \mathbf{e}_1^T {}^G \hat{\mathbf{R}}_l^T \mathbf{A}_x \\ -\mathbf{e}_3^T {}^G \hat{\mathbf{R}}_l^T \mathbf{A}_x \\ {}^G \hat{\mathbf{n}}_{\pi} \mathbf{e}_3^T {}^G \hat{\mathbf{R}}_{\pi}^T \mathbf{A}_x \end{bmatrix} \quad (60)$$

where  $\mathbf{A}_x = [\mathbf{0}_{2 \times 1} \quad \mathbf{I}_2]^\top$ . As compared to  $\mathbf{N}$  in (54) without global  $x$  measurements, both the global translation in  $x$  direction and the rotation around the gravity direction become observable. Analogously, if a global  $y$ -axis measurement is available, translation along  $y$  and the rotation around the gravity will become observable [69].

Proceeding similarly, if the global translation in  $z$  direction is directly measured, e.g., by a barometer, we have an additional global  $z$ -axis measurement  $z^{(z)} = \mathbf{e}_3^\top {}^G\mathbf{p}_I$ . In this case, the block row of the observability matrix becomes

$$\mathbf{H}_{I_k} \Phi_{(k,1)} = \begin{bmatrix} \mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} \\ \mathbf{H}_{I_k}^{(z)} \Phi_{(k,1)} \end{bmatrix}. \quad (61)$$

Since  $\mathbf{e}_3$  is parallel to  ${}^G\mathbf{g}$ , we have  $\mathbf{e}_3^\top [{}^G\mathbf{p}_{I_1}] {}^G\mathbf{g} = 0$ . Therefore, the system's unobservable directions become

$$\mathbf{N}_z = \begin{bmatrix} \mathbf{N}_g & \mathbf{0}_{12 \times 2} \\ -[{}^G\hat{\mathbf{p}}_{I_1}] {}^G\mathbf{g} & \mathbf{A}_z \\ -[{}^G\hat{\mathbf{p}}_{\mathbf{f}}] {}^G\mathbf{g} & \mathbf{A}_z \\ -{}^G\mathbf{g} & \frac{1}{G\hat{d}_l} {}^G\hat{\mathbf{v}}_{\mathbf{e}} \mathbf{e}_1^\top {}^G\hat{\mathbf{R}}_l^\top \mathbf{A}_z \\ 0 & -\mathbf{e}_3^\top {}^G\hat{\mathbf{R}}_l^\top \mathbf{A}_z \\ -[{}^G\hat{\mathbf{p}}_\pi] {}^G\mathbf{g} & {}^G\hat{\mathbf{n}}_\pi \mathbf{e}_3^\top {}^G\hat{\mathbf{R}}_\pi^\top \mathbf{A}_z \end{bmatrix} \quad (62)$$

where  $\mathbf{A}_z = [\mathbf{I}_2 \quad \mathbf{0}_{2 \times 1}]^\top$ . Clearly, only translation in  $z$  becomes observable, while different from the previous case of the global  $x$  or  $y$  measurements, the rotation around the gravity direction remains unobservable.

## VII. ANALYSIS OF DEGENERATE MOTIONS

While it has been reported in [24], [26] that pure translation and constant acceleration are degenerate for monocular VINS with point features, in this section, we perform a comprehensive study of degenerate motions for the aided INS with heterogeneous features. It is important to identify these degenerate motions in order to keep estimators healthy. Unlike the previous work [26], we explicitly consider the extrinsic calibration between the aiding sensor and IMU.

In particular, to ease our analysis, we use the range and bearing parameterization (i.e., spherical coordinates) for the point feature:

$$\mathbf{x}_f := \begin{bmatrix} r_f \\ \theta \\ \phi \end{bmatrix} \Rightarrow r_f \mathbf{b}_f = r_f \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \sin \phi \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} =: \mathbf{p}_f. \quad (63)$$

In this case (point features), the block row of the observability matrix can be computed as [see (32)]

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} = \mathbf{H}_{proj,k} {}^C\hat{\mathbf{R}}_G {}^{I_k} \hat{\mathbf{R}} \times [\Gamma_1 \ \Gamma_2 \ \Gamma_3 \ \Gamma_4 \ -\mathbf{I}_3 \ \hat{\mathbf{b}}_f \ {}^G\hat{r}_f \cos \hat{\phi} \hat{\mathbf{b}}_1^\perp \ {}^G\hat{r}_f \hat{\mathbf{b}}_2^\perp] \quad (64)$$

where

$$\hat{\mathbf{b}}_1^\perp = [-\sin \hat{\theta} \ \cos \hat{\theta} \ 0]^\top \quad (65)$$

$$\hat{\mathbf{b}}_2^\perp = [-\cos \hat{\theta} \sin \hat{\phi} \ -\sin \hat{\theta} \sin \hat{\phi} \ \cos \hat{\phi}]^\top. \quad (66)$$

By inspection, the unobservable directions can be found as

$$\mathbf{N}_{rb} = \begin{bmatrix} \mathbf{N}_g & \mathbf{0}_{12 \times 3} \\ -[{}^G\hat{\mathbf{p}}_{I_1}] {}^G\mathbf{g} & \mathbf{I}_3 \\ 0 & \hat{\mathbf{b}}_f^\top \\ g & \frac{1}{G\hat{r}_f \cos \hat{\phi}} (\hat{\mathbf{b}}_1^\perp)^\top \\ 0 & \frac{1}{G\hat{r}_f} (\hat{\mathbf{b}}_2^\perp)^\top \end{bmatrix} =: [\mathbf{N}_{rb,r} \ \mathbf{N}_{rb,p}] \quad (67)$$

where  $\mathbf{N}_{rb,p}$  and  $\mathbf{N}_{rb,r}$  are the unobservable directions associated with the global translation and the global rotation around the gravity direction, which, as expected, agrees with the preceding analysis (34).

### A. Pure Translation

Based on the above analysis of point measurements, we show that given point, line, and plane measurements (49), if the sensor undergoes pure translation, the system gains the following *additional* unobservable directions [by noting that the state vector (9) includes the IMU state, one point in spherical coordinates (63), one line, and one plane]:

$$\mathbf{N}_R = \begin{bmatrix} {}^{I_1} \hat{\mathbf{R}} \\ \mathbf{0}_3 \\ -[{}^G\hat{\mathbf{v}}_{I_1}] \\ {}^{I_1} \hat{\mathbf{R}} [{}^G\mathbf{g}] \\ -[{}^G\hat{\mathbf{p}}_{I_1}] \\ -\Theta \\ -\mathbf{I}_3 \\ \mathbf{0}_{1 \times 3} \\ -[{}^G\mathbf{p}_\pi] \end{bmatrix} \quad (68)$$

where

$$\Theta = \begin{bmatrix} 0 & 0 & 0 \\ \cos \hat{\theta} \tan \hat{\phi} & \sin \hat{\theta} \tan \hat{\phi} & -1 \\ -\sin \hat{\theta} & \cos \hat{\theta} & 0 \end{bmatrix}. \quad (69)$$

Similar to [26], this unobservable direction can be easily verified [see (50)]

$$\mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} \mathbf{N}_R = \begin{bmatrix} \mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_R \\ \mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} \mathbf{N}_R \\ \mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} \mathbf{N}_R \end{bmatrix} = \mathbf{0}. \quad (70)$$

Specifically, using (13) and (33), we can easily verify each block row of (70) as follows:

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_R = \mathbf{H}_{proj,k} {}^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} \left( \Gamma_4 {}^{I_1} \hat{\mathbf{R}} - \frac{1}{2} \delta t_k^2 \mathbf{I}_3 \right) [{}^G \mathbf{g}] \\ = \mathbf{0}$$

$$\mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} \mathbf{N}_R = \mathbf{H}_{l,k} \frac{\partial \tilde{\mathbf{l}}'}{\partial {}^C \tilde{\mathbf{L}}} \frac{\partial {}^C \tilde{\mathbf{L}}}{\partial {}^I \tilde{\mathbf{L}}} \begin{bmatrix} {}^{I_k} \hat{\mathbf{R}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \\ \times \begin{bmatrix} [{}^G \hat{\mathbf{v}}_e] \left( \Phi_{54} {}^{I_1} \hat{\mathbf{R}} + \frac{1}{2} \delta t_k^2 \mathbf{I}_3 \right) [{}^G \mathbf{g}] - \\ [{}^G \hat{\mathbf{v}}_e] [{}^G \hat{\mathbf{p}}_{I_k}] - [{}^G \hat{\mathbf{p}}_{I_k}] [{}^G \hat{\mathbf{v}}_e] + [{}^G \hat{\mathbf{p}}_{I_k}] [{}^G \hat{\mathbf{v}}_e] \\ [{}^G \mathbf{v}_e] - [{}^G \mathbf{v}_e] \end{bmatrix} \\ = \mathbf{0}$$

$$\mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} \mathbf{N}_R = \mathbf{H}_{\pi,k} \frac{\partial [{}^C \tilde{\mathbf{n}}_\pi^\top \ {}^C \tilde{d}_l]^\top}{\partial [{}^I \tilde{\mathbf{n}}_\pi^\top \ {}^I \tilde{d}_l]^\top} \begin{bmatrix} {}^{I_k} \hat{\mathbf{R}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ \times \begin{bmatrix} [{}^G \hat{\mathbf{n}}_\pi] - \frac{1}{G \hat{d}_\pi} (\mathbf{I}_3 - {}^G \hat{\mathbf{n}}_\pi^\top {}^G \hat{\mathbf{n}}_\pi) [{}^G \hat{\mathbf{p}}_\pi] \\ - {}^G \hat{\mathbf{n}}_\pi^\top \left( \Phi_{54} {}^{I_1} \hat{\mathbf{R}} + \frac{1}{2} \delta t_k^2 \mathbf{I}_3 \right) [{}^G \mathbf{g}] \end{bmatrix} = \mathbf{0}$$

where we have also employed the following identities ( $\mathbf{a}$  and  $\mathbf{b}$  are general  $3 \times 1$  vectors):

$$[\mathbf{a} \mathbf{b}] = \mathbf{b} \mathbf{a}^\top - \mathbf{a} \mathbf{b}^\top \quad (71)$$

$$[\mathbf{a}] [\mathbf{b}] = \mathbf{b} \mathbf{a}^\top - \mathbf{a}^\top \mathbf{b} \mathbf{I} \quad (72)$$

$$\Phi_{54} {}^{I_1} \hat{\mathbf{R}} = -\frac{1}{2} \delta t_k^2 \mathbf{I}_3. \quad (73)$$

We see from  $\Theta$  that its first row corresponding to the range of the point feature [see (63)] is filled with zeros, and thus this unobservable direction (70) relates to the bearing of the feature. Note also that the global rotation of the sensor becomes unobservable, while only the global yaw is unobservable for general motions [see (70) and (54)]. It is important to notice that no assumption is made about the type of sensors used, and thus, the aided INS with generic sensors (not including global sensors) with pure translation will all gain additional unobservable directions of  $\mathbf{N}_R$ .

### B. Constant Local Acceleration

As it is not straightforward to have direct plane measurements (24) for INS aided by a monocular camera, to ease our analysis of VINS, from now on we focus on using only point measurement (15) to verify the degenerate motion. In particular, if the camera moves with constant local acceleration, i.e.,  ${}^I \mathbf{a}$  is constant, then the system will have one more unobservable direction given by

$$\mathbf{N}_a = [\mathbf{0}_{1 \times 6} \ {}^G \hat{\mathbf{v}}_{I_1}^\top \ - {}^I \hat{\mathbf{a}}^\top \ {}^G \hat{\mathbf{p}}_{I_1}^\top \ {}^G \hat{r}_f \mathbf{e}_1^\top]^\top. \quad (74)$$

Since a monocular camera provides only bearing measurements,  $\mathbf{H}_{proj,k} = \mathbf{H}_{b,k} = \begin{bmatrix} {}^C \hat{\mathbf{b}}_{\perp 1,k}^\top \\ {}^C \hat{\mathbf{b}}_{\perp 2,k}^\top \end{bmatrix}$ , where  ${}^C \hat{\mathbf{b}}_{\perp i,k}$  ( $i = 1, 2$ ) are orthogonal to  ${}^C \hat{\mathbf{b}}_f$  (see Appendix B). In this case, as shown in [26],  $\Gamma_4 {}^I \mathbf{a} = {}^G \hat{\mathbf{p}}_{I_k} - {}^G \hat{\mathbf{p}}_{I_1} - {}^G \hat{\mathbf{v}}_{I_1} \delta t_k$ . Thus, we have

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_a \\ = \mathbf{H}_{b,k} {}^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} \left( - {}^G \hat{\mathbf{v}}_{I_1} \delta t_k - \Gamma_4 {}^I \hat{\mathbf{a}} - {}^G \hat{\mathbf{p}}_{I_1} + {}^G \hat{r}_f {}^G \hat{\mathbf{b}}_f \right) \\ = \mathbf{H}_{b,k} {}^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} \left( {}^G \hat{\mathbf{p}}_f - {}^G \hat{\mathbf{p}}_{I_k} \right) = \mathbf{H}_{b,k} \left( {}^{C_k} \hat{\mathbf{p}}_f - {}^C \hat{\mathbf{p}}_I \right). \quad (75)$$

In general, (75) will not be zero. However, for a practical aided INS, the translation calibration  ${}^C \hat{\mathbf{p}}_I$  may be much smaller than the feature range  ${}^{C_k} \hat{\mathbf{p}}_f$ , and thus we have

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_a = \mathbf{H}_{b,k} \left( {}^{C_k} \hat{\mathbf{p}}_f - {}^C \hat{\mathbf{p}}_I \right) \simeq \mathbf{H}_{b,k} {}^{C_k} \hat{\mathbf{p}}_f \stackrel{(91)}{=} \mathbf{0}. \quad (76)$$

The above equity holds if  $\| {}^{C_k} \hat{\mathbf{p}}_f \| \gg \| {}^C \hat{\mathbf{p}}_I \|$ . This analysis is different from that of [24], [26] and provides the following two insights:

- 1) The rigid transformation between the monocular camera and IMU does affect the degenerate motion analysis. In particular, the translation part  ${}^C \hat{\mathbf{p}}_I$  makes the scale observable.
- 2) In practice, due to the compact sensor housing ( $\| {}^{C_k} \hat{\mathbf{p}}_f \| \gg \| {}^C \hat{\mathbf{p}}_I \|$ ), constant acceleration may make the system “close to” the unobservable direction  $\mathbf{N}_a$ .

### C. Pure Rotation

If the sensor has only rotational motion, then  ${}^C \hat{\mathbf{p}}_{I_k} = \mathbf{0}_{3 \times 1}$ . For monocular-camera-based point measurements (15), the system will gain the following extra unobservable directions corresponding to the feature scale:

$$\mathbf{N}_s = [\mathbf{0}_{1 \times 15} \ {}^G r_f \mathbf{e}_1^\top]^\top \quad (77)$$

which can be seen as follows [see (75)]:

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_s = \mathbf{H}_{b,k} {}^C \hat{\mathbf{R}}_G^{I_k} \hat{\mathbf{R}} {}^G \hat{\mathbf{p}}_f \\ = \mathbf{H}_{b,k} \left( {}^{C_k} \hat{\mathbf{p}}_f - {}^C \hat{\mathbf{p}}_I \right) \simeq \mathbf{0} \quad (78)$$

which holds only when  $\| {}^{C_k} \hat{\mathbf{p}}_f \| \gg \| {}^C \hat{\mathbf{p}}_I \|$ .

### D. Moving Toward Point Feature

With the point measurements (15), if the camera moves straightly toward a point feature, the system will gain one more unobservable direction related to the point scale (range)

$$\mathbf{N}_1 = [\mathbf{0}_{1 \times 15} \ \mathbf{e}_1^\top]^\top. \quad (79)$$

This degenerate motion indicates that the sensor is moving along the point feature’s bearing direction, that is,  ${}^C \hat{\mathbf{p}}_{I_k} = \alpha {}^G \hat{\mathbf{b}}_f$ , where  $\alpha$  denotes the scale of the sensor’s motion. Then, we can arrive at

$${}^{I_k} \mathbf{p}_f = {}^{I_k} r_f {}^{I_k} \mathbf{b}_f = {}^I \mathbf{R} \left( {}^G r_f - \alpha \right) {}^G \mathbf{b}_f. \quad (80)$$

TABLE II  
SUMMARY OF DEGENERATE MOTIONS FOR AIDED INS

Motion	Sensor	Unobservable
1. Pure translation	General	Orientation ${}^T_G \mathbf{R}$
2. Constant acceleration	Mono cam	System scale
3. Pure rotation	Mono cam	Feature scale
4. Toward the feature	Mono cam	Feature scale

Similar to the case of pure rotation, we can verify the additional unobservable direction  $\mathbf{N}_1$  as follows [see (75)]:

$$\begin{aligned} \mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_1 &= \frac{1}{G \hat{r}_f - \alpha} \mathbf{H}_{b,k} {}^C_I \hat{\mathbf{R}}^{I_k} \hat{\mathbf{p}}_f \\ &= \frac{1}{G \hat{r}_f - \alpha} \mathbf{H}_{b,k} ({}^{C_k} \hat{\mathbf{p}}_f - {}^C \hat{\mathbf{p}}_I) \simeq \mathbf{0}. \end{aligned} \quad (81)$$

Again, the above equity holds only when  $\|{}^{C_k} \hat{\mathbf{p}}_f\| \gg \|{}^C \hat{\mathbf{p}}_I\|$ .

### E. Summary

As summarized in Table II, pure translation is degenerate for all sensor types (without global measurements), causing the system orientation unobservable, without any assumption of the rigid transformation between the aiding sensor and IMU. The other three degenerate motions will cause the scale to be unobservable for the case of monocular camera (i.e., bearing-only measurements). However, constant acceleration will cause the whole system (e.g., position, velocity, acceleration bias, and features) scale to be unobservable while pure rotation and moving toward a feature will only make the feature scale unobservable. We also want to stress that these three degenerate motions can only be justified if  $\|{}^{C_k} \hat{\mathbf{p}}_f\| \gg \|{}^C \hat{\mathbf{p}}_I\|$  (which may often hold in practice).

## VIII. NUMERICAL VALIDATIONS

We have implemented both MSCKF-based VIO and VI-SLAM with different features to validate our analysis (see [2], [72]). In particular, the state vector of the VI-SLAM contains

$$\mathbf{x} = [\mathbf{x}_I^\top \ \mathbf{x}_c^\top \ {}^G \mathbf{x}_f]^\top \quad (82)$$

$$\mathbf{x}_c = [{}^I_G \bar{q}^\top \ {}^G \mathbf{p}_{I_1}^\top \ \cdots \ {}^I_m \bar{q}^\top \ {}^G \mathbf{p}_{I_m}^\top]^\top \quad (83)$$

where  $\mathbf{x}_c$  represents a fixed-size sliding window consisting of  $m$  cloned IMU poses. The system propagation is similar to [67]. For feature measurements with respect to a feature within a sliding window, we can get

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_c, {}^G \mathbf{x}_f) + \mathbf{n} \quad (84)$$

$$\Rightarrow \tilde{\mathbf{z}} \simeq \mathbf{H}_{\mathbf{x}_c} \tilde{\mathbf{x}}_c + \mathbf{H}_{\mathbf{x}_f} {}^G \tilde{\mathbf{x}}_f + \mathbf{n} \quad (85)$$

$$\Rightarrow \mathbf{Q}_e^\top \tilde{\mathbf{z}} = \mathbf{Q}_e^\top \mathbf{H}_{\mathbf{x}_c} \tilde{\mathbf{x}}_c + \mathbf{Q}_e^\top \mathbf{H}_{\mathbf{x}_f} {}^G \tilde{\mathbf{x}}_f + \mathbf{Q}_e^\top \mathbf{n} \quad (86)$$

$$\Rightarrow \mathbf{Q}_n^\top \tilde{\mathbf{z}} = \mathbf{Q}_n^\top \mathbf{H}_{\mathbf{x}_c} \tilde{\mathbf{x}}_c + \mathbf{Q}_n^\top \mathbf{n} \quad (87)$$

where  $\mathbf{H}_{\mathbf{x}_c}$  and  $\mathbf{H}_{\mathbf{x}_f}$  are the Jacobians with respect to the state and the feature, respectively. Note that  $\mathbf{H}_{\mathbf{x}_f} = [\mathbf{Q}_e \ \mathbf{Q}_n] \begin{bmatrix} \mathbf{R}_\Delta \\ \mathbf{0} \end{bmatrix}$ . If this feature is already in the state vector, we can use (86) and (87) for standard EKF update. If this feature is observed

TABLE III  
DEGENERATE MOTIONS FOR TRIANGULATING LINE FEATURES  
USING BEARING-ONLY MEASUREMENTS

Motion	Solvable	Unsolvable
Along line direction $\mathbf{v}_e$	$\mathbf{n}_e$	$\mathbf{v}_e$ and $d$
Toward line $[\mathbf{v}_e] \mathbf{n}_e$	$\mathbf{n}_e$	$\mathbf{v}_e$ and $d$
Pure rotation	$\mathbf{n}_e$	$\mathbf{v}_e$ and $d$
Perpendicular to plane $\mathbf{n}_e$	$\mathbf{n}_e, \mathbf{v}_e$ and $d$	-
Random motion	$\mathbf{n}_e, \mathbf{v}_e$ and $d$	-

for the first time and  $\mathbf{Q}_e^\top \mathbf{H}_{\mathbf{x}_c}$  is invertible, we can use an efficient algorithm proposed in our previous work [9] for the SLAM feature covariance initialization with (86). During the implementation of MSCKF, we use null space operation [73] to marginalize  ${}^G \mathbf{x}_f$  from the state vector and only use (87) for efficient update.

### A. Line Feature Triangulation

Within the MSCKF framework, we need to estimate the feature's state using the measurements in the sliding window. Note that we do not need to triangulate plane features since we assume a direct plane measurements while an efficient triangulation of point features was developed in [2]. In this article, we focus on sliding-window-based line feature triangulation, with which the basic geometric elements (e.g.,  $\mathbf{v}_e$ ,  $\mathbf{v}_e$  and  $d_l$ ) for a line feature will be determined. A collection of line segment endpoint pairs for a line feature,  $\mathbf{p}_l$ , will be generated, when it is detected and tracked over the sliding window. With these measurements, we develop two triangulation algorithms detailed in Appendix D. Based on the triangulation algorithms, we also identify three degenerate motions for monocular camera that the line feature triangulation might fail (see Table III). We will avoid these degenerate motions when designing simulation trajectories. After linear triangulation, nonlinear least-squares are performed to refine the line estimates utilizing the collected endpoint measurements.

We perform Monte-Carlo simulations to verify the proposed line feature triangulation algorithms and the identified degenerate motions. Specifically, in simulation, eight lines were placed about 2 m in front of the camera (see Fig. 2) and they were observed by a monocular camera from 20 poses in space. Similar to realistic line segment detector (LSD) [74], the simulated monocular camera collected the two endpoint measurements of lines in its view, with each endpoint measurement corrupted by 2 pixel Gaussian noises, while we assume no correspondences between these endpoints. Three different camera motions (including straight line motion, planar motion, and 3-D sinusoidal motion) were simulated to verify the degenerate motions. During triangulation, we corrupted the true camera poses (both the orientation and position) with random noises

$$\bar{q}_m = \begin{bmatrix} \frac{1}{2} \mathbf{n}^{(\theta)} \\ 1 \end{bmatrix} \otimes \bar{q}, \quad \mathbf{p}_{Cm} = \mathbf{p}_C + \mathbf{n}^{(p)} \quad (88)$$

where  $\mathbf{n}^{(\theta)}$  and  $\mathbf{n}^{(p)}$  are the white Gaussian noises added to the camera pose estimates, while  $\bar{q}_m$  and  $\mathbf{p}_{Cm}$  represent the corrupted camera orientation and position estimates used

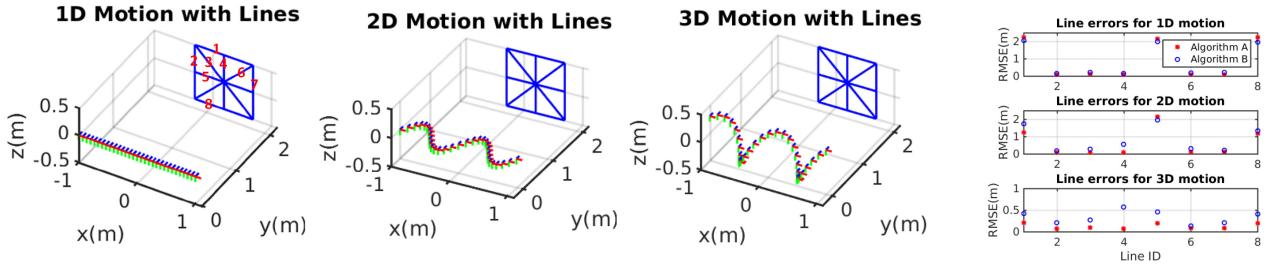


Fig. 2. Simulation setup for linear line triangulation. Eight lines are simulated with IDs in the left figure. Three motion patterns are simulated, including (left) straight line motion, (middle-left) 2-D planar motion, and (middle-right) 3-D motion. Note that 2-D planar motion is in the plane formed by the initial camera position and line 5. On the far right, line triangulation RMSE (computed in CP form) across the Monte-Carlo simulations for the three tested motion profiles.

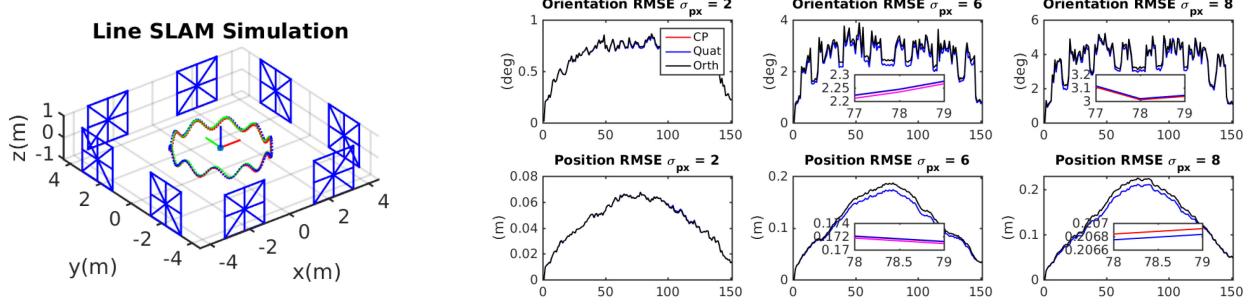


Fig. 3. Simulation setup for visual line SLAM (left). Average orientation and position RMSE for visual SLAM with line features in different representations over the 30 Monte-Carlo simulation runs (right).

by the triangulation algorithms. When evaluating the triangulation errors, we transformed the estimated line parameters into CP form and computed the errors in Euclidean space.

For each motion profile, we generated 30 sets of data and computed the root-mean-square error (RMSE) [75] for the line accuracy evaluation (in CP form), with results shown in Fig. 2. Specifically, since lines 1, 5, and 8 are horizontal lines, when the camera performs straight line motion along this direction (shown in Fig. 2 left), these line features cannot be accurately triangulated. For the planar trajectory, the camera moved in the same plane formulated by the camera center and line 5; therefore, line 5's triangulation still failed. For lines 1 and 8, their accuracy was slightly improved over the 1-D motion case because this planar motion is not strictly degenerate for them. Finally, in the 3-D motion case, all lines were successfully triangulated with relatively low errors due to the fact that all degenerate cases were avoided.

### B. Comparing Line Feature Representations

It is interesting to compare the performance of different line representations. To this end, we performed Monte-Carlo simulations in a VI-SLAM scenario, in which a line map (with 64 lines in total) in an indoor room was generated while a monocular camera was simulated to follow a sinusoidal trajectory (150 poses were simulated in total), as shown in the right of Fig. 3. To focus on evaluating feature representations, we simulated relative pose odometry measurements for the

camera (which were also corrupted with pose noises (88)). To evaluate the robustness of line representations, we used three different noise levels (2, 6, and 8 pixels) to corrupt the line endpoint measurements (as shown in the right of Fig. 3). The camera traversed a single loop of this trajectory as shown in the left of Fig. 3. We employed bundle adjustment (BA) to solve this VL-SLAM problem [76], [77]. In this test, we allowed five Gauss–Newton iterations for each representation for a fair comparison. We ran 30 Monte-Carlo simulations and computed the RMSE [78] for the camera poses to evaluate the accuracy.

As shown in Fig. 3, all three representations yield similar performances. However, as we increase the measurement noise levels, the Plücker representation with orthonormal error states tends to perform slightly worse than the others (e.g., the CP and quaternion representations). Note that in all the noise levels tested, the CP and quaternion line representations perform similarly. These results suggest that one of these two representations should be used in practice, in particular for low-cost sensors with noisy measurements.

### C. Monte-Carlo Simulation Results

To validate our observability analysis of aided INS using heterogeneous geometric features, we perform extensive Monte-Carlo simulations of VINS,<sup>3</sup> (i) VI-SLAM and (ii) VIO,

<sup>3</sup>Similar results as presented in this section would be expected if other aiding sensors are used, for example, acoustic-aided INS was developed in our recent work [12].

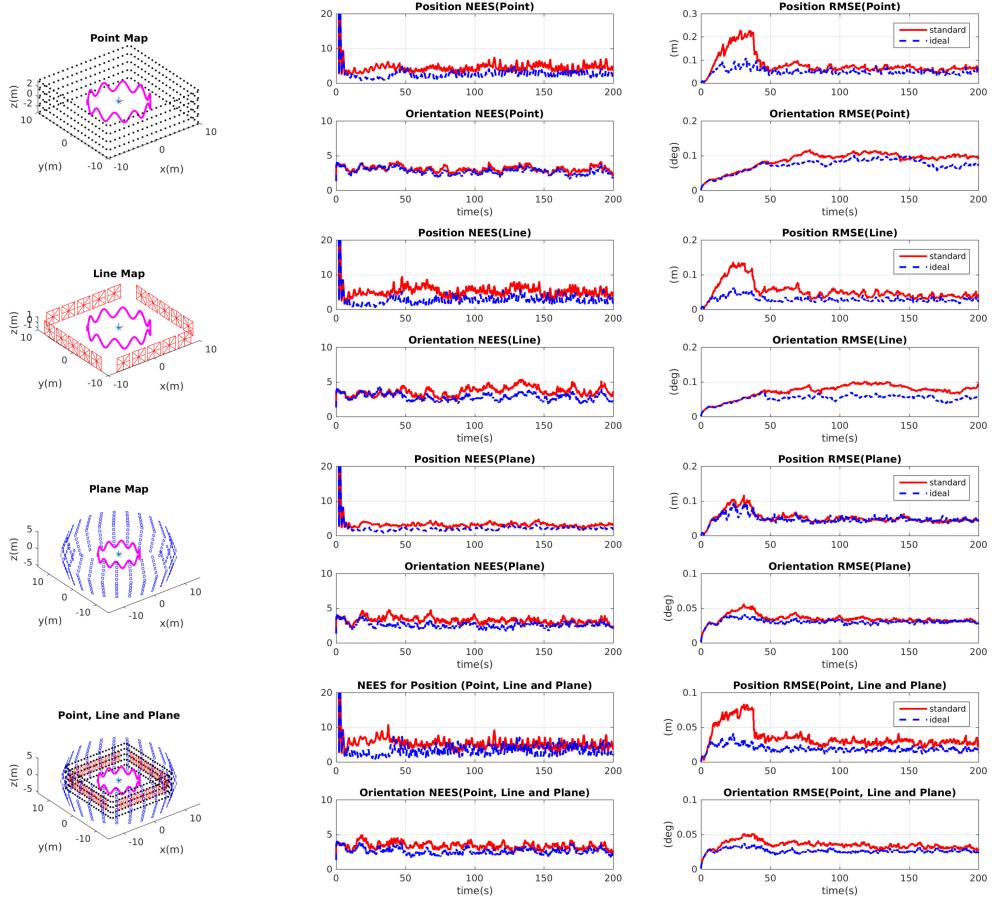


Fig. 4. Monte-Carlo results of EKF-based VI-SLAM using different geometric features of points, lines, and planes.

TABLE IV  
SIMULATION PARAMETERS

Params	Value	Params	Value
$\sigma_g$	1.1220e-4	$\sigma_{pixel}$	1.5pixel
$\sigma_{wg}$	5.6323e-6	$\sigma_\pi$	0.08m
$\sigma_a$	5.0119e-4	IMU Hz	100Hz
$\sigma_{wa}$	3.9811e-5	Camera Hz	5Hz

which are among the most popular localization technologies in part due to their ubiquitous availability and complementary sensing modalities. To this end, we have adapted both the MSCKF/SLAM hybrid system and MSCKF-based VIO algorithms to fuse measurements of points, lines, planes, and their different combinations. To the best of our knowledge, algorithmically, we, for the first time, introduce and evaluate the EKF-based VI-SLAM/VIO approaches with heterogeneous features (which are common in structured environments). The simulation parameters are given in Table IV.

In particular, we have compared two different EKFs in both VI-SLAM and VIO: (i) the *ideal* EKF that uses true states as the linearization points in computing filter Jacobians and has been shown to have correct observability properties and expected to be consistent, thus being used as the benchmark in simulations as in the literature (e.g., [14], [15], [17], [19], [43]);

and (ii) the *standard* EKF that uses current state estimates as the linearization points in computing filter Jacobians and has been shown to be overconfident (inconsistent) [14], [15], [17]. The metrics used to evaluate estimation performance are the RMSE and the average normalized (state) estimation error squared (NEES) [22]. The RMSE provides a measure of accuracy, while the NEES is a standard criterion for evaluating estimator consistency, which (implicitly) indicates the correctness of the EKF system observability.

The simulated trajectories and different geometric features are shown in the left side of Figs. 4 and 5, where we simulate a camera/IMU sensor suite moving on the sinusoidal trajectories to collect measurements of different features. For the results in Fig. 4, we developed the EKF-based VI-SLAM algorithm, which simultaneously performs visual-inertial localization and mapping by keeping different features in the state vector. In contrast, for the results in Fig. 5, we adapted the MSCKF-based VIO [2], which estimates only the sensor poses while marginalizing out different (not only points) features with null space operation. It is clear from these results of both VI-SLAM and VIO in Figs. 4 and 5 that the standard EKF/MSCKF performs worse than the benchmark ideal filter, which agrees with the literature (with point features only) [14], [15], [17]. This again reflects the importance of system observability for consistent state estimation.

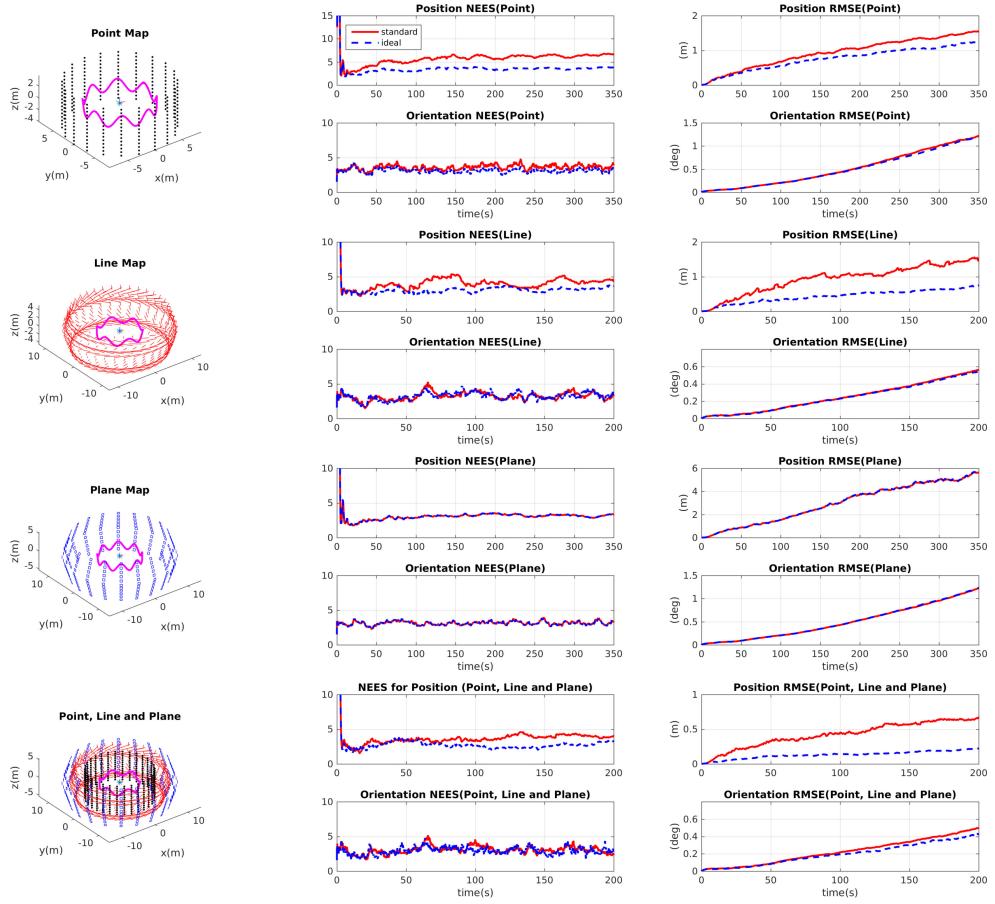


Fig. 5. Monte-Carlo results of MSCKF-based VIO using different geometric features of points, lines, and planes.

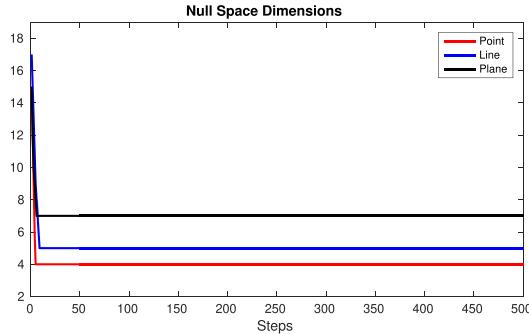


Fig. 6. The dimensions of the null space of the observability matrix (i.e., unobservable subspace) that are numerically computed during the simulation of the ideal-EKF-based VI-SLAM with a single point, line, or plane feature.

#### D. Null Space Verification

Moreover, in order to directly validate the unobservable subspace of the aided INS found in our analysis, using the same simulation setup as above but with a single feature, we have constructed the observability matrix of the ideal EKF-based VI-SLAM with a single point (or line or plane) and numerically computed the dimension of its null space, which is shown in Fig. 6. Clearly, the dimension of the unobservable subspace for the (ideal) VI-SLAM with a single point (line or plane) is 4 (5 or 7), which agrees with our analysis.

## IX. CONCLUSION

In this article, we performed observability analysis for aided INS with different geometric features including points, lines, and planes, which were detected from generic range and/or bearing measurements, encompassing VINS as a special case. As in practice, most aided INS estimators were built based on the linearized systems, whose observability properties directly impact the estimation performance, this article primarily focused on observability analysis of the linearized aided INS with points, lines, planes, and their combinations. In particular, in the case of point features, we analytically showed that the aided INS (both linearized and nonlinear) using generic range and/or bearing measurements has four unobservable directions. In the case of lines (planes), we prove that there exist at least five (or seven) unobservable directions with a single line (plane) feature, and for the first time, analytically derived the unobservable directions for multiple lines and planes. We generalized this observability analysis for linearized aided INS with different combinations of point, line, and plane features and summarized important results in Table V. Based on this analysis, we also systematically investigated the effects of global measurements on the observability of aided INS, and found, as expected, that global information improved the system observability. Moreover, we identified several types of degenerate motions that negatively impact the system observability and should be avoided if possible.

TABLE V  
SUMMARY OF OBSERVABILITY ANALYSIS OF AIDED INS

Features	Unobservable Directions
Single or multiple points	4
Non-parallel lines	
Planes with non-parallel intersections	
Point and line	
Point and plane	
Single line non-parallel to planes	
Plane intersections non-parallel to lines	
Point, line and plane	
Single line	5
Single line parallel to single plane	
Two non-parallel planes	
Single plane	7

(otherwise, extra sensors may be needed). To validate our analysis, we developed EKF-based VI-SLAM and MSCKF-based VIO using heterogeneous geometric features of points, lines, and planes, and evaluated their performance extensively in Monte–Carlo simulations.

In the future, we will leverage the insights gained from this observability analysis to design consistent estimators for aided INS with different geometric features by enforcing proper observability constraints as in our prior work [23]. We will also investigate the (stochastic) observability of aided INS under adversarial attacks [79] or with unknown inputs [54] in order to design secure estimators for robot navigation.

#### APPENDIX A ERROR STATES OF CP LINE

The connection between the error state  $\hat{\mathbf{p}}_l$  of a CP line and that of the corresponding quaternion line  $[\delta\theta_l^\top \tilde{d}_l]^\top$  can be computed as [see (6)]

$$\begin{aligned}
 \mathbf{p}_l &= \hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l = (\hat{d}_l + \tilde{d}_l) \delta\bar{q}_l \otimes \hat{\bar{q}} = (\hat{d}_l + \tilde{d}_l) \mathfrak{R}(\hat{\bar{q}}) \begin{bmatrix} \frac{1}{2}\delta\theta_l \\ 1 \end{bmatrix} \\
 &\Rightarrow \mathfrak{R}(\hat{\bar{q}})^\top (\hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l) = \begin{bmatrix} \frac{1}{2}(\hat{d}_l + \tilde{d}_l)\delta\theta_l \\ \hat{d}_l + \tilde{d}_l \end{bmatrix} \simeq \begin{bmatrix} \frac{1}{2}\hat{d}_l\delta\theta_l \\ \hat{d}_l + \tilde{d}_l \end{bmatrix} \\
 &\Rightarrow \mathfrak{R}(\hat{\bar{q}})^\top (\hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l) - \begin{bmatrix} \mathbf{0} \\ \hat{d}_l \end{bmatrix} \simeq \begin{bmatrix} \frac{1}{2}\hat{d}_l\mathbf{I}_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta\theta_l \\ \tilde{d}_l \end{bmatrix} \\
 &\Rightarrow \begin{bmatrix} \delta\theta_l \\ \tilde{d}_l \end{bmatrix} \simeq \begin{bmatrix} \frac{2}{\hat{d}_l}\mathbf{I}_3 & 0 \\ 0 & 1 \end{bmatrix} \left( \mathfrak{R}(\hat{\bar{q}})^\top (\hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l) - \begin{bmatrix} \mathbf{0} \\ \hat{d}_l \end{bmatrix} \right) \\
 &\Rightarrow \begin{bmatrix} \delta\theta_l \\ \tilde{d}_l \end{bmatrix} \simeq \begin{bmatrix} \frac{2}{\hat{d}_l} (\hat{q}_l\mathbf{I}_3 - \lfloor \hat{\mathbf{q}}_l \rfloor) & -\frac{2}{\hat{d}_l} \hat{\mathbf{q}}_l \\ \hat{\mathbf{q}}_l^\top & \hat{q}_l \end{bmatrix} \tilde{\mathbf{p}}_l \quad (89)
 \end{aligned}$$

where  $\mathfrak{R}(\cdot)$  is the right quaternion multiplication matrix [67].

#### APPENDIX B BEARING MEASUREMENTS OF POINTS FOR A MONOCULAR CAMERA

Geometrically, a monocular camera provides only bearing information of a point feature, whose bearing measurement model is given by

$$\mathbf{z}^{(b)} = \begin{bmatrix} \mathbf{e}_1^\top {}^C \mathbf{p}_f / \mathbf{e}_3^\top {}^C \mathbf{p}_f \\ \mathbf{e}_2^\top {}^C \mathbf{p}_f / \mathbf{e}_3^\top {}^C \mathbf{p}_f \end{bmatrix} + \mathbf{n}^{(b)} = \begin{bmatrix} {}^C x_f / {}^C z_f \\ {}^C y_f / {}^C z_f \end{bmatrix} + \mathbf{n}^{(b)} \quad (90)$$

where  $\mathbf{e}_i \in \mathbb{R}^{3 \times 1}$  ( $i = 1, 2, 3$ ) are the canonical basis unit vectors, i.e.,  $[\mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3] = \mathbf{I}_3$ . As in [80], we use the following bearing measurement model for a point feature:

$$\begin{aligned}
 \mathbf{z}^{(b)} &= \mathbf{h}_b \left( {}^C \mathbf{p}_f, \mathbf{n}^{(b)} \right) \\
 &= \begin{bmatrix} {}^C \mathbf{b}_{\perp 1}^\top \\ {}^C \mathbf{b}_{\perp 2}^\top \end{bmatrix} {}^C \mathbf{p}_f + \mathbf{e}_3^\top {}^C \mathbf{p}_f \begin{bmatrix} {}^C \mathbf{b}_{\perp 1}^\top \\ {}^C \mathbf{b}_{\perp 2}^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \mathbf{n}^{(b)} \quad (91)
 \end{aligned}$$

where  ${}^C \mathbf{b}_{\perp i}, i \in \{1, 2\}$  are two orthogonal vectors to the bearing vector  ${}^C \mathbf{b}_f$ , and can be easily constructed as in [80]. To obtain the Jacobians, we linearize the model about the current estimate of feature-to-camera relative position  ${}^C \hat{\mathbf{p}}_f$  as follows:

$$\begin{aligned}
 \tilde{\mathbf{z}}^{(b)} &\simeq \mathbf{H}_b {}^C \tilde{\mathbf{p}}_f + \mathbf{H}_n \mathbf{n}^{(b)} \\
 &= \begin{bmatrix} {}^C \hat{\mathbf{b}}_{\perp 1}^\top \\ {}^C \hat{\mathbf{b}}_{\perp 2}^\top \end{bmatrix} {}^C \tilde{\mathbf{p}}_f + \mathbf{e}_3^\top {}^C \hat{\mathbf{p}}_f \begin{bmatrix} {}^C \hat{\mathbf{b}}_{\perp 1}^\top \\ {}^C \hat{\mathbf{b}}_{\perp 2}^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \mathbf{n}^{(b)}. \quad (92)
 \end{aligned}$$

#### APPENDIX C PROOF OF LEMMA 4

First of all, it is not difficult to verify that the null space  $\mathbf{N}_{pl\pi 1}$  corresponding to the rotation around the gravity direction. What we need to verify is that  $\mathbf{N}_{pl\pi 2:4}$  is in the unobservable subspace. To this end, for simplicity, we write  $k$ -th block row of the observability matrix as

$$\mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} = \begin{bmatrix} \mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \\ \mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} \\ \mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} \end{bmatrix} \quad (93)$$

where  $\mathbf{H}_{I_k}^{(p)}$ ,  $\mathbf{H}_{I_k}^{(l)}$ , and  $\mathbf{H}_{I_k}^{(\pi)}$  are the Jacobians with respect to the IMU state and the point, line, and plane features, respectively. We can easily verify the following:

$$\mathbf{H}_{I_k}^{(p)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} = \mathbf{0} \quad (94)$$

$$\mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} {}^G \hat{\mathbf{R}}_l = \mathbf{0} \quad (95)$$

$$\mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} {}^G \hat{\mathbf{R}}_\pi = \mathbf{0}. \quad (96)$$

Similarly, since  ${}^G \hat{\mathbf{R}}_\pi$  and  ${}^G \hat{\mathbf{R}}_l$  are rotation matrices of full rank, by left-multiplying both sides of (95) and (96) with  ${}^G \hat{\mathbf{R}}_\pi^\top$  and

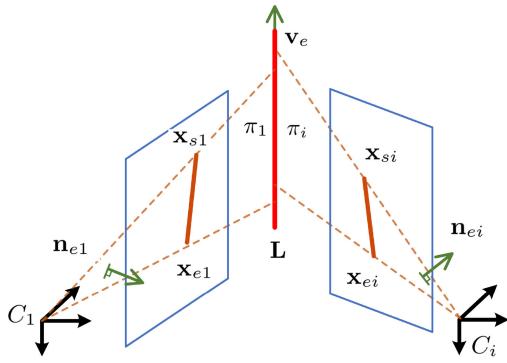


Fig. 7. Sliding window-based line feature triangulation.

$C_i \hat{\mathbf{R}}_l^\top$ , respectively, we have

$$\mathbf{H}_{I_k}^{(l)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} = \mathbf{0} \quad (97)$$

$$\mathbf{H}_{I_k}^{(\pi)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} = \mathbf{0}. \quad (98)$$

Thus, we reach:  $\mathbf{H}_{I_k}^{(pl\pi)} \Phi_{(k,1)} \mathbf{N}_{pl\pi 2:4} = \mathbf{0}$ .

#### APPENDIX D LINE TRIANGULATION ALGORITHMS

##### A. Algorithm A

Denoting the endpoints for a line in the  $i$ th image in the sliding window as  $\mathbf{x}_{si}$  and  $\mathbf{x}_{ei}$  (see Fig. 7), we obtain the normal direction of the plane  $\pi_i$  formed by the line  $L$  and the  $i$ th camera center

$$C_i \mathbf{n}_{ei} = \frac{[\mathbf{x}_{si} | \mathbf{x}_{ei}]}{\|[\mathbf{x}_{si} | \mathbf{x}_{ei}]\|}. \quad (99)$$

Since line  $L$  resides on every plane  $\pi_i$ , we have the following constraint:

$$\underbrace{\begin{bmatrix} \vdots \\ C_i \mathbf{n}_{ei}^\top C_i \mathbf{R}^\top \\ \vdots \end{bmatrix}}_{\mathbf{B}} C_1 \mathbf{v}_{e1} = \mathbf{0}. \quad (100)$$

Therefore,  $C_1 \mathbf{v}_{e1}$  can be found as the unit vector minimizing the error on this constraint, which is given by the eigenvector corresponding to the smallest eigenvalue of  $\mathbf{B}^\top \mathbf{B}$ .

The transformation of a line expressed in frame  $C_i$  to a representation in frame  $C_1$  is given by

$$\begin{bmatrix} C_1 d_l C_i \mathbf{n}_{ei} \\ C_1 \mathbf{v}_{e1} \end{bmatrix} = \begin{bmatrix} C_i \mathbf{R} & [C_1 \mathbf{p}_{C_i}] C_i \mathbf{R} \\ \mathbf{0}_3 & C_i \mathbf{R} \end{bmatrix} \begin{bmatrix} C_i d_l C_i \mathbf{n}_{ei} \\ C_i \mathbf{v}_{e1} \end{bmatrix} \quad (101)$$

$$\Rightarrow C_1 d_l C_i \mathbf{n}_{ei} - C_i d_l C_i \mathbf{R} C_i \mathbf{n}_{ei} = [C_1 \mathbf{p}_{C_i}] C_i \mathbf{R} C_i \mathbf{v}_{e1} \quad (102)$$

$$\Rightarrow C_1 d_l \mathbf{b}_i^\top C_i \mathbf{n}_{ei} = \mathbf{b}_i^\top [C_1 \mathbf{p}_{C_i}] C_i \mathbf{R} C_i \mathbf{v}_{e1} \quad (103)$$

where  $\mathbf{b}_i = [C_1 \mathbf{v}_{e1}] C_i \mathbf{R} C_i \mathbf{n}_{ei}$  is a unit vector perpendicular to  $C_i \mathbf{R} C_i \mathbf{n}_{ei}$  and  $C_1 \mathbf{v}_{e1}$ . Given all the measurements  $i = 2, \dots, m$ , we build a linear system as

$$C_1 d_l \begin{bmatrix} \vdots \\ \mathbf{b}_i^\top C_1 \mathbf{n}_{e1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{b}_i^\top [C_1 \mathbf{p}_{C_i}] C_i \mathbf{R} C_i \mathbf{v}_{e1} \\ \vdots \end{bmatrix}. \quad (104)$$

By solving the above system, we obtain  $C_1 d_l$ . After this step, we have recovered each of the required line parameters:  $C_1 \mathbf{n}_{e1}$ (100),  $C_1 \mathbf{v}_{e1}$ (101), and  $C_1 d_l$ (104). Note that in our algorithm, we have made no assumptions on the correspondences of the endpoints.

##### B. Algorithm B

One of the classical methods to triangulate line features is based on the two intersecting planes (e.g.,  $\pi_1$  and  $\pi_i$ ). The dual Plücker matrix  $\mathbf{L}^*$  (see [81]) can be computed as

$$\mathbf{L}^* = \pi_1 \pi_i^\top - \pi_i \pi_1^\top = \begin{bmatrix} [C_1 \mathbf{v}_{e1}] & C_1 d_l C_1 \mathbf{n}_{e1} \\ -C_1 d_l (C_1 \mathbf{n}_{e1})^\top & 0 \end{bmatrix} \quad (105)$$

where  $\pi_1 = [C_1 \mathbf{n}_{e1}^\top \ 0]^\top$  and  $\pi_i = [C_1 \mathbf{n}_{ei}^\top \ C_1 \mathbf{n}_{ei}^\top C_1 \mathbf{p}_{C_i}]^\top$ .  $C_1 d_l$ ,  $C_1 \mathbf{n}_{e1}^\top$ , and  $C_1 \mathbf{v}_{e1}^\top$  represent the line geometric elements computed based on  $\pi_1$  and  $\pi_i$ . In this work, we offer a generalization of this method for  $m$  measurements. In particular, we solve for the line parameters using

$$C_1 \mathbf{n}_{e1} = \sum_{i=2}^m C_1 \mathbf{n}_{ei}^{(i)} \left/ \left\| \sum_{i=2}^m C_1 \mathbf{n}_{ei}^{(i)} \right\| \right. \quad (106)$$

$$C_1 \mathbf{v}_{e1} = \sum_{i=2}^m C_1 \mathbf{v}_{ei}^{(i)} \left/ \left\| \sum_{i=2}^m C_1 \mathbf{v}_{ei}^{(i)} \right\| \right. \quad (107)$$

$$C_1 d_l = \frac{\sum_{i=2}^m C_1 d_l^{(i)}}{m-1}. \quad (108)$$

##### C. Degenerate Motion Analysis for Line Triangulation

When using a monocular camera, the ability to perform line feature triangulation is heavily dependent on the sensor motion. In particular, we identify degenerate motions that cause the line feature parameters to become unsolvable, thereby causing triangulation to fail (see Table III and Fig. 8). Let  $C$  denote the center of the camera frame and  $L$  the line feature, a plane  $\pi$ . Formulating a plane  $\pi$  (in Fig. 8) with camera center  $C$  and line  $L$ , we have the following remarks:

- 1) If the monocular camera moves along the direction  $\mathbf{v}_e$  of  $L$  or toward  $L$  with direction  $[\mathbf{v}_e] \mathbf{n}_e$ , the camera stays in the same plane,  $\pi$ . As a result, each of the  $C_i \mathbf{n}_{ei}$  will be parallel to each other, causing ambiguity in the solution for  $C_1 \mathbf{v}_{e1}$ , because the rank of matrix  $\mathbf{B}$  in (101) becomes 1. In addition, without  $C_1 \mathbf{v}_{e1}$ ,  $C_1 d_l$  becomes unsolvable.

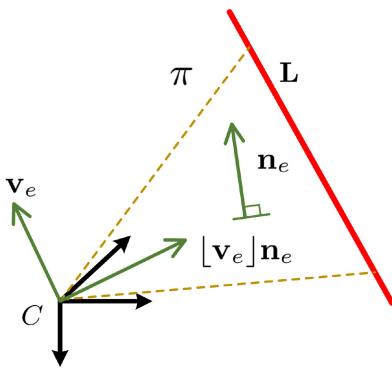


Fig. 8. Degenerate motions for line feature triangulation.

- 2) If the monocular camera undergoes pure rotation (no translation), the camera also stays in the plane  $\pi$ , causing degeneracy as in the previous case.
- 3) The effective motion for line triangulation is the motion along  $\mathbf{n}_e$ , which is perpendicular to the plane  $\pi$ .

Note that for a monocular camera, any combination of the listed three degenerate motions will also cause triangulation failure. Interestingly, for stereo cameras, if both cameras remain in the plane during the motion (such as when the platform translation and camera-to-camera offset remain in the plane), we will still have degenerate motion. This is because triangulation requires that we measure  $\mathbf{L}$  from different views along  $\mathbf{n}_e$ . In this case, even stereo vision cannot guarantee proper line triangulation.

## REFERENCES

- [1] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.
- [2] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation,” in *Proc. Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.
- [3] A. Mourikis, N. Trawny, S. Roumeliotis, A. 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] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual-inertial odometry using nonlinear optimization,” *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, Dec. 2015. [Online]. Available: <https://doi.org/10.1177/0278364914554813>
- [5] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [6] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C. Chu, and H. Hu, “Trifo-vio: Robust and efficient stereo visual inertial odometry using points and lines,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 1–5, 2018, pp. 3686–3693.
- [7] A. Zihao Zhu, N. Atanasov, and K. Daniilidis, “Event-based visual inertial odometry,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Honolulu, HI, USA, Jul. 21–26, 2017, pp. 5391–5399.
- [8] C. X. Guo and S. I. Roumeliotis, “IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 6–10, 2013, pp. 2935–2942.
- [9] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang, “Tightly-coupled aided inertial navigation with point and plane features,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, Canada, May 20–24, 2019, pp. 1–7.
- [10] J. Zhang and S. Singh, “Enabling aggressive motion estimation at low-drift and accurate mapping in real-time,” in *Proc. Int. Conf. Robot. Autom.*, Singapore: IEEE, May 29–Jun. 3, 2017, pp. 5051–5058.
- [11] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, “LIPS: Lidar-inertial 3D plane slam,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Madrid, Spain, Oct. 1–5, 2018, pp. 123–130.
- [12] Y. Yang and G. Huang, “Acoustic-inertial underwater navigation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, May 29–Jun. 3, 2017, pp. 4927–4933.
- [13] S. Rahman, A. Q. Li, and I. Rekleitis, “Sonar visual inertial slam of underwater structures,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Brisbane, Australia, May 21–25, 2018, pp. 1–7.
- [14] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Consistency analysis and improvement of vision-aided inertial navigation,” *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 158–176, Feb. 2014.
- [15] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis, “Camera-IMU-based localization: Observability analysis and consistency improvement,” *Int. J. Robot. Res.*, vol. 33, pp. 182–201, 2014.
- [16] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.
- [17] G. Huang, M. Kaess, and J. Leonard, “Towards consistent visual-inertial navigation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, May 31–Jun. 7, 2014, pp. 4926–4933.
- [18] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.
- [19] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, “Observability-based rules for designing consistent EKF SLAM estimators,” *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 502–528, Apr. 2010.
- [20] G. Huang, K. Eckenhoff, and J. Leonard, “Optimal-state-constraint EKF for visual-inertial navigation,” in *Proc. Int. Symp. Robot. Res.*, Sestri Levante, Italy, Sep. 12–15, 2015, pp. 125–139.
- [21] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, “Convergence and consistency analysis for a 3-D invariant-EKF slam,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 733–740, Apr. 2017.
- [22] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley and Sons, 2001.
- [23] G. Huang, “Improving the consistency of nonlinear estimators: Analysis, algorithms, and applications,” Ph.D. dissertation, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA, 2012. [Online]. Available: <http://people.csail.mit.edu/ghuang/paper/thesis.pdf>
- [24] A. Martinelli, “Visual-inertial structure from motion: Observability and resolvability,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 4235–4242.
- [25] Y. Yang and G. Huang, “Aided inertial navigation with geometric features: Observability analysis,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Brisbane, Australia, May 21–25, 2018, pp. 2334–2340.
- [26] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, “Vins on wheels,” in *Proc. Int. Conf. Robot. Autom.*, Singapore: IEEE, May 29–Jun. 3, 2017, pp. 5155–5162.
- [27] 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.
- [28] 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. Exp. Robot.*, Quebec City, Canada, Jun. 17–20, 2012.
- [29] Y. Yang and G. Huang, “Aided inertial navigation: Unified feature representations and observability analysis,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, Canada, May 20–24, 2019.
- [30] J. Farrell, *Aided Navigation: GPS With High Rate Sensors*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2008.
- [31] M. Li, B. Kim, and A. I. Mourikis, “Real-time motion estimation on a cellphone using inertial sensing and a rolling-shutter camera,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 6–10, 2013, pp. 4697–4704.
- [32] C. Guo, D. Kottas, R. DuToit, A. Ahmed, R. Li, and S. Roumeliotis, “Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps,” in *Proc. Robot. Sci. Syst. Conf.*, Berkeley, CA, Jul. 13–17, 2014, pp. 1–9.
- [33] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Vision-based state estimation for autonomous rotorcraft MAVs in complex environments,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 6–10, 2013, pp. 1750–1756.
- [34] D. G. Kottas and S. I. Roumeliotis, “Efficient and consistent vision-aided inertial navigation using line observations,” in *Proc. Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 6–10, 2013, pp. 1540–1547.

- [35] H. Yu and A. I. Mourikis, "Vision-aided inertial navigation with line features and a rolling-shutter camera," in *Proc. Int. Conf. Robot. Intell. Syst.*, Hamburg, Germany, Oct. 2015, pp. 892–899.
- [36] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/4/1159>
- [37] C. X. Guo *et al.*, "Large-scale cooperative 3D visual-inertial mapping in a Manhattan world," in *Proc. IEEE Int. Conf. Robot. Autom.* Stockholm, Sweden: IEEE, May. 16–21, 2016, pp. 1071–1078.
- [38] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera navigation using point and plane features," in *Proc. Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 3–7, 2013, pp. 3164–3171.
- [39] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A laser-aided inertial navigation system (L-INS) for human localization in unknown indoor environments," in *Proc. Int. Conf. Robot. Autom.*, Anchorage, Alaska, May 3–8, 2010, pp. 5376–5382.
- [40] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, May 2015, pp. 4605–4611.
- [41] D. Kottas and S. Roumeliotis, "Exploiting urban scenes for vision-aided inertial navigation," in *Proc. Robot. Sci. Syst. Conf.*, Berlin, Germany, Jun. 24–28, 2013.
- [42] A. Bartoli and P. Sturm, "Structure from motion using lines: Representation, triangulation and bundle adjustment," *Comput. Vision Image Understanding*, vol. 100, no. 3, pp. 416–441, Dec. 2005.
- [43] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May 19–23, 2008, pp. 473–479.
- [44] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. 11th Int. Symp. Exp. Robot.*, Athens, Greece, Jul. 14–17, 2008, pp. 373–382.
- [45] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "On the complexity and consistency of UKF-based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 12–17, 2009, pp. 4401–4408.
- [46] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1226–1243, Oct. 2013.
- [47] G. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent EKF estimators for multi-robot cooperative localization," *Auton. Robots*, vol. 30, no. 1, pp. 99–122, Jan. 2011.
- [48] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability constrained sliding window filter for SLAM," in *Proc. Int. Conf. Intell. Robots Syst.*, San Francisco, CA, Sep. 25–30, 2011, pp. 65–72.
- [49] 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.
- [50] J. Hernandez, K. Tsotsos, and S. Soatto, "Observability, identifiability and sensitivity of vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA: IEEE, May 26–30, 2015, pp. 2319–2325.
- [51] A. Martinelli, "State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 239–255, Apr. 2011.
- [52] A. Martinelli, "Visual-inertial structure from motion: Observability vs minimum number of sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, May 2014, pp. 1020–1027.
- [53] A. Martinelli, "Nonlinear unknown input observability: The general analytic solution," *arXiv:1704.03252*, 2017.
- [54] A. Martinelli, "Nonlinear unknown input observability: Extension of the observability rank condition," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 222–237, Jan. 2019.
- [55] A. Martinelli, "Closed-form solution to cooperative visual-inertial structure from motion," *ArXiv*, vol. abs/1802.08515, 2018.
- [56] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Trans. Autom. Control*, vol. 22, no. 5, pp. 728–740, Oct. 1977.
- [57] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, Oct. 29–Nov. 2, 2007, pp. 2427–2434.
- [58] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, Jan. 2011.
- [59] G. Panahandeh, C. X. Guo, M. Jansson, and S. I. Roumeliotis, "Observability analysis of a vision-aided inertial navigation system using planar features on the ground," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 3–8, 2013, pp. 4187–4194.
- [60] G. Panahandeh, S. Hutchinson, P. Händel, and M. Jansson, "Planar-based visual inertial navigation: Observability analysis and motion estimation," *J. Intell. & Robotic Syst.*, vol. 82, no. 2, pp. 277–299, May 2016.
- [61] 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.
- [62] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis, "Towards consistent vision-aided inertial navigation," in *Algorithmic Foundations of Robotics X* (ser. Springer Tracts in Advanced Robotics), E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds., Berlin, Heidelberg: Springer, 2013, vol. 86, pp. 559–574.
- [63] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Degenerate motion analysis for aided ins with online spatial and temporal sensor calibration," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2070–2077, Apr. 2019.
- [64] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, ETH Zurich, Switzerland, 2012.
- [65] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.
- [66] J. Maley and G. Huang, "Unit quaternion-based parameterization for point features in visual navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Madrid, Spain, Oct. 2018, pp. 6880–6886.
- [67] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," Tech. Rep., Department of Computer Science & Engineering, University of Minnesota, MN, USA, Mar. 2005.
- [68] X. Zuo, J. Xie, Y. Liu, and G. Huang, "Robust visual SLAM with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sep. 24–28, 2017, pp. 1775–1782.
- [69] Y. Yang and G. Huang, "Observability analysis for aided ins with heterogeneous features of points, lines and planes," Tech. Rep., RPNG, University of Delaware, Newark, DE, USA, 2018. Available: [udel.edu/~yuyang/downloads/tr\\_observabilityIII.pdf](http://udel.edu/~yuyang/downloads/tr_observabilityIII.pdf)
- [70] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge University Press, 2000.
- [71] P. F. Proena and Y. Gao, "Probabilistic RGB-D odometry based on points, lines and planes under depth uncertainty," *Robot. Auton. Syst.*, vol. 104, pp. 25–39, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889017303378>
- [72] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Proc. Robot. Sci. Syst. Conf.*, Sydney, Australia, Jul. 2012.
- [73] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sep. 24–28, 2017.
- [74] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.
- [75] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York, NY, USA: Academic Press, 1988.
- [76] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – A modern synthesis," in *Vision Algorithms: Theory and Practice* (ser. LNCS), W. Triggs, A. Zisserman, and R. Szeliski, Eds. Berlin, Heidelberg, Germany: Springer Verlag, 2000, pp. 298–375.
- [77] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 3607–3613.
- [78] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: John Wiley & Sons, 2004.
- [79] Y. Yang and G. Huang, "Map-based localization under adversarial attacks," in *Proc. Int. Symp. Robot. Res.*, Puerto Varas, Chile, Dec. 11–14, 2017, pp. 1–16.
- [80] Z. Yang and S. Shen, "Monocular visual-inertial fusion with online initialization and camera-IMU calibration," in *Proc. IEEE Int. Symp. Safety Secur. Rescue Robot. (SSRR)*, West Lafayette, IN, USA, Oct. 2015, pp. 1–8.
- [81] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge University Press, 2004.



**Yulin Yang** received the B.Eng. degree from Shandong University, Shandong, China, in 2009, and the M.Sc. degree from Xi'an Jiaotong University, China, in 2012, both in mechanical engineering. He is currently working toward the Ph.D. degree in mechanical engineering with University of Delaware, Newark, DE, USA.

From 2012 to 2015, he was a Research & Development Engineer at Siemens High Voltage Research Center in Shanghai, China. His research topics focus on visual inertial navigation, SLAM, and nonlinear estimation.



**Guoquan Huang** received the B.Eng. degree in automation (electrical engineering) from the University of Science and Technology, Beijing, China, in 2002, and the M.Sc. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2009 and 2012, respectively.

He is currently an Assistant Professor of Mechanical Engineering (ME), Electrical and Computer Engineering (ECE), and Computer and Information Sciences (CIS) with the University of Delaware (UD), Newark, DE, USA, where he is leading the Robot Perception and Navigation Group (RPNG). He is also an Adjunct Professor with the Zhejiang University, Hangzhou, China. He was a Senior Consultant (2016–2018) at the Huawei 2012 Laboratories and a Postdoctoral Associate (2012–2014) at MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA. His research interests include sensing, localization, mapping, and perception of autonomous ground, aerial, and underwater robots.

Dr. Huang received the 2006 Academic Excellence Fellowship from the University of Minnesota, the 2011 Chinese Government Award for Outstanding Self-Financed Students Abroad, the 2015 UD Research Award (UDRF), the 2016 NSF CRII Award, the 2017 UD Makerspace Faculty Fellow, the 2018 SATEC Robotics Delegation, the 2018 Google Daydream Faculty Research Award, and was the Finalist for the 2009 Best Paper Award from the Robotics: Science and Systems Conference (RSS).