Analysis of Optimization-Based Monocular Point–Plane VINS: Formulation, Information, and Marginalization

Yizhi Zhou[1], Xuan Wang[1,†], and Chuchu Chen[2,†]

***Abstract*— Geometric structures such as planes are widely used in monocular visual–inertial navigation systems (VINS) to exploit structural regularities in man-made environments. Existing point–plane formulations typically augment the state with planes and introduce point–plane constraints, but how different geometric formulations affect the underlying information structure, computational complexity, and estimator consistency remains unclear, especially under sliding-window marginalization. In this paper, we analyze optimization-based monocular point–plane VINS from an information-theoretic perspective. Using the Fisher information matrix, we characterize how point, plane, and their combinations alter the information structure and unobservable subspace, and how marginalization reshapes estimator observability. We further study degenerate motion scenarios to identify when plane constraints provide complementary information and when their benefit is limited. Based on theoretical analysis and experiments, we provide guidelines for constructing efficient point–plane factor-graph formulations that balance accuracy and computational cost.**

I. Introduction

Visual–Inertial Navigation Systems (VINS) perform state estimation by fusing visual and inertial measurements, and are widely used in robotics [1]. Beyond point features, many VINS systems exploit planes [2] and point–plane relationships [3], [4], providing additional geometric constraints that improve estimation conditioning and robustness. Reliable plane extraction is essential for point-plane VINS. Planes can be obtained from RGB-D or stereo sensors [4]–[8], or from monocular images using learning-based or multi-view geometric methods [3], [9]–[11]. Once extracted, planes are jointly optimized with points in the factor graph, which can improve estimation accuracy in certain environments [4], [10]. However, directly incorporating planes with points may significantly increase computational and memory costs, limiting deployment on resource-constrained platforms. Since planes provide more structured geometric information and enable more compact map representations than points, properly exploiting plane-related constraints may improve the efficiency of point–plane VINS. But, how plane-related constraints should be formulated and how different formulations affect the trade-off between accuracy and efficiency remain unclear. Moreover, to bound the problem size, marginalization [12] is widely used in sliding-window VINS. However, marginalized states are linearized at a previous estimate and compressed into a prior factor. When the remaining states are later relinearized, the original gauge structure is no longer preserved, which may cause inconsistency [13]. The interaction between marginalization and different point–plane formulations is still not well understood.

In this paper, we focus on monocular VINS with plane constraints incorporated into a fixed sliding-window optimization framework. We conduct a comprehensive analysis to investigate how plane constraints can be formulated within an optimization-based point–plane VINS system, and how different formulations affect computational complexity and estimation accuracy. We also examine how marginalization affects the information and observability properties under different point–plane VINS formulations, and its impact on estimator consistency. We further provide practical guidelines for designing point–plane formulations that effectively exploit planes to balance estimation accuracy and computational efficiency. The main contributions of this paper include:

- We present a systematic analysis of different plane constraint formulations in monocular point–plane VINS, and study how they affect the information structure, observability properties, and estimation performance.
- We analyze how sliding-window marginalization interacts with different point–plane formulations and study its effect on system observability and consistency.
- Based on our theoretical and experimental results, we derive practical guidelines for designing efficient point–plane formulations that balance estimation accuracy and computational cost.

II. Related Work

A. RGB-D and Stereo Systems with Plane Features

Planes have been widely used in stereo and depth-based systems [5], [14]–[18], since direct depth observations enable more reliable plane detection than monocular systems. Early works extract planes from depth measurements and represent them as geometric primitives in the optimization backend [14], [15], [19]. Kaess et al. [5] proposed a minimal plane representation suitable for factor graph–based SLAM, enabling planes to be used as primary features in graph optimization. Building on this idea, several RGB-D VINS and SLAM systems jointly estimate poses, points, and planes while exploiting structural constraints such as plane parallelism and orthogonality [6], [11] under Manhattan-world assumptions. Stereo-based approaches similarly exploit triangulated depth to detect and track planes, allowing them to be treated as independent state variables and jointly optimized with poses and points [4], [7], [20]. While these methods demonstrate the benefits of planar structures and associated geometric constraints, they generally require

Y. Zhou and X. Wang (e-mail: xwang64@gmu.edu) are with the Department of Electrical and Computer Engineering at George Mason University.

C. Chen is with the Department of Mechanical and Aerospace Engineering at George Washington University (e-mail: chuchu.chen@gwu.edu).

[†]Equal advising.

additional depth-sensing hardware and incorporate all point and plane variables into the factor graph, which increases system complexity and limits their efficiency on resource-constrained platforms. Only a few works attempt to reduce the computational cost of point–plane SLAM through equivalent reduced formulations [21] or by replacing dense point-level constraints with plane-level representations [7], [16].

B. Monocular System with Plane Features

In monocular systems, plane extraction is challenging due to the lack of direct depth measurements. Several learning-based approaches infer planes from single images using convolutional neural networks (CNNs) or segmentation masks [9], [10], [22]. However, they require large labeled datasets and may generalize poorly across environments. Several geometry-based methods extract planes from multi-view observations [23], where planes are reconstructed from cross-view correspondences. Some approaches rely on the Manhattan-world assumption, where planes are detected based on alignment with mutually orthogonal directions [9]. Once extracted, planes are incorporated into the state and jointly optimized with point features. Some methods further impose point-on-plane or planarity constraints to enforce structural consistency [3], [10], [23]. While most approaches jointly optimize planes and points within a unified framework, others decouple their estimation and do not refine plane parameters together with point states [24]. Although several studies report improved performance after introducing plane-related constraints (e.g., point-on-plane and plane observation factors), Arndt et al. [25] noted that these gains may stem from structural assumptions and plane extraction quality rather than the constraints themselves. Their experiments provide empirical insights, but the analysis remains purely empirical. How different plane-constraint formulations affect the estimator's behavior and information structure, especially under sliding-window marginalization, and how they impact the trade-off between estimation accuracy and computational complexity remain unclear.

III. Problem Statement

A. Batch-MAP Formulation for Point-Plane VINS

We consider an optimization-based point–plane VINS framework, in which state estimation is formulated as a batch maximum a posteriori (MAP) estimation problem over a sliding window of states and measurements. The posterior distribution of the state trajectory can be factorized as

$$p \propto p(\mathbf{x}_0) \prod_{i=0}^{k-1} p(\mathbf{x}_{i+1} \mid \mathbf{x}_i, \mathbf{u}_i) \prod_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:k}} p(\mathbf{z}_{ij} \mid \mathbf{x}_i, \mathbf{x}_{f_j}) \quad (1)$$

where $\mathbf{x}_i$ denotes the IMU state at time t_i, $\mathbf{x}_{f_j}$ denotes the state of feature j, and $\mathcal{Z}_{0:k}$ contains all visual observations within the sliding window. Under the Gaussian distribution assumption, maximizing (1) is equivalent to minimizing the following cost: [1]

$$\begin{aligned}\mathcal{C}(\mathbf{x}) = \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2_{\mathbf{P}_0} + \sum_{i=0}^{k-1} \|\mathbf{x}_{i+1} \boxminus f(\mathbf{x}_i, \mathbf{u}_i)\|^2_{\mathbf{Q}_i} \\ + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:k}} \left\|\mathbf{z}_{ij} \boxminus h(\mathbf{x}_i, \mathbf{x}_{f_j})\right\|^2_{\mathbf{R}_{ij}} \end{aligned} \quad (2)$$

where $\|\mathbf{a}\|^2_{\mathbf{W}} \triangleq \mathbf{a}^\top \mathbf{W}^{-1} \mathbf{a}$; Set $\mathcal{Z}_{0:k}$ denotes all measurements between $[t_0, t_k]$; $f(\cdot)$ and $h(\cdot)$ denote the state transition and measurement models. The system state $\mathbf{x}$ contains the IMU state $\mathbf{x}_k$ and the feature state $\mathbf{x}_f$ as

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix}\mathbf{x}_0^\top & \cdots & \mathbf{x}_k^\top & \mathbf{x}_f^\top\end{bmatrix}^\top, \mathbf{x}_f = \begin{bmatrix}\cdots & \mathbf{x}_{f_j}^\top & \cdots\end{bmatrix}^\top \\ \mathbf{x}_k &= \begin{bmatrix}{}^{I_k}_{G}\mathbf{q}^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{G}\mathbf{v}_{I_k}^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top\end{bmatrix}^\top \end{aligned} \quad (3)$$

Here, ${}^{I_k}_{G}\mathbf{q}$ denotes the orientation of the IMU frame $\{I_k\}$ with respect to the global frame $\{G\}$, while ${}^{G}\mathbf{p}_{I_k}$ and ${}^{G}\mathbf{v}_{I_k}$ denote the IMU position and velocity expressed in G. $\mathbf{b}_g$ and $\mathbf{b}_a$ represent the gyroscope and accelerometer biases. The feature state $\mathbf{x}_f$ may correspond to point features, plane features, or their combinations. A point feature is represented as ${}^{G}\mathbf{p}_f \in \mathbb{R}^3$. For plane features, we adopt the closest-point (CP) representation [2], defined as ${}^{G}\mathbf{p}_{\Pi} = {}^{G}d\,{}^{G}\mathbf{n}$. ${}^{G}\mathbf{n}$ is the unit normal and ${}^{G}d$ is the signed distance from the origin to the plane. ${}^{G}\mathbf{p}_{\Pi}$ represents the plane in the global frame $\{G\}$ [26]. Given the state (3), the continuous-time system dynamics are given by

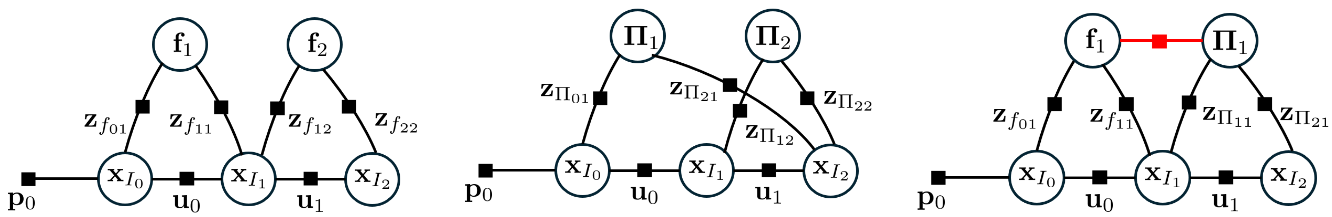
$$\begin{aligned} {}^{I_k}_{G}\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\left({}^{I}\omega\right){}^{I_k}_{G}\mathbf{q}, \quad {}^{G}\dot{\mathbf{p}}_{I_k} = {}^{G}\mathbf{v}_{I_k}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{wa} \\ {}^{G}\dot{\mathbf{v}}_{I_k} = {}^{G}\mathbf{a}, \quad \dot{\mathbf{b}}_g = \mathbf{n}_{wg}, \quad {}^{G}\dot{\mathbf{x}}_f = \mathbf{0} \end{aligned} \quad (4)$$

where $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are zero-mean white Gaussian noise modeling the IMU biases. Based on the system dynamics (4), we can compute the state-transition matrix $\boldsymbol{\Phi}_k$ [3].

The batch MAP formulation in (1) provides a unified factor graph for point-plane VINS, where points and planes are modeled as landmarks and their geometric relations as constraints. Most existing systems include all available features and constraints in a single optimization problem. While such formulations often improve accuracy, they significantly increase computational and memory costs due to the growing state dimension and dense coupling between point and plane variables. Points provide detailed geometric information but are computationally expensive when many landmarks are maintained, whereas planes offer more compact structural constraints with less local detail. A natural solution is to retain only a subset of point and plane features in the state. However, different feature selection strategies lead to different factor-graph formulations, as illustrated in Fig. 1, introducing distinct trade-offs between estimation accuracy, computational complexity, and estimator consistency.

Problem of interest: In the following section, we analyze how different point–plane VINS formulations lead to distinct information structures and explain these trade-offs.

[1] Throughout the paper, $\hat{\mathbf{x}}$ denotes the estimate of $\mathbf{x}$. The operators $\boxplus$ and $\boxminus$ denote manifold retraction and local difference, respectively, and reduce to standard $+$ and $-$ for vector variables.



(a) VINS with only point features. (b) VINS with only plane features (c) VINS with both point and plane features.

Fig. 1: Comparison of VINS formulations using points, planes, and their combination. The diagrams illustrate simplified factor graphs for analysis purposes. The point-based formulation includes IMU motion factors and pose–point factors. The plane-based formulation includes IMU motion factors and plane measurement factors. The combined formulation incorporates IMU motion factors, pose–point factors, plane measurement factors, and point–plane regularization factors [26].

B. State Marginalization

As the robot moves and observes new features, incorporating all measurements in (2) causes the computational cost to grow with the state dimension (scaling as $\mathcal{O}(n^3)$), which is unsuitable for real-time estimation. To bound the problem size, state marginalization [12], [13], [27] is commonly applied, as illustrated in Fig. 2. However, since marginalized states are linearized and fixed, the resulting estimator no longer preserves the same information structure as the full batch formulation (2). As a result, the information matrix differs from that of the batch-MAP solution, which may lead to inconsistency. Inconsistent estimators can become overconfident and produce biased or unreliable estimates [13]. **Problem of interest:** In the following section, we also analyze how marginalization affects different point–plane VINS formulations.

C. Measurement Model and Factors

We present the visual measurement models and the corresponding factors for point and plane features used in our formulation. Standard factors, such as IMU motion and pose–point factors, are omitted for brevity as they follow conventional VINS formulations [26]. The analysis therefore focuses on the pose–plane and point–plane factors.

Point measurements: For a point feature f_1, the corresponding measurement model is

$$\mathbf{z}_{f_1,k} = \pi\left({}^{C_k}\mathbf{p}_{f_1}\right) + \mathbf{n}_{f_1} = \begin{bmatrix} {}^{C_k}x/{}^{C_k}z \\ {}^{C_k}y/{}^{C_k}z \end{bmatrix} + \mathbf{n}_{f_1} \tag{5}$$

where ${}^{C_k}\mathbf{p}_{f_1} = [{}^{C_k}x,\ {}^{C_k}y,\ {}^{C_k}z]^\top$ denotes the point in the camera frame, obtained from ${}^{C_k}\mathbf{p}_{f_1} = {}^{C}_{I}\mathbf{R}{}^{G}_{I_k}\mathbf{R}^\top({}^{G}\mathbf{p}_{f_1} - {}^{G}\mathbf{p}_{I_k}) + {}^{C}\mathbf{p}_I$, and ${}^{G}_{I_k}\mathbf{R}$ denotes the rotation from I_k to G.

Plane measurements: Under the CP representation, the plane measurement model is given by [3]

$$\begin{aligned} \mathbf{z}_{\Pi,k} &= {}^{C_k}d\,{}^{C_k}\mathbf{n} + \mathbf{n}_\Pi,\ {}^{C_k}\mathbf{n} = {}^{C}_{I}\mathbf{R}{}^{G}_{I_k}\mathbf{R}^\top\,{}^{G}\mathbf{n} \\ {}^{C_k}d &= {}^{G}d + {}^{C}\mathbf{p}_I{}^\top\,{}^{C_k}\mathbf{n} - {}^{G}\mathbf{p}_{I_k}^\top\,{}^{G}\mathbf{n} \end{aligned} \tag{6}$$

where ${}^{C_k}\mathbf{n}$ and ${}^{C_k}d$ denote the plane normal and distance in the camera frame. The corresponding pose–plane factor residual is $\mathbf{r}_{\Pi,k} = \mathbf{z}_{\Pi,k} - {}^{C_k}d\,{}^{C_k}\mathbf{n}$, which connects the IMU state $\mathbf{x}_k$ and the plane state ${}^{G}\mathbf{p}_\Pi$.

Regularization constraints: Consider a point feature ${}^{G}\mathbf{p}_f$ lying on the plane ${}^{G}\mathbf{p}_\Pi$ at t_k. The corresponding point–plane geometric relationship can be modeled as a regularization-based measurement [3]

$$z_{d,k} = \left({}^{G}\mathbf{p}_f^\top\,{}^{G}\mathbf{n} - {}^{G}d\right) + n_d \tag{7}$$

where n_d denotes the noise that softens the constraint and becomes zero in the ideal case. The corresponding point-plane factor residual is defined as $r_{d,k} = z_{d,k} - ({}^{G}\mathbf{p}_f^\top\,{}^{G}\mathbf{n} - {}^{G}d)$, which connects the point $\mathbf{p}_f$ and the plane state ${}^{G}\mathbf{p}_\Pi$.

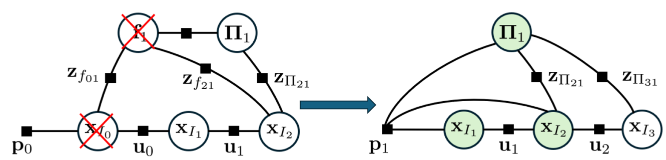


Fig. 2: Example of sliding-window marginalization in point–plane VINS. The left shows the original factor graph, where the oldest states are marginalized. The right shows the updated fixed-size sliding window after marginalization and state propagation, where the condensed prior factor encodes the information of the removed states.

IV. Analysis of Point-Plane VINS

In this section, we analyze how different point–plane VINS formulations alter the estimation information structure, estimator observability, and marginalization behavior. We adopt the Fisher Information Matrix (FIM) [28] to characterize their information content and observability properties.

A. Optimization with Point Features

We consider the case where the geometric feature state consists of g point features. Recall the state representation in (3). The system state at time step k is given by

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_0^\top & \cdots & \mathbf{x}_k^\top & \mathbf{x}_f^\top\end{bmatrix}^\top,\ \mathbf{x}_f = \begin{bmatrix}{}^{G}\mathbf{p}_{f_1}^\top & \cdots & {}^{G}\mathbf{p}_{f_g}^\top\end{bmatrix}^\top$$

Assuming that all features are observed n times along the trajectory, the corresponding information matrix is given by

$$\begin{aligned} \mathbf{F}_f &= \sum_{k=0}^{n-1}\bar{\mathbf{F}}_k^\top\mathbf{Q}^{-1}\bar{\mathbf{F}}_k + \sum_{k=0}^{n}\bar{\mathbf{H}}_{f,k}^\top\mathbf{R}^{-1}\bar{\mathbf{H}}_{f,k} \\ \bar{\mathbf{F}}_k &= [\mathbf{0}_3 \quad \cdots \quad \mathbf{\Phi}_k \quad \mathbf{I}_{9+3g} \quad \cdots \quad \mathbf{0}_3] \\ \bar{\mathbf{H}}_{f,k} &= [\mathbf{0}_3 \quad \cdots \quad \mathbf{H}_{f,k} \quad \cdots \quad \mathbf{0}_3] \end{aligned} \tag{8}$$

where $\mathbf{Q}$ and $\mathbf{R}$ denote the covariances associated with the IMU motion residuals and point feature measurement residuals in (2), respectively. $\mathbf{\Phi}_k$ denotes the state transition matrix, and $\mathbf{H}_{f,k}$ denotes the Jacobian of the point feature measurement, which is given by

$$
\begin{aligned}
\mathbf{H}_{f,k} &= \begin{bmatrix}\mathbf{H}_{f_1,k}^\top & \cdots & \mathbf{H}_{f_g,k}^\top\end{bmatrix}^\top \\
\mathbf{H}_{f_j,k} &= {}^{C}_{I}\mathbf{R}\ [\mathbf{\Gamma}_{1,k} \quad -\mathbf{I}_3 \quad \mathbf{0}_3 \quad \cdots \quad \mathbf{I}_3 \quad \cdots \quad \mathbf{0}_3] \\
\mathbf{\Gamma}_{1,k} &= \left\lfloor {}^{G}_{I_k}\mathbf{R}^\top({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{I_k})\times\right\rfloor, \quad \forall j \in \{1,\cdots,g\}
\end{aligned} \tag{9}
$$

To quantify the information contributed by point features, we analyze the nullspace of the information matrix $\mathbf{F}_f$. It can be shown that $\mathbb{N}(\mathbf{F}_f)$ is four-dimensional [13]:

$$
\begin{aligned}
\mathbb{N}(\mathbf{F}_f) &= \begin{bmatrix}\mathbb{N}_{f_0}^\top & \cdots & \mathbb{N}_{f_n}^\top & \mathbb{N}_{ff_1}^\top & \cdots \mathbb{N}_{ff_g}^\top\end{bmatrix} \\
\mathbb{N}_{f_k}^\top &= \begin{bmatrix}({}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g})^\top & (\lfloor {}^{G}\mathbf{p}_{I_k}\times\rfloor{}^{G}\mathbf{g})^\top & (\lfloor {}^{G}\mathbf{v}_{I_k}\times\rfloor{}^{G}\mathbf{g})^\top \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3\end{bmatrix} \\
\mathbb{N}_{ff_j} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{f_j}\times\rfloor{}^{G}\mathbf{g} & \mathbf{I}_3\end{bmatrix}, \quad \forall j \in \{1,\cdots,g\}
\end{aligned}
$$

The four-dimensional nullspace corresponds to the global yaw and position unobservability of monocular VINS with point features. The above analysis holds for the full batch-MAP formulation before marginalization. In a sliding-window implementation, old pose states are marginalized and their information is incorporated into a prior on the remaining states (Fig. 2), resulting in an information matrix denoted by $\mathbf{F}'_f$. It can be shown that [13]:

$$
\dim(\mathbb{N}(F_f)) > \dim(\mathbb{N}(F'_f)) \tag{10}
$$

The reduced nullspace dimension indicates that marginalization alters the information structure due to linearization mismatch [13]. Since the prior is constructed at an earlier linearization point, subsequent relinearization breaks the original gauge structure, causing certain unobservable directions to appear constrained and leading to estimator inconsistency. This issue can be mitigated using First-Estimate Jacobians (FEJ) [3], where the Jacobians associated with marginalized IMU states (e.g., orientation, position, and velocity) and the related point measurement terms are evaluated at their first linearization point and kept fixed when forming the prior.

B. Optimization with Plane Features

With a single plane feature ${}^{G}\mathbf{p}_{\Pi_1}$, the system state is

$$
\mathbf{x} = \begin{bmatrix}\mathbf{x}_0^\top & \cdots & \mathbf{x}_k^\top & \mathbf{x}_f^\top\end{bmatrix}^\top, \mathbf{x}_f = {}^{G}\mathbf{p}_{\Pi_1} \tag{11}
$$

Assume the plane is observed n times along the trajectory. Following the same batch formulation as in the point feature case, linearizing the plane measurement model (6) with respect to (11) yields

$$
\begin{aligned}
\mathbf{H}_{\Pi_1,k} &= [\mathbf{\Gamma}_{2,k} \quad \mathbf{\Gamma}_{3,k} \quad \mathbf{0}_3 \quad \mathbf{\Gamma}_{4,k}],\ \mathbf{\Gamma}_{2,k} = \mathbf{\Gamma}_{5,k}\lfloor {}^{G}_{I_k}\mathbf{R}^{\top G}\mathbf{n}\times\rfloor \\
\mathbf{\Gamma}_{3,k} &= -{}^{G}_{I_k}\mathbf{R}^{\top G}\mathbf{n}^{G}\mathbf{n}^\top,\ \mathbf{\Gamma}_{5,k} = {}^{G}d - {}^{G}\mathbf{p}_{I_k}^{\top}{}^{G}\mathbf{n} \\
\mathbf{\Gamma}_{4,k} &= \frac{{}^{G}_{I_k}\mathbf{R}^\top}{{}^{G}d}\Big(\mathbf{\Gamma}_{5,k}\mathbf{I}_3 - {}^{G}\mathbf{n}^{G}\mathbf{p}_I^\top + 2({}^{G}\mathbf{n}^{\top G}\mathbf{p}_I)\,{}^{G}\mathbf{n}^{G}\mathbf{n}^\top\Big)
\end{aligned}
$$

The corresponding information matrix can then be written as

$$
\begin{aligned}
\mathbf{F}_{\Pi_1} &= \sum_{k=0}^{n-1}\bar{\mathbf{F}}_k^\top\mathbf{Q}^{-1}\bar{\mathbf{F}}_k + \sum_{k=1}^{n}\bar{\mathbf{H}}_{\Pi_1,k}^\top\mathbf{R}^{-1}\bar{\mathbf{H}}_{\Pi_1,k} \\
\bar{\mathbf{H}}_{\Pi_1,k} &= [\mathbf{0}_3 \quad \cdots \quad \mathbf{H}_{\Pi_1,k} \quad \cdots \quad \mathbf{0}_3]
\end{aligned} \tag{12}
$$

The nullspace of $\mathbf{F}_{\Pi_1}$ can be computed as

$$
\begin{aligned}
\mathbb{N}(\mathbf{F}_{\Pi_1}) &= \begin{bmatrix}\mathbb{N}_{\Pi_0}^\top & \cdots \mathbb{N}_{\Pi_n}^\top & \mathbb{N}_{\Pi\Pi_1}^\top\end{bmatrix} \\
\mathbb{N}_{\Pi_j}^\top &= \begin{bmatrix}({}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g})^\top & (\lfloor {}^{G}\mathbf{p}_{I_k}\times\rfloor{}^{G}\mathbf{g})^\top & (\lfloor {}^{G}\mathbf{v}_{I_k}\times\rfloor{}^{G}\mathbf{g})^\top \\ \mathbf{0}_3 & {}^{G}_{\Pi_1}\mathbf{R} & \mathbf{0}_3 \\ \mathbf{N}_{\Pi\Pi} & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix} \\
&\triangleq \begin{bmatrix}\mathbf{N}_{\Pi_1}^\top & \mathbf{N}_{\Pi_{2:4}}^\top & \mathbf{N}_{\Pi_{5:7}}^\top\end{bmatrix}^\top \\
\mathbb{N}_{\Pi\Pi_1} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_1}\times\rfloor{}^{G}\mathbf{g} & {}^{G}\mathbf{n}\,\mathbf{e}_3^\top & \mathbf{0}_{3\times1}\end{bmatrix}
\end{aligned} \tag{13}
$$

where ${}^{G}_{\Pi_1}\mathbf{R} \triangleq \begin{bmatrix}{}^{G}\mathbf{n}_1^\perp & {}^{G}\mathbf{n}_2^\perp & {}^{G}\mathbf{n}\end{bmatrix}$ denotes the plane orientation matrix, with ${}^{G}\mathbf{n}_1^\perp$ and ${}^{G}\mathbf{n}_2^\perp$ forming an orthonormal basis of the subspace orthogonal to the plane normal ${}^{G}\mathbf{n}$; $\mathbf{e}_3 \triangleq [0\ 0\ 1]^\top$. The nullspace derivation follows the same procedure as in the point-feature case, with the formulation extended from point features to plane features [13]. From (13), a system observing a single plane feature has at least seven unobservable directions, including the global yaw and position, two translations parallel to the plane, and rotation about the plane normal. Following the same sliding-window marginalization as in the point-feature case, the resulting information matrix is denoted by $\mathbf{F}'_{\Pi_1}$. We further prove that

$$
\dim\big(\mathbb{N}(\mathbf{F}_{\Pi_1})\big) > \dim\big(\mathbb{N}(\mathbf{F}'_{\Pi_1})\big) \tag{14}
$$

We prove that the global yaw and the rotation about the plane normal become spuriously observable after marginalization (see Remark 1). To preserve estimator consistency, FEJ is applied by fixing both the marginalized IMU Jacobians and the plane-related Jacobians at their first linearization point.

Remark 1 (Proof Sketch): From (13), any unobservable direction $\boldsymbol{\eta} \in \mathbb{N}(\mathbf{F}_{\Pi_1})$ satisfies $\mathbf{F}_{\Pi_1}\boldsymbol{\eta} = 0$. In a sliding-window implementation, marginalization constructs a prior via the Schur complement at a linearization point $\mathbf{x}^\star$. After subsequent relinearization at a new state $\tilde{\mathbf{x}} \neq \mathbf{x}^\star$, the remaining factors are relinearized at $\tilde{\mathbf{x}}$ while the prior Jacobian remains fixed at $\mathbf{x}^\star$, leading to linearization mismatch. As a result, some original unobservable directions are no longer preserved, i.e., $\mathbf{F}'_{\Pi_1}\boldsymbol{\eta} \neq 0$ for some $\boldsymbol{\eta} \in \mathbb{N}(\mathbf{F}_{\Pi_1})$. Therefore $\boldsymbol{\eta} \notin \mathbb{N}(\mathbf{F}'_{\Pi_1})$, which implies $\dim(\mathbb{N}(\mathbf{F}_{\Pi_1})) > \dim(\mathbb{N}(\mathbf{F}'_{\Pi_1}))$. Due to space limitations, only a proof sketch is provided here. The proofs for the remaining cases follow the same reasoning and are omitted for brevity.

The above analysis can be extended to scenarios with multiple plane features. Specifically, consider the case with m planes, where the geometric feature state is

$$
\mathbf{x}_f = \begin{bmatrix}{}^{G}\mathbf{p}_{\Pi_1}^\top & {}^{G}\mathbf{p}_{\Pi_2}^\top & \cdots & {}^{G}\mathbf{p}_{\Pi_m}^\top\end{bmatrix}^\top \tag{15}
$$

In this case, the structure of the information matrix depends not only on the number of planes but also on their geometric relationships, particularly the relative orientations of the plane normals. We therefore analyze several representative plane configurations in the following.

Mutually parallel planes: if all the planes are parallel to each other, the corresponding information matrix, denoted as $\mathbf{F}_{\Pi}^{(1)}$, has the following nullspace

$$
\begin{aligned}
\mathbb{N}(\mathbf{F}_{\Pi}^{(1)}) &= \begin{bmatrix}\mathbb{N}_{\Pi_0}^\top & \cdots & \mathbb{N}_{\Pi_n}^\top & \mathbb{N}_{\Pi\Pi_1}^\top & \cdots & \mathbb{N}_{\Pi\Pi_m}^\top\end{bmatrix} \\
\mathbb{N}_{\Pi\Pi_j} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_j}\times\rfloor{}^{G}\mathbf{g} & {}^{G}\mathbf{n}\,\mathbf{e}_3^\top & \mathbf{0}_{3\times1}\end{bmatrix}
\end{aligned} \tag{16}
$$

where ${}^{G}\mathbf{n}$ is a unit normal vector shared by all planes. The system still exhibits seven unobservable directions, indicating that multiple parallel planes provide no additional independent constraints compared to the single-plane case and therefore do not improve observability. Consequently, the marginalization analysis remains identical to the single-plane case and is omitted.

Non-parallel planes: For non-parallel planes whose pairwise intersection lines are all parallel, the information matrix $\mathbf{F}_{\Pi}^{(2)}$ admits the following nullspace:

$$\begin{aligned}\mathbb{N}(\mathbf{F}_{\Pi}^{(2)}) &= \begin{bmatrix}\mathbb{N}_{\Pi_0}^{(1:5)\top} & \cdots & \mathbb{N}_{\Pi_n}^{(1:5)\top} & \mathbb{N}_{\Pi\Pi_1}^{\top} & \cdots & \mathbb{N}_{\Pi\Pi_m}^{\top}\end{bmatrix} \\ \mathbb{N}_{\Pi_j}^{(1:5)} &= \begin{bmatrix}\mathbf{N}_{\Pi_1}^{\top} & \mathbf{N}_{\Pi_{2:4}}^{\top} & \mathbf{N}_{\Pi_5}^{\top}\end{bmatrix}^{\top} \quad \forall k \in \{1,\cdots,n\} \\ \mathbb{N}_{\Pi\Pi_k} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_j}\times\rfloor {}^{G}\mathbf{g} & {}^{G}\mathbf{n} & \mathbf{0}_{3\times 1}\end{bmatrix}, \quad \forall j \in \{1,\cdots,m\}\end{aligned}$$

where $\mathbf{N}_{\Pi_1}$ and $\mathbf{N}_{\Pi_2}$ are defined in (13). The system has at least five unobservable directions, including the global yaw angle, global position, and motion along the parallel lines.

If all the planes have non-parallel intersection lines, the information matrix, denoted as $\mathbf{F}_{\Pi}^{(3)}$, will have a four-dimensional nullspace as

$$\begin{aligned}\mathbb{N}(\mathbf{F}_{\Pi}^{(3)}) &= \begin{bmatrix}\mathbb{N}_{\Pi_0}^{(1:4)\top} & \cdots & \mathbb{N}_{\Pi_n}^{(1:4)\top} & \mathbb{N}_{\Pi\Pi_1}^{\top} & \cdots & \mathbb{N}_{\Pi\Pi_m}^{\top}\end{bmatrix} \\ \mathbb{N}_{\Pi_j}^{(1:4)} &= \begin{bmatrix}\mathbf{N}_{\Pi_1}^{\top} & \mathbf{N}_{\Pi_2}^{\top}\end{bmatrix}^{\top} \quad \forall k \in \{1,\cdots,n\} \\ \mathbb{N}_{\Pi\Pi_k} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_j}\times\rfloor {}^{G}\mathbf{g} & {}^{G}\mathbf{n}\,\mathbf{e}_3^{\top}\end{bmatrix}, \quad \forall j \in \{1,\cdots,m\}\end{aligned} \tag{17}$$

which corresponds to the global position and global yaw. Following the same procedures, we prove that the global yaw becomes spuriously observable after marginalization.

C. Optimization with Points and Planes

We extend the analysis to geometric states $\mathbf{x}_f$ containing both point and plane features. In (1), these features are jointly estimated as landmarks. When a point lies on a plane, a point–plane geometric relationship arises, introducing coupled constraints between the two states. For analytical clarity, we consider the minimal case consisting of one point and one plane feature, as defined in (18). It is important to note that in (1), whether points and planes can be jointly represented as landmarks depends on how planes are extracted. If a plane is fitted from points, the same geometric structure must be represented either as a plane landmark or as point landmarks, but not both, to avoid information reuse. In contrast, if a plane is extracted independently, point and plane landmarks can be jointly incorporated. We consider this latter case, where a point lies on a plane but is not involved in the plane extraction, and analyze its effect on estimation performance.

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_0^{\top} & \cdots & \mathbf{x}_k^{\top} & \mathbf{x}_f^{\top}\end{bmatrix}^{\top}, \mathbf{x}_f = \begin{bmatrix}{}^{G}\mathbf{p}_{f_1}^{\top} & {}^{G}\mathbf{p}_{\Pi_1}^{\top}\end{bmatrix}^{\top} \tag{18}$$

and the measurement model becomes $\mathbf{z}_{\Pi f} = \begin{bmatrix}\mathbf{z}_{f_1,k}^{\top} & \mathbf{z}_{\Pi_1,k}^{\top}\end{bmatrix}^{\top}$. The corresponding measurement Jacobians can be computed based on (9) and (12). Assuming that both features are observed over a time window of n measurements, we construct the corresponding information matrix associated with the feature state, which is denoted by $\mathbf{F}_{f\Pi}$, where the nullspace of $\mathbf{F}_{f\Pi}$ can be computed as

$$\begin{aligned}\mathbb{N}(\mathbf{F}_{f\Pi}) &= \begin{bmatrix}\mathbb{N}_{f_0}{}^{\top} & \cdots & \mathbb{N}_{f_n}{}^{\top} & \mathbb{N}_{ff_1}^{\top} & \mathbb{N}_{f\Pi_1}^{\top}\end{bmatrix} \\ \mathbb{N}_{f\Pi_1} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_1}\times\rfloor {}^{G}\mathbf{g} & {}^{G}\mathbf{n}\,\mathbf{e}_3^{\top}\,{}^{\Pi}_{G}\mathbf{R}\end{bmatrix}\end{aligned} \tag{19}$$

where the definitions of $\mathbb{N}_{f_0}$ and $\mathbb{N}_{ff_1}$ are given in (10). The system has four unobservable degrees of freedom: the global yaw angle, the 3-D global position of the IMU. We also prove that, under sliding-window marginalization, the dimension of the nullspace of the marginalized information matrix, denoted as $\mathbf{F}'_{f\Pi}$, is smaller than that of the full-batch information matrix $\mathbf{F}_{f\Pi}$. As in the previous cases, FEJ is also applied to fix the Jacobians of marginalized IMU states and the related point–plane measurement terms at their first estimate to maintain the correct gauge structure.

D. Points on Planes

When a point lies on a plane, it is natural to introduce a point–plane constraint. However, if the plane is fitted from the same point measurements, the estimated point and plane are statistically dependent, and the point–plane relation is already reflected in the plane-fitting process. Adding an explicit point–plane factor in this case may therefore double-count the same geometric information. To isolate the information provided by the point–plane factor, we consider below the case where the point and plane are modeled as independent variables and jointly optimized in the factor graph. For analytical simplicity, we extend the analysis in Sec. IV-C by introducing an additional point–on–plane constraint. Specifically, we consider the case where the state contains only one point feature and one plane feature, as defined in (18). We linearize the point–plane regularization constraint (7), yielding the following Jacobian:

$$\begin{aligned}\mathbf{H}_{d,k} &\triangleq \begin{bmatrix}\mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & {}^{G}\mathbf{n}^{\top} & \mathbf{\Gamma}_{5,k}\end{bmatrix} \\ \mathbf{\Gamma}_{5,k} &= \frac{1}{{}^{G}d}\,{}^{G}\mathbf{p}_f^{\top}(\mathbf{I}_3 - {}^{G}\mathbf{n}{}^{G}\mathbf{n}^{\top}) - {}^{G}\mathbf{n}^{\top}\end{aligned} \tag{20}$$

The resulting information matrix, denoted by $\mathbf{F}_{f\Pi}^{(1)}$, admits the following nullspace

$$\begin{aligned}\mathbb{N}(\mathbf{F}_{f\Pi}^{(1)}) &= \begin{bmatrix}\mathbb{N}_{f_0}{}^{\top} & \cdots & \mathbb{N}_{f_n}{}^{\top} & \mathbb{N}_{ff_1}^{\top} & \mathbb{N}_{f\Pi_1}^{\top}\end{bmatrix}, \\ \mathbb{N}_{f\Pi_1} &= \begin{bmatrix}\lfloor {}^{G}\mathbf{p}_{\Pi_1}\times\rfloor {}^{G}\mathbf{g} & {}^{G}\mathbf{n}\,\mathbf{e}_3^{\top}\,{}^{\Pi}_{G}\mathbf{R}\end{bmatrix}\end{aligned} \tag{21}$$

The marginalization analysis is analogous to that in Sec. IV-C and is therefore omitted.

E. Discussion of Degenerate Motion

The previous sections analyze the information characteristics of point and plane features under general motion. We now consider motion degeneration, which introduces additional rank deficiency in the information matrix [29]. This analysis examines how motion patterns affect point and plane constraints and guides factor-graph formulation.

Motion toward the point: When the platform moves toward a point along the viewing direction, the depth component of the point position along the camera optical axis becomes unobservable. In this case, incorporating plane constraints can provide additional geometric information to help constrain the depth direction.

Motion parallel to a plane: When the motion is parallel to a plane, the distance between the robot and the plane remains constant, introducing an additional unobservable direction along the plane normal. Under such motion, plane constraints offer limited benefit relative to their computational cost.

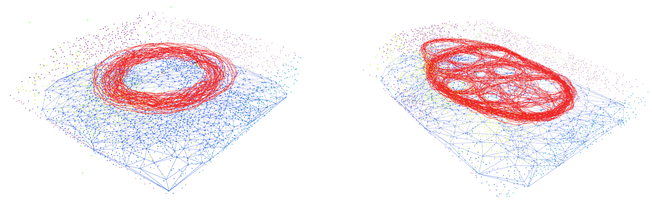


Fig. 3: Simulated indoor environment with plane surfaces, generated point features, and a predefined robot trajectory for evaluating different point–plane configurations.

TABLE I: Simulation parameters used in the Monte Carlo experiments.

Parameter	Value	Parameter	Value
Gyro. Noise	1.696×10^{-4}	Gyro. Bias	1.939×10^{-5}
Accel. Noise	2.000×10^{-3}	Accel. Bias	3.000×10^{-3}
Cam. Freq. (Hz)	10	IMU Freq. (Hz)	400
Num. Clones	11	Total Planes	6
Avg. Feats on Plane	150	Max SLAM Pts	15

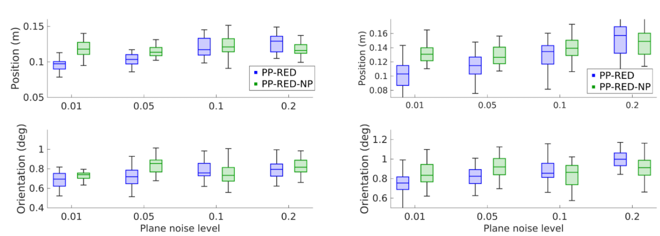
V. Experiments

A. Simulation Analysis

Simulation setup: To validate the theoretical analysis, we conduct Monte Carlo simulations with a monocular point–plane VINS built on OpenVINS [1] and the `ov_plane` simulator [3]. A robot follows a pre-defined 6-DOF B-spline trajectory in a simulated environment (Fig. 3). The environment contains 500 points: 250 on planar surfaces and 250 in free space. The planes represent the walls, floor, and ceiling, with known point–plane associations. Table I summarizes the simulation settings. Estimation performance is evaluated using Absolute Trajectory Error (ATE). Based on the generated trajectories and measurements, we implement a sliding-window point–plane VINS that jointly optimizes the navigation states and geometric features using IMU, point, and plane factors. Marginalization is applied to maintain a bounded problem size. Point features are initialized through triangulation after reaching a minimum track length and are marginalized when no longer tracked, whereas planes are directly observed and require no triangulation. Since this work focuses on point–plane formulations in the VINS backend, the simulator provides plane observations and point–plane associations, allowing us to isolate the effects of different feature-state and factor choices.

Study 1 (validation of different formulations): This study validates the theoretical analysis by examining *four key questions* regarding the role of plane features in point–plane VINS: (i) the impact of plane constraints on estimation accuracy and efficiency; (ii) whether a *plane-only* formulation can achieve higher efficiency while maintaining comparable accuracy; (iii) how the number and geometric configuration of planes influence estimation performance; and (iv) whether a compact formulation using a small number of points together with planes can achieve a favorable accuracy–efficiency trade-off. To this end, we design the following experimental settings: (i) a point-only setting (PT-OL); (ii) a plane-only setting with six planes and no point features, including two configurations: at least three mutually non-parallel planes (PL-NP), and two mutually non-parallel planes with the remaining planes forming parallel pairs (PL-PAR); (iii) a full point–plane setting where all points, planes, and associated constraints are included in the optimization (PP-ALL); (iv) a reduced point–plane setting where only a subset of off-plane points (100 randomly selected points, about 20% of those in (iii)) and two mutually non-parallel planes are maintained in the state (PP-RED); (v) a configuration similar to (iv) but without point-on-plane constraints (PP-RED-NP).

Results from 50 Monte Carlo runs with different estimator configurations are summarized in Table II. First, incorporating plane constraints in the joint point–plane optimization leads to only marginal improvement in estimation accuracy compared with the point-only formulation, while noticeably increasing computational cost due to the additional plane states and pose–plane factors. This observation is consistent with the theoretical analysis in Sec. IV-B. Second, when a sufficient number of mutually non-parallel planes are observed, the plane-only formulation achieves up to $15.8\times$ backend speedup compared with the joint point–plane formulation, while maintaining comparable position accuracy, although a moderate degradation in orientation accuracy is observed. This speedup results from the reduced number of landmark states and factors in the backend optimization. Third, the geometric configuration of planes plays a critical role in estimation performance: configurations dominated by parallel planes lead to significant accuracy degradation, which is consistent with the observability analysis in Sec. IV-B. Fourth, the reduced formulation (PP-RED) achieves comparable accuracy using only a small subset of points and planes, offering a favorable accuracy–efficiency trade-off, whereas the full formulation (PP-ALL) provides only marginal accuracy gains at a much higher computational cost. In addition, incorporating point-on-plane constraints yields only limited improvement in pose estimation, since they do not alter the system observability (Sec. IV-D). Their effectiveness also depends on the accuracy of plane extraction, as demonstrated in the following experiments. Based on these observations, when sufficient planes are available, accurate estimation can be achieved using planes with only a small number of points, significantly improving efficiency,



(a) Optimization with FEJ (b) Optimization without FEJ

Fig. 4: Impact of plane extraction accuracy on estimation performance. Position (top) and orientation (bottom) errors are evaluated under different plane noise levels. Each box summarizes 30 Monte Carlo runs.

TABLE II: Monte Carlo results under different feature configurations and trajectory lengths. Each entry reports Position ATE (m) / Orientation ATE (deg).

Formulation	Simulated trajectory length (m)					
	100	200	300	500	600	800
PT-OL	0.1289 / 0.7866	0.1342 / 0.8756	0.1206 / 0.7994	0.1245 / 0.7907	0.1195 / 0.7348	0.1103 / 0.7746
PL-NP	**0.1174** / 1.0623	0.1408 / 1.0783	0.1255 / 1.1084	0.1246 / 0.9877	0.1208 / 0.9943	0.1211 / 1.1002
PL-PAR	0.1312 / 1.6425	0.1387 / 1.7013	0.1276 / 1.6189	0.1298 / 1.5527	0.1249 / 1.5234	0.1227 / 1.5871
PP-ALL	0.1155 / **0.7014**	0.1265 / 0.7364	**0.1104** / 0.7478	**0.1170** / **0.7529**	0.1124 / **0.7187**	**0.1078** / 0.8845
PP-RED	**0.1095** / 0.7786	0.1273 / 0.8622	0.1187 / 0.8049	0.1216 / 0.8213	0.1109 / 0.8097	0.1118 / **0.8486**
PP-RED-NP	0.1229 / 0.8024	**0.1263** / **0.7237**	0.1179 / **0.7433**	0.1204 / 0.8288	**0.1071** / 0.8298	0.1142 / 0.8674

as planes are fewer and easier to track over long trajectories.

Study 2 (Impact of plane extraction accuracy): The goal of this study is to investigate how plane extraction accuracy affects the performance of point–plane VINS and to evaluate the role of FEJ in validating our observability analysis. To emulate plane extraction errors, we vary the Gaussian noise level in the plane measurement model in (6) with standard deviations of 0.01, 0.05, 0.1, and 0.2. Two formulations are considered for comparison: (PP-RED) and (PP-RED-NP). As shown in Fig. 4a, the estimation errors increase only slightly as the plane noise grows, indicating that plane extraction accuracy has limited impact on overall estimation performance, since the dominant information is still provided by point features and inertial measurements. When the plane noise becomes large, incorporating point-on-plane constraints provides little benefit and may even slightly degrade accuracy due to biased geometric constraints. As shown in Fig. 4b, FEJ improves estimation accuracy, even with inaccurate plane initialization and imperfect linearization points. This improvement is attributed to enhanced estimator consistency, as analyzed in the previous section and further confirmed by the Normalized Estimation Error Squared (NEES) results in our technical report [26].

Study 3 (Impact of degenerate motion): In this study, we focus on the (PP-RED) formulation. We simulate trajectories parallel to certain planes, leading to degenerate configurations. We then compare the estimation performance with and without incorporating these planes in the optimization. The results show that including such planes does not noticeably improve estimation accuracy.

B. Real-World Experiments

To evaluate the impact of planes and point–on–plane constraints in real-world scenarios, we compare three configurations on the EuRoC [30] and RPNG [3] datasets: a point-only formulation (PT-OL), a point–plane formulation without point-on-plane constraints (PP-NP), and a point–plane formulation with point-on-plane constraints (PP-ALL). All configurations share the same feature tracks and system parameters for a fair comparison. Since plane features are not directly observed, we adopt the plane extraction method in `ov_plane` [3] to detect planes and estimate their parameters from the visual measurements. The formal analysis assumes unbiased plane observations, correct point–plane associations, and no information reuse when planes are fitted from points. These assumptions are enforced in simulation using simulator-provided planes and associations. The real-world experiments, where planes are extracted from visual measurements, empirically evaluate the practical behavior under imperfect plane extraction and association. Table III reports representative EuRoC results, and the complete results on all evaluated EuRoC and RPNG sequences are provided in [26]. Overall, plane features improve the point-only formulation in most cases. However, point–on–plane constraints are sensitive to plane detection and point–plane association errors: points incorrectly associated with a plane may introduce biased geometric constraints. As a result, these constraints do not always provide additional gains and may slightly degrade performance in some sequences, as shown in Table III. Since they also introduce extra optimization factors and computational cost, point–on–plane constraints should be used with caution in practice.

TABLE III: Evaluation on the EuRoC dataset. Each entry reports Position ATE (m) / Orientation ATE (deg).

Seq.	PT-OL	PP-NP	PP-ALL
V101	0.0834 / 0.8123	0.0791 / 0.7814	**0.0756 / 0.7427**
V102	0.0947 / 1.5628	**0.0896 / 1.4973**	0.0912 / 1.5156
V103	0.1521 / 2.4136	0.1475 / 2.3618	**0.1418 / 2.2987**
V201	0.1246 / 1.0435	0.1208 / 1.0027	**0.1162 / 0.9614**
V202	0.0982 / 1.3374	**0.0915 / 1.2612**	0.0931 / 1.2784

VI. Discussion and Limitations

This paper analyzes how different point–plane optimization formulations in sliding-window monocular VINS affect information structure, marginalization, estimation accuracy, and computational cost. Based on our analysis, planes can replace many point landmarks with fewer geometric primitives, reducing backend optimization cost while maintaining comparable accuracy. This benefit, however, depends on plane selection: mutually non-parallel planes are preferred because they provide more independent geometric information.

The main limitation of this work is that the theoretical analysis assumes accurate plane observations and correct point–plane associations. In real-world systems, plane extraction and association may be noisy or incorrect, so point–on–plane factors may require additional outlier rejection and uncertainty modeling. In addition, this work focuses on local sliding-window estimation and does not consider plane-based loop closure or long-term map correction. Moreover, nullspace analysis characterizes gauge freedoms and rank deficiency, but not the strength, conditioning, or directional distribution of information in the observable subspace. Formulations with the same nullspace may therefore exhibit different accuracy, orientation performance, and robustness. The effects of marginalization and FEJ may also vary with motion and the linearization points [13], [31]. Another limitation is that our analysis treats point and plane measurements as independent features. In practical systems, however, planes are often fitted from the same visual measurements used to construct point factors, introducing unmodeled cross-correlations and possible information reuse [3]. Explicitly modeling these correlations, or avoiding reuse of the underlying point information, is left for future work.

VII. Conclusion and Future Work

This paper analyzed different point–plane optimization formulations for sliding-window monocular VINS, focusing on their effects on information structure, observability, marginalization, estimation accuracy, and computational cost. We compared point-only, plane-only, and joint point–plane formulations under different feature selections and plane configurations. Our results show that the benefit of planar structures depends on their formulation and geometric informativeness, rather than on simply adding more variables or constraints. Compact point–plane formulations can provide a favorable accuracy–efficiency trade-off, while point–on–plane constraints should be used cautiously when plane extraction and association are imperfect. Future work will extend the analysis to line features with loop closures.

References

[1] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.

[2] Y. Yang and G. Huang, "Observability analysis of aided ins with heterogeneous features of points, lines, and planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1399–1418, 2019.

[3] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Monocular visual-inertial odometry with planar regularities," in *Proc. of the IEEE International Conference on Robotics and Automation*, London, UK, 2023.

[4] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2105–2112.

[5] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4605–4611.

[6] M. Hosseinzadeh, Y. Latif, T. Pham, N. Suenderhauf, and I. Reid, "Structure aware slam using quadrics and planes," in *Computer Vision – ACCV 2018*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 410–426.

[7] L. Zhou, D. Koppel, H. Ju, F. Steinbruecker, and M. Kaess, "An efficient planar bundle adjustment algorithm," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2020, pp. 136–145.

[8] L. Zhang, H. Yin, W. Ye, and J. Betz, "Dp-vins: Dynamics adaptive plane-based visual-inertial slam for autonomous vehicles," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–16, 2024.

[9] S. Yang, Y. Hou, Y. Chen, H. Bao, and G. Zhang, "Recovering 3d planes from a single image via convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[10] F. Wu and G. Beltrame, "Direct sparse odometry with planes," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 557–564, 2022.

[11] Y. Zhang, F. Tang, Z. Xu, Y. Wu, and P. Ma, "Pgd-vio: A plane-aided rgb-d inertial odometry with graph-based drift suppression," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4276–4283, 2025.

[12] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[13] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Optimization-based vins: Consistency, marginalization, and fej," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1517–1524.

[14] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense rgb-d-inertial slam with map deformations," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6741–6748.

[15] C. X. Guo and S. I. Roumeliotis, "Imu-rgbd camera navigation using point and plane features," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3164–3171.

[16] D. Chen, S. Wang, W. Xie, S. Zhai, N. Wang, H. Bao, and G. Zhang, "Vip-slam: An efficient tightly-coupled rgb-d visual inertial planar slam," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5615–5621.

[17] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5110–5117.

[18] H. Wang, H. Wei, Z. Xu, Z. Lv, P. Zhang, N. An, F. Tang, and Y. Wu, "Rss: Robust stereo slam with novel extraction and full exploitation of plane features," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5158–5165, 2024.

[19] X. Li, Z. Wen, Z. Dong, H. Duan, T. Chen, and Z. Hong, "Snh-slam: Implicit dense slam based on scalable neural-hash representation," *IEEE Transactions on Multimedia*, pp. 1–12, 2026.

[20] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: A stereo slam system through the combination of points and line segments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.

[21] Y. Zhou, H. Zhang, M. Yang, and S. Scherer, "Efficient second-order plane adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. –.

[22] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-slam: Low-drift monocular slam in indoor environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6583–6590, 2020.

[23] S. Du, H. Guo, Y. Chen, Y. Lin, X. Meng, L. Wen, and F.-Y. Wang, "Gpo: Global plane optimization for fast and accurate monocular slam initialization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6254–6260.

[24] A. Concha and J. Civera, "Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5686–5693.

[25] C. Arndt, R. Sabzevari, and J. Civera, "Do planar constraints improve camera pose estimation in monocular slam?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[26] Y. Zhou, X. Wang, and C. Chen, "Tech report for "analysis of optimization-based monocular point–plane vins: Formulation, information, and marginalization"," 2026, supplementary material.

[27] M. Li, H. Zhang, T. Shen, Z. Zhou, and Z. Zhou, "Sm-vins: A fast and decoupled monocular visual–inertial sensors slam system with stepwise marginalization," *IEEE Sensors Journal*, vol. 24, no. 20, pp. 33 240–33 251, 2024.

[28] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[29] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Degenerate motion analysis for aided ins with online spatial and temporal sensor calibration," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2070–2077, 2019.

[30] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[31] C. Chen, Y. Yang, P. Geneva, and G. Huang, "Fej2: A consistent visual-inertial state estimator design," in *Proc. International Conference on Robotics and Automation*, 2022.