PLPO-KF: A Unified Pose-Only Kalman Filter With Point-Line Features for Visual-Inertial Odometry

Liqiang Wang, *Graduate Student Member, IEEE*, Hailiang Tang, Yan Wang, Guan Wang, Tisheng Zhang, *Member, IEEE*, and Xiaoji Niu, *Member, IEEE*

***Abstract*—The pose-only Kalman filter (PO-KF) for visual-inertial odometry (VIO) has demonstrated comparable localization accuracy to optimization-based systems while retaining the computational efficiency of filter-based approaches. However, the performance of point-based PO-KF degrades in textureless environments due to the scarcity of point features, while line features can serve as valuable supplements. To exploit the benefits of pose-only line-feature representation, we extend PO-KF with a pose-only line-feature measurement model and propose PLPO-KF, a unified pose-only representation-based Kalman filter for point-line-based VIO. Specifically, to eliminate reliance on 3-D line reconstruction and enable immediate line updates, we develop a trifocal tensor-based pose-only line-feature measurement model with a concise and clear measurement equation. In addition, we introduce an Inertial Navigation System (INS)-enhanced line optical-flow tracking method to ensure robust and accurate line feature association across multiple frames. Extensive experimental results on both public and private datasets demonstrate that PLPO-KF consistently outperforms state-of-the-art (SOTA) point-based and point-line-based VIO systems. Ablation studies further exhibit the effectiveness of both the proposed line pose-only measurement model and the INS-enhanced line tracking method.**

***Index Terms*—Line feature tracking, point-line-based system, pose-only representation, visual-inertial odometry (VIO).**

I. Introduction

VISUAL-INERTIAL odometry (VIO) systems combine measurements from a monocular camera and an inertial measurement unit (IMU) to estimate camera poses. Owing to their low cost, compact size, and high accuracy, VIO systems have become widely used in various Internet of Things (IoT) applications [1], [2]. Among existing approaches, feature-based tightly coupled VIO has emerged as the mainstream solution due to its robustness and precision. In this paradigm, visual features are extracted from images and fused with IMU poses, after which camera poses are updated using either optimization-based or filter-based estimators [3].

Points are the most widely used features in VIO systems due to their robustness and general applicability. However, in textureless environments, point features may become sparse, leading to degraded localization accuracy. To address this limitation, line features have been introduced as complementary features to enhance localization performance [4]. A critical challenge for line-based VIO is line extraction and association (tracking). Numerous geometry-based and learning-based methods have been developed to extract line features from single images [5]; yet, existing line association techniques remain insufficiently stable and accurate for VIO. Among them, optical-flow-based line tracking [6] demonstrates superior efficiency and accuracy but suffers from sensitivity to initial values, often causing failures in highly dynamic scenarios. Fortunately, the IMU in a VIO system provides high-frequency measurements and prior camera poses via the Inertial Navigation System (INS) before each new image [7], [8], [9]. Leveraging these prior poses together with historical line measurements enables the prediction of new line positions and provides better initialization for line tracking. Motivated by these challenges, we propose an INS-enhanced line optical-flow tracking method to improve the accuracy and robustness of line tracking under dynamic conditions.

When considering estimation methods in VIO systems, optimization-based approaches are generally regarded as more accurate, whereas filter-based methods, such as the extended Kalman filter (EKF) and multistate constraint kalman filter (MSCKF), are more efficient for real-time operation on low-cost hardware [3]. In our previous work, we demonstrated that the pose-only representation-based Kalman filter (PO-KF) achieves localization performance comparable to optimization-based methods while preserving the real-time efficiency of filter-based VIO [10]. These advantages make filter-based VIO particularly attractive for IoT platforms. Nevertheless, PO-KF still suffers in textureless environments due to the sparsity of point features. Prior research [11] has shown that line features provide an effective complement in such cases. There-

Received 10 November 2025; revised 7 December 2025; accepted 8 December 2025. Date of publication 15 December 2025; date of current version 26 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 42374034 and Grant 42504027, in part by the Key Research and Development Program of Hubei Province under Grant 2024BAB024, in part by the Major Program of Hubei Province under 2025BEA002 and Grant 2023BAA02602, and in part by the High Quality Development Project of MIIT under Grant 2024-182. *(Corresponding author: Tisheng Zhang.)*

Liqiang Wang is with the GNSS Research Center, Wuhan University, Wuhan 430079, China, and also with the Electronic Information School, Wuhan University, Wuhan 430079, China (e-mail: wlq@whu.edu.cn).

Hailiang Tang, Yan Wang, and Guan Wang are with the GNSS Research Center, Wuhan University, Wuhan 430079, China (e-mail: thl@whu.edu.cn; wystephen@whu.edu.cn; wanguan@whu.edu.cn).

Tisheng Zhang and Xiaoji Niu are with the GNSS Research Center, Wuhan University, Wuhan 430079, China, also with the the Electronic Information School, Wuhan University, Wuhan 430079, China, also with Hubei Technology Innovation Center for Spatiotemporal Information and Positioning Navigation, Wuhan 430079, China, and also with Hubei Luojia Laboratory, Wuhan 430079, China (e-mail: zts@whu.edu.cn; xjniu@whu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2025.3643950

fore, incorporating pose-only line-feature representations into PO-KF can combine the benefits of line features and the pose-only formulation. POPL-VIO [12] addresses this by deriving a pose-only line-feature representation from Plücker coordinates [13] and integrating it into our PO-KF. However, Plücker coordinates are not a minimal representation, which results in the complexity of the pose-only formulation in [12]. To overcome the above limitations, we propose a trifocal tensor-based pose-only measurement model for line features. Since the line trifocal tensor directly constrains line measurements and camera poses [14], the measurement equation in our model is more compact and concise. Building on this model, we design a unified pose-only representation-based Kalman filter with both point and line features, named PLPO-KF. The main contributions of this article are summarized as follows.

1) We propose an INS-enhanced line optical-flow tracking method that improves the accuracy and robustness of line tracking by leveraging prior INS poses to predict new line positions and provide better initialization for the tracking solution.
2) We propose a unified pose-only representation-based Kalman filter with point-line features, termed PLPO-KF, which extends our previous point-based PO-KF by incorporating a trifocal-tensor-based pose-only line-feature representation.
3) We conduct extensive evaluations of PLPO-KF on both public and private datasets. Experimental results demonstrate that PLPO-KF achieves superior localization accuracy than state-of-the-art (SOTA) point-based and point-line-based VIO systems.

The remainder of this paper is organized as follows. Section II reviews related works. Section III introduces the preliminaries of MSCKF-based VIO with line features. Section IV presents an overview of the proposed PLPO-KF, while Section V details the INS-enhanced line optical-flow tracking method. Section VI derives the proposed pose-only line-feature measurement model, and Section VII provides a comprehensive evaluation of PLPO-KF's localization performance. Finally, Section VIII concludes this article and discusses future work.

II. Related Works

VIO has attracted considerable research attention over the past two decades [3]. Among various approaches, feature-based VIO systems have been demonstrated to provide more robust and accurate localization, where point features are the most commonly relied-upon features. Nonetheless, point features often become sparse in textureless environments, leading to degraded performance. To address this limitation, researchers have explored incorporating additional geometric features, particularly line features, into VIO to improve accuracy and robustness [11], [15], [16]. In this part, we review related works on line feature association (tracking) and point-line-based VIO systems.

A. Line Extraction and Association

Line extraction is a fundamental problem in computer vision, and numerous studies have focused on fast and accurate detection of line features from images [5]. Classical geometry-based approaches include edge detection and Hough transform methods, gradient and region-growing-based methods such as LSD and ELSED [17], and edge-drawing with line-fitting methods such as EDLines [18]. More recently, deep learning-based methods such as L-CNN [19] and HAWP [20] have been proposed to enhance line extraction. However, these approaches mainly focus on extracting line features from individual images, without addressing line association across multiple frames.

Line association (tracking) across images is equally important [21]. Existing methods can be broadly categorized into three types: descriptor-based, point-based, and line-flow (LF)-based associations.

1) Descriptor-Based Association: This category computes line segment descriptors using either traditional or deep learning methods and matches descriptors between image frames. The gradient-based line binary descriptor (LBD) and deep learning-based descriptors, such as DLD [22] and WLD [23], compute line descriptors using the neighborhood of a line feature. SOLD^2 [24] leverages deep learning to adaptively select sampling points and compute line descriptors. The descriptor-based association methods are generally more accurate. However, the computation of descriptors is generally computationally intensive, making them less suitable for real-time VIO applications.

2) Point-Based Association: Points sampled along or near line features are used to establish line associations across frames. For example, Xu et al. [25] employed optical flow to track sampled points, associating two line features if most point features were successfully tracked between frames. AirSLAM [26] instead uses nearby points for line association, considering two line features matched if their nearby points satisfy predefined thresholds. Point-based methods are generally more efficient than descriptor-based approaches, but they often ignore the structural information of line features, making them more prone to mismatches. Furthermore, the choice of sampling points is usually heuristic and may not generalize well to line features of different lengths.

3) LF-Based Association: Wang et al. [27] proposed an LF-based simultaneous localization and mapping (SLAM) approach that exploits the spatiotemporal consistency of lines to form line flows, predict and extract new lines, and update existing flows [27]. However, this method requires 3-D positions of line features to establish spatial constraints, which limits its applicability. EPLF-VINS [6] introduced an optical-flow-based line tracking method that models 2-D line motion between frames as a combination of start-point translation and line rotation. The translation and rotation are iteratively estimated by enforcing photometric consistency across frames. Building upon this, MLINE-VINS [28] further accounts for length variations of line features, modeling them as linear changes and jointly estimating translation, rotation, and length parameters.

Compared with descriptor-based and point-based approaches, LF-based tracking directly estimates the new line position, thereby avoiding the negative impact of unstable line extraction. Moreover, it has a lower computational cost while preserving geometric information, making it well-suited

for consecutive-frame association in VIO systems. However, the iterative solution relies on accurate initialization, which may not hold under highly dynamic scenarios, leading to frequent tracking failures. To address this limitation, we propose an INS-enhanced line optical-flow tracking method, which improves stability and robustness across diverse motion conditions.

B. Point-Line-Based VIO Systems

Point-line-based VIO systems can be broadly categorized into optimization-based and filter-based methods, depending on the employed state estimators. Among optimization-based methods, PL-VIO [11] extends VINS-Mono [29] by incorporating line features, using LBD descriptors for line matching, and minimizing the pixel distance from observed line endpoints to the projected 2-D line features as the optimization objective. PL-VINS [30] further refines the line segment extraction method to ensure real-time performance. EPLF-VINS [6] and MLINE-VINS [28] employ optical-flow-based line tracking methods, significantly improving tracking efficiency and localization accuracy. AirSLAM [26] combines deep learning-based line features with nearby-point-based association, leveraging deep learning and relocalization for superior performance. Si et al. [31] proposed POL-VIO, which adopts a new measurement model based on midpoint position error and angular deviation between observed and projected lines. Xu et al. [25] and Ye et al. [28] integrated and classify lines as structural or non-structural to construct distinct constraints for state optimization. Filter-based point-line-based VIO systems are relatively few. Yang and Huang [32] integrated point and line features into the MSCKF framework and analyzed system observability. Zou et al. [15] developed StructVIO and StructSLAM, which leverage structural lines and employ EKF for state estimation.

The pose-only representation for point features, introduced by Cai et al. [33], decouples feature positions from camera poses and provides significant efficiency gains in visual 3-D reconstruction. This formulation has been adopted in optimization-based VIO systems such as PO-VINS [34] and PIPO-SLAM [35], yielding notable improvements in optimization efficiency. Building upon this, our prior work extends the point pose-only representation to a filter-based framework, proposing PO-KF [10], which achieves comparable localization accuracy to optimization-based methods while maintaining real-time performance. Zhang et al. [36] further analyzed how the immediate update strategy of the pose-only representation enhances filter-based localization accuracy.

Applying the same idea to line-based VIO systems, pose-only line-feature representations are expected to improve localization performance over traditional methods. Following this concept, Yang et al. [12] developed POPL-VIO, which derives a pose-only representation from Plücker coordinates. Although this approach achieves higher accuracy than the point-only MSCKF, Plücker coordinates are not minimal representations and lead to a rather complex representation for line features. In contrast, the line trifocal tensor directly constrains line measurements and camera poses, providing a more concise and intuitive line formulation. Motivated by this, we propose a trifocal tensor-based pose-only representation for line features and introduce PLPO-KF, a unified pose-only Kalman filter (PO-KF) VIO system that integrates both point and line features.

III. Preliminary of MSCKF-Based VIO With Line Features

To better introduce the proposed PLPO-KF, we present preliminaries on MSCKF-based VIO systems with line features in this section, including line representations, the trifocal tensor, and the MSCKF-based line-feature measurement model.

A. Line Representation

Lines can be represented in various forms. Plücker coordinates are suitable for coordinate transformation, while an orthogonal formulation provides a minimum parametric representation. Since cameras observe lines in 2-D images, the 2-D line parameterizations are also presented.

1) Plücker Coordinates Representation: The Plücker coordinates of a 3-D line are defined by the plane normal $\boldsymbol{n}$ (spanned by the line and the origin) and the line direction $\boldsymbol{v}$ [13]. In the world frame w, a line $\mathcal{L}$ is written as follows:

$$\mathcal{L}^w = \begin{bmatrix} \boldsymbol{n}^w \\ \boldsymbol{v}^w \end{bmatrix}. \tag{1}$$

The distance from the origin to the line is $d^w = (\|\boldsymbol{n}^w\|)/(\|\boldsymbol{v}^w\|)$.

The Plücker transformation maps line coordinates between frames. For a camera pose $(\mathbf{R}_c^w, \boldsymbol{p}_c^w)$, the mapping from the camera frame to the world frame is given by

$$\mathcal{L}^w = \begin{bmatrix} \boldsymbol{n}^w \\ \boldsymbol{v}^w \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^w & [\boldsymbol{p}_c^w]_\times \mathbf{R}_c^w \\ \mathbf{0} & \mathbf{R}_c^w \end{bmatrix} \begin{bmatrix} \boldsymbol{n}^c \\ \boldsymbol{v}^c \end{bmatrix} = \mathcal{T}_c^w \mathcal{L}^c. \tag{2}$$

2) Orthogonal Representation: Plücker coordinates use six parameters for a 3-D line, which is redundant since a line has four degrees of freedom. Hence, Kottas and Roumeliotis [37] proposed an orthogonal representation with a distance scalar d_l and a unit quaternion $\mathbf{q}_l$ that encodes the line orientation. A convenient construction of the associated rotation matrix is

$$\mathbf{R}_l(\mathbf{q}_l) = \left[\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|} \quad \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \quad \frac{\boldsymbol{n}\times\boldsymbol{v}}{\|\boldsymbol{n}\times\boldsymbol{v}\|} \right]. \tag{3}$$

The error distribution of the rotation matrix and distance scalar can be expressed as follows:

$$\hat{\mathbf{R}}_l^w = \mathbf{R}_l^w \left(\boldsymbol{I} - [\boldsymbol{\theta}_l]_\times \right), \quad \hat{d}_l = d_l - \delta d_l \tag{4}$$

where $\boldsymbol{\theta}_l$ denotes the error state of the line direction, and δd_l represents the distance error. The minimal 4-D error state of the line orthogonal representation is $\delta \boldsymbol{p}_l = \left[\boldsymbol{\theta}_l^T \ \delta d\right]^T$.

3) 2-D Line Parameterization: When observing a 3-D line in the 2-D image plane, we obtain the start point $\boldsymbol{p}_s^p$ and endpoint $\boldsymbol{p}_e^p$, and compute its 2-D line parameters $\boldsymbol{l}^p = \left[l_1^p \ l_2^p \ l_3^p\right]$. The camera intrinsic parameters, including focal lengths, f_x and f_y, and principal point, c_x and c_y, can convert the line's endpoints from the unified camera coordinates to the pixel frame. Therefore, the transformation of 2-D line parameters from the unified camera frame to the pixel frame is

$$\begin{bmatrix} l_1^p \\ l_2^p \\ l_3^p \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 \\ -\frac{c_x}{f_x} & -\frac{c_y}{f_y} & 1 \end{bmatrix} \begin{bmatrix} l_1^u \\ l_2^u \\ l_3^u \end{bmatrix}. \tag{5}$$

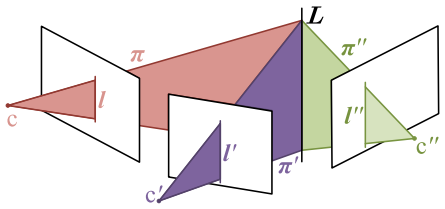

Fig. 1. Line 2-D projections of 3-D line $\boldsymbol{L}$ in three camera views.

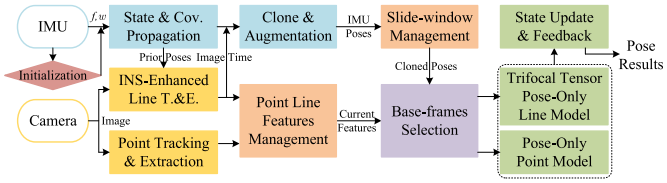

Fig. 2. Overview of the proposed PLPO-KF. T.&E.: Tracking and extraction.

B. Trifocal Tensor for Line Features

The trifocal tensor describes the geometric relationship among the projections of a 3-D line observed in three views. As illustrated in Fig. 1, the projections of a 3-D line $\boldsymbol{L}$ in the image planes of cameras c, c', and c'' are denoted as $\boldsymbol{l}$, $\boldsymbol{l}'$, and $\boldsymbol{l}''$, respectively. Taking camera c as the reference frame, the projection matrices of the three cameras are given by

$$\boldsymbol{P} = \left[\boldsymbol{I}_3 \ \boldsymbol{0}_{3\times 1}\right], \quad \boldsymbol{P}' = \left[\mathbf{A} \ \boldsymbol{a}_4\right], \quad \boldsymbol{P}'' = \left[\mathbf{B} \ \boldsymbol{b}_4\right] \tag{6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the rotation (or homography-at-infinity) matrices from camera c to c' and c'', while $\boldsymbol{a}_4$ and $\boldsymbol{b}_4$ represent the epipoles induced by camera c in c' and c'', respectively.

As shown in Fig. 1, each 2-D line feature on the image plane can be back-projected into a 3-D plane. These planes can be expressed using the camera matrices as:

$$\boldsymbol{\pi} = \boldsymbol{P}^T\boldsymbol{l}, \quad \boldsymbol{\pi}' = \boldsymbol{P}'^T\boldsymbol{l}', \quad \boldsymbol{\pi}'' = \boldsymbol{P}''^T\boldsymbol{l}'' \tag{7}$$

where $\boldsymbol{l}$ denotes the homogeneous representation of a 2-D line feature and $\boldsymbol{\pi}$ denotes the homogeneous representation of the corresponding 3-D plane.

Since the three planes $\boldsymbol{\pi}$, $\boldsymbol{\pi}'$, and $\boldsymbol{\pi}''$ intersect at the same 3-D line, the matrix formed by stacking them must have rank 2, i.e., $\text{rank}([\boldsymbol{\pi}\ \boldsymbol{\pi}'\ \boldsymbol{\pi}'']) = 2$. Thus, one plane can be expressed as a linear combination of the other two. For example,

$$\boldsymbol{\pi} = \alpha\boldsymbol{\pi}' + \beta\boldsymbol{\pi}''. \tag{8}$$

Substituting the above expressions into (8), the relationship among the three 2-D line features can be derived as follows:

$$\boldsymbol{l} = \left(\boldsymbol{l}'\mathbf{A}\boldsymbol{b}_4^T\boldsymbol{l}''\right)^T - \boldsymbol{l}'^T\boldsymbol{a}_4\mathbf{B}^T\boldsymbol{l}''. \tag{9}$$

Let $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ denote the ith column of $\mathbf{A}$ and $\mathbf{B}$, respectively. Define $\boldsymbol{T}_i = \boldsymbol{a}_i\boldsymbol{b}_4^T - \boldsymbol{a}_4\boldsymbol{b}_i^T$, the above relationship can be simplified as:

$$l_i = \boldsymbol{l}'^T\boldsymbol{T}_i\boldsymbol{l}'', \quad i = 1, 2, 3. \tag{10}$$

The set $\boldsymbol{T}_1, \boldsymbol{T}_2$, and $\boldsymbol{T}_3$ constitutes the trifocal tensor for line features. Hence, (9) can be expressed compactly as follows:

$$\boldsymbol{l} = \{l_1, l_2, l_3\} = \boldsymbol{l}'^T\{\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3\}\boldsymbol{l}''. \tag{11}$$

C. MSCKF-Based Line-Feature Measurement Model

The update strategy of the MSCKF-based line-feature VIO is similar to that of the point-feature MSCKF. Specifically, the system state is updated either when a tracked line feature fails or when a line feature reaches the maximum tracking length. The selected line features are first triangulated using the cloned poses, and all corresponding measurements are then used to update the system through the line measurement model. This section introduces the line-feature measurement model adopted in the MSCKF-based VIO.

Projecting a triangulated 3-D line feature $\boldsymbol{L}$ onto each measurement pixel frame, the line measurement residuals are obtained by computing the distances between the observed start and end points and their corresponding projected 2-D line features in the image plane. The residuals of this line feature in the c_i-frame can be expressed as follows [38], [39]:

$$\boldsymbol{z}_l^{p_i} = \mathbf{h}_\ell\left(\mathbf{h}_p\left(\mathbf{h}_t\left(\mathbf{h}_\mathcal{L}\left(\boldsymbol{p}_l^w\right), \mathbf{R}_w^{c_i}, \boldsymbol{p}_{c_i}^w\right), \boldsymbol{K}\right), \tilde{\boldsymbol{p}}_s^{p_i}, \tilde{\boldsymbol{p}}_e^{p_i}\right) \tag{12}$$

where $\boldsymbol{p}_l^w$ denotes the orthogonal representation of the triangulated 3-D line, $\tilde{\boldsymbol{p}}_s^{p_i}$ and $\tilde{\boldsymbol{p}}_e^{p_i}$ are the observed start and end points in the c_i-frame, $\boldsymbol{K}$ is the camera projection matrix, $\mathbf{h}_\mathcal{L}$ represents the line transformation function, $\mathbf{h}_t$ is the coordinate transformation function, $\mathbf{h}_p$ is the projection function, and $\mathbf{h}_\ell$ computes the distances from the endpoints to the corresponding line feature.

Stacking the measurements of all features and simplifying the equation, we obtain the complete line-feature MSCKF measurement equation as follows:

$$\boldsymbol{z}^p = \boldsymbol{H}_x\boldsymbol{x} + \boldsymbol{H}_l\delta\boldsymbol{p}_l^w + \boldsymbol{n}^p \tag{13}$$

where $\boldsymbol{H}_x$ and $\boldsymbol{H}_l$ are the Jacobian matrices with respect to the state vector and the line feature position, respectively, $\boldsymbol{z}^p$ denotes the measurement innovation, and $\boldsymbol{n}^p$ represents the measurement noise. Finally, a nullspace projection is applied to eliminate the line feature position from (13), followed by measurement compression and a Kalman update to update the system state.

IV. System Overview of PLPO-KF

The system overview of the proposed PLPO-KF is depicted in Fig. 2. PLPO-KF is a point-line-based, filter-based VIO system that extends our previous work, PO-KF [10]. The processing flow of PLPO-KF is as follows. After system initialization, IMU data are processed for navigation state recurrence, covariance propagation, state clone, and augmentation. When a new image arrives, both point and line feature tracking and extraction (T.&E.) are executed in parallel, where INS-enhanced optical-flow-based methods are employed for line tracking. The cloned IMU poses are maintained using a sliding window strategy, while the point and line features are managed accordingly. When enough visual measurements are available for the state update, the base frames for both point and line features are selected, and their pose-only measurement models are then constructed. Finally, these visual measurements are incorporated to update the system state. The feature

Algorithm 1 PLPO-KF Processing Pipeline

Require: IMU stream, image stream, system hyper-parameters(installation parameters, noise, window size W et al.).
Ensure: Estimated trajectory, covariance, and calibrated parameters.
Initialize: IMU state, covariance, system parameters.
while data streams active **do**
  IMU buffering: buffer IMU data until new image arrives.
  if new image (t_l) arrives **then**
    IMU propagation: propagate IMU state and covariance to t_l.
    State clone & augmentation: clone current IMU pose into the sliding window, augment into the state vector and covariance.
    Point T.&E.: track existing points; extract new points as needed.
    Line T.&E.: predict line parameters and track existing lines; NCC-based expansion; extract new lines as needed.
    Feature management: update point and line features.
    if enough visual measurements for update **then**
      Base-frame selection: select base-frames for all features.
      Pose-Only modeling: construct point pose-only models and trifocal tensor line pose-only models.
      Outlier rejection: perform chi-square test on each feature.
      Kalman update: update state with all visual measurements.
    end if
    if number of clones > W **then**
      Marginalization: remove measurements associated with the oldest clone; remove the oldest clone from the state; marginalize the corresponding covariance rows/columns
    end if
  end if
end while

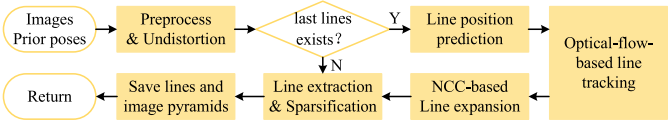

Fig. 3. Processing flow of the proposed line optical-flow tracking.

management and state marginalization processes will proceed if the number of cloned poses exceeds the maximum size. The detailed processing pipeline is summarized in Algorithm 1.

Since PLPO-KF inherits the same system architecture and all functionalities of PO-KF, it shares the same IMU kinematic and state equations as PO-KF, together with online calibration of camera-IMU extrinsics and time delay. In addition, feature management and base-frame selection for line features follow principles similar to those used for point features in PO-KF. The modules dedicated to INS-enhanced line T.&E., as well as the pose-only line measurement model, are described in detail in the following sections.

V. INS-Enhanced Line Optical-Flow Tracking

This section presents the proposed INS-enhanced line T.&E. module in PLPO-KF. The processing pipeline is shown in Fig. 3. When a new image arrives, it is first preprocessed using histogram equalization and undistortion. The subsequent procedure depends on whether line features from the previous frame are available. If no previous line features exist, we directly extract new line features using the ELSED method [40], followed by sparsification with occupancy masks. If previous line features are available, we predict their positions in the new image and refine them using an optical-flow-based line tracking method. The tracked line features are then expanded using a normalized cross-correlation (NCC) strategy. To maintain a sufficient number of features, new line features are further extracted and sparsified. Finally, we save the current detected line features and images for the next tracking.

Among the processing steps in Fig. 3, the key components of the proposed method are line position prediction, line optical-flow tracking, and line expansion. For clarity, we first describe the basic line optical-flow tracking method, followed by the INS-aided line position prediction and the NCC-based line expansion algorithm.

A. Line Optical-Flow Tracking

The fundamental principle of optical-flow tracking is the photometric consistency assumption, which states that the photometric intensity of a point remains constant between two consecutive frames

$$\begin{aligned} I\left(u', v', t+\delta t\right) &= I\left(u, v, t\right) + I_u \delta u + I_v \delta v + I_t \delta t \\ 0 &= I_u \delta u + I_v \delta v + I_t \delta t \end{aligned} \tag{14}$$

where I denotes the photometric intensity, (u, v) and (u', v') denote the pixel coordinates at time t and $t+\delta t$. I_u, I_v, and I_t are the image gradients along the u and v directions and the temporal gradient, respectively.

Sampling n points along the line feature, denoting the line's angle at time t as θ, and denoting the ith point in this line feature as $\boldsymbol{p}_i = (u_i, v_i)$, the corresponding ith point at time $t+\delta t$ can be expressed as follows:

$$\begin{aligned} u_i' &= u_i + g_1 + l_i \cos\left(\theta + g_3\right) - l_i \cos\left(\theta\right) \\ v_i' &= v_i + g_2 + l_i \sin\left(\theta + g_3\right) - l_i \sin\left(\theta\right) \end{aligned} \tag{15}$$

where l_i is the distance from the ith point to the line's start point, g_1 and g_2 are the offsets in the u- and v-directions of the line's start point, and g_3 is the angle offset of the line feature from time t to $t+\delta t$. Notably, although the length l_i is actually not constant in image frames at time t and $t+\delta t$, we neglect this change considering the limited and complicated change. In addition, the angle change g_3 is also considered tiny between two consecutive frames.

Substituting (15) into (14), we obtain the constraint equation for the ith point as follows:

$$\begin{bmatrix} I_{u,i} & I_{v,i} \end{bmatrix} \begin{bmatrix} 1 & 0 & -l_i \sin\left(\theta\right) \\ 0 & 1 & l_i \cos\left(\theta\right) \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = -I_{t,i} \delta t. \tag{16}$$

We extract a patch around all n sampling points and substitute all points within these patches into (16). Then the line optical-flow parameters g_1–g_3 are iteratively estimated using the Gauss–Newton method.

B. Line Prediction Using Prior INS Pose

The initial values of the line optical-flow parameters g_1–g_3 play a crucial role in ensuring convergence of the Gauss–Newton method. We predict the line positions using the INS prior poses and calculate initial estimates of the optical-flow parameters in this section.

Based on historical measurements, we classify line features into two categories: 1) lines with sufficient measurements and parallax and 2) lines with insufficient measurements or parallax. We then describe the prediction methods for these two cases separately.

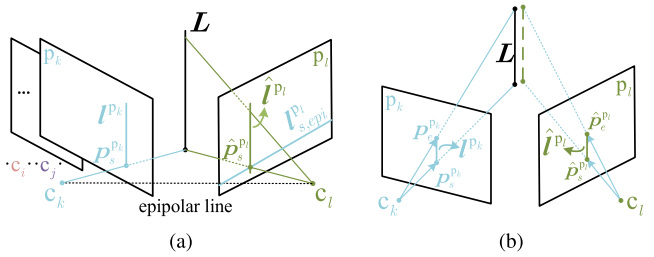

Fig. 4. Predicted line positions in the new camera frame using the prior INS pose. (a) Predicted line with line trifocal tensor. (b) Predicted line with only rotation compensation.

1) Lines With Sufficient Measurement and Parallax: For line features with sufficient parallax, the trifocal tensor is employed to predict their 2-D parameters. To enhance prediction accuracy, we select as base frames the two historical camera frames with the largest parallax angle. Taking the current camera frame as the reference frame and substituting the line's 2-D parameters together with the poses of the two base frames into (9), We obtain the predicted line 2-D parameters in the current camera frame $\hat{\boldsymbol{l}}^{p_l}$, as illustrated in Fig. 4(a).

Denoting the last camera frame as c_k (c_k can be one of the two base frames) and the line 2-D parameters in this frame as $\boldsymbol{l}^{p_k}$, we can predict the initial value of the angle offset g_3 as follows:

$$g_3(0) = \arctan\left(-\frac{\hat{l}_1^{p_l}}{\hat{l}_2^{p_l}}\right) - \arctan\left(-\frac{l_1^{p_k}}{l_2^{p_k}}\right) \tag{17}$$

where angular wrap-around is properly compensated.

Since $\hat{\boldsymbol{l}}^{p_l}$ defines an infinite line, we can only calculate the initial rotation, not the initial position offset. Denote the start point in the c_k frame as $\boldsymbol{p}_s^{c_k}$ and its counterpart in the current camera as $\boldsymbol{p}_s^{c_l}$. Their epipolar constraint can be described as:

$$\underbrace{\left(\boldsymbol{p}_s^{c_k}\right)^T \left[\boldsymbol{p}_{c_l}^{c_k}\right]_\times \mathbf{R}_{c_l}^{c_k}}_{\left(\boldsymbol{l}_{s,\text{epi}}^{c_l}\right)^T} \boldsymbol{p}_s^{c_l} = 0. \tag{18}$$

The epipolar constraint is also the epipolar line parameters for the line's start point $\boldsymbol{p}_s^{c_l}$, denoted as $\boldsymbol{l}_{s,\text{epi}}^{c_l}$. We transform the line parameters to the corresponding pixel frame p_l using (5), yielding $\boldsymbol{l}_{s,\text{epi}}^{p_l}$. As shown in the p_l frame of Fig. 4(a), the intersection point of the epipolar line $\boldsymbol{l}_{s,\text{epi}}^{p_l}$ and the predicted line $\hat{\boldsymbol{l}}^{p_l}$ is the predicted start point $\hat{\boldsymbol{p}}_s^{p_l}$. Then, we can obtain the initial values of the start-point offsets as follows:

$$\left[g_1(0)\ g_2(0)\right]^T = \hat{\boldsymbol{p}}_s^{p_l} - \boldsymbol{p}_s^{p_k}. \tag{19}$$

2) Lines With Insufficient Measurement or Parallax: We lack depth information of line features with insufficient measurements and parallax, which prevents us from predicting the accurate line 2-D parameters in the current camera frame. Nevertheless, we can still compensate for the rotation information using the prior INS pose and predict coarse optical-flow parameters, as illustrated in Fig. 4(b).

The visual measurements of line's endpoints are only their direction in the camera frame, denoted as $\boldsymbol{p}_s^{c_k}$ and $\boldsymbol{p}_e^{c_k}$. To compensate for the prior INS rotation, we transfer the direction of the line's endpoints from camera frame c_k to the current camera frame as follows:

$$\begin{aligned} \hat{\boldsymbol{p}}_s^{p_l} &= \boldsymbol{K}\mathbf{R}_{c_k}^{c_l}\boldsymbol{p}_s^{c_k} \\ \hat{\boldsymbol{p}}_e^{p_l} &= \boldsymbol{K}\mathbf{R}_{c_k}^{c_l}\boldsymbol{p}_e^{c_k}. \end{aligned} \tag{20}$$

Then, we can obtain the initial values of the start point offsets using (19) and predict the initial value of the angle offset g_3 as follows:

$$g_3(0) = \arctan\left(\frac{\hat{p}_{e,v}^{p_l} - \hat{p}_{s,v}^{p_l}}{\hat{p}_{e,u}^{p_l} - \hat{p}_{s,u}^{p_l}}\right) - \arctan\left(-\frac{l_1^{p_k}}{l_2^{p_k}}\right) \tag{21}$$

with angular wrap-around properly compensated. These predicted optical-flow parameters are then employed to initialize the iterative solver.

3) Analysis of the INS-Enhanced Line Tracking: Initial values play a crucial role in the convergence of iterative line optical-flow tracking. In the basic method, where no prior information is available, the initial parameters are simply set to 0. As a result, the basic approach becomes highly vulnerable to large interframe motions. In contrast, the proposed INS-enhanced method leverages INS prior poses to predict the initial optical-flow parameters, thereby significantly improving both the convergence and accuracy of line optical-flow tracking.

To intuitively demonstrate the advantages of the proposed method, we present the tracking results of the basic optical-flow method under dynamic conditions in Fig. 5(a) and those of the proposed INS-enhanced method in Fig. 5(b). The orange lines in the right image of Fig. 5(a) are the initial positions for the basic line optical-flow solution, which are actually the positions of the previous image. The difference between the orange lines and their actual positions is quite large, indicating the large interframe dynamic. As a result of the inaccurate initial positions, the basic optical-flow method yields numerous tracking failures (red lines). In contrast, the predicted orange lines in Fig. 5(b) almost coincide with their

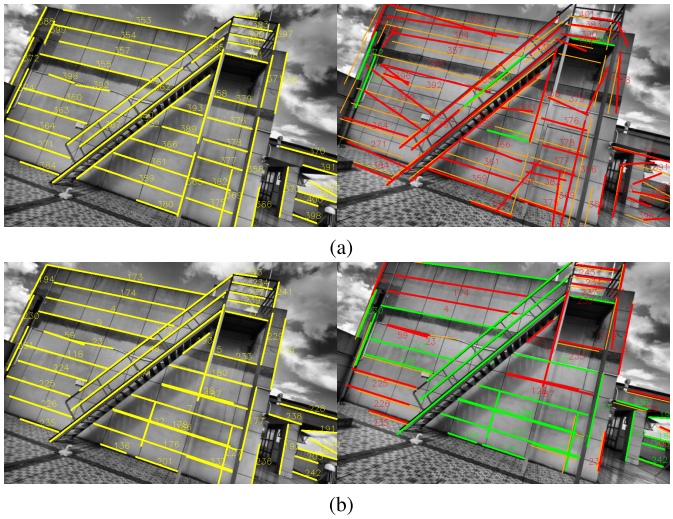

Fig. 5. Line optical-flow tracking results under dynamic conditions. (a) Basic method. (b) INS-enhanced method. Previous frame (left) and current frame (right). Previous lines (yellow); predicted lines (orange); failed tracks (red); successfully tracked lines (green); line IDs (numbers).

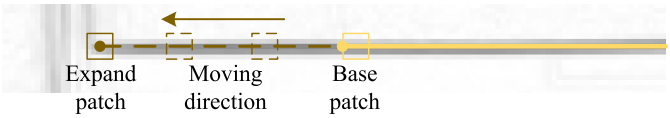

Fig. 6. Patch NCC-based line expansion in the gradient magnitude image. The yellow line and point represent the tracked line and its endpoint; the brown dotted line and point represent the extended part and the new endpoint.

corresponding positions in the new image. Consequently, the INS-enhanced method produces a higher number of successfully tracked lines (green). Furthermore, even when tracking fails, the INS-enhanced results remain close to their true positions, whereas the failures of the basic method appear disordered and scattered. These observations demonstrate that the INS-enhanced method not only improves the accuracy and convergence of line optical-flow tracking but also mitigates system instability caused by the random divergence of tracking solutions.

C. Patch NCC-Based Line Expansion

Due to viewpoint changes and limitations of line extraction methods, the tracked line features in new images are often shorter than their true counterparts. To avoid progressive shortening over time, we extend the tracked line features after line tracking. The process is illustrated in Fig. 6, where line expansion is performed on the gradient magnitude image.

At each endpoint, we take a patch as the reference patch for NCC comparison. The candidate patch is then shifted along the line direction by one patch size, and its NCC with the reference patch is computed as follows:

$$\mathrm{NCC}(T,I)=\frac{\sum_{i=1}^{n}T_iI_i}{\sqrt{\sum_{i=1}^{n}T_i^2}\sqrt{\sum_{i=1}^{n}I_i^2}} \tag{22}$$

where T and I are the reference and candidate patches, and n is the patch pixel size.

If the NCC value exceeds a predefined threshold, the candidate patch is accepted as a valid extension, and the process continues. When the NCC falls below the threshold or the patch exceeds the image boundary, the process stops, and the center of the last valid patch is taken as the new endpoint. The same procedure is applied to the other endpoint, resulting in the extended line feature.

VI. Pose-Only Line-Feature Measurement Model

A. Pose-Only Line-Feature Representation

Line 2-D parameters are straightforwardly employed in line measurement equations. Therefore, we employ a trifocal tensor-based pose-only representation that describes the line's new 2-D parameters instead of its 3-D position. To deduce the pose-only line-feature representation, we denote the current camera frame as c_l, and the two base frames as c_i and c_j, and present the line representation in Fig. 7(a). The line measurements in their respective pixel frames are expressed as $\boldsymbol{l}^{p_i}$, $\boldsymbol{l}^{p_j}$, and $\boldsymbol{l}^{p_l}$. Taking the c_l camera frame as the reference

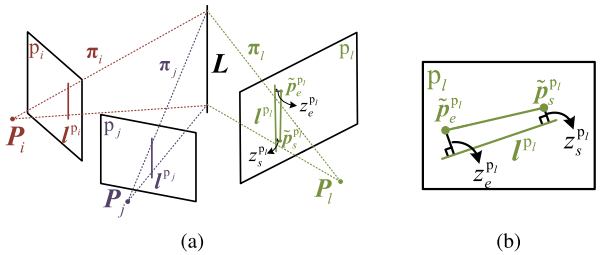

Fig. 7. (a) Trifocal tensor-based line-feature representation and measurement innovation. (b) Detail illustration of the line-feature measurement innovation.

frame, we obtain the camera matrices of the three views for this line

$$\boldsymbol{P}_i=\left[\mathbf{R}_{c_l}^{c_i}\ \boldsymbol{p}_{c_l}^{c_i}\right],\quad \boldsymbol{P}_j=\left[\mathbf{R}_{c_l}^{c_j}\ \boldsymbol{p}_{c_l}^{c_j}\right],\quad \boldsymbol{P}_l=\left[\boldsymbol{I}_3\ \mathbf{0}_{3\times1}\right]. \tag{23}$$

Substituting the camera matrices into (9), we derive the line's 2-D parameters in the current image frame as follows:

$$\boldsymbol{l}^{p_l}=\left(\left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_{c_l}^{c_i}\left(\boldsymbol{p}_{c_l}^{c_j}\right)^T\boldsymbol{l}^{p_j}\right)^T-\left(\boldsymbol{l}^{p_i}\right)^T\boldsymbol{p}_{c_l}^{c_i}\left(\mathbf{R}_{c_l}^{c_j}\right)^T\boldsymbol{l}^{p_j}. \tag{24}$$

According to the definition in (10), we obtain the trifocal tensor matrices for the current line feature $\boldsymbol{l}^{p_l}$ as follows:

$$\boldsymbol{T}_m=\mathbf{R}_{c_l}^{c_i}\boldsymbol{e}_m\left(\boldsymbol{p}_{c_l}^{c_j}\right)^T-\boldsymbol{p}_{c_l}^{c_i}\left(\mathbf{R}_{c_l}^{c_j}\boldsymbol{e}_m\right)^T \tag{25}$$

where $m=1,2,3$, $\boldsymbol{e}_m$ is the mth column of a 3-D identity matrix. Substituting (25) back into (24), we can express the trifocal tensor-based pose-only line-feature representation as follows:

$$\boldsymbol{l}^{p_l}=\left\{l_1^{p_l},l_2^{p_l},l_3^{p_l}\right\}=\left(\boldsymbol{l}^{p_i}\right)^T\left\{\boldsymbol{T}_1,\boldsymbol{T}_2,\boldsymbol{T}_3\right\}\boldsymbol{l}^{p_j}. \tag{26}$$

The derived trifocal-based pose-only line-feature representation is obviously concise and intuitive, making it easily integrated in the subsequent line measurement model.

B. Measurement Update

We construct the line measurement model at the newest lth pixel frame using the observed line endpoints $\tilde{\boldsymbol{p}}_s^{p_l}$ *and* $\tilde{\boldsymbol{p}}_e^{pl}$ and the pose-only line-feature representation $\boldsymbol{l}^{p_l}$. The measurement residuals in the lth pixel frame are defined as the distances from the observed endpoints to the corresponding projected 2-D line feature, as illustrated in Fig. 7(b). The measurement equation is formulated as folllows:

$$\boldsymbol{z}_l^{p_l}=\mathbf{h}_\ell\left(\mathbf{h}_r\left(\boldsymbol{l}^{p_i},\boldsymbol{l}^{p_j},\mathbf{R}_w^{c_i},\boldsymbol{p}_{c_i}^w,\mathbf{R}_w^{c_j},\boldsymbol{p}_{c_j}^w,\mathbf{R}_w^{c_l},\boldsymbol{p}_{c_l}^w\right),\tilde{\boldsymbol{p}}_s^{p_l},\tilde{\boldsymbol{p}}_e^{p_l}\right) \tag{27}$$

where $\boldsymbol{z}_l^{p_l}$ is the measurement innovation, $\mathbf{h}_\ell$ is the point-to-line distance function, and $\mathbf{h}_r$ is the pose-only line-feature representation function. Illustratively, Fig. 7(a) and (b) visualizes the $\mathbf{h}_r$ and $\mathbf{h}_\ell$ formulations, respectively.

The pose-only line-feature measurement model (27) composes two functions, which yield a more compact and intuitive form than the MSCKF-based line-feature measurement model (12). Using the trifocal tensor-based line-feature representation also makes our measurement model also more concise than the Plücker-based pose-only model in [12].

The camera poses required in (27) are computed from the camera-IMU extrinsic parameters, and the cloned IMU

poses. Hence, the proposed pose-only line-feature measurement model depends only on the state vector and can be simplified as follows:

$$\boldsymbol{z}_l^{p_l} = \boldsymbol{H}_{x,l}\boldsymbol{x} + \boldsymbol{n}^{p_l} \tag{28}$$

where $\boldsymbol{H}_{x,l}$ is the Jacobian with respect to the state vector and $\boldsymbol{n}^{p_l}$ is the measurement noise. Then, we derive the complete measurement equation and Jacobian matrix.

1) Jacobian of the Distance Function: Denoting the projected line in the lth camera image frame as $\boldsymbol{l}^{p_l} = \mathbf{h}_r$, the distance from the observed line endpoints to the projected 2-D line feature are

$$z_s^{p_l} = \frac{\tilde{\boldsymbol{p}}_s^{p_l} \cdot \boldsymbol{l}^{p_l}}{\sqrt{\left(l_1^{p_l}\right)^2 + \left(l_2^{p_l}\right)^2}}, \quad z_e^{p_l} = \frac{\tilde{\boldsymbol{p}}_e^{p_l} \cdot \boldsymbol{l}^{p_l}}{\sqrt{\left(l_1^{p_l}\right)^2 + \left(l_2^{p_l}\right)^2}} \tag{29}$$

where z_s and z_e are the distances from the observed start and end points to the projected line in the current pixel frame, respectively.

The Jacobian of the start point distance z_s with respect to the projected line feature is as follows:

$$\boldsymbol{J}_{\boldsymbol{l}^{p_l}}^{z_s} = \frac{\left(\tilde{\boldsymbol{p}}_s^{p_l}\right)^T}{\ln} - \frac{\tilde{\boldsymbol{p}}_s^{\mathbf{p}_l} \cdot \boldsymbol{l}^{p_l}}{\ln^2}\begin{bmatrix}\frac{l_1^{p_l}}{\ln} & \frac{l_2^{p_l}}{\ln} & 0\end{bmatrix} \tag{30}$$

where $\ln = ((l_1^{p_l})^2 + (l_2^{p_l})^2)^{1/2}$. The Jacobian of z_e with respect to $\boldsymbol{l}^{p_l}$ can be derived analogously.

By stacking both endpoints, the line measurement innovation in the lth frame is expressed as: $\boldsymbol{z}_l^{p_l} = -\begin{bmatrix}z_s^{p_l} & z_e^{p_l}\end{bmatrix}^T$, and the Jacobian with respect to the projected line $\boldsymbol{l}^{p_l}$ becomes

$$\boldsymbol{J}_{\boldsymbol{l}^{p_l}}^{\boldsymbol{z}_l^{p_l}} = -\begin{bmatrix}\boldsymbol{J}_{\boldsymbol{l}^{p_l}}^{z_s} \\ \boldsymbol{J}_{\boldsymbol{l}^{p_l}}^{z_e}\end{bmatrix}. \tag{31}$$

2) Jacobian of the Line Pose-Only Representation to Camera Poses: As derived in (26) of Sec. VI-A, the line 2-D parameters $\boldsymbol{l}^{p_l}$ are functions of the c_i, c_j, and c_l camera poses. The mth element of $\boldsymbol{l}^{p_l}$ is expressed as follows:

$$l_m^{p_l} = \left(\boldsymbol{l}^{p_i}\right)^T \boldsymbol{T}_m \boldsymbol{l}^{p_j}. \tag{32}$$

The error-state perturbation method is employed in (32) to derive the Jacobian of the mth element in $\boldsymbol{l}^{p_l}$ with respect to the camera poses. Denoting the error-state of the ith camera pose as $\delta\boldsymbol{T}_{c_i}^w = \begin{bmatrix}\boldsymbol{\theta}_{c_i}^T & (\delta\boldsymbol{p}_{c_i}^w)\end{bmatrix}^T$ and substituting the error-state perturbation into (25) and (32), we obtain the Jacobian matrices as follows:

$$\begin{aligned}
\boldsymbol{J}_{T_{c_i}^w}^{l_m^{p_l}} &= \begin{bmatrix}\left(\boldsymbol{l}^{p_i}\right)^T\left[\mathbf{R}_{c_l}^{c_i}\boldsymbol{e}_m\left(\boldsymbol{p}_{c_l}^{c_j}\right)^T\boldsymbol{l}^{p_j}\right]_\times & \boldsymbol{e}_m^T\mathbf{R}_{c_j}^{c_l}\boldsymbol{l}^{p_j}\left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_w^{c_i} \\ -\left(\boldsymbol{l}^{p_i}\right)^T\left[\boldsymbol{p}_{c_l}^{c_i}\boldsymbol{e}_m^T\mathbf{R}_{c_j}^{c_l}\boldsymbol{l}^{p_j}\right]_\times & \end{bmatrix} \\
\boldsymbol{J}_{T_{c_j}^w}^{l_m^{p_l}} &= \begin{bmatrix}-\left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_{c_l}^{c_i}\boldsymbol{e}_m\left(\boldsymbol{p}_{c_l}^{c_j}\right)^T[\boldsymbol{l}^{p_j}]_\times & -\boldsymbol{l}^{p_i}\mathbf{R}_{c_l}^{c_i}\boldsymbol{e}_m\left(\mathbf{R}_{c_j}^w\boldsymbol{l}^{p_j}\right)^T \\ +\left(\boldsymbol{l}^{p_i}\right)^T\boldsymbol{p}_{c_l}^{c_i}\boldsymbol{e}_m^T\mathbf{R}_{c_j}^{c_l}[\boldsymbol{l}^{p_j}]_\times & \end{bmatrix} \\
\boldsymbol{J}_{T_{c_l}^w}^{l_m^{p_l}} &= \begin{bmatrix}-\left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_{c_l}^{c_i}\left[\boldsymbol{e}_m\left(\boldsymbol{p}_{c_l}^{c_j}\right)\boldsymbol{l}^{p_j}\right]_\times & \left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_{c_l}^{c_i}\boldsymbol{e}_m\left(\mathbf{R}_{c_j}^w\boldsymbol{l}^{p_j}\right)^T \\ -\left(\boldsymbol{l}^{p_i}\right)^T\boldsymbol{p}_{c_l}^{c_i}\boldsymbol{e}_m^T\left[\mathbf{R}_{c_j}^{c_l}\boldsymbol{l}^{p_j}\right]_\times & -\boldsymbol{e}_m^T\mathbf{R}_{c_j}^{c_l}\boldsymbol{l}^{p_j}\left(\boldsymbol{l}^{p_i}\right)^T\mathbf{R}_w^{c_i}\end{bmatrix}.
\end{aligned} \tag{33}$$

Stacking the Jacobian matrices of the three elements in $\boldsymbol{l}^{p_l}$, we can get the Jacobian matrix of $\boldsymbol{l}^{p_l}$ with respect to the c_i camera pose:

$$\boldsymbol{J}_{T_{c_i}^w}^{l^{p_l}} = \begin{bmatrix}\left(\boldsymbol{J}_{T_{c_i}^w}^{l_1^{p_l}}\right)^T & \left(\boldsymbol{J}_{T_{c_i}^w}^{l_2^{p_l}}\right)^T & \left(\boldsymbol{J}_{T_{c_i}^w}^{l_3^{p_l}}\right)^T\end{bmatrix}^T. \tag{34}$$

The Jacobian matrices of $\boldsymbol{l}^{p_l}$ to the c_j and c_l poses can be obtained in the same way and denoted as $\boldsymbol{J}_{T_{c_j}^w}^{l^{p_l}}$ and $\boldsymbol{J}_{T_{c_l}^w}^{l^{p_l}}$.

3) Jacobian of Camera Poses to the State Vector: The error-states of IMU poses are augmented in the state vector instead of the camera poses. Therefore, we need to construct the Jacobian matrices with respect to the state vector. The ith camera pose is computed from the ith IMU pose and the camera-IMU extrinsic parameters as follows:

$$\begin{aligned}
\mathbf{R}_{c_i}^w &= \mathbf{R}_{b_i}^w\left(\mathbf{R}_b^c\right)^T \\
\boldsymbol{p}_{c_i}^w &= \boldsymbol{p}_{b_i}^w - \mathbf{R}_{c_i}^w\boldsymbol{p}_b^c.
\end{aligned} \tag{35}$$

Denoting the error-state of the ith IMU pose as $\delta\boldsymbol{T}_{b_i}^w = \begin{bmatrix}\boldsymbol{\theta}_{b_i}^T & (\delta\boldsymbol{p}_{b_i}^w)^T\end{bmatrix}^T$, denoting that of the camera-IMU extrinsics as $\delta\boldsymbol{T}_{\mathrm{b,c}} = \begin{bmatrix}\boldsymbol{\theta}_{\mathrm{b,c}}^T & (\delta\boldsymbol{p}_b^c)^T\end{bmatrix}^T$, and substituting them into (35), we can express the Jacobian of camera pose to the IMU pose and extrinsics as follows:

$$\begin{aligned}
\begin{bmatrix}\boldsymbol{\theta}_{c_i} \\ \delta\boldsymbol{p}_{c_i}^w\end{bmatrix} &= \begin{bmatrix}\mathbf{R}_b^c & \mathbf{0}_3 \\ -\mathbf{R}_{b_i}^w\left[\boldsymbol{p}_c^b\right]_\times & \boldsymbol{I}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_{b_i} \\ \delta\boldsymbol{p}_{b_i}^w\end{bmatrix} \\
&\quad + \begin{bmatrix}-\mathbf{R}_b^c & \mathbf{0}_3 \\ \mathbf{R}_{b_i}^w\left[\boldsymbol{p}_c^b\right]_\times & -\mathbf{R}_{c_i}^w\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_{\mathrm{b,c}} \\ \delta\boldsymbol{p}_b^c\end{bmatrix}.
\end{aligned} \tag{36}$$

According to the definition of the state vector of PLPO-KF, we can express the Jacobian matrix of the ith camera poses to the complete state vector as follows:

$$\boldsymbol{J}_x^{T_{c_i}^w} = \begin{bmatrix}\cdots & -\mathbf{R}_b^c & \mathbf{0}_3 & \cdots & \mathbf{R}_b^c & \mathbf{0}_3 & \cdots \\ \cdots & \mathbf{R}_{b_i}^w\left[\boldsymbol{p}_c^b\right]_\times & -\mathbf{R}_{c_i}^w & \cdots & -\mathbf{R}_{b_i}^w\left[\boldsymbol{p}_c^b\right]_\times & \boldsymbol{I}_3 & \cdots\end{bmatrix}. \tag{37}$$

4) Complete Measurement Jacobian: Finally, we calculate the complete Jacobian matrix of the pose-only line-feature measurement model based on the chain rule, as displayed in the following equation:

$$\boldsymbol{H}_{x,l} = \boldsymbol{J}_{\boldsymbol{l}^{p_l}}^{\boldsymbol{z}_l^{p_l}}\left(\boldsymbol{J}_{T_{c_i}^w}^{l^{p_l}}\boldsymbol{J}_x^{T_{c_i}^w} + \boldsymbol{J}_{T_{c_j}^w}^{l^{p_l}}\boldsymbol{J}_x^{T_{c_j}^w} + \boldsymbol{J}_{T_{c_l}^w}^{l^{p_l}}\boldsymbol{J}_x^{T_{c_l}^w}\right). \tag{38}$$

The expression of the complete Jacobian matrix further demonstrates the conciseness and intuitiveness of the proposed pose-only line-feature measurement model. Then, we can directly update the state vector using the line's measurement and the computed measurement Jacobian matrix.

C. Base Frame Selection for Line Feature Update

Similar to the point-based pose-only measurement model, two base frames are required for each line feature to construct the measurement equation. Let the base frames and the current frame be indexed as i, j, and l. The first base frame, i.e., the ith frame, is fixed as the oldest frame in the sliding window to mitigate accumulated IMU drift.

Following our previous work [10], we use the product of parallax magnitudes as the base-frame selection indicator. The line parallax between ith and jth frames is defined as:

$$\vartheta_{i,j} = \frac{\| \boldsymbol{n}_{e_i}^{c_i} \times \mathbf{R}_{c_j}^{c_i}\boldsymbol{n}_{e_j}^{c_j} \|}{\| \boldsymbol{n}_{e_i}^{c_i} \|\| \mathbf{R}_{c_j}^{c_i}\boldsymbol{n}_{e_j}^{c_j} \|} \tag{39}$$

and the base-frame selection metric for a line feature is:

$$d_{L,\Omega} = \vartheta_{i,j} \cdot \vartheta_{i,l} \cdot \vartheta_{j,l}. \tag{40}$$

The optimal second base frame j^* is selected from the candidate frames in the sliding window by the following conditions:

$$j^* = \arg\max_j d_{L,\Omega}. \tag{41}$$

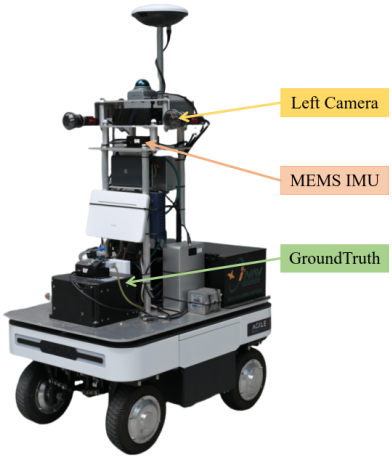

Fig. 8. Wheeled robot platform for the i2Nav-Robot dataset.

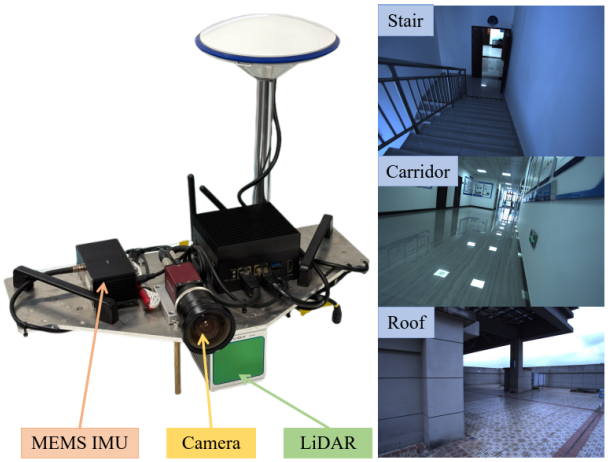

Fig. 9. Handheld platform and test scenes for the HandNav dataset.

VII. Experiments and Results

A. Experimental Setup

We conduct a comprehensive evaluation of the proposed PLPO-KF on both public and self-collected datasets. The public datasets include EuRoC and i2Nav-Robot, captured by a drone and a low-speed wheeled robot, respectively. The self-collected dataset was recorded using a handheld platform.

1) Public EuRoC Dataset: The EuRoC dataset [41] contains drone sequences recorded in challenging indoor environments, including low-light conditions and pure rotations. The left monocular images and IMU data of all 11 sequences are utilized in our evaluation. The provided ground truth is used to assess absolute position error.

2) Public i2Nav-Robot Dataset: The i2Nav-Robot dataset [42], released by our team, was collected using a low-speed wheeled robot (as shown in Fig. 8) across multiple campus scenarios. Notably, the image timestamps correspond to the start of exposure. In this evaluation, we only employ the left monocular images (downsampled to 800 $\times$ 600) and the IMU data from three *parking* sequences. These sequences contain structured environments, span a total duration of 3458 s, and cover 4249 m of trajectory. The provided continuous ground truth is used for evaluating both absolute and relative pose errors.

3) Private HandNav Dataset: We further validate the proposed method on a self-collected handheld dataset (HandNav). The platform (see Fig. 9) is equipped with an MEMS IMU and a monocular camera, while light detection and ranging (LiDAR)-based mapping provides ground truth. The IMU runs at 200 Hz, and the camera captures 900 $\times$ 600 images at 20 Hz. The dataset includes ten sequences collected in four environments (stair, corridor, roof, and a hybrid scenario), spanning 2925 s and 3507 m. The LiDAR-derived ground truth is used to evaluate the absolute position error.

4) Evaluation Method: We build PLPO-KF upon our previous work PO-KF. We compare PLPO-KF against several SOTA open-source VIO systems, including point-based methods (OpenVINS [39], VINS-Mono [29], ORB-SLAM3 [43]) and point-line-based methods (PL-VIO [11], PL-VINS [30], and EPLF-VINS [6]). In PLPO-KF, the sliding window size is set to 20 camera poses, with approximately 100 feature points and 40 line features extracted per image. We use the default or appropriately tuned parameters for the open-source baselines. For a fair comparison, loop closure is disabled for both VINS-Mono and ORB-SLAM3 in our evaluation. It is also worth noting that VINS-Mono, ORB-SLAM3, PL-VIO, PL-VINS, and EPLF-VINS rely on non-linear optimization for state estimation, while both OpenVINS and our PLPO-KF adopt filter-based architectures.

For quantitative evaluation, we employ the EVO toolkit [44], which provides metrics including absolute translation error (ATE), absolute rotation error (ARE), relative translation error (RTE), and relative rotation error (RRE). In addition, we specifically evaluate the performance of the proposed INS-enhanced line optical-flow tracking method and analyze the localization benefits introduced by the trifocal tensor-based line pose-only measurement model.

B. Evaluation of Localization Accuracy

1) Public EuRoC Dataset: We first evaluate the localization performance of the proposed PLPO-KF on the EuRoC dataset, in comparison with SOTA open-source VIO algorithms. Table I reports the absolute position errors across all 11 test sequences of the other methods. The best result for each sequence is highlighted in bold, and the second-best is underlined. Due to the relatively short duration and limited motion range of EuRoC sequences, all methods achieve small errors with limited variation.

It is observed that ORB-SLAM3 achieves significantly better performance than other methods on almost all sequences of this dataset. This is because ORB-SLAM3 maintains co-visible keyframes and a local map, enabling frequent relocalization even without loop closure on EuRoC, which differs from a strict odometry-only setting. Therefore, in the following paragraph, we additionally compare PLPO-KF only with other VIO baselines to fairly assess odometry performance.

Among odometry systems, the filter-based OpenVINS diverged on the *MH04* sequence, whereas other optimization-based methods and our filter-based PLPO-KF successfully

TABLE I
ABSOLUTE POSITION ERRORS (M) ON THE EUROC DATASET

Method	*MH01*	*MH02*	*MH03*	*MH04*	*MH05*	*V1-01*	*V1-02*	*V1-03*	*V2-01*	*V2-02*	*V2-03*	*RMS*
OpenVINS	0.195	0.135	0.250	failed	0.418	0.110	0.080	0.090	0.099	0.093	0.150	-
VINS-MONO	0.191	0.173	0.196	0.361	0.275	0.066	0.093	0.162	0.079	0.133	0.242	*0.198*
ORB-SLAM3	**0.064**	**0.055**	**0.045**	0.159	**0.098**	**0.038**	0.055	**0.053**	**0.047**	**0.031**	**0.047**	***0.072***
PL-VIO	0.179	0.154	0.258	0.309	0.238	0.074	0.107	0.189	0.082	0.144	0.257	*0.196*
PL-VINS	0.210	0.185	0.217	0.240	0.273	0.069	0.108	0.162	0.084	0.103	0.191	*0.179*
EPLF-VINS	0.172	0.088	0.182	0.235	0.215	0.066	0.056	0.132	0.084	0.098	0.168	*0.148*
PLPO-KF	0.066	0.178	0.152	**0.129**	0.278	0.064	**0.037**	0.082	0.077	0.087	0.210	*0.142*

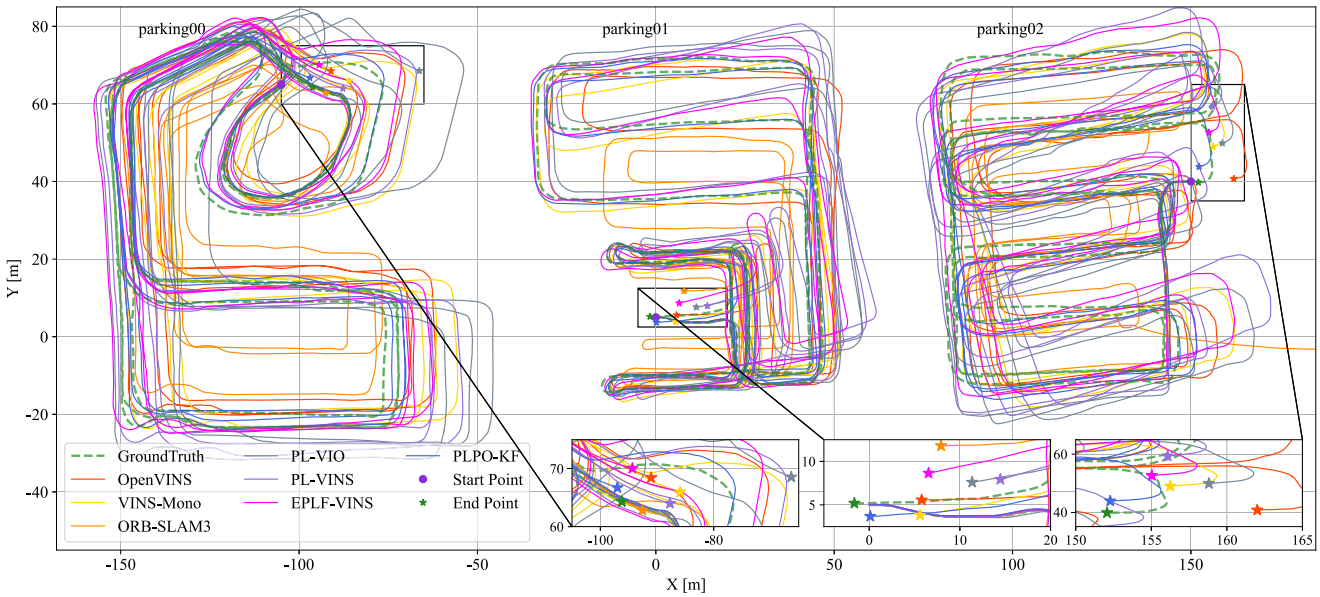

Fig. 10. Resulting trajectories of the Robot *parking* sequences. All trajectories start from the same point and end at different stars.

completed all sequences, demonstrating stronger robustness of PLPO-KF than OpenVINS. In terms of localization accuracy, point-line-based methods (PL-VIO, PL-VINS, and EPLF-VINS) statistically outperform the point-only VINS-Mono, confirming the advantage of incorporating line features in VIO systems. Among them, EPLF-VINS achieves lower errors than PL-VIO and PL-VINS, mainly due to improved line-tracking continuity from its LF-based tracker. Similarly, our PLPO-KF benefits from the INS-enhanced LF (IE-LF) tracking, which provides more reliable line tracking and further refines accuracy through its pose-only line measurement model. Consequently, PLPO-KF achieves the second-best or even best performance on 8 of the 11 sequences and delivers better statistical localization accuracy compared with any other open-source pure odometry algorithms.

2) Public i2Nav-Robot Dataset: We further evaluate PLPO-KF on the i2Nav-Robot dataset. Since ORB-SLAM3, PL-VIO, PL-VINS, and EPLF-VINS do not estimate the camera-IMU time-delay online, we manually compensated for a constant delay in the *parking01* and *parking02* sequences. For the *parking00* sequence, which spans indoor and outdoor environments with variable time delay, we did not apply this compensation to the *parking00* data. In contrast, OpenVINS, VINS-Mono, and PLPO-KF all perform online delay estimation.

To visually demonstrate the position errors of different methods, Fig. 10 shows the resulting trajectories aligned to the ground-truth starting point. In terms of trajectory consistency with the ground truth, all baseline methods show significant position and heading deviations to varying degrees. The *parking01* and *parking02* sequences were captured entirely indoors, leading to more image blur and fewer reliable features. As a result, all tested methods, including our proposed PLPO-KF, exhibit considerable heading deviation after an extended distance. Since ORB-SLAM3 relies more heavily on visual information, while the wheeled robot's dynamics are relatively weak, it exhibits noticeable scale drift on these sequences. Particularly, ORB-SLAM3 shows a significant divergence in the latter part of the *parking02* sequence. Additionally, the point-line-based baselines (PL-VIO, PL-VINS, and EPLF-VINS) still suffer from residual time-delay error, resulting in larger trajectory deviations than the point-only VINS-Mono and OpenVINS. As shown in the subplots in Fig. 10, the endpoints of PLPO-KF for the three test sequences are also the closest to the ground truth, further confirming the superior localization performance of our proposed PLPO-KF.

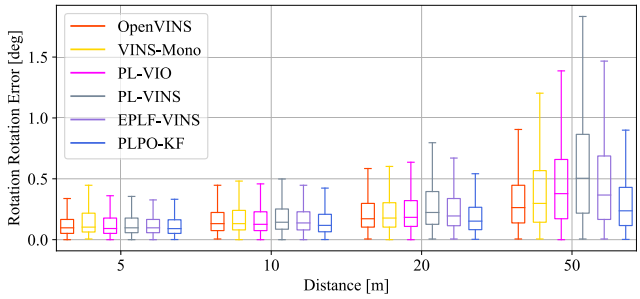

Fig. 11. RREs across different trajectory lengths on the i2Nav-Robot dataset.

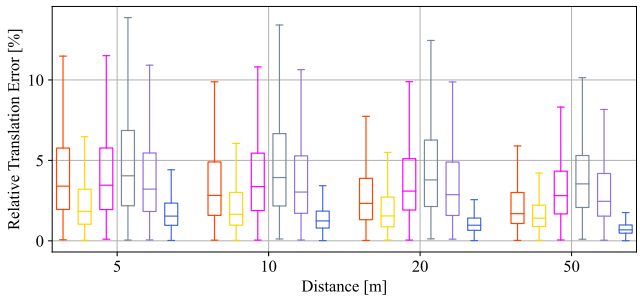

Fig. 12. RTEs across different trajectory lengths on the i2Nav-Robot dataset.

TABLE II
ABSOLUTE POSE ERRORS (DEG/M) ON THE ROBOT DATASET

Method	*parking00*	*parking01*	*parking02*	*RMS*
OpenVINS	0.84 / 3.30	1.04 / 3.01	**0.93** / 3.67	*0.94 / 3.34*
VINS-Mono	1.16 / 4.40	2.37 / 2.11	3.56 / 4.18	*2.56 / 3.71*
ORB-SLAM3	0.98 / 10.85	1.32 / 12.68	failed	-
PL-VIO	1.32 / 4.68	2.90 / 4.26	3.97 / 6.33	*2.94 / 5.17*
PL-VINS	1.94 / 5.17	4.90 / 5.52	5.75 / 7.80	*4.50 / 6.27*
EPLF-VINS	0.95 / 3.33	3.39 / 3.87	3.73 / 4.43	*2.96 / 3.90*
PLPO-KF	**0.49** / **0.88**	**0.57** / **0.89**	1.32 / **1.05**	***0.88 / 0.94***

The point-line-based baselines still exhibit larger deviations due to residual delay, while the other two point-only methods (VINS-Mono and OpenVINS) perform better. As illustrated, PLPO-KF consistently yields trajectories closest to the ground truth.

Quantitative pose errors on the i2Nav-Robot dataset are summarized in Figs. 11 and 12, and Table II. Due to the scale divergence, ORB-SLAM3 shows substantially larger relative pose errors on this dataset. Therefore, its results are not included in the relative error box plots to better compare with other baselines. For relative pose errors in Fig. 11 and Fig. 12, the proposed PLPO-KF outperforms all baselines across almost all metrics except for the 5-m RTE of EPLF-VNS. In terms of absolute pose error in Table II, PLPO-KF achieves the lowest errors on all sequences except the rotation error of *parking02* sequence. Statistically, PLPO-KF achieves the smallest absolute pose errors of 0.88deg and 0.94 m, outperforming all baseline methods.

3) Private HandNav Dataset: We finally evaluate PLPO-KF on the self-collected HandNav dataset. We also compensate for the constant camera-IMU time delay in PL-VIO, PL-VINS, and EPLF-VINS. Fig. 13 shows trajectories for the *hybrid01* sequence, and Fig. 14 presents

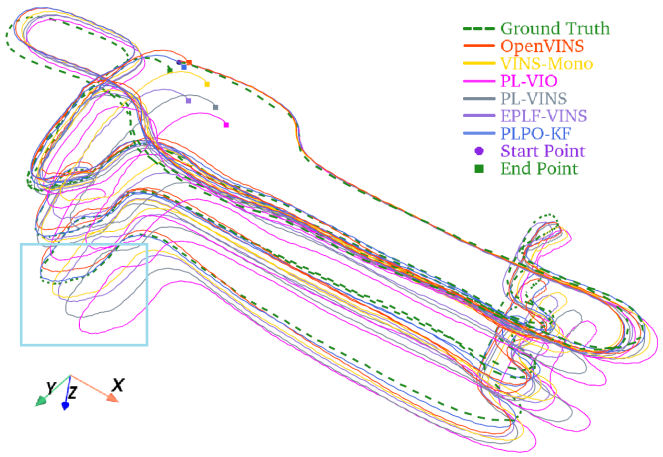

Fig. 13. Resulting trajectories of the HandNav *hybrid01* sequence. All trajectories start from the same point and end at different starts, each shown in a distinct color.

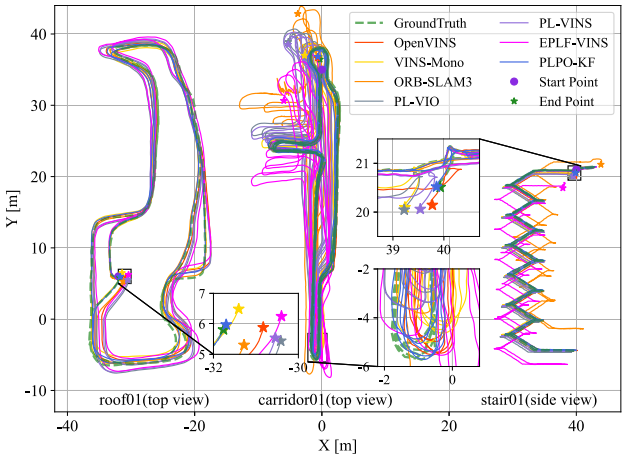

Fig. 14. Resulting trajectories of the HandNav *roof01 corridor01* and *stair01* sequences. All trajectories start from the same point and end at different squares, each shown in a distinct color.

resulting trajectories for *roof01*, *corridor01*, and *stair01* sequences. The images in the HandNav dataset contain limited texture and exhibit significant motion blur, leading to frequent tracking failures in ORB-SLAM3. As a result, ORB-SLAM3 fails on all *hybrid* sequences and shows severe trajectory drift in the *corridor* and *stair* sequences. In addition, due to residual time delay, the three point-line-based baselines (PL-VIO, PL-VINS, EPLF-VINS) again show degraded localization results than the point-only VINS-Mono and OpenVINS. In contrast, the proposed PLPO-KF obtains trajectories that align most closely with the ground truth, especially in the sky-blue rectangular area in Fig. 13. Furthermore, the resulting trajectories of the *roof01*, *corridor01*, and *stair01* sequences in Fig. 14 also show that the PLPO-KF achieves obviously better trajectory fitness with the ground truth than all baseline methods. The closest endpoints with the ground truth in Figs. 13 and 14 further confirm the superior localization performance of PLPO-KF.

Table III summarizes absolute position errors across all 10 HandNav sequences. PLPO-KF consistently outperforms all

TABLE III

ABSOLUTE POSITION ERRORS (M) ON THE HANDNAV DATASET

Method	*stair00*	*stair01*	*corridor00*	*corridor01*	*roof00*	*roof01*	*roof02*	*hybrid00*	*hybrid01*	*hybrid02*	*RMS*
OpenVINS	0.239	0.204	**0.275**	0.284	0.631	0.509	0.531	0.203	0.421	0.721	*0.440*
VIINS-Mono	0.220	0.521	0.676	0.557	0.839	0.483	0.682	0.272	0.874	1.019	*0.661*
ORB-SLAM3	failed	1.901	2.924	2.919	failed	0.334	1.425	failed	failed	failed	-
PL-VIO	0.229	0.615	0.841	1.189	1.500	0.752	0.939	0.338	1.758	1.325	*1.059*
PL-VINS	0.226	0.400	0.731	1.186	1.358	0.767	0.887	0.356	1.406	1.055	*0.927*
EPLF-VINS	0.294	1.095	0.799	1.904	1.182	0.826	1.036	0.412	0.949	0.940	*1.032*
PLPO-KF	**0.147**	**0.099**	0.328	**0.268**	**0.324**	**0.180**	**0.363**	**0.135**	**0.324**	**0.435**	***0.282***

TABLE IV

LINE TRACK PERFORMANCE AND TRIANGULATION RESIDUALS (AVG/CDF95) OF THE TWO LINE TRACKING METHODS

Test	Track Rate[%]		Track Length		Track Parallax		Tri. Error[pixel]	
	LF	IE-LF	LF	IE-LF	LF	IE-LF	LF	IE-LF
test00	49	**70**	5.65 / 19	**7.88 / 25**	0.12 / 0.48	**0.19 / 0.70**	**1.86 / 3.96**	1.87 / 4.12
test01	59	**73**	8.77 / 33	**8.95 / 33**	0.15 / 0.63	**0.18 / 0.72**	**1.98 / 4.37**	2.00 / 4.44
test02	53	**65**	6.09 / 17	**7.22 / 22**	0.10 / 0.41	**0.14 / 0.57**	1.72 / **3.72**	**1.67** / 3.75
RMS	*54*	*69*	*6.97 / 24*	*8.05 / 27*	*0.13 / 0.52*	*0.17 / 0.66*	*1.86 / **4.03***	***1.85** / 4.11*

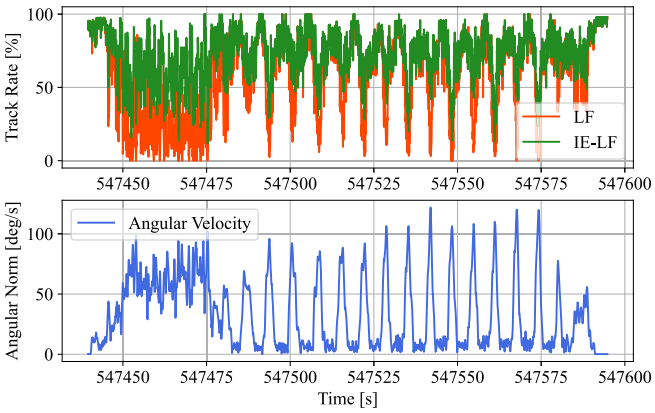

Fig. 15. Line tracking rate and camera angular speed of the proposed and baseline line tracking methods.

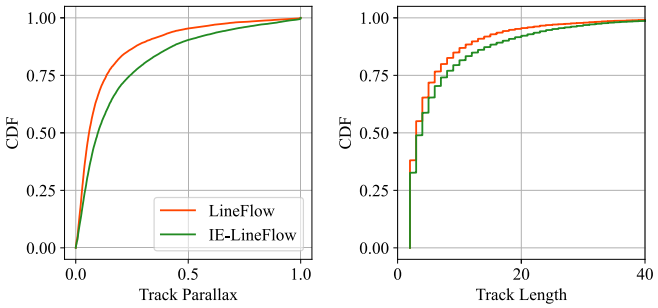

Fig. 16. Tracking length and tracking parallax of the proposed and baseline line tracking methods.

baseline methods, although it yields a slightly higher error than OpenVINS on the *corridor00* sequence. Statistically, PLPO-KF achieves an average absolute position error of only 0.282 m, significantly lower than all other baselines.

According to the results across three diverse datasets, the proposed PLPO-KF consistently achieves superior localization accuracy compared with both point-only and point-line-based SOTA VIO methods.

C. Impact of INS-Enhanced Line Tracking

To specifically evaluate the proposed IE-LF tracking algorithm, we collected three sets of dynamically rich test data using the handheld platform and compared it against the basic LF algorithm. Line tracking performance is assessed in terms of frame-to-frame tracking rate, total tracking length, and total tracking parallax. In addition, since tracking accuracy is as important as tracking length, we further validate the accuracy using line feature triangulation residuals.

1) Frame-to-Frame Tracking Rate: The frame-to-frame tracking rate on the *front-test01* data is shown in Fig. 15, along with the smoothed carrier angular velocity curve. As expected, when carrier dynamics increase (i.e., high angular velocity), the LF algorithm exhibits a sharp drop in tracking success rate, while the proposed IE-LF algorithm maintains a much higher level of robustness. The average frame-to-frame tracking rates across all three datasets are summarized in Table IV. These results show that incorporating INS prior poses improves the average tracking success rate from 54% to 69%.

2) Total Tracking Length and Accuracy: We next evaluate the long-term tracking performance of the proposed IE-LF algorithm. Taking *front-test00* as an example, Fig. 16 shows the cumulative distribution function (cdf) curves of tracking length and parallax for both algorithms. The IE-LF consistently achieves longer tracking lengths and larger parallaxes, as indicated by its cdf curve lying below that of the LF algorithm. The mean and 95th percentile (CDF95) statistics of tracking length and parallax are summarized in Table IV. Across all datasets, IE-LF increases the average tracking length by approximately 15% and the average tracking parallax by about 30%.

Finally, we compute the line triangulation residuals of line features obtained by both IE-LF and LF, with results also presented in Table IV. The statistical residuals of the two algorithms are nearly identical, demonstrating that the

TABLE V
ATEs (m) OF THE THREE TEST METHODS (PO-KF, POKF-LM, AND PLPO-KF) ON THE EuRoC DATASET

Method	*MH01*	*MH02*	*MH03*	*MH04*	*MH05*	*V1-01*	*V1-02*	*V1-03*	*V2-01*	*V2-02*	*V2-03*	*RMS*
PO-KF	0.076	**0.153**	0.162	0.126	0.312	0.062	**0.037**	0.095	0.062	0.087	0.208	*0.147*
POKF-LM	0.074	0.156	0.155	**0.125**	0.310	**0.060**	0.042	0.096	**0.059**	**0.084**	**0.202**	*0.145*
PLPO-KF	**0.066**	0.178	**0.152**	0.129	**0.278**	0.064	**0.037**	**0.082**	0.077	0.087	0.210	***0.142***

TABLE VI
ABSOLUTE AND RELATIVE POSE ERRORS OF THE THREE METHODS (PO-KF/POKF-LM/PLPO-KF) ON THE ROBOT PARKING DATASET

Sequence	ARE[deg]	RRE[deg]		ATE[m]	RTE[%]	
		10m	50m		10m	50m
parking00	0.52 / 0.54 / **0.49**	**0.16 / 0.16 / 0.16**	0.27 / 0.27 / **0.26**	0.94 / 0.96 / **0.88**	1.75 / 1.70 / **1.44**	0.85 / 0.84 / **0.76**
parking01	0.76 / 0.77 / **0.57**	**0.20 / 0.20 / 0.20**	0.38 / 0.38 / **0.37**	1.32 / 1.23 / **0.89**	2.14 / 2.08 / **1.71**	1.05 / 1.02 / **0.87**
parking02	1.48 / 1.56 / **1.32**	0.25 / 0.24 / **0.22**	0.51 / 0.51 / **0.49**	1.33 / 1.36 / **1.05**	1.94 / 1.92 / **1.73**	0.97 / 0.98 / **0.95**
RMS	*1.01 / 1.05 / **0.88***	*0.21 / 0.21 / **0.19***	*0.40 / 0.40 / **0.38***	*1.21 / 1.20 / **0.94***	*1.95 / 1.91 / **1.63***	*0.96 / 0.95 / **0.86***

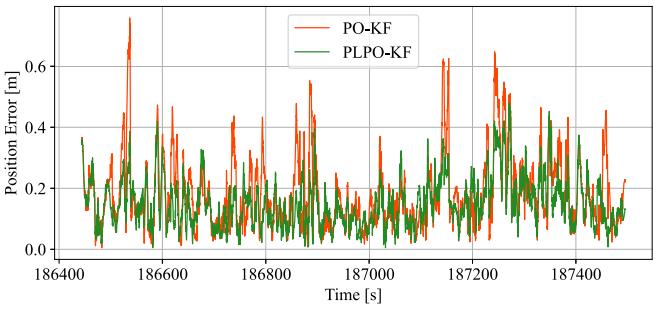

Fig. 17. 10-m relative position error on the *parking01* sequence.

proposed IE-LF does not sacrifice tracking accuracy for enhanced tracking robustness and tracking length.

D. Impact of the Pose-Only Line-Feature Model

In this section, we specifically evaluate the proposed trifocal tensor-based pose-only line-feature measurement model. Our previous work, PO-KF, serves as the baseline to assess the contribution of line features. We also implement an MSCKF-based line-feature measurement model on top of PO-KF, denoted as POKF-LM, to benchmark against the proposed pose-only line-feature measurement model. The evaluation is also conducted on the EuRoC, i2Nav-Robot, and HandNav datasets, and performance is assessed in terms of localization accuracy. For a fair comparison, PO-KF, POKF-LM, and PLPO-KF share identical system parameters.

1) Public EuRoC Dataset: We first evaluate the line measurement models on the EuRoC dataset. The absolute position errors of PO-KF, POKF-LM, and PLPO-KF on the EuRoC dataset are summarized in Table V. Across the 11 sequences, both POKF-LM and PLPO-KF achieve better position accuracy on more sequences compared with PO-KF, confirming the contribution of line features to localization accuracy. Furthermore, the comparison between POKF-LM and PLPO-KF shows that PLPO-KF consistently outperforms POKF-LM, highlighting that the proposed pose-only line-feature measurement model provides greater benefits for localization accuracy than the traditional MSCKF-based line-feature measurement model.

2) Public i2Nav-Robot Dataset: We then evaluate the three methods on the i2Nav-Robot dataset, with absolute and relative pose errors presented in Table VI. For clarity, only the 10- and 50-m relative errors are reported. A direct comparison between PLPO-KF and PO-KF shows that PLPO-KF consistently achieves smaller absolute and relative pose errors across all three sequences, which benefits from the sufficient measurements of PLPO-KF with both point and line features. To illustrate this improvement, we present the 10-m relative position error of *parking01* in Fig. 17, where PLPO-KF achieves lower errors than PO-KF throughout most of the sequence. In addition, the insufficient line parallax in wheeled robots will lead to more triangulation failures and poor line triangulation accuracy, resulting in the MSCKF-based POKF-LM does not always outperforming PO-KF. In contrast, PLPO-KF adaptively selects three frames with enough parallax to formulate a pose-only line-feature measurement model, ensuring reliable line updates and consistently improving localization accuracy. The comparison between PLPO-KF and POKF-LM clearly demonstrates the superiority of the proposed pose-only line-feature model over the traditional MSCKF-based model.

3) Private HandNav Dataset: Finally, we evaluate the three methods on the HandNav dataset. The absolute position errors are summarized in Table VII. Across the ten sequences, both PLPO-KF and POKF-LM consistently achieve lower position errors than PO-KF. This further validates the contribution of line features to localization accuracy. Notably, PLPO-KF achieves particularly significant improvements over PO-KF in the *hybrid01* and *hybrid02* sequences, which involve larger trajectory ranges that better showcase the benefits of the pose-only line-feature measurement model. In contrast, the *hybrid00* sequence exhibits only marginal improvements due to its limited trajectory range spanning just two floors. Quantitatively, PLPO-KF reduces the position error of PO-KF from 0.330 to 0.282 m. Moreover, when compared with POKF-LM, PLPO-KF consistently achieves higher accuracy across all sequences, confirming the superior performance of the proposed trifocal tensor-based pose-only line-feature measurement model.

TABLE VII
ABSOLUTE POSITION ERRORS (M) OF THE LINE MEASUREMENT MODEL TEST ON THE HANDNAV DATASET

Method	*stair00*	*stair01*	*corridor00*	*corridor01*	*roof00*	*roof01*	*roof02*	*hybrid00*	*hybrid01*	*hybrid02*	*RMS*
PO-KF	0.173	0.164	0.426	0.282	0.329	0.203	0.390	0.154	0.414	0.522	*0.330*
POKF-LM	0.164	0.143	0.419	0.275	0.331	0.201	0.389	0.148	0.373	0.505	*0.319*
PLPO-KF	**0.147**	**0.099**	**0.328**	**0.268**	**0.324**	**0.180**	**0.363**	**0.135**	**0.324**	**0.435**	***0.282***

TABLE VIII
AVERAGE RUNTIMES PER FRAME (MS) ON THE HANDNAV DATASET

Method	Point Track	Line Track	Prop.&Update	Total
OpenVINS	3.17	-	1.14	*4.31*
VINS-Mono	11.24	-	11.78	*23.02*
ORB-SLAM3	-	-	-	*46.48*
PO-KF	5.95	-	0.66	*6.61*
PL-VIO	11.04	66.81	13.46	*93.31*
PL-VINS	11.65	18.32	13.39	*43.36*
EPLF-VINS	12.37	11.06	13.95	*37.38*
PLPO-KF	5.93	11.43	0.83	*18.19*

Track includes extraction and association; Prop. denotes IMU propagation.

4) Discussion on Limitations of Line Features: Although the above evaluations demonstrate the advantages of incorporating line features, several inherent limitations should be acknowledged. In extremely textureless, motion-blurred, or low-light environments, line features may become sparse and unreliable, thereby reducing their contribution to the overall localization accuracy. In addition, certain degenerate motions or scene geometries, for instance, pure forward motion in very long and straight corridors, can make line features poorly observable. This loss of observability may weaken the geometric constraints, leading to degraded pose estimation accuracy or even potential filter divergence.

Based on these observations and our practical experience, line features should be regarded as a complementary modality rather than a complete replacement for point features, even when outlier rejection and degeneration–detection strategies are employed. Their benefits are most pronounced in environments with structural information.

E. Runtime Analysis

We further evaluate the computational efficiency of the proposed PLPO-KF through a runtime analysis. All methods are tested on the HandNav dataset using a desktop PC equipped with an AMD R7950X CPU and 32 GB of RAM. The average processing times of the key modules per frame are summarized in Table VIII. For clarity, the processing times of point and line features are reported separately. The runtimes of IMU propagation and state update are combined, since preintegration factors and visual measurement factors are jointly optimized in the optimization-based methods. ORB-SLAM3 performs two pose optimizations (initial pose estimation for each frame and local mapping optimization for keyframes), so we report only its total runtime per frame. In addition, the runtime of ORB-SLAM3 is averaged over five sequences, while the results of the other methods are averaged over all ten HandNav sequences.

From the runtime results, we first observe that the filter-based point-only methods (OpenVINS and PO-KF) achieve the highest computational efficiency. PO-KF shows a slightly higher runtime than OpenVINS due to differences in point-feature types and the number of tracked features. In contrast, the optimization-based point-only methods (VINS-Mono and ORB-SLAM3) incur substantially higher computational cost as a result of their iterative optimization procedures. Due to line extraction and tracking, the point-line-based methods generally require more computation. For example, the additional runtime of PLPO-KF compared with PO-KF also mainly comes from line processing.

The runtime of line tracking is variable among different point-line-based methods. PL-VIO employs an LSD-based line tracker, leading to the highest runtime overhead. Although PL-VINS improves the efficiency of LSD-based tracking, its line-processing cost remains considerably higher than the optical-flow-based line tracking used in EPLF-VINS and in our PLPO-KF. The INS-enhanced line tracking in PLPO-KF makes only a slight increase in line tracking time compared with EPLF-VINS. In addition, the estimation methods also notably impact the system runtime. Specifically, due to the filter-based framework, PLPO-KF retains significantly lower propagation and state update time than the optimization-based approaches. In general, these module-level runtimes clearly illustrate the high effectiveness of combining a filter-based estimation framework with efficient optical-flow-based line tracking.

Overall, PLPO-KF achieves a lower per-frame runtime than all baseline methods except the filter-based point-only approaches. Compared with the SOTA point-line-based methods, PLPO-KF demonstrates a substantially lower computational cost, confirming its superior efficiency.

In summary of the above experiments, the results demonstrate that PLPO-KF outperforms both point-based and point-line-based SOTA VIO methods, validate the effectiveness of the proposed IE-LF tracking algorithm and the trifocal tensor-based pose-only line-feature measurement model, as well as verify the superior computational efficiency of PLPO-KF than SOTA point-line-based methods.

VIII. CONCLUSION

In this paper, we propose PLPO-KF, a unified pose-only representation-based Kalman filter with point-line features for VIO, by integrating an INS-enhanced line tracking method and a trifocal tensor-based pose-only line-feature measurement model into our previous PO-KF framework. The proposed INS-enhanced line tracking leverages INS prior poses to provide reliable initial values, ensuring stable and accurate

line tracking. Meanwhile, the trifocal tensor-based pose-only line-feature measurement model offers a more concise and effective formulation compared with conventional models. Extensive evaluations on three diverse datasets demonstrate that PLPO-KF achieves superior localization accuracy over SOTA point-based and point-line-based VIO systems, including OpenVINS, VINS-Mono, ORB-SLAM3, PL-VIO, PL-VINS, and EPLF-VINS. Additional experiments further validate the robustness of the proposed IE-LF tracking method, the effectiveness of the trifocal tensor-based pose-only line-feature measurement model, and the superior computational efficiency of the proposed PLPO-KF.

In future work, we plan to investigate learning-based approaches for more accurate line detection and tracking, extend the proposed pose-only line-feature measurement model toward efficient visual line mapping, and further explore leveraging line-feature maps for absolute visual localization.

REFERENCES

[1] L. Liu et al., "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6469–6486, Apr. 2021.

[2] S. Javed et al., "State-of-the-art and future research challenges in UAV swarms," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19023–19045, Jun. 2024.

[3] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9572–9582.

[4] P. Hruby, S. Liu, R. Pautrat, M. Pollefeys, and D. Barath, "Handbook on leveraging lines for two-view relative pose estimation," 2023, *arXiv:2309.16040*.

[5] K. Ivanov, G. Ferrer, and A. Kornilova, "EVOLIN benchmark: Evaluation of line detection and association," 2023, *arXiv:2303.05162*.

[6] L. Xu, H. Yin, T. Shi, D. Jiang, and B. Huang, "EPLF-VINS: Real-time monocular visual-inertial SLAM with efficient point-line flow features," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 752–759, Feb. 2023.

[7] L. Wang, H. Tang, T. Zhang, Q. Chen, J. Shi, and X. Niu, "Improving the navigation performance of the MEMS IMU array by precise calibration," *IEEE Sensors J.*, vol. 21, no. 22, pp. 26050–26058, Nov. 2021.

[8] X. Niu, H. Tang, T. Zhang, J. Fan, and J. Liu, "IC-GVINS: A robust, real-time, INS-centric GNSS-visual-inertial navigation system," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 216–223, Jan. 2023.

[9] H. Tang, X. Niu, T. Zhang, L. Wang, and J. Liu, "LE-VINS: A robust solid-state-LiDAR-enhanced visual-inertial navigation system for low-speed robots," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[10] L. Wang, H. Tang, T. Zhang, Y. Wang, Q. Zhang, and X. Niu, "PO-KF: A pose-only representation-based Kalman filter for visual inertial odometry," *IEEE Internet Things J.*, vol. 12, no. 10, pp. 14856–14875, May 2025.

[11] 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, p. 1159, Apr. 2018.

[12] Z. Yang, S. Chen, and H. Zhang, "POPL-VIO: A novel pose-only measurement model for point-line-based visual inertial odometry," in *Proc. China Autom. Congr. (CAC)*, Nov. 2024, pp. 6064–6069.

[13] J. M. Selig, "Line geometry," in *Geometrical Methods in Robotics*. New York, NY, USA: Springer, 1996, pp. 81–100.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., New York, NY, USA: Cambridge Univ. Press, 2003.

[15] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-inertial odometry with structural regularity of man-made environments," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 999–1013, Aug. 2019.

[16] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging planar regularities for point line visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5120–5127.

[17] I. Suárez, J. M. Buenaposada, and L. Baumela, "ELSED: Enhanced line SEgment drawing," 2021, *arXiv:2108.03144*.

[18] C. Akinlar and C. Topal, "Edlines: Real-time line segment detection by edge drawing (ed)," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2837–2840.

[19] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: End-to-end traffic line detection with line proposal unit," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 248–258, Jan. 2020.

[20] N. Xue et al., "Holistically-attracted wireframe parsing," 2020, *arXiv:2003.01663*.

[21] L. Wang, T. Zhang, Y. Wang, H. Tang, and X. Niu, "Enhancing visual navigation performance by prior pose-guided active feature points distribution," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.

[22] M. Lange, F. Schweinfurth, and A. Schilling, "DLD: A deep learning based line descriptor for line feature matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5910–5915.

[23] M. Lange, C. Raisch, and A. Schilling, "WLD: A wavelet and learning based line descriptor for line feature matching," in *Vision, Modeling, and Visualization*, J. Krüger, M. Niessner, and J. Stückler, Eds., The Eurographics Association, 2020, doi: 10.2312/vmv.20201186.

[24] R. Pautrat, J.-T. Lin, V. Larsson, M. R. Oswald, and M. Pollefeys, "SOLD2: Self-supervised occlusion-aware line description and detection," 2021, *arXiv:2104.03362*.

[25] B. Xu, P. Wang, Y. He, Y. Chen, Y. Chen, and M. Zhou, "Leveraging structural information to improve point line visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3483–3490, Apr. 2022.

[26] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "AirSLAM: An efficient and illumination-robust point-line visual SLAM system," *IEEE Trans. Robot.*, vol. 41, pp. 1673–1692, 2025.

[27] Q. Wang, Z. Yan, J. Wang, F. Xue, W. Ma, and H. Zha, "Line flow based simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1416–1432, Oct. 2021.

[28] C. Ye, H. Li, W. Lin, and X. Yang, "MLINE-VINS: Robust monocular visual-inertial SLAM with flow Manhattan and line features," 2025, *arXiv:2503.01571*.

[29] 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.

[30] Q. Fu et al., "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," 2020, *arXiv:2009.07462*.

[31] H. Si, H. Yu, K. Chen, and W. Yang, "Point–line visual–inertial odometry with optimized line feature processing," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.

[32] Y. Yang and G. Huang, "Observability analysis of aided INS with heterogeneous features of points, lines, and planes," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1399–1418, Dec. 2019.

[33] Q. Cai, L. Zhang, Y. Wu, W. Yu, and D. Hu, "A pose-only solution to visual reconstruction and navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 73–86, Jan. 2023.

[34] H. Tang, X. Niu, T. Zhang, L. Wang, G. Wang, and J. Liu, "PO-VINS: An efficient pose-only LiDAR-enhanced visual-inertial state estimator," 2023, *arXiv:2305.12644*.

[35] Y. Ge, L. Zhang, Y. Wu, and D. Hu, "PIPO-SLAM: Lightweight visual-inertial SLAM with preintegration merging theory and pose-only descriptions of multiple view geometry," *IEEE Trans. Robot.*, vol. 40, pp. 2046–2059, 2024.

[36] Q. Zhang, W. Ouyang, J. Han, Q. Cai, M. Zhu, and Y. Wu, "An immediate update strategy of multi-state constraint Kalman filter for visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 4125–4131, Apr. 2025.

[37] D. G. Kottas and S. I. Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1540–1547.

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

[39] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4666–4672.

[40] I. Suárez, J. M. Buenaposada, and L. Baumela, "ELSED: Enhanced line SEgment drawing," *Pattern Recognit.*, vol. 127, Jul. 2022, Art. no. 108619.

[41] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[42] H. Tang et al., "I2Nav-robot: A large-scale indoor-outdoor robot dataset for multi-sensor fusion navigation and mapping," 2025, *arXiv:2508.11485*.

[43] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[44] M. Grupp. (2017). *Evo: Python Package for the Evaluation of Odometry and SLAM*. [Online]. Available: https://github.com/MichaelGrupp/evo

Liqiang Wang (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from Wuhan University, Wuhan, China, in 2020 and 2023, respectively, where he is currently pursuing the Ph.D. degree with the Global Navigation Satellite System (GNSS) Research Center.

His primary research interests include GNSS/Inertial Navigation System (INS) integrations, visual simultaneous localization and mapping (SLAM) and multisensor fusion navigation systems.

Hailiang Tang received the M.E. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2020 and 2023, respectively.

He is currently a Post-Doctoral Fellow with the Global Navigation Satellite System (GNSS) Research Center, Wuhan University. His research interests include autonomous robotics systems, visual and light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM), GNSS/Inertial Navigation System (INS) integration, and deep learning.

Yan Wang received the M.E. degree in computer applied technology from China University of Mining and Technology, Xuzhou, China, in 2019, and the Ph.D. degree from the Global Navigation Satellite System (GNSS) Research Center, Wuhan University, Wuhan, China, in 2023.

He is currently a Post-Doctoral Fellow with the GNSS Research Center, Wuhan University. His research interests focus on indoor navigation, sensor fusion algorithms, and computer vision.

Guan Wang received the B.E. and M.E. degrees from Wuhan University, Wuhan, China, in 2022 and 2025, respectively.

He is currently working with the Huawei Riemann Laboratory, Shenzhen, China. His primary research interests include visual simultaneous localization and mapping (SLAM) and point-line-based visual-inertial odometry (VIO).

Tisheng Zhang (Member, IEEE) received the B.Sc. and Ph.D. degrees in communication and information systems from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

He was a Post-Doctoral Researcher with The Hong Kong Polytechnic University, Hong Kong, from 2017 to 2018. He is currently a Professor with the Global Navigation Satellite System (GNSS) Research Center, Wuhan University. His research interests focus on the fields of GNSS receiver, multisensor positioning, and robot navigation.

Xiaoji Niu (Member, IEEE) received the bachelor's and Ph.D. degrees (Hons.) from the Department of Precision Instruments, Tsinghua University, Beijing, China, in 1997 and 2002, respectively.

He performed his post-doctoral research at the University of Calgary, Calgary, AB, Canada, and worked as a Senior Scientist with SiRF Technology (Shanghai), Shanghai, China. He is currently leading the Integrated and Intelligent Navigation (i2Nav) Group. He is currently a Professor with the Global Navigation Satellite System (GNSS) Research Center, Wuhan University, Wuhan, China. He has published more than 200 academic articles and owns more than 30 patents. His research interests focus on GNSS/Inertial Navigation System (INS) integration, low-cost navigation sensor fusion, and relevant new applications.