PO-KF: A Pose-Only Representation-based Kalman Filter for Visual Inertial Odometry

Liqiang Wang, Hailiang Tang, Tisheng Zhang, Yan Wang, Quan Zhang, and Xiaoji Niu

***Abstract*—Visual-inertial state estimation is widely employed in the Internet of Things, with filter-based visual-inertial odometry (VIO) being a popular algorithm due to its balance between computational efficiency and localization accuracy. However, the localization performance of the commonly used multi-state constraint Kalman filter (MSCKF)-based VIO is suffering from linearization errors in feature three-dimensional (3D) positions and delayed measurement updates. Targeting more accurate and robust localization, we incorporate the pose-only representation into the filter-based VIO and propose a pose-only representation-based Kalman filter (PO-KF) in this paper. Leveraging the decoupling of camera poses and feature positions in the pose-only representation, the proposed PO-KF explicitly eliminates feature 3D coordinates from its measurement equation. As a result, the linearization errors caused by feature positions can be removed efficiently, while immediate updates of visual measurements can be conducted. We also introduce an information matrix-derived base-frame selection algorithm to identify the most suitable base-frames for each feature. Extensive experiments on multiple datasets demonstrate that PO-KF outperforms state-of-the-art VIO systems. Notably, PO-KF achieves nearly a 50% reduction in relative pose errors compared to MSCKF-based VIO. Further experiments demonstrate that PO-KF also exhibits superior robustness while maintaining real-time performance comparable to MSCKF-based VIO.**

***Index Terms*—Visual-inertial odometry(VIO), pose-only representation, state estimation, Kalman filter.**

I. Introduction

REAL-TIME, accurate, and robust localization is the fundamental requirement for the real-world Internet of Things (IoT) systems, such as autonomous vehicles, drones, and mobile augmented Reality (AR) / virtual Reality (VR) devices [1], [2]. Among various localization methods, visual-inertial odometry (VIO), which combines a monocular camera and a consumer-grade inertial measurement unit (IMU), has been extensively employed due to its advantage of low cost, high precision, and small size [3]. To date, feature points-based, visual-inertial tight-coupled VIO has been recognized for its stronger robustness and higher accuracy. For the state estimation in the aforementioned VIO, there are two predominant algorithms: optimization-based solutions and filter-based solutions [4]. Optimization-based solutions, known for iterated relinearization, generally achieve more accurate state estimation but at a higher computational cost. In contrast, filter-based solutions leverage an efficient Kalman filter update, consuming less computation power. Unlike the Extended Kalman Filter (EKF), which augments both feature positions and camera poses into the state vector, the Multi-State Constraint Kalman Filter (MSCKF) [5] augments only camera poses. Consequently, MSCKF significantly reduces the dimension of the state vector and computational complexity, making it the most representative filter-based solution. With the significant advantages of low computational cost, filter-based solutions are popular in IoT platforms.

To construct the measurement equation in MSCKF, the three-dimensional (3D) feature positions are first triangulated and then projected onto the image planes. Since the 3D feature positions are absent in the state vector, MSCKF performs nullspace projection to eliminate these positions from its measurement equation [5], [6]. However, in this way, the update and relinearization of feature positions are prevented [4] in MSCKF. Therefore, to achieve the most accurate 3D position, a feature will be triangulated until it reaches its maximum tracking length or experiences tracking failure, known as delayed feature initialization [6], [7] in MSCKF. Although delayed feature initialization helps minimize the linearization errors on feature 3D positions, it also postpones the use of visual measurements and the correction of the state vector until the feature is triangulated. These delays in MSCKF may lead to large accumulative errors in the system state, which can be particularly challenging for consumer-grade IMU in IoT applications. Moreover, the linearization errors on the feature 3D positions remain in the measurement equation and nullspace projection process. Equally importantly, once the feature triangulation fails, the measurement equation cannot be constructed, rendering most short-tracking features ineffective in the system state.

The pose-only imaging representation [8], equivalent to the two-view visual geometry, decouples camera poses from 3D feature positions. This decoupling facilitates the formulation of cost functions only tied to the camera poses in the optimization problem, leading to an analytical solution for the spatial feature coordinates reconstruction [9]. Recognizing this advantage, we are inspired to integrate the pose-only

Manuscript received Month Day, Year; revised Month Day, Year.

This research is funded by the National Key R&D Program of China (No. 2022YFB3903802), the National Natural Science Foundation of China (No.42374034), the Key Research and Development Program of Hubei Province (No. 2024BAB024), the Major Program of Hubei Province (No. 2023BAA02602), and High quality development project of MIIT(No. 2024-182). (Corresponding authors: Xiaoji Niu; Tisheng Zhang.)

Liqiang Wang, Hailiang Tang, and Yan Wang are with GNSS Reaserch Center of Wuhan University, Wuhan 430079, China (Email: wlq@whu.edu.cn; thl@whu.edu.cn; wystephen@whu.edu.cn).

Tisheng Zhang, Quan Zhang, and Xiaoji Niu are with GNSS Reaserch Center of Wuhan University, Wuhan 430079, China, and also with Hubei Luojia Laboratory, Wuhan 430079, China (Email: zts@whu.edu.cn; zhangquan@whu.edu.cn; xjniu@whu.edu.cn).

representation into the measurement equations of filter-based VIO and explicitly eliminate feature 3D positions. In this way, the reprojection measurement equation becomes only related to the cloned poses and no longer rely on successful feature triangulation. Consequently, the linearization errors caused by the feature positions are removed from the measurement equation. As feature triangulation is indispensable for the measurement equation, the delayed measurement update becomes unnecessary, replaced by an immediate update when enough poses are available to represent the feature.

Based on the above motivations, we propose PO-KF, a pose-only representation-based Kalman filter for visual-inertial odometry. To construct a pose-only measurement model with optimal pose constraints, we also propose an information matrix-derived base-frame selection algorithm. Additionally, we introduce the specialized-designed strategy during zero-velocity states for PO-KF to enhance its localization robustness. Leveraging these methods, PO-KF has the potential to tackle the aforementioned concerns of MSCKF and produce a smooth and robust localization trajectory. The main contributions of this paper are highlighted as follows:

- We propose a pose-only representation-based Kalman filter for visual-inertial odometry. The features' positions, represented by poses of the base-frames, are employed in the construction of measurement equations for PO-KF, explicitly eliminating the feature 3D positions from the measurement equation and immediately updating the system state using visual measurements.
- We propose a base-frame selection algorithm for the pose-only representation. By constructing an information matrix that incorporates the depth constraint of the feature using the base-frames and current frame, we consider the determinant value of the matrix as an indicator to identify the most suitable base-frames.
- We thoroughly evaluate the localization performance of PO-KF using publicly available and privately obtained datasets. Compared to MSCKF-based VIO, PO-KF reduces the 100m relative rotation and position error by 52% and 38% respectively, while demonstrating superior localization robustness and similar real-time performance.

The remaining sections of this paper are organized as follows. Section II provides a summary of related works. Section III presents the preliminary knowledge of MSCKF-based VIO. In Section IV, we introduce the overview of our proposed PO-KF. Following that, Section V details the pose-only measurement model, and Section VI introduces our base-frame selection algorithm. Section VII is dedicated to the comprehensive evaluation and analysis of the localization performance of PO-KF. Finally, Section VIII concludes the paper and outlines future works.

II. Related Works

VIO is a classical topic for estimating precise six degrees of freedom pose by integrating visual and inertial measurements. Considerable algorithms for VIO have emerged in the past two decased [3], [4] and our focus centers on the feature-based tight-coupled VIO system. In this section, we briefly review the VIO systems and feature representations for VIO.

A. Visual-Inertial Odometry Systems

The classification of the feature-based tight-coupled VIO systems revolves around state estimation methods, distinguishing them into optimization-based and filter-based solutions [4]. By leveraging iterated relinearization, the batch nonlinear optimization method proves highly applicability for the nonlinear state estimation of VIO. Thanks to the rapid development of high-performance computing, optimization-based VIO has successfully achieved real-time solutions and gained widespread popularity in related research. The representative optimization-based VIO, such as OKVIS [10] and IC-GVINS [11], fuse a series of past camera poses and current inertial states by a keyframe-based sliding window optimization approach, aiming at accurate trajectory estimation. Besides, some optimization-based solutions, like VINS-Mono [12] and ORB-SLAM3 [13], also extend advanced capabilities based on VIO, including online relocalization and global pose optimization. For stable localization accuracy in dynamic environments, DGM-VINS [14], DynaVINS [15], and SRVIO [16] achieve precise feature detection and robust loop closure by leveraging multiple geometric constraints or nerual networks.

Different from the optimization-based VIO, filter-based solutions linearize the system state once in its state propagation and update, respectively, leading to a more efficient VIO [17]. An earlier study [18] implemented VIO using an EKF, which expanded the state vector with feature depths and simultaneously estimated camera poses and feature depths. However, augmenting the feature depth dramatically increases the dimension of the state vector and compromises the efficiency advantages of a filter. To address this, Mourikis et al. proposed MSCKF [5], which only cloned camera poses [19] into the state vector and effectively constrained the state dimension. In this way, MSCKF maintains the low computational cost of filter-based VIO and achieves equivalent localization accuracy, becoming the most representative filter-based VIO algorithm. With a balance of real-time performance and localization accuracy, MSCKF has expanded its applications in drones [20] and wheeled carriers [21], [22].

Despite its superior efficiency, the localization accuracy of MSCKF is still impacted by the limitations of once linearization of the filter. To mitigate the linearization error in the measurement equations, the iterated EKF was introduced into MSCKF [23]. Although the iterated MSCKF achieves improved accuracy, it also complicates the measurement update process. Delayed feature initialization [6], [7] in MSCKF minimizes linearization errors on feature positions, but the one-time triangulation limits the accuracy of feature 3D positions. To address this limitation, hybrid MSCKF/SLAM VIO [24], [25] augments and estimates the 3D positions of long-tracking features, thereby enhancing the localization accuracy by the reduced feature 3D position errors and the prolonged tracking measurements. However, the hybrid system suffers from a complex system structure and increased computation cost. Moreover, the delayed updates of visual measurements caused by delayed feature initialization in these systems are neglected, which also affects the localization performance.

Several recent studies have aimed to enhance the robust-

ness of MSCKF, focusing on online calibration, observability consistency, and numerical stability. The online camera-IMU extrinsic calibration [26] and time offset calibration [27] ensure the normal operation of MSCKF in devices without precise hardware parameters. Various observability-based methodologies [28]–[30] are also employed in MSCKF to ensure correct observability properties. In terms of numerical stability, SR-ASWF [31] introduces a square-root inverse version of MSCKF and enables it to run on platforms with limited resources. Additionally, the state divergence caused by limited parallax during zero-velocity states is also a concern in VIO [32]. Odometry and non-holonomic constraints [33] [34] are effective auxiliary information for VIO during the stationary periods [21], [35], but only be applicable in wheeled carriers. Based on accurate zero-velocity detection, the zero-velocity update(ZUPT) is a common method for VIO across various devices [36]. However, as a virtual measurement, ZUPT strongly depends on accurate zero-velocity detection results and suitable measurement noise.

B. Feature Representations

Visual features play a crucial role in VIO and significantly affect system precision and stability. Among various visual features, point features are most commonly used in the general texture scene [6], [11], [13], [37]. In certain textureless environments, robust geometric features such as line features in PL-VIO [38], structure lines in struct-VIO [39], and plane features in PLP-VIO [40] are also utilized to maintain the localization accuracy. Despite the robustness of geometric features, point features remain the most common and efficient choice.

Feature positions in VIO are typically parameterized in a Cartesian coordinate system using three elements, triangulated from multi-view measurements and corresponding camera poses. Given that accurate depth is unknown in monocular camera measurements, alternative representations such as inverse depth parametrization [41] and parallax angle-based representation [42] have been proposed to handle features with minimal parallax. Additionally, feature points can also be represented using spherical coordinates or with a unit-bearing vector and a range scalar [43]. Although these representations prove effective in certain cases, errors in their specific values will persist after the one-time triangulation in MSCKF. The same challenge also arises in MSCKF when employing the aforementioned geometric and learning-based features.

Recent research has demonstrated that the two-view visual geometry is equivalent to a pair of pose-only constraints [8], effectively decoupling camera poses from 3D feature positions. Consequently, the pose-only constraint implicitly represents a feature's position using only visual measurements and poses of the two views. This pose-only representation has been extended to multi-view imaging geometry, targeting to efficiently solve the 3D visual construction problem [9]. By decoupling camera poses from feature positions, the pose-only solution dramatically simplifies the optimization problem in visual construction, providing an analytical reconstruction for spatial feature coordinates. Similarly, the pose-only representation is also employed in the optimization-based VIO system, such as PO-VINS [44] and PIPO-SLAM [45], showcasing superior computational efficiency while maintaining localization accuracy. Although filter-based VIOs address the same state estimation problem as optimization-based VIOs, their distinct system frameworks lead to different implementation strategies to incorporate pose-only representation, especially in the update strategy and base-frame selection. However, the incorporation of pose-only representation into the filter-based VIO remains unexplored. With the decoupling of camera poses and feature 3D positions in the pose-only representation, the filter-based VIO can explicitly eliminate feature coordinates from the measurement equation. Therefore, the challenges of linearization errors in feature positions and delayed visual measurement updates encountered in the traditional MSCKF-based VIO are expected to be addressed expertly, significantly improving its localization performance.

In summary, filter-based VIO solutions such as MSCKF are distinguished by their remarkable computational efficiency. However, concerns persist regarding their localization performance, including delayed measurement updates, linearization errors on feature positions, and insufficient localization robustness. Currently employed feature representations fail to tackle these challenges, while the pose-only representation shows the potential to address them, a potential that remains unexplored. Therefore, to resolve these concerns in MSCKF, we incorporate the pose-only representation into the filter-based VIO and propose PO-KF.

III. Preliminary of MSCKF-Based VIO

A. Coordinate Systems

The employed coordinate systems in this paper are detailed in Table I, where the u-frame has a unified z coordinate, the p-frame is a two-dimensional frame, and the w-frame aligns with the gravity direction.

TABLE I
DEFINITIONS OF THE EMPLOYED COORDINATE SYSTEMS

Coordinate system	Notation	Origin	Axes
IMU frame	b-frame	IMU center	F-R-D
camera frame	c-frame	camera's optical center	R-D-F
unified c-frame	u-frame	camera's optical center	R-D-F
pixel frame	p-frame	image's top-left corner	R-D
world frame	w-frame	first IMU position	F-R-V

F: forward, R: right, D: down, V: vertical

B. IMU Kinematic and State Equations

1) IMU Kinematic Equation: Since the Earth's rotation is generally ignored in low-cost MEMS IMU, we consider the w-frame as IMU's reference frame and denote the measured acceleration and angular velocity as $\boldsymbol{f}_{\mathrm{wb}}^{\mathrm{b}}$ and $\boldsymbol{\omega}_{\mathrm{wb}}^{\mathrm{b}}$, respectively. When modelling the IMU measurement errors, we only consider the dominant error, namely the IMU bias ($\boldsymbol{b}_g$, $\boldsymbol{b}_a$), and measurement white noise ($\boldsymbol{n}_g$, $\boldsymbol{n}_a$). Besides, in the inertial navigation system (INS), we compensate for the estimated IMU bias in the measurements at each step. Therefore, the IMU measurements in our INS algorithm are expressed as:

$$\hat{\boldsymbol{\omega}}_{\mathrm{wb}}^{\mathrm{b}} = \boldsymbol{\omega}_{\mathrm{wb}}^{\mathrm{b}} + \delta\boldsymbol{b}_g + \boldsymbol{n}_g, \quad \hat{\boldsymbol{f}}_{\mathrm{wb}}^{\mathrm{b}} = \boldsymbol{f}_{\mathrm{wb}}^{\mathrm{b}} + \delta\boldsymbol{b}_a + \boldsymbol{n}_a. \tag{1}$$

Taking the IMU measurements as input, an INS solves the navigation state by integrating the kinematic equations. Building upon the motion kinematic model [46], we derive the following IMU kinematic equations:

$$\dot{\boldsymbol{p}}_{\mathrm{b}}^{\mathrm{w}} = \boldsymbol{v}_{\mathrm{b}}^{\mathrm{w}}, \quad \dot{\boldsymbol{v}}_{\mathrm{b}}^{\mathrm{w}} = \mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \boldsymbol{f}_{\mathrm{wb}}^{\mathrm{b}} + \boldsymbol{g}^{\mathrm{w}}, \quad \dot{\boldsymbol{R}}_{\mathrm{b}}^{\mathrm{w}} = \boldsymbol{R}_{\mathrm{b}}^{\mathrm{w}} \left[\boldsymbol{\omega}_{\mathrm{wb}}^{\mathrm{w}}\right]_{\times}, \quad (2)$$

where, $\boldsymbol{p}_{\mathrm{b}}^{\mathrm{w}}$ and $\boldsymbol{v}_{\mathrm{b}}^{\mathrm{w}}$ denote the IMU's position and velocity in the w-frame, respectively, $\boldsymbol{g}^{\mathrm{w}}$ is the global gravity, the rotation matrix $\mathbf{R}_{\mathrm{b}}^{\mathrm{w}}$ signifies the IMU's attitude, and $[\cdot]_{\times}$ represents the skew-symmetric matrix of the corresponding vector.

Building upon Eq.(2) and employing the two-sample approximation [47], we can derive the refined discrete-time INS update equations using the discrete-time IMU measurements.

2) IMU State Equation: The position error, velocity error, attitude error, and bias errors of the gyroscope and accelerometer are incorporated in the IMU error-state, described as:

$$\boldsymbol{x}_{\mathrm{b},k} = \begin{bmatrix} (\delta\boldsymbol{\theta}_{b,k})^T & \left(\delta\boldsymbol{p}_{\mathrm{b}_k}^{\mathrm{w}}\right)^T & \left(\delta\boldsymbol{v}_{\mathrm{b}_k}^{\mathrm{w}}\right)^T & (\delta\boldsymbol{b}_{g,k})^T & (\delta\boldsymbol{b}_{a,k})^T \end{bmatrix}^T, \quad (3)$$

where, $\delta\boldsymbol{\theta}_{b,k}$ represents the attitude error, $\delta\boldsymbol{p}_{\mathrm{b}_k}^{\mathrm{w}}$ and $\delta\boldsymbol{v}_{\mathrm{b}_k}^{\mathrm{w}}$ are the position and velocity errors, respectively. Additionally, $\delta\boldsymbol{b}_{g,k}$ and $\delta\boldsymbol{b}_{a,k}$ denote the gyroscope and accelerometer bias errors. The error-states of the above parameters are defined as:

$$\begin{aligned} \hat{\mathbf{R}}_{\mathrm{b}}^{\mathrm{w}} &= \mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \left(\boldsymbol{I} - [\boldsymbol{\theta}_{\mathrm{b}}]_{\times}\right) \\ \hat{\boldsymbol{x}} &= \boldsymbol{x} - \delta\boldsymbol{x} \end{aligned}, \quad (4)$$

where, $\mathbf{R}_{\mathrm{b}}^{\mathrm{w}}$ is the true attitude, and $\boldsymbol{x}$ represents the other true states. Correspondingly, $\hat{\mathbf{R}}_{\mathrm{b}}^{\mathrm{w}}$ and $\hat{\boldsymbol{x}}$ denote the estimated states with errors, while $\boldsymbol{\theta}_{\mathrm{b}}$ and $\delta\boldsymbol{x}$ are the error-states.

By performing error perturbation on Eq.(2) and differential operations on Eq.(4), we obtain the continuous-time state equations for position, velocity, and attitude, as shown below:

$$\begin{aligned} \dot{\boldsymbol{\theta}}_{\mathrm{b}} &= -\left[\boldsymbol{\omega}_{\mathrm{wb}}^{\mathrm{w}}\right]_{\times} \boldsymbol{\theta}_{\mathrm{b}} - \delta\boldsymbol{b}_g - \boldsymbol{n}_g \\ \delta\dot{\boldsymbol{v}}_{\mathrm{b}}^{\mathrm{w}} &= -\mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \left[\boldsymbol{f}_{\mathrm{wb}}^{\mathrm{b}}\right]_{\times} \boldsymbol{\theta}_{\mathrm{b}} - \mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \delta\boldsymbol{b}_a - \mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \boldsymbol{n}_a. \\ \delta\dot{\boldsymbol{p}}_{\mathrm{b}}^{\mathrm{w}} &= \delta\boldsymbol{v}_{\mathrm{b}}^{\mathrm{w}} \end{aligned} \quad (5)$$

The bias errors of the gyroscope and accelerometer are modelled as the first-order Gauss-Markov process [47]. Based on the state equations in Eq.(5) and the model of IMU bias errors, we derive the state transition matrix $\boldsymbol{F}_I$ and noise-driven matrix $\boldsymbol{G}_I$ for the IMU's error-state.

C. Multi-State Constraint Kalman Filter

1) State Vector and Equation: In addition to the IMU error-state, the error-states of the cloned n historical IMU poses are also included in the state vector of MSCKF. Denoting the IMU pose at the i-th timestamp as $\boldsymbol{T}_{\mathrm{b}_i}^{\mathrm{w}} = \{\mathbf{R}_{\mathrm{b}_i}^{\mathrm{w}}, \boldsymbol{p}_{\mathrm{b}_i}^{\mathrm{w}}\}$, we obtain the complete state vector of MSCKF as:

$$\boldsymbol{x}_k = \begin{bmatrix} (\boldsymbol{x}_{\mathrm{b},k})^T & \delta\boldsymbol{T}_{\mathrm{b}_1}^{\mathrm{w}} & \cdots & \delta\boldsymbol{T}_{\mathrm{b}_i}^{\mathrm{w}} & \cdots & \delta\boldsymbol{T}_{\mathrm{b}_n}^{\mathrm{w}} \end{bmatrix}^T, \quad (6)$$

where, $\delta\boldsymbol{T}_{\mathrm{b}_i}^{\mathrm{w}} = \begin{bmatrix} (\boldsymbol{\theta}_{\mathrm{b}_i})^T & \left(\delta\boldsymbol{p}_{\mathrm{b}_i}^{\mathrm{w}}\right)^T \end{bmatrix}$ is the error-states of the i-th IMU pose. The camera poses, $\boldsymbol{T}_{\mathrm{c}_i}^{\mathrm{w}} = \{\mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}}, \boldsymbol{p}_{\mathrm{b}_i}^{\mathrm{w}}\}$, required in the measurement model, are transformed from the cloned IMU poses using the camera-IMU extrinsic parameters $\{\mathbf{R}_{\mathrm{c}}^{\mathrm{b}}, \boldsymbol{p}_{\mathrm{c}}^{\mathrm{b}}\}$.

Given that the cloned IMU poses are non-dynamic quantities and remain unchanged over time, the continuous-time differential equation of the i-th cloned IMU pose is $\delta\dot{\boldsymbol{T}}_{\mathrm{b}_i}^{\mathrm{w}} = \mathbf{0}_6$. Therefore, the state transition matrix and noise-driven matrix for the cloned IMU poses are both zero matrices.

Stacking the above state equations together and discretizing them, we obtain the discrete-time full-state equation as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{\omega}_k, \quad (7)$$

where, $\boldsymbol{\Phi}_k$ and $\boldsymbol{G}_k$ are the discrete-time state transition and noise-driven matrices, $\boldsymbol{\omega}_k$ is the equivalent discretization of the driving white noise, with the equivalent intensity $\boldsymbol{Q}_k$.

2) State Propagation and Augmentation: Since the state vector includes the IMU state and the cloned historical IMU poses, $\boldsymbol{\Phi}_k$ and $\boldsymbol{G}_k$ can be split as follows:

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi}_{I,k} & \mathbf{0}_{15\times(6n)} \\ \mathbf{0}_{(6n)\times 15} & \boldsymbol{I}_{6n} \end{bmatrix}, \quad \boldsymbol{G}_k = \begin{bmatrix} \boldsymbol{G}_{I,k} \\ \mathbf{0}_{6n\times 12} \end{bmatrix}, \quad (8)$$

where, $\boldsymbol{\Phi}_{I,k} = \exp\left(\int_{t_{k-1}}^{t_k} \boldsymbol{F}_I(t)dt\right) \approx \boldsymbol{I} + \boldsymbol{F}_I \Delta t_k$. Utilizing the complete transition matrix, we propagate the system state vector at t_{k-1} as $\boldsymbol{x}_{k|k-1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k$.

We partition the state covariance matrix $\boldsymbol{P}$ using the similar split form as presented in Eq.(8). Then the covariance matrix can be propagated with the following expression [5]:

$$\boldsymbol{P}_{k|k-1} = \begin{bmatrix} \boldsymbol{\Phi}_{I,k}\boldsymbol{P}_{I,k-1}\boldsymbol{\Phi}_{I,k}^T & \boldsymbol{\Phi}_{I,k}\boldsymbol{P}_{IT,k-1} \\ \boldsymbol{P}_{IT,k-1}^T\boldsymbol{\Phi}_{I,k}^T & \boldsymbol{P}_{T,k-1} \end{bmatrix} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T. \quad (9)$$

When a new image is received, we propagate the current IMU state and covariance to the image timestamp. Then, the current IMU pose is added to the state vector and the state covariance matrix is augmented as follows [5]:

$$\boldsymbol{P}_k \leftarrow \begin{bmatrix} \boldsymbol{I}_N \\ \boldsymbol{J}_{6\times N} \end{bmatrix} \boldsymbol{P}_k \begin{bmatrix} \boldsymbol{I}_N \\ \boldsymbol{J}_{6\times N} \end{bmatrix}^T, \quad (10)$$

where, $N = 15 + 6n$ is the dimension of the state vector, and $\boldsymbol{J}_{6\times N}$ is the Jacobian matrix of the augmented poses to the system state vector. When marginalization, the oldest cloned IMU pose will be removed from the sliding-window and the corresponding rows and columns in the covariance matrix will be eliminated.

3) Measurment Update: When a feature point is lost or reaches the maximum tracking length in this frame, all its visual measurements are employed to triangulate its 3D position $\boldsymbol{p}_f^{\mathrm{w}}$ and perform measurement updates in MSCKF [5], [6]. The reprojection measurement equation of this feature in the c_i-frame is formulated as follows [5], [6]:

$$\boldsymbol{z}_f^{\mathrm{p}_i} = \tilde{\boldsymbol{p}}_f^{\mathrm{p}_i} - \mathbf{h}_p\left(\mathbf{h}_d\left(\mathbf{h}_u\left(\mathbf{h}_t\left(\mathbf{R}_{\mathrm{w}}^{\mathrm{c}_i}, \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}}, \boldsymbol{p}_f^{\mathrm{w}}\right)\right), \boldsymbol{K}\right), \boldsymbol{\zeta}\right), \quad (11)$$

where, $\tilde{\boldsymbol{p}}_f^{\mathrm{p}_i}$ and $\boldsymbol{z}_f^{\mathrm{p}_i}$ are the visual measurement and the reprojection error, $\mathbf{h}_p$ and $\mathbf{h}_d$ are the projection and distortion functions in the camera imaging process, $\mathbf{h}_u$ is the normalized process of the feature's position in the c_i-frame, $\mathbf{h}_t$ is the coordinates transformation function, and $\boldsymbol{K}$ and $\boldsymbol{\zeta}$ are the camera's projection and distortion parameters.

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

$$\boldsymbol{z}^{\mathrm{p}} = \boldsymbol{H}_x \boldsymbol{x} + \boldsymbol{H}_f \delta\boldsymbol{p}_f^{\mathrm{w}} + \boldsymbol{n}^{\mathrm{p}}, \quad (12)$$

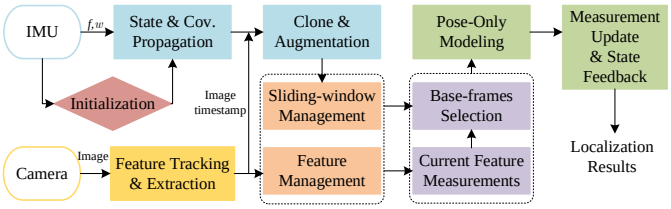


Fig. 1. Overview of the proposed PO-KF.

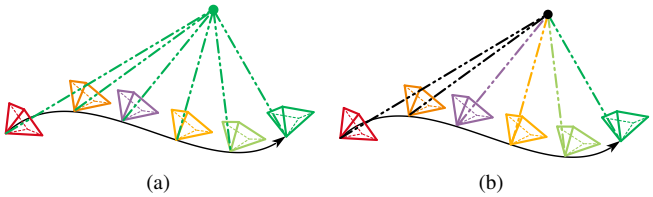


Fig. 2. Two measurement update strategies, with camera colors denoting the timestamp. (a) MSCKF update strategy: all measurements are updated at the green timestamp; (b) PO-KF update strategy: black measurements denote base-frames, and other measurements are updated at their own timestamp.

where, $\boldsymbol{H}_x$ and $\boldsymbol{H}_f$ are the measurement Jacobian matrices with respect to the state vector and feature's position, respectively. Additionally, $\boldsymbol{z}^{\mathrm{p}}$ and $\boldsymbol{n}^{\mathrm{p}}$ denote the measurement innovation and noise, respectively.

Then, nullspace projection is performed to eliminate the feature position from Eq.(12). Before proceeding with the standard Kalman update, measurement compression is also performed to reduce the measurement dimension.

D. Zero-Velocity Update

When carriers remain stationary for an extended period, the camera frames corresponding to the cloned timestamps yield nearly identical poses. The limited parallax causes feature triangulation to fail, and no measurements are available to update the system state, leading to quick state divergence. Fortunately, the ZUPT, including zero-velocity update and zero-heading-rate update, serves as effective supplementary measurement information for VIO in stationary situations. Leveraging the covariance of several original IMU measurements and the disparity of visual measurements, we can accurately detect stationary periods [48]. Subsequently, we calculate the measurement innovations and matrices from the following measurement equations:

$$\boldsymbol{z}_v = \boldsymbol{v}_{\mathrm{b}}^{\mathrm{w}} - \boldsymbol{n}_v, \quad z_\psi = \boldsymbol{e}_3 \mathbf{R}_{\mathrm{b}}^{\mathrm{w}} \boldsymbol{\omega}_{\mathrm{wb}}^{\mathrm{b}} + n_\psi, \tag{13}$$

where, $\boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, $\boldsymbol{n}_v$ and n_ψ are the measurement noises. To enhance the ZUPT constraint, we also incorporate the integral measurement model of the two measurement equations [49] in our implementation. It is also worth mentioning that appropriate measurement noises must be empirically preset, as the above measurements are virtual observations. Besides, the noise intensities have a considerable impact on the constraint strength of the estimated state.

IV. System Design of PO-KF

The system overview of our proposed PO-KF is depicted in Fig.1. After system initialization, IMU measurements and image data are processed for INS state recurrence and feature extraction and tracking. Then, the solved INS states and their covariance are propagated and augmented. Once enough cloned poses and visual measurements have been accumulated for state updates, the best base-frames are selected and pose-only measurement models are constructed. Finally, the selected visual measurements are used to update the system state.

Compared to MSCKF-based VIO, the proposed PO-KF shares similar state propagation and augmentation processes but differs in its measurement models and update strategies. Fig.2 illustrates the update strategies of MSCKF and PO-KF, with various timestamps represented by different colors. As shown in Fig.2a, MSCKF constructs measurement models using all measurements of a feature and updates the system state at the time of the last measurement. In contrast, PO-KF formulates its measurement equation using the newest measurement and the selected base-frames once the feature tracking length reaches 3 frames.

In addition to the pose-only measurement model and base-frame selection modules, PO-KF also extends the state equation by incorporating camera-IMU extrinsic calibration to enhance the system robustness. Besides, specialized strategies are designed for managing the cloned poses and visual features to support the measurement updates in PO-KF. These strategies include a dedicated strategy for handling zero-velocity states. The details of each module shown in Fig.1 will be explained in the following subsections.

1) Initialization: For robust and accurate initialization, PO-KF automatically employs both static initialization [6] [50] and dynamic initialization [51] to determine the initial velocity, roll and pitch angles, and IMU biases. The initial absolute position and absolute heading angle are set to 0, while their corresponding covariances are also initialized to 0. The covariances for the remaining states are assigned empirical values.

2) Feature Tracking and Extraction: The visual front-end of PO-KF extracts the Shi-Tomasi corner points from the new images as visual features and utilizes Optical-Flow-Tracking for efficient and accurate feature tracking. To ensure a balanced feature distribution, the image in divided into several grids, and feature points are extracted evenly within each grid. Additionally, the distance between feature points is regulated to prevent clustering.

3) State Propagation and Augmentation: On the basis of state propagation and augmentation procedures of standard MSCKF, PO-KF incorporates online calibration of the camera-IMU extrinsic parameters and time offset [26], [27]. We model the extrinsic parameters and time offset as random walk processes and augment their error-states, denoted as $\delta \boldsymbol{T}_{\mathrm{c}}^{\mathrm{b}} = \begin{bmatrix} (\boldsymbol{\theta}_{\mathrm{c,b}})^T & (\delta \boldsymbol{p}_{\mathrm{c}}^{\mathrm{b}})^T \end{bmatrix}^T$ and δt_d, into the state vector.

The error-state definitions of the camera-IMU relative position and the time offset follow the definition in Eq.(4). The error-state of the camera-IMU relative rotation is defined as:

$$\hat{\mathbf{R}}_{\mathrm{c}}^{\mathrm{b}} = \mathbf{R}_{\mathrm{c}}^{\mathrm{b}} \left(\boldsymbol{I} - [\boldsymbol{\theta}_{\mathrm{c,b}}]_\times \right), \tag{14}$$

where, $\mathbf{R}_{\mathrm{c}}^{\mathrm{b}}$ and $\hat{\mathbf{R}}_{\mathrm{c}}^{\mathrm{b}}$ denote the true and estimated rotation between the IMU and camera, and $\boldsymbol{\theta}_{\mathrm{b,c}}$ is the error-state.

These error-states of extrinsic and time offset are augmented into the state vector $\boldsymbol{x}_k$ and their driven noises are added into the system noises $\boldsymbol{\omega}_k$. The new total state vector $\boldsymbol{x}_k$ is:

$$\boldsymbol{x}_k = \begin{bmatrix} (\boldsymbol{x}_{\mathrm{b},k})^T & \delta\boldsymbol{T}_{\mathrm{c}}^{\mathrm{b}} & \delta t_d & \delta\boldsymbol{T}_{\mathrm{b}_1}^{\mathrm{w}} & \cdots & \delta\boldsymbol{T}_{\mathrm{b}_i}^{\mathrm{w}} & \cdots & \delta\boldsymbol{T}_{\mathrm{b}_n}^{\mathrm{w}} \end{bmatrix}^T. \tag{15}$$

The state transition matrix and the noise-driven matrix are modified corresponding to the new state vector and system noise. During covariance augmentation, we introduce $\boldsymbol{J}_t$, the Jacobian matrix of the cloned pose with respect to the time offset under the uniform velocity assumption, to update the $\boldsymbol{J}_{6\times N}$ in Eq.(10). This allows the time offset to be updated when the Kalman update is performed without modelling the time offset in the measurement equation [27]. In this way, the updated $\boldsymbol{J}_{6\times(N+7)}$ and $\boldsymbol{J}_t$ are expressed as:

$$\begin{aligned} \boldsymbol{J}_{6\times(N+7)} &= \begin{bmatrix} \boldsymbol{I}_6 & \mathbf{0}_{6\times 9} & \mathbf{0}_{6\times 6} & \boldsymbol{J}_t & \mathbf{0}_{6\times 6n} \end{bmatrix} \\ \boldsymbol{J}_t &= \begin{bmatrix} \left(\boldsymbol{\omega}_{\mathrm{wb},k}^{\mathrm{b}}\right)^T & \left(\boldsymbol{v}_{\mathrm{b},k}^{\mathrm{w}}\right)^T \end{bmatrix}^T \end{aligned}. \tag{16}$$

4) Feature Database and Sliding-Window Management: The proposed PO-KF dynamically manages the sliding-window and feature database. Sliding-window management in PO-KF is similar to that in MSCKF, which adds the IMU pose at the newest image time into the sliding-window and deletes the oldest cloned pose when the sliding-window is full. The IMU pose at every image time is included in the sliding-window.

Feature management in PO-KF differs from that in MSCKF. In MSCKF, all visual measurements of a feature are employed together to construct the measurement model and update the system state, as illustrated in Fig. 2. On the contrary, PO-KF constructs the feature's measurement equation only in the newest image plane. This implies that earlier visual measurements in the feature database are not considered system observations. Consequently, the elimination of the feature's all measurements, which is performed after the measurement update in MSCKF, is not required in PO-KF. Instead, PO-KF adds the newest extracted visual measurements to the feature management module and removes feature measurements under the following two conditions. First, the oldest measurement of a feature is eliminated when its measurements exceed the sliding-window size. Second, all measurements of a feature are removed when it is lost.

5) Strategy for Zero-Velocity States: We introduce special management strategies for the sliding-window and feature database in PO-KF to effectively handle zero-velocity states without ZUPT. During the stationary periods, the newest two cloned IMU poses are very similar under the regular management strategies, resulting in limited parallax within the sliding-window. To guarantee sufficient parallax, we only need to marginalize the newest cloned IMU pose and the newest visual measurements in PO-KF, rather than the oldest ones, during the zero-velocity states. These special strategies enable PO-KF to maintain enough parallax in the remaining IMU poses within the sliding-window, even if the carrier keeps stationary for an extended period. Correspondingly, PO-KF can construct a valid measurement equation for the newest visual measurement and sustain accurate localization without relying on ZUPT.

6) Base-Frame Selection and Measurement Model: Subsequently, the current measurements are picked out from the feature database to construct the pose-only measurement model. Specifically, the best two base-frames for each feature are firstly selected from the sliding-window. Then, they are employed in the pose-only measurement model to acquire the complete measurement matrix, innovation, and noise. Finally, the Kalman update is performed, and the system states are updated using the estimated error-states.

Detailed information on the construction of the pose-only measurement model and the algorithm design of the base-frame selection is provided in the following two sections.

V. Pose-Only Visual Measurement Model

This section introduces the state measurement model based on the pose-only representation, including the pose-only representation, the detailed measurement equations, and an analysis of the associated advantage.

A. Pose-Only Representation

The pose-only representation, equivalent to the classical multi-view geometry, is raised in [8], [9] to estimate camera motion efficiently and reconstruct the spatial feature coordinates analytically. In this subsection, we first present the derivation of the pose-only representation.

Consider a 3D feature point in the w-frame, with its coordinates denoted as $\boldsymbol{p}_f^{\mathrm{w}}$, is observed in several images. The coordinate of this feature in the i-th camera frame is denoted as $\boldsymbol{p}_f^{\mathrm{c}_i}$ while that in the unified camera frame is denoted as $\boldsymbol{x}_f^{\mathrm{u}_i}$. The feature position can be derived using the camera pose as follows:

$$\boldsymbol{p}_f^{\mathrm{w}} = \mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}}\boldsymbol{p}_f^{\mathrm{c}_i} + \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}} = z_f^{\mathrm{c}_i}\mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}}\boldsymbol{x}_f^{\mathrm{u}_i} + \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}}, \tag{17}$$

where, $\{\mathbf{R}_{\mathrm{w}}^{\mathrm{c}_i}, \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}}\}$ is the i-th camera pose in the w-frame and $z_f^{\mathrm{c}_i} = p_{f,z}^{\mathrm{c}_i}$ is the feature depth in the c_i-frame. The normalized measurement in the u_i-frame $\boldsymbol{x}_f^{\mathrm{u}_i}$ is derived from back projection and undistortion from the origin measurement.

Selecting the feature's measurement in the i-th and j-th images, we obtain the two-view pose-only constraints between the image pair (i, j):

$$z_f^{\mathrm{c}_i}\mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i} + \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} = z_f^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_j}. \tag{18}$$

Left-multipling $\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times$ on both sides of Eq.(18), we yield the following expression:

$$z_f^{\mathrm{c}_i}\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i} = -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j}. \tag{19}$$

Taking the magnitude of Eq.(19), it yields the feature depth:

$$z_f^{\mathrm{c}_i} = \frac{\| -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j}\|}{\| \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i}\|} = \frac{\lambda_f^{\mathrm{c}_j}}{\theta_{i,j}} \triangleq d_{f,i}^{(i,j)}, \tag{20}$$

where, $d_{f,i}^{(i,j)}$ is the feature depth constrained by the image pair (i,j), and $\theta_{i,j}$ denotes the parallax of the two base-frames. Similarly, the feature depth in the c_j-frame is marked as $d_{f,j}^{(i,j)}$.

Now, we represent the features' 3D positions with only camera poses and their normalized measurements as follows:

$$\boldsymbol{p}_f^{\mathrm{c}_i} = d_{f,i}^{(i,j)} \boldsymbol{x}_f^{\mathrm{u}_i}, \boldsymbol{p}_f^{\mathrm{c}_j} = d_{f,j}^{(i,j)} \boldsymbol{x}_f^{\mathrm{u}_j}. \tag{21}$$

B. Measurement Update

When the visual measurement of the feature $\boldsymbol{p}_f^{\mathrm{w}}$ at l-th image comes, we construct its measurement equation in the p_l-frame using the current measurement $\tilde{\boldsymbol{p}}_f^{\mathrm{p}_l}$ and the 3D position represented by the image pair (i,j). We name the image pair (i,j) as the two base-frames of the feature's current measurement. The two base-frames appear in the feature's current reprojection equation, replacing the feature's position. The measurement equation of this feature $\boldsymbol{p}_f^{\mathrm{w}}$ in the p_l-frame are as follows:

$$\boldsymbol{z}_f^{\mathrm{p}_l} = \tilde{\boldsymbol{p}}_f^{\mathrm{p}_l} - \mathbf{h}_p\left(\mathbf{h}_d\left(\mathbf{h}_u\left(\mathbf{h}_t\left(d_{f,i}^{(i,j)}, \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_i}, \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}}, \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_l}, \boldsymbol{p}_{\mathrm{c}_l}^{\mathrm{w}}\right)\right), \boldsymbol{K}\right), \boldsymbol{\zeta}\right), \tag{22}$$

where, $\boldsymbol{z}_f^{\mathrm{p}_l}$ is known as the measurement innovation, $\mathbf{h}_p$, $\mathbf{h}_d$, $\mathbf{h}_u$ and $\mathbf{h}_t$ are defined in Sec.III-C3, and $\boldsymbol{K}$ and $\boldsymbol{\zeta}$ are the camera intrinsic parameters.

As described in Eq.(20), the feature's depth $d_{f,i}^{(i,j)}$ is represented with the poses and measurements of its two base-frames. The frame poses are a function of the cloned IMU poses and the camera-IMU extrinsic parameters, all of which are included in the state vector $\boldsymbol{x}$. Therefore, except for the known camera's intrinsic parameters and the feature's measurements, the pose-only measurement model is only related to the state vector and has nothing to do with the feature's 3D position. Consequently, Eq.(22) can be simplified as follows:

$$\boldsymbol{z}_f^{\mathrm{p}_l} = \boldsymbol{H}_{x,f} \boldsymbol{x} + \boldsymbol{n}_f^{\mathrm{p}_l}, \tag{23}$$

where, $\boldsymbol{H}_x$ is the Jacobian matrix with respect to the complete state vector, and $\boldsymbol{n}_f^{\mathrm{p}_l}$ is the measurement noise. In the following content, we present the complete expression of the measurement equation and derive the complete Jacobian matrix.

1) Jacobian of the camera pose to the state vector: Denoting the error-state of the i-th camera pose as $\delta \boldsymbol{T}_{\mathrm{c}_i}^{\mathrm{w}} = \begin{bmatrix} \boldsymbol{\theta}_{\mathrm{c}_i}^T & \left(\delta \boldsymbol{p}_{\mathrm{c}_i}^{w}\right)^T \end{bmatrix}^T$, we derive its relation with the error-states of the i-th IMU pose and the extrinsic parameters as:

$$\begin{bmatrix} \boldsymbol{\theta}_{\mathrm{c}_i} \\ \delta \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{R}_{\mathrm{b}}^{\mathrm{c}} & \mathbf{0}_3 \\ \mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}} \left[\boldsymbol{p}_{\mathrm{b}}^{\mathrm{c}}\right]_\times & -\mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}} & -\mathbf{R}_{\mathrm{b}_i}^{\mathrm{w}} \left[\boldsymbol{p}_{\mathrm{c}}^{\mathrm{b}}\right]_\times & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_{\mathrm{c,b}} \\ \delta \boldsymbol{p}_{\mathrm{c}}^{\mathrm{b}} \\ \boldsymbol{\theta}_{\mathrm{b}_i} \\ \delta \boldsymbol{p}_{\mathrm{b}_i}^{\mathrm{w}} \end{bmatrix}. \tag{24}$$

Therefore, we get the Jacobian matrix of the i-th camera pose with respect to the complete state vector as:

$$\boldsymbol{J}_{x}^{T_{\mathrm{c},i}^{\mathrm{w}}} = \begin{bmatrix} \mathbf{0}_{3\times 15} & \boldsymbol{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{R}_{\mathrm{b}}^{\mathrm{c}} & \mathbf{0}_3 & \cdots \\ \mathbf{0}_{3\times 15} & \mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}} \left[\boldsymbol{p}_{\mathrm{b}}^{\mathrm{c}}\right]_\times & -\mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}} & \cdots & -\mathbf{R}_{\mathrm{b}_i}^{\mathrm{w}} \left[\boldsymbol{p}_{\mathrm{c}}^{\mathrm{b}}\right]_\times & \boldsymbol{I}_3 & \cdots \end{bmatrix}. \tag{25}$$

The Jacobian matries of the j-th and l-th camera poses, denoted as $\boldsymbol{J}_{x}^{T_{\mathrm{c},j}^{\mathrm{w}}}$ and $\boldsymbol{J}_{x}^{T_{\mathrm{c},l}^{\mathrm{w}}}$, can be derived in the same way.

2) Jacobian of the feature depth to the camera poses: The feature depth $d_{f,i}^{(i,j)}$ is represented using the poses of the two-base frames, as shown in Eq.(20). We denote the Jacobian matrices to the two base-frames as $\boldsymbol{J}_{T_{\mathrm{c},i}^{\mathrm{w}}}^{d_{f,i}}$ and $\boldsymbol{J}_{T_{\mathrm{c},j}^{\mathrm{w}}}^{d_{f,i}}$, of which the detailed derivation is presented in Appendix A.

3) Jacobian of the transformation function: Denoting feature's position in the c_l-frame as $\boldsymbol{p}_f^{\mathrm{c}_l}$, the coordinate transformation function are expressed as:

$$\boldsymbol{p}_f^{\mathrm{c}_l} = \mathbf{h}_t = \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_l}\left(d_{f,i}^{(i,j)} \mathbf{R}_{\mathrm{c}_i}^{\mathrm{w}} \boldsymbol{x}_f^{\mathrm{u}_i} + \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{w}} - \boldsymbol{p}_{\mathrm{c}_l}^{\mathrm{w}}\right). \tag{26}$$

During the transform process, we verify the validity of the depth $d_{f,i}^{(i,j)}$ and ensure a positive feature depth in the c_l-frame for numerical stability. Feature measurements resulting in invalid or negative feature depths will be disregarded. Performing error disturbance on Eq.(26), we derive the Jacobian matrices of $\boldsymbol{p}_f^{\mathrm{c}_l}$ to the above variables, as shown below:

$$\begin{aligned} \boldsymbol{J}_{d_{f,i}}^{p_f^{\mathrm{c}_l}} &= \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_l} \boldsymbol{x}_f^{\mathrm{u}_i}, \\ \boldsymbol{J}_{T_{c,i}^{w}}^{p_f^{\mathrm{c}_l}} &= \begin{bmatrix} -d_{f,i}^{(i,j)} \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_l} \left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times & \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_l} \end{bmatrix}, \\ \boldsymbol{J}_{T_{c,l}^{w}}^{p_f^{\mathrm{c}_l}} &= \begin{bmatrix} \left[\boldsymbol{p}_f^{\mathrm{c}_l}\right]_\times & -\mathbf{R}_{\mathrm{w}}^{\mathrm{c}_l} \end{bmatrix}. \end{aligned} \tag{27}$$

4) Jacobian of the normalized function: The normalized process calculates the normalized feature coordinates in the u_l-frame. The normalized coordinate, $\boldsymbol{x}_f^{\mathrm{u}_l} = \mathbf{h}_u(\boldsymbol{p}_f^{\mathrm{c}_l})$, and the Jacobian matrix with respect to $\boldsymbol{p}_f^{\mathrm{c}_l}$ are as follows:

$$\boldsymbol{x}_f^{\mathrm{u}_l} = \begin{bmatrix} x_f^{\mathrm{u}_l} \\ y_f^{\mathrm{u}_l} \end{bmatrix} = \begin{bmatrix} \frac{p_{f,x}^{\mathrm{c}_l}}{p_{f,z}^{\mathrm{c}_l}} \\ \frac{p_{f,y}^{\mathrm{c}_l}}{p_{f,z}^{\mathrm{c}_l}} \end{bmatrix}, \boldsymbol{J}_{p_f^{\mathrm{c}_l}}^{x_f^{\mathrm{u}_l}} = \begin{bmatrix} \frac{1}{p_{f,z}^{\mathrm{c}_l}} & 0 & -\frac{p_{f,x}^{\mathrm{c}_l}}{\left(p_{f,z}^{\mathrm{c}_l}\right)^2} \\ 0 & \frac{1}{p_{f,z}^{\mathrm{c}_l}} & -\frac{p_{f,y}^{\mathrm{c}_l}}{\left(p_{f,z}^{\mathrm{c}_l}\right)^2} \end{bmatrix}. \tag{28}$$

5) Jacobian of the distortion and projection functions: We denote the distorted normalized feature coordinates as $\boldsymbol{x}_{f,d}^{\mathrm{u}_l} = \mathbf{h}_d(\boldsymbol{x}_f^{\mathrm{u}_l}) = \begin{bmatrix} x_{f,d}^{\mathrm{u}_l} & y_{f,d}^{\mathrm{u}_l} \end{bmatrix}^T$, and denote the distorted feature measurement in the p_l-frame as $\boldsymbol{p}_f^{\mathrm{p}_l} = \mathbf{h}_p(\boldsymbol{x}_{f,d}^{\mathrm{u}_l}) = \begin{bmatrix} u^{\mathrm{p}_l} & v^{\mathrm{p}_l} \end{bmatrix}^T$. The Jacobian matrices of the distortion and projection functions are denoted as $\boldsymbol{J}_{x_f^{\mathrm{u}_l}}^{x_{f,d}^{\mathrm{u}_l}}$ and $\boldsymbol{J}_{x_{f,d}^{\mathrm{u}_l}}^{p_f^{\mathrm{p}_l}}$, respectively. The detailed distortion and projection process and the derivations of the Jacobian matrices are given in Appendix B.

6) Complete measurement Jacobian matrix: Finally, we solve the complete Jacobian matrix of the pose-only measurement model based on the chain rule, as shown below:

$$\begin{aligned} \boldsymbol{H}_{x,f} = & \boldsymbol{J}_{x_{f,d}^{\mathrm{u}_l}}^{p_f^{\mathrm{p}_l}} \boldsymbol{J}_{x_f^{\mathrm{u}_l}}^{x_{f,d}^{\mathrm{u}_l}} \boldsymbol{J}_{p_f^{\mathrm{c}_l}}^{x_f^{\mathrm{u}_l}} \cdot \\ & \left(\boldsymbol{J}_{d_{f,i}}^{p_f^{\mathrm{c}_l}} \left(\boldsymbol{J}_{T_{\mathrm{c},i}^{\mathrm{w}}}^{d_{f,i}} \boldsymbol{J}_{x}^{T_{\mathrm{c},i}^{\mathrm{w}}} + \boldsymbol{J}_{T_{\mathrm{c},j}^{\mathrm{w}}}^{d_{f,i}} \boldsymbol{J}_{x}^{T_{\mathrm{c},j}^{\mathrm{w}}}\right) + \boldsymbol{J}_{T_{\mathrm{c},i}^{\mathrm{w}}}^{p_f^{\mathrm{c}_l}} \boldsymbol{J}_{x}^{T_{\mathrm{c},i}^{\mathrm{w}}} + \boldsymbol{J}_{T_{\mathrm{c},l}^{\mathrm{w}}}^{p_f^{\mathrm{c}_l}} \boldsymbol{J}_{x}^{T_{\mathrm{c},l}^{\mathrm{w}}}\right). \end{aligned} \tag{29}$$

By now, we construct the complete measurement model for the feature $\boldsymbol{p}_f^{\mathrm{w}}$ at timestamp t_k. Before including in the overall measurement model, feature measurements are also checked by the standard chi-square test. By stacking the measurement equations of all features, we obtain the total measurement matrix, innovation, and noise. Then we can directly calculate the updated system state vector using the standard Kalman update equation without nullspace projection.

C. Analysis of the Model's Advantage

The proposed PO-KF is essentially a filter-based VIO, where the state propagation and measurement update are performed once at each timestamp. Therefore, we choose

MSCKF, the most popular filter-based VIO among the state-of-the-art (SOTA) methods, as the baseline for analysis.

The advantages of the proposed PO-KF compared with MSCKF are twofold. The first is to eliminate the linearization error of features' 3D positions, and the second is the immediate updating of visual measurements. Here, we analyze the two advantages in detail.

1) Elimination of feature's 3D positions: As indicated in Eq.(12), feature 3D positions are essential in formulating the measurement equation of MSCKF. As all visual measurements are used for feature triangulation, both limited parallax and unstable numerical solutions lead to failures. Consequently, when feature triangulation fails, the corresponding all visual measurements become unusable for the system state, limiting the availability of measurements. However, PO-KF can formulate the measurement equation as long as the parallax $\theta_{i,j}$ between the two base-frames is non-zero. Furthermore, even with a well-constructed measurement equation, linearization errors on feature 3D positions are still introduced into the measurement matrix and nullspace projection process of MSCKF. In contrast, the PO-KF measurement equation, as defined in Eq.(23), theoretically eliminates feature 3D positions, thereby avoiding the linearization errors on these positions. Generally, PO-KF yields more valid visual measurements and a more accurate measurement equation.

2) Immediate updating of visual measurements: In the update strategy of MSCKF shown in Fig.2a, a feature is triangulated only when it is lost or reaches the maximum tracking length, known as delayed feature initialization. That means that all visual measurements of this feature are employed together to construct measurement equations and update the system state at the green camera timestamp. In contrast, PO-KF formulates the measurement equation as long as a feature has 3 or more measurements, independent of the feature's 3D position. As a result, after the feature's second observation, the newest visual measurement is immediately used to update the system state, as illustrated in Fig.2b. The immediate updating capability of PO-KF is advantageous, hopefully allowing it to maintain a small error state and thus reduce linearization errors on the system state in the measurement equation.

3) Quantative Validation: We analyze the measurement counts of both MSCKF and PO-KF using a group of common robot data. As depicted in Fig.3, the upper subplot illustrates the number of measurements inputted into the measurement model at each timestamp, while the lower subplot displays the measurement counts employed to update the system. The input and updated measurement numbers in MSCKF demonstrate larger fluctuations compared to PO-KF due to delayed feature initialization. The average input measurement sizes of PO-KF and MSCKF, measuring 106 and 108, are quite similar since they employ the same visual front-end. However, the average updated measurement counts of MSCKF and PO-KF are 62 and 83, respectively. The reduction of measurement counts in MSCKF primarily results from the unsuccessful feature triangulation, whereas in PO-KF, it results from features having fewer than 3 measurements. The greater reduction in measurement counts in MSCKF implies that 19 feature points, each with 3 or more measurements, are discarded due to failed

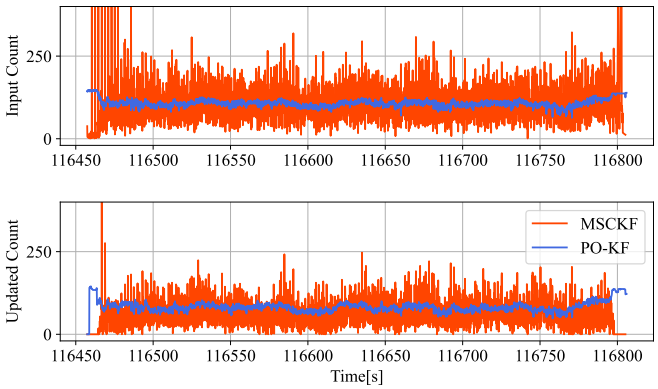


Fig. 3. Input and updated measurement counts of measurement update modules in MSCKF and PO-KF.

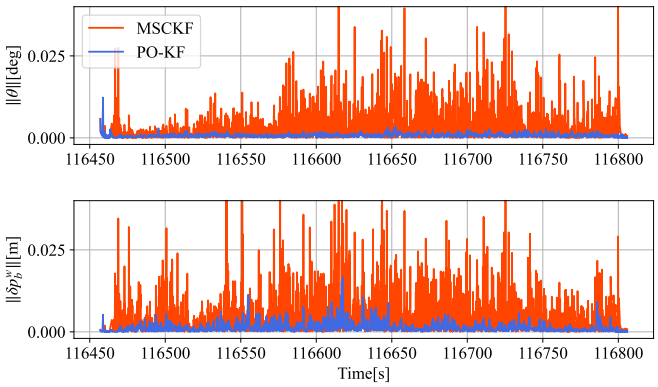


Fig. 4. Magnitudes of the updated current IMU pose error-states in MSCKF and PO-KF.

triangulation. Thus, the elimination of feature positions from the measurement equation ensures more visual measurements in PO-KF compared to MSCKF.

To validate the advantages of the immediate update, we conducted a comparison of the correction error-state calculated from the Kalman update process for both MSCKF and PO-KF. Fig.4 displays the correction magnitude of the current IMU's position and attitude. Benefiting from the constant updated measurement count, the state vector of PO-KF in Fig.4 exhibits a more stable correction magnitude compared to MSCKF. Consequently, the immediate update also ensures that the system state of PO-KF remains closer to the truth state. Therefore, the magnitude of calculated position and attitude errors in PO-KF are significantly smaller than those in MSCKF, which further reflects the advantages of PO-KF.

VI. Information Matrix-Derived Base-Frame Selection

The pose-only measurement model requires two base-frames to represent a feature's depth and construct its measurement equation. When the feature is tracked for more than 3 frames, there are multiple options for selecting its base-frames. Differing from taking the maximum parallax as the selection criterion in optimization-based solutions, we introduce a dedicated base-frame selection algorithm for PO-

KF in this section, designed to choose the optimal two base-frames for each feature measurement.

A. Selection Formulation

The derived pose-only measurement model for a single feature at a time essentially represents a constraint involving camera poses of the current image frame and the previous two base-frames. In the context of the current observed frame, denoted as n ($n \geq 3$), the selection of base frames involves considering historical observed frames from 1 to $n-1$. Since the INS diverges with time, we opt to select the 1-st observed frame as the feature's first base-frame to best mitigate the INS divergence. Thus, as depicted in Fig.5, the base-frame selection problem becomes selecting one base-frame from the 2nd and $(n-1)$-th historically observed frames to formulate the most effective constraint on the camera poses.

The constraint on the camera poses of the current and the two base-frames is regarded as the indicator of the base-frame selection. Denoting the indices of the three camera frames as i, j, and l, we formulate the constraint between the three camera poses and the feature's position as follows:

$$\begin{aligned} z^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_j} &= \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j} z^{\mathrm{c}_i}\boldsymbol{x}_f^{\mathrm{u}_i} + \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} \\ z^{\mathrm{c}_l}\boldsymbol{x}_f^{\mathrm{u}_l} &= \mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_l} z^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_j} + \boldsymbol{p}_{\mathrm{c}_j}^{\mathrm{c}_l} \\ z^{\mathrm{c}_i}\boldsymbol{x}_f^{\mathrm{u}_i} &= \mathbf{R}_{\mathrm{c}_l}^{\mathrm{c}_i} z^{\mathrm{c}_l}\boldsymbol{x}_f^{\mathrm{u}_l} + \boldsymbol{p}_{\mathrm{c}_l}^{\mathrm{c}_i} \end{aligned}. \tag{30}$$

The total dimension of the feature's three measurements is 6, whereas that of the three camera poses is 18. The above equation set is an underdetermined equation concerning the three camera poses, and as such, a unique solution cannot be determined. As depicted in Fig.5, the feature's measurements establish relative constraints among the three observed camera poses and the feature's position, i.e., the feature's depths in the three camera frames. Thus, to avoid the aforementioned underdetermined problem, we put our focus on the feature depths and consider the equation set as a constraint with respect to the three depths. Consequently, the base-frame selection problem is transformed into selecting the second base-frame to obtain the most accurate feature depths in the three camera frames.

B. Selection Indicator

To derive the equations set corresponding to the feature's three depths, we left multiply $\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times$, $\left[\boldsymbol{x}_f^{\mathrm{u}_l}\right]_\times$, and $\left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times$ on both sides of the three equations in Eq.(30) and simplify each equation individually. Thus we obtain the following equations:

$$\begin{aligned} z^{\mathrm{c}_i}\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i} &= -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} \\ z^{\mathrm{c}_j}\left[\boldsymbol{x}_f^{\mathrm{u}_l}\right]_\times \mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_l}\boldsymbol{x}_f^{\mathrm{u}_j} &= -\left[\boldsymbol{x}_f^{\mathrm{u}_l}\right]_\times \boldsymbol{p}_{\mathrm{c}_j}^{\mathrm{c}_l} \\ z^{\mathrm{c}_l}\left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \mathbf{R}_{\mathrm{c}_l}^{\mathrm{c}_i}\boldsymbol{x}_f^{\mathrm{u}_l} &= -\left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \boldsymbol{p}_{\mathrm{c}_l}^{\mathrm{c}_i} \end{aligned}. \tag{31}$$

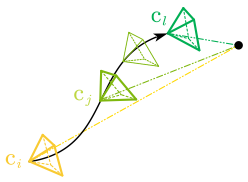


Fig. 5. PO-KF measurement constraint.

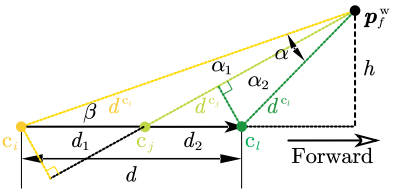


Fig. 6. The diagram of base-frame selection.

Transform the equation set into an overdetermined equation with respect to the feature's three depths, as shown below:

$$\underbrace{\begin{bmatrix} \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} & \left[\boldsymbol{x}_f^{\mathrm{u}_l}\right]_\times \mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_l}\boldsymbol{x}_f^{\mathrm{u}_j} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & \left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \mathbf{R}_{\mathrm{c}_l}^{\mathrm{c}_i}\boldsymbol{x}_f^{\mathrm{u}_l} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} z^{\mathrm{c}_i} \\ z^{\mathrm{c}_j} \\ z^{\mathrm{c}_l} \end{bmatrix}}_{\boldsymbol{x}} = -\underbrace{\begin{bmatrix} \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} \\ \left[\boldsymbol{x}_f^{\mathrm{u}_l}\right]_\times \boldsymbol{p}_{\mathrm{c}_j}^{\mathrm{c}_l} \\ \left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \boldsymbol{p}_{\mathrm{c}_l}^{\mathrm{c}_i} \end{bmatrix}}_{\boldsymbol{b}}, \tag{32}$$

where, $\boldsymbol{x}$ denotes the vector of the feature's three depths, $\boldsymbol{A}$ is its coefficient matrix, and $\boldsymbol{b}$ represents the constant vector. The equation set is expressed as $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. When solving for the feature's depths using the least squares method, we transform the equation into $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$. The current coefficient matrix of $\boldsymbol{x}$ is referred to as the information matrix of the variables, denoted as $\boldsymbol{\Omega} = \boldsymbol{A}^T\boldsymbol{A}$, which contains the accuracy of the variables. To quantitatively compare the accuracy when selecting different base-frames, we take the determinant value of the information matrix $d_\Omega = \det(\boldsymbol{\Omega})$ as the indicator.

Denoting $\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i} = \boldsymbol{\theta}_{i,j}^{\mathrm{u}_j}$ as the parallax and $\theta_{i,j} = \|\boldsymbol{\theta}_{i,j}^{\mathrm{u}_j}\|$ as the parallax magnitude in Eq.(20), we obtain the expression of the information matrix as:

$$\boldsymbol{\Omega} = \begin{bmatrix} \|\boldsymbol{\theta}_{i,j}^{\mathrm{u}_j}\|^2 & 0 & 0 \\ 0 & \|\boldsymbol{\theta}_{j,l}^{\mathrm{u}_l}\|^2 & 0 \\ 0 & 0 & \|\boldsymbol{\theta}_{i,l}^{\mathrm{u}_i}\|^2 \end{bmatrix} = \begin{bmatrix} (\theta_{i,j})^2 & 0 & 0 \\ 0 & (\theta_{j,l})^2 & 0 \\ 0 & 0 & (\theta_{i,l})^2 \end{bmatrix}. \tag{33}$$

The determinant value of the information matrix is:

$$d_\Omega = (\theta_{i,j}\cdot\theta_{j,l}\cdot\theta_{i,l})^2 \propto \theta_{i,j}\cdot\theta_{j,l}\cdot\theta_{i,l}. \tag{34}$$

That is, the indicator can be simplified as the product of parallax magnitudes of every two camera frames. Then we identify the second base-frame that results in the largest d_Ω throughout the historical camera frames between the 2nd and $(n-1)$-th frames.

C. Selection Algorithm Analysis

To intuitively know the selection result of the proposed algorithm, we conduct an analysis based on common linear motion in this subsection. Fig.6 illustrates the selected base-frames and current image frame, where β is the angle between the $x_f^{\mathrm{u}_i}$ and the trajectory, α_1, α_2, and α are the angles between the three visual measurements. The parallax magnitude between the i-th and j-th camera frames is as follows:

$$\theta_{i,j} = \|\boldsymbol{\theta}_{i,j}^{\mathrm{u}_j}\| = \|\boldsymbol{x}_f^{\mathrm{u}_j}\|\|\boldsymbol{x}_f^{\mathrm{u}_i}\|\sin(\alpha_1), \tag{35}$$

where $\|\boldsymbol{x}_f^{\mathrm{u}_i}\| = \frac{1}{\cos\beta}$, $\|\boldsymbol{x}_f^{\mathrm{u}_j}\| = \frac{1}{\cos(\beta+\alpha_1)}$, as $\boldsymbol{x}_f^{\mathrm{u}_i}$ and $\boldsymbol{x}_f^{\mathrm{u}_j}$ have a normalized z-coordinate of 1.

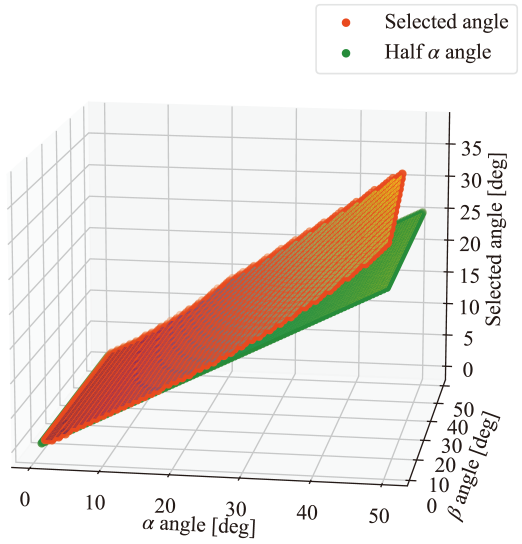


Fig. 7. The selected angles in the simulation test. The red plane represents the selected angle, the green plane denotes half of α.

We can get the other two parallax magnitudes in the same way. Multiplying the magnitudes together and excluding the constant α and β, we obtain the simplified base-frame selection indicator. The base-frame selection is to select the α_1 that satisfies the following requirement:

$$\arg\max_{\alpha_1}(d_\Omega) \propto \arg\max_{\alpha_1}\left(\frac{\sin(\alpha_1)\sin(\alpha-\alpha_1)}{\cos^2(\beta+\alpha_1)}\right). \quad (36)$$

We conducted a simulation to analyze the selected angle α_1. Based on the field of view of a typical camera, we set α, β, and $\alpha+\beta$ within the range of $[0, 50]$ deg, and set α_1 within the range of $[0, \alpha]$ deg. The selected angle α_1 of the simulation is presented in Fig.7, where the chosen angle demonstrates a clear correlation with half of α, especially when α is below 30 deg. Since feature depths in outdoor scenes are generally much larger than the baseline between the i-th and the j-th cameras, the angle α between visual measurements is typically smaller than 30 deg. Consequently, we treat the selected angle as $\alpha_1 = \alpha_2 = \frac{1}{2}\alpha$ for the following analysis.

Denoting the two baselines between the c_i, c_j, and c_l frames as d_1 and d_2, we establish their correlation with the feature distances in the i-th and j-th frames as follows:

$$\frac{d_1}{d_2} = \frac{d^{\mathrm{c}_i}\sin\alpha_1}{d^{\mathrm{c}_l}\sin\alpha_2} = \frac{d^{\mathrm{c}_i}}{d^{\mathrm{c}_l}}. \quad (37)$$

Since most features are far from the camera, the two distances d^{c_i} and d^{c_l} are generally large but similar, resulting in d_1 being quite close to d_2. If we further assume the carrier moves with uniform velocity, the selected second base-frame most likely lands near the middle of the historical frames. Any irregular movements, such as acceleration or deceleration, would cause the selected frame to shift forward or backward from the middle frame. Nonetheless, nearly uniform linear motion is the most common movement for a wheeled carrier or a pedestrian, resulting in the overall selected base-frame being around the middle of the historical frames.

VII. Experiments and Results

A. Experiments Description

The proposed PO-KF is evaluated on both publicly available and privately obtained datasets, covering three distinct carriers. Specifically, the public datasets comprise the KAIST urban dataset [52] and the TUM-VI dataset [53] collected from commercial cars and pedestrians. The private dataset is obtained from a low-speed wheeled robot.

1) Public Dataset: The public KAIST urban dataset is a vehicle's multi-sensor dataset in a complex urban environment, where the carrier moves fast with a maximum speed of 15 m/s. Our experiment only utilizes the industrial-grade MEMS IMU measurements and the left camera images. The camera captures images at a resolution of 1280*560 and a frame rate of 10Hz, while the IMU records data at a rate of 100Hz. Due to the less-smooth ground truth in this dataset, it is primarily suitable for evaluating the absolute pose errors. Our experiments encompass five sequences in this dataset, namely *urban28*, *urban30*, *urban32*, *urban38*, and *urban39*. These sequences collectively span a total time of 8321 seconds and cover a cumulative distance of 45525 meters.

The public TUM-VI dataset is a widely used benchmark with a diverse set of sequences for evaluating visual-inertial odometry. In our evaluation, we use the left camera images with 512*512 resolution at 20Hz and IMU measurements at 200Hz. We quantitatively evaluate localization accuracy only on the *room* series sequences, which are notable for their continuous ground truth within this dataset. Additionally, we include the *magistrale* series sequences for qualitative comparison. The selected sequences span a total duration of 4261 seconds and cover a total distance of 4625 meters.

2) Private Robot Dataset: The multi-sensor wheeled robot platform, capable of reaching a maximum speed of 1.5m/s, is illustrated in Fig.8 and serves as the test carrier for our private dataset. Our experiments utilize the industrial-grade MEMS IMU with a data rate of 200Hz, and the left gray camera with a resolution of 800*600 and a frame rate of 10Hz. The ground truth system includes a navigation-grade IMU and a GNSS RTK receiver, providing smoothed trajectories for the accurate evaluation of the absolute and relative pose errors of the tested system. The private dataset is systematically collected within a typical outdoor campus scene, as depicted in Fig.9. Comprising eight distinct sequences of robot data denoted as *Robot-A*∼*Robot-H*, the dataset spans a cumulative time of 11550 seconds and a total distance of 14821 meters.

3) Evaluation Method: We implemented our PO-KF with the design details in Sec. IV. For a comparative analysis, we also developed an MSCKF-based VIO system on the SOTA open-sourced platform OpenVINS [6] as our baseline method. Importantly, the baseline system employs the same visual front-end and state equations as PO-KF, ensuring a fair and consistent comparison. For clarity, we will refer to this baseline system, i.e., the modified OpenVINS, simply as 'OpenVINS' in the following experiments. Furthermore, to provide a comprehensive evaluation, we benchmark our PO-KF against SOTA open-source optimization-based VINS platforms, VINS-Mono [37] and IC-VINS (the GNSS-free

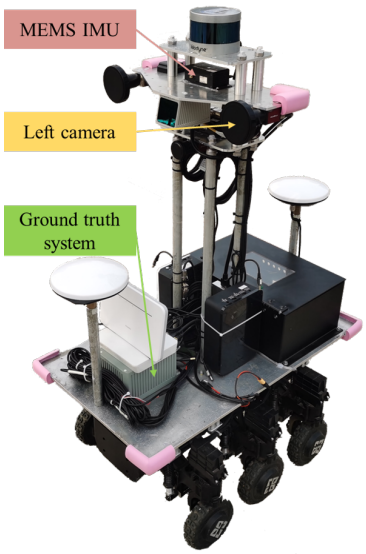


Fig. 8. The wheeled robot for the private dataset.

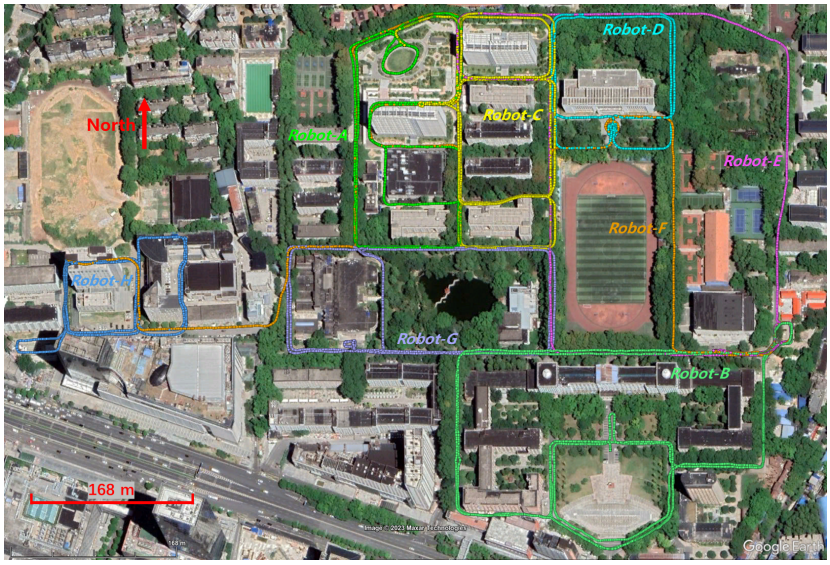


Fig. 9. The test trajectories of the private dataset.

version included in the IC-GVINS [11]). Throughout the VIO system evaluations, we have carefully tuned and applied appropriate hyperparameters. Notably, the OpenVINS and PO-KF systems share the same hyperparameters across the experiments. Specifically, the visual measurement noise for all tested methods is set to 1 pixel. To mitigate the impact of outliers, a chi-square test with 95% confidence is employed in both OpenVINS and PO-KF, while robust kernels are utilized in VINS-Mono and IC-VINS.

For quantitative evaluation of localization accuracy, we utilize the EVO [54] tools, which include metrics such as absolute translation error (ATE) and absolute rotation error (ARE) over the entire test sequence, as well as relative translation error (RTE) and relative rotation error (RRE) across various trajectory lengths. In the following subsections, we systematically evaluate PO-KF in terms of localization accuracy across different datasets, the effectiveness of the base-frame selection algorithm, localization robustness under two challenging conditions, and real-time performance.

B. Experiments on Localization Accuracy

1) Public KAIST Urban Dataset: As VINS-Mono reboots when the carrier is stationary in this dataset, we compare PO-KF only with OpenVINS and IC-VINS across this dataset. For a clearer comparison of localization accuracy, we present the result trajectories of *urban38* data in Fig.10, aligning the start points of both result trajectories with the ground truth trajectory. The overall trajectory in Fig.10 evidently demonstrate that PO-KF outperforms both OpenVINS and IC-VINS in terms of trajectory fitness.

The trajectory illustrated in the subfigures of Fig.10 evidently demonstrates that PO-KF outperforms both OpenVINS and IC-VINS regarding trajectory smoothness. As shown in the subfigures, the occasional jumps observed in the OpenVINS trajectory, mainly result from the delayed updates inherent to the MSCKF-based method. Although IC-VINS

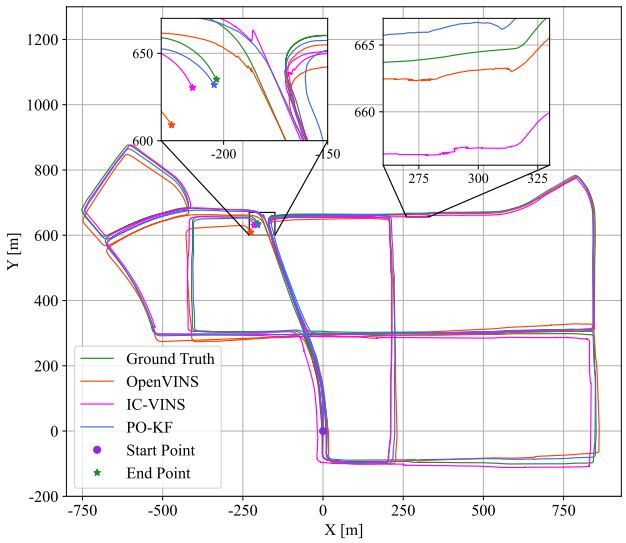


Fig. 10. The result trajectories of the KAIST *urban38* data. All trajectories start from the same start initial point and end at distinct stars, each represented by different colors.

iteratively refines feature positions, its measurement updates are also delayed until sufficient observations are available for feature triangulation. In contrast, the immediate measurement update in PO-KF maintains error-states at a lower level and ensures a smoother trajectory. Moreover, in terms of trajectory alignment, the trajectory produced by PO-KF in Fig.10 exhibits the closest endpoint and best overall alignment with the truth trajectory compared to OpenVINS and IC-VINS, which highlights the superior localization accuracy of PO-KF.

We summarize the absolute pose errors for OpenVINS, IC-VINS, and PO-KF on the KAIST dataset in Table II. Across the five groups of data in Table II, PO-KF consistently produces significantly smaller pose errors compared to the

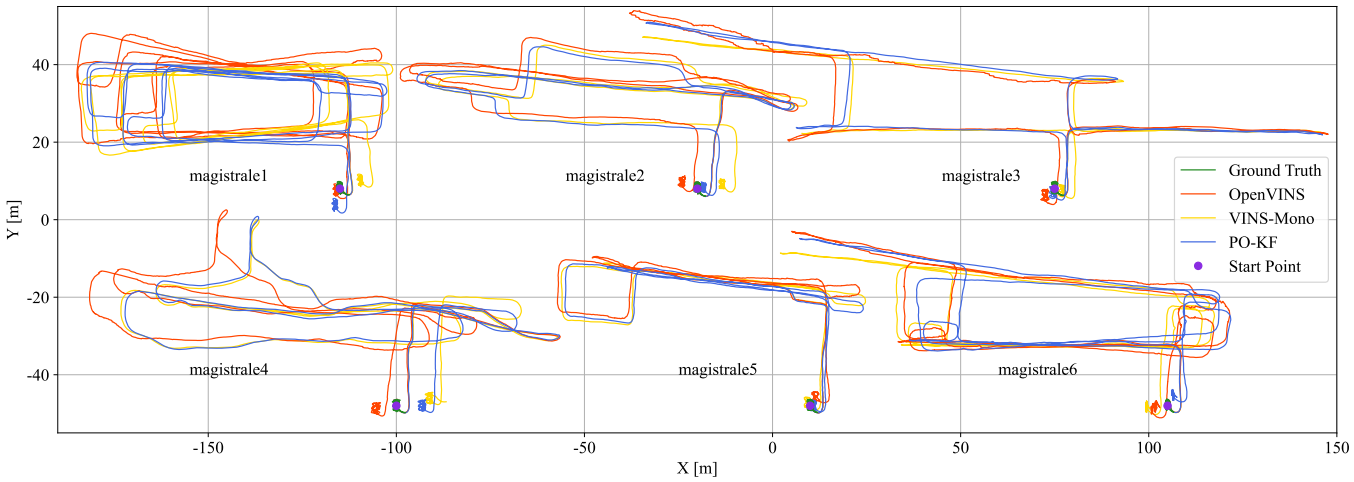


Fig. 11. The result trajectories of the TUM-VI *magistrale* series sequences.

MSCKF-based OpenVINS. Additionally, PO-KF even demonstrates superior localization accuracy across a greater number of datasets compared to the optimization-based IC-VINS. The root-mean-square (RMS) values of AREs and ATEs for PO-KF are both smaller than those of OpenVINS and IC-VINS, further showcasing its superiority in localization performance. Statistically, the filter-based PO-KF achieves a 17% reduction in the ARE and a 39% reduction in ATE compared to the MSCKF-based OpenVINS.

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

Sequence	OpenVINS	IC-VINS	PO-KF
urban28	2.88 / 29.95	2.16 / **12.70**	**2.01** / 14.40
urban30	2.40 / 16.67	2.44 / **12.48**	**2.19** / 13.16
urban32	1.79 / 19.19	**1.56** / 12.86	1.61 / **9.59**
urban38	1.86 / 11.32	**0.89** / 10.37	1.44 / **8.42**
urban39	1.70 / 11.86	1.87 / 13.83	**1.59 / 11.63**
RMS	*2.17 / 19.04*	*1.86 / 12.50*	***1.79 / 11.65***

2) Public TUM-VI Dataset: We further evaluate the localization performance of PO-KF on the TUM-VI dataset. IC-VINS is excluded from testing this dataset as it lacks support for fisheye camera images. We compute the ATEs for the *room* series sequences using their ground truth and summarize the position errors in Table III. Except for *room2* and *room4* in this series, PO-KF consistently outperforms both OpenVINS and VINS-Mono in absolute translation accuracy. Statistically, PO-KF demonstrates the best localization accuracy among the three methods, reflecting its SOTA localization capabilities. Statically, compared to the MSCKF-based OpenVINS, PO-KF reduces the ATE by 36% for this series data.

The ground truth for the *magistrale* series data is available only at the beginning and end of the sequences. To evaluate the proposed PO-KF, we align the beginning part of the result trajectories with the ground truth and plot the trajectories in Fig.11. In the result trajectories for *magistrale1* and *magistrale4*, the endpoints of OpenVINS are closer to the ground truth compared to those of VINS-Mono and PO-KF. However,

TABLE III
ATES (M) ON THE ROOM SEQUENCES OF THE TUM-VI DATASET

Sequence	OpenVINS	VINS-Mono	PO-KF
room1	**0.05**	0.07	**0.05**
room2	**0.06**	0.07	0.08
room3	0.09	0.11	**0.08**
room4	0.18	**0.04**	0.05
room5	0.10	0.20	**0.07**
room6	0.12	0.08	**0.06**
RMS	*0.11*	*0.11*	***0.07***

the OpenVINS trajectories in both sequences exhibit inconsistent headings and positions when passing through the same corridor multiple times. In contrast, VINS-Mono and PO-KF maintain more consistent headings and positions. For *magistrale2* and *magistrale5*, VINS-Mono and PO-KF demonstrate consistent headings, positions and endpoints, whereas OpenVINS experiences significant deviations. Although the three methods exhibit trajectory inconsistencies for *magistrale3* and *magistrale6*, PO-KF achieves the closest endpoints with the ground truth. Furthermore, the trajectories of VINS-Mono and PO-KF are smoother than those of OpenVINS, particularly *magistrale3* and *magistrale6* sequences. Generally, the results indicate that PO-KF outperforms OpenVINS in localization performance and achieves comparable performance with the optimization-based method, VINS-Mono.

3) Private Robot dataset: The localization accuracy of PO-KF is also evaluated using our private robot dataset. We initially display the test trajectories of *Robot-A* and *Robot-B* data in Fig.12 and Fig.13. The left subfigures in Fig.12 illustrate the local positioning results, obviously showcasing the superior trajectory smoothness of PO-KF and VINS-Mono than IC-VINS and OpenVINS. Due to the accumulated positioning errors over time, all four result trajectories gradually diverge from the ground truth. Nonetheless, the endpoints of PO-KF in both Fig.12 and Fig.13 exhibit smaller deviations from the ground truth compared to those of OpenVINS, VINS-Mono, and IC-VINS. Furthermore, as shown in the sky-blue rectangle of Fig.13, the maximum trajectory deviation generated by PO-

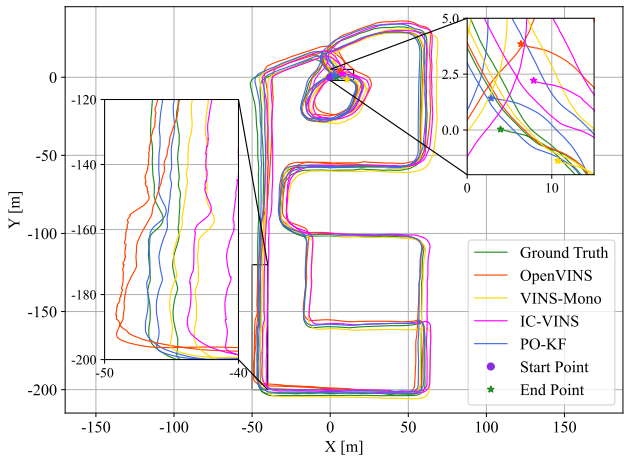


Fig. 12. The result trajectories of the *Robot-A* data.

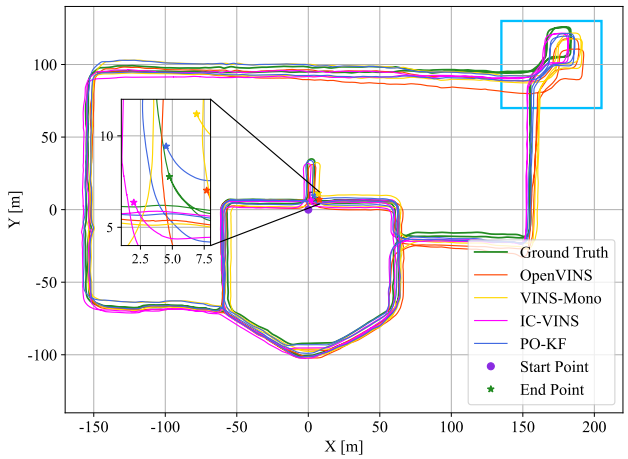


Fig. 13. The result trajectories of the *Robot-B* data. The sky-blue rectangle highlights the smaller trajectory drift of PO-KF.

KF are only slightly larger than those of the optimization-based IC-VINS, while indicating obvious improvements over OpenVINS and VINS-Mono.

We quantify the ATEs and AREs for OpenVINS, VINS-Mono, IC-VINS, and PO-KF across all eight data groups in this dataset, as presented in Table IV. Throughout this dataset, our filter-based PO-KF achieves rotation accuracy comparable to both the optimization-based VINS-Mono and IC-VINS, as well as translation accuracy translation accuracy comparable to IC-VINS. Notably, PO-KF even demonstrates significantly higher translation accuracy compared to VINS-Mono. Furthermore, except for the likely outliers in *Robot-C*, PO-KF consistently shows substantially lower absolute pose errors than OpenVINS. Statistically, PO-KF achieves similar localization accuracy with IC-VINS and outperforms both OpenVINS and VINS-Mono. Specifically, compared to OpenVINS, PO-KF achieves a reduction in ARE of 36% and ATE of 34% statistically.

Leveraging the continuous and reliable truth trajectories

TABLE IV
ABSOLUTE POSE ERRORS (DEG / %) ON THE ROBOT DATASET

Sequence	OpenVINS	VINS-Mono	IC-VINS	PO-KF
Robot-A	1.69 / 3.15	**0.64** / 3.06	1.02 / **1.62**	0.96 / 1.71
Robot-B	1.45 / 3.46	0.89 / 3.04	0.72 / **2.04**	**0.67** / 2.23
Robot-C	0.76 / 2.04	**0.55** / 2.48	**0.55 / 1.11**	1.05 / 2.32
Robot-D	1.31 / 1.99	**0.76** / 1.91	**0.76** / 1.15	0.77 / **1.06**
Robot-E	1.00 / 2.39	0.80 / 2.75	**0.43 / 1.02**	0.46 / 1.51
Robot-F	0.87 / 2.02	**0.56** / 1.96	0.76 / 1.80	**0.56 / 1.11**
Robot-G	0.90 / 2.32	1.05 / 4.11	0.65 / 1.25	**0.50 / 1.20**
Robot-H	0.86 / 1.13	1.06 / 1.86	**0.73** / 1.12	**0.73 / 1.09**
RMS	*1.15 / 2.41*	*0.81 / 2.74*	***0.72 / 1.43***	*0.74 / 1.60*

TABLE V
RMS OF RELATIVE POSE ERRORS (DEG / %) OF DIFFERENT TRAJECTORY LENGTHS ON THE ROBOT DATASET

Length	OpenVINS	VINS-Mono	IC-VINS	PO-KF
10m	0.15 / 2.63	**0.09** / 2.11	0.15 / 2.07	**0.09 / 1.63**
50m	0.39 / 1.62	**0.19** / 1.59	0.26 / 1.15	0.20 / **0.98**
100m	0.61 / 1.29	0.31 / 1.27	0.36 / 0.91	**0.29 / 0.81**
200m	0.93 / 1.01	0.49 / 0.97	0.54 / 0.71	**0.44 / 0.65**

in our robot dataset, we conduct a thorough analysis by calculating the RREs and RTEs for different trajectory lengths on this dataset. The relative pose evaluation helps eliminate variations in positioning errors across different trajectories and mitigates the impact of brief outliers on the statistical localization accuracy. Namely, the relative pose errors offer more convincing insights into evaluating the performance of a recursive localization system. Considering the scale of the robot test scene, we specifically examine trajectory lengths of 10m, 50m, 100m, and 200m to evaluate the relative pose accuracy. We plot the boxplots of the RREs results in Fig.14 and the RTEs results in Fig.15.

The boxplots indicate that the proposed filter-based PO-KF achieves the best relative rotation and translation accuracy across various trajectory lengths within this dataset. Specifically, PO-KF obviously surpasses the optimization-based IC-VINS in relative rotation accuracy and significantly outperforms the optimization-based VINS-Mono in relative translation accuracy. Additionally, compared to the MSCKF-based OpenVINS, PO-KF consistently achieves significantly lower RREs and RTEs across all trajectory lengths in this dataset. We statistics the RMS values of these relative pose errors for each trajectory length, as summarized in Table V, further confirming the superiority of PO-KF over OpenVINS,VINS-Mono, and IC-VINS. Notably, the RMS values of relative pose errors for PO-KF across the four trajectory lengths exhibit nearly a half error reduction compared to OpenVINS. Specifically, PO-KF achieves a 52% reduction in RRE and a 38% reduction in RTE compared to the MSCKF-based OpenVINS.

In summary, experiments conducted across the above three diverse datasets reveal that PO-KF achieves superior localization accuracy compared to both MSCKF-based OpenVINS and optimization-based VINS-Mono. Particularly, PO-KF achieves relative pose errors reduced to nearly half of those observed in OpenVINS.

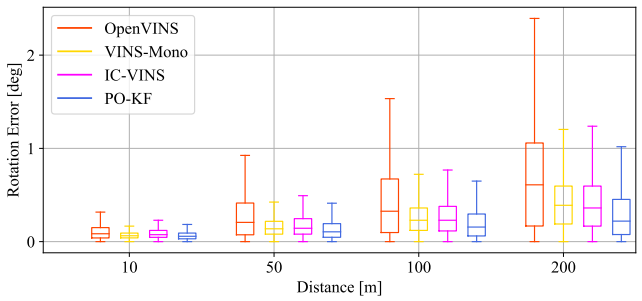


Fig. 14. The RREs of Different Trajectory lengths on the robot dataset.

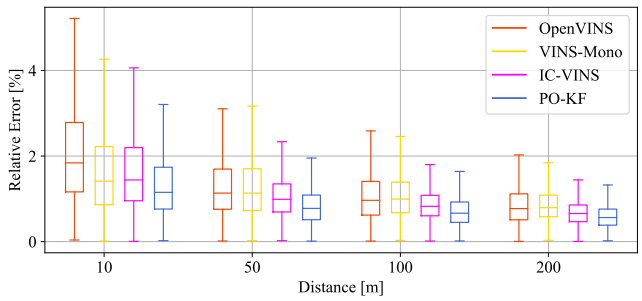


Fig. 15. The RTEs of Different Trajectory lengths on the robot dataset.

C. Experiments on Base-Frame Selection

The base-frame selection, a key aspect of our pose-only measurement model, plays a crucial role in achieving the excellent pose accuracy of PO-KF. In this subsection, we validate the contribution of our base-frame selection algorithm through experiments conducted on our robot dataset.

1) Selection Result: Initially, we validate the selected results of the proposed base-frame selection algorithm. As detailed in Sec.VI-A, the first base-frame is anchored to the oldest frame among the historical frames. Our focus here is to verify the consistency of the selected second base-frame with the analysis results discussed in Sec.VI-C. Taking the robot data *Robot-F* as an example, we record the selected second base-frame indices and the historical frame size (i.e., feature tracking length) of each feature point. Subsequently, we cluster feature points by their tracking lengths and normalize the selected base-frame indices into the $(0, 1)$ range, relative to their respective tracking lengths. Given the high randomness of selected indices for short-tracking features, our analysis specifically focuses on features with tracking lengths exceeding 10 frames. We calculate and illustrate the proportions of selected indices for features with tracking lengths between 10 and 20 frames in Fig.16, where a normalized index of 0.5 represents the middle frame within a feature's historical frames.

The findings from Fig.16 clearly indicate that the middle frames are the most frequently selected by our algorithm across various tracking lengths. Considering that the robot's motion is predominantly characterized by uniform linear movement, the theoretical selection outcome, as derived in Sec.VI-C, is expected to be consistently concentrated around the middle frames. Deviations from the middle frames are observed when other types of motion are introduced. Additionally, since uniform linear motion is more prevalent during shorter tracking intervals (e.g., 10 frames), the selection result of these shorter tracking lengths in Fig.16 are more centralized around the middle frame. In general, the selection results shown in Fig.16 are well-aligned with the analysis presented in Sec.VI-C, providing robust validation for our base-frame selection algorithm.

2) Contribution to Accuracy: Next, we assess the impact of the proposed base-frame selection algorithm on localization accuracy. For comparison purposes, in addition to the baseline method OpenVINS, we introduce two alternative base-frame

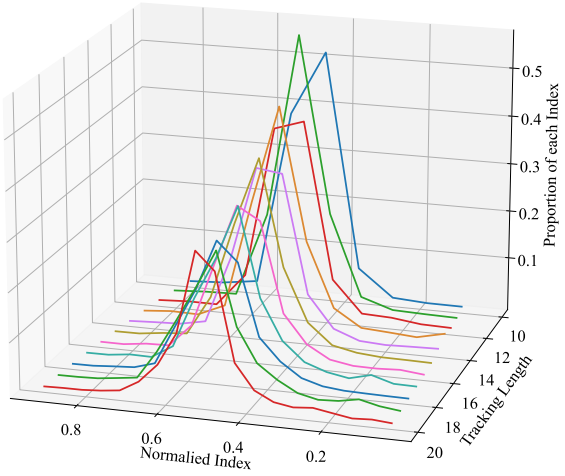


Fig. 16. Indices of the selected second base-frames on *Robot-F* data. Colors indicate different feature tracking lengths.

selection strategies for PO-KF:

a) POKF-D (PO-KF with deliberate disturbance): This method deliberately selects a disturbed frame based on the result from our proposed algorithm. Let the base-frame selected by our algorithm be denoted as j, with alternative frames ranging from 2 to $n-1$. If j is closer to the $n-1$-th frame, POKF-D selects the middle frame between the j-th and 2nd frames as the second base-frame. Otherwise, it selects the middle frame between the j-th and $(n-1)$-th frames.

b) POKF-M (PO-KF with maximum parallax): This method adopts a base-frame selection strategy employed in optimization-based systems. Specifically, POKF-M selects the second newest frame, i.e., the $(n-1)$-th frame, as the second base-frame, since this frame typically exhibits the largest parallax relative to the first base-frame.

Subsequently, we assess the localization accuracy of POKF-D and POKF-M using our robot dataset. The localization errors of OpenVINS, PO-KF, POKF-D, and POKF-M are compared together for detailed analysis. First, we illustrate the relative pose errors for different trajectory lengths across the eight data groups in Fig.17 and Fig.18. These figures intuitively demonstrate that PO-KF achieves smaller pose errors compared to OpenVINS, POKF-D, and POKF-M across all

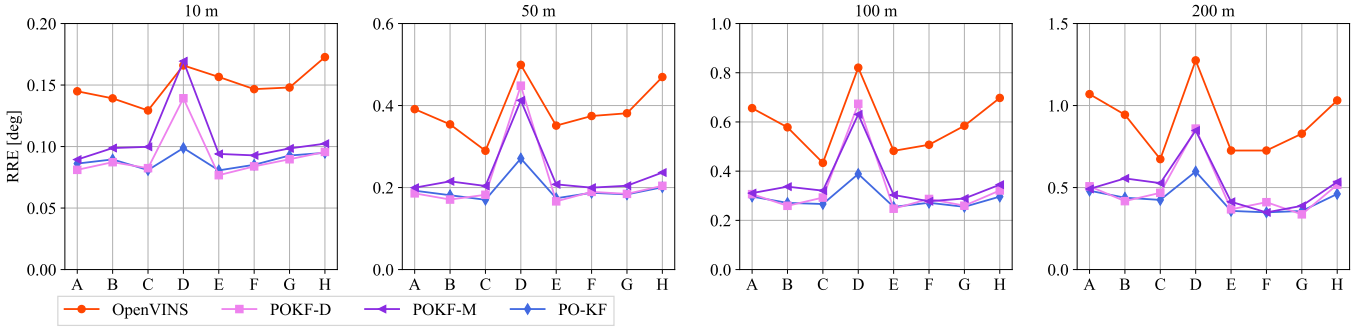


Fig. 17. Relative rotation errors of different base-frame selection methods on the robot dataset. The X-axis represents the different data sequences.

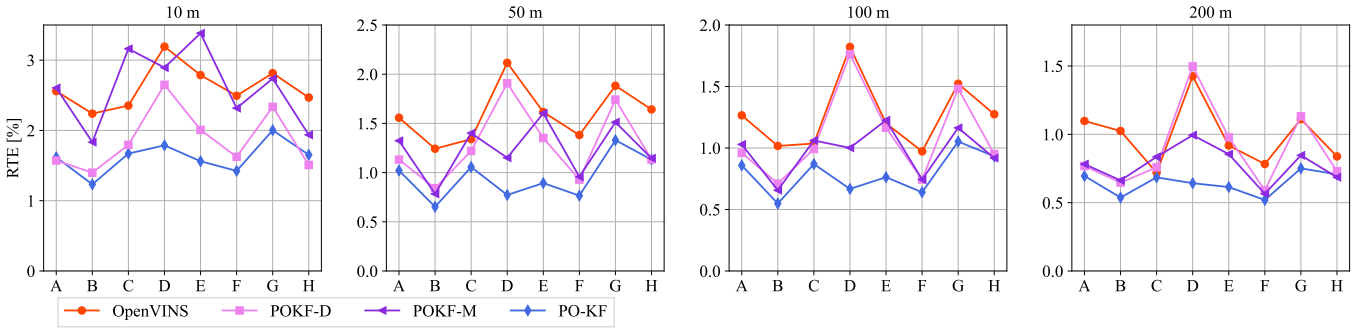


Fig. 18. Relative translation errors of different base-frame selection methods on the robot dataset. The X-axis represents the different data sequences.

TABLE VI
THE RMS OF RELATIVE POSE ERRORS (DEG / %) OF DIFFERENT TRAJECTORY LENGTHS ON THE ROBOT DATASET

Length	OpenVINS	POKF-D	POKF-M	PO-KF
10m	0.15 / 2.63	**0.09** / 1.91	0.11 / 2.66	**0.09 / 1.63**
50m	0.39 / 1.62	0.23 / 1.33	0.24 / 1.26	**0.20 / 0.98**
100m	0.61 / 1.29	0.36 / 1.15	0.37 / 0.99	**0.29 / 0.81**
200m	0.93 / 1.01	0.51 / 0.93	0.53 / 0.79	**0.44 / 0.65**

TABLE VII
ABSOLUTE POSE ERRORS (DEG / M) OF DIFFERENT BASE-FRAME SELECTION METHODS ON THE ROBOT DATASET

Sequence	OpenVINS	POKF-D	POKF-M	PO-KF
Robot-A	1.69 / 3.15	0.98 / **1.62**	0.98 / 1.86	**0.96** / 1.71
Robot-B	1.45 / 3.46	**0.65 / 1.73**	0.83 / 2.51	0.67 / 2.23
Robot-C	**0.76 / 2.04**	0.96 / 2.72	1.35 / 2.45	1.04 / 2.32
Robot-D	1.31 / 1.99	1.25 / 1.71	1.07 / 1.34	**0.77 / 1.06**
Robot-E	1.00 / 2.39	0.60 / 1.98	0.51 / 1.79	**0.46 / 1.51**
Robot-F	0.86 / 2.02	0.58 / 1.27	**0.55** / 1.17	0.56 / **1.11**
Robot-G	0.90 / 2.31	**0.46** / 1.87	0.54 / 1.59	0.50 / **1.20**
Robot-H	0.86 / 1.13	**0.71** / 1.15	0.76 / 1.18	0.73 / **1.09**
RMS	*1.15 / 2.41*	*0.81 / 1.81*	*0.87 / 1.80*	***0.74 / 1.60***

data groups. Specifically, compared to PO-KF, the enlargement of the RTEs in POKF-D and POKF-M is more significant than the RREs. This reflects that the camera essentially measures the relative angles between feature points and itself, making rotation accuracy less sensitive to the choice of base-frames. Conversely, the accuracy of the feature's position highly relies on the spatial distribution of the cameras, including the base-frames and the current camera frame. Consequently, inappropriate base-frames in POKF-D and POKF-M lead to significantly larger position errors. Particularly, POKF-M, especially for the 10m RTE, generally exhibits larger errors compared to both PO-KF and POKF-D. This indicates that the base-frame utilized in optimization-based systems is not optimal for a filter-based system like PO-KF.

For a quantitative comparison, we calculate and present the RMS values of RREs and RTEs across all data groups in the robot dataset in Table VI. POKF-D and POKF-M consistently exhibit larger relative pose errors compared to PO-KF across all trajectory lengths, underscoring the significant contribution of our base-frame selection algorithm to the superior localization of PO-KF. Additionally, Table VII provides the absolute pose errors for the four methods. Generally, both POKF-D and POKF-M increase the absolute pose error compared to PO-KF. Statistically, POKF-D and POKF-M raise the ARE of PO-KF from 0.74deg to 0.81deg and 0.87deg, respectively, and the ATE from 1.60m to 1.81m and 1.80m, respectively. Despite these error increases, POKF-D and POKF-M still outperform OpenVINS in terms of both relative and absolute pose errors in Table VI and Table VII, confirming the excellent localization accuracy of PO-KF.

In summary of the above evaluation results, PO-KF exhibits the best localization accuracy among the four methods, emphasizing the substantial contribution of the proposed base-frame selection algorithm on its superior localization performance.

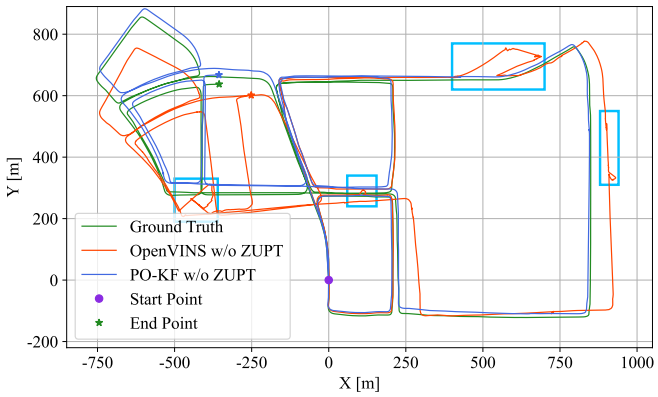


Fig. 19. The result trejectories without ZUPT of the KAIST *urban39* data. The sky-blue rectangles highlight the trajectory drifts of OpenVINS during the zero-velocity states.

D. Experiments on Robustness

As discussed in Sec.IV-5, the sliding-window and feature management strategy designed for stationary periods enables PO-KF to handle potential drift during zero-velocity states without relying on ZUPT. Furthermore, PO-KF addresses the necessity of 3D feature position in the measurement equation, making it easy to correct the state vector for short-tracking features within a small sliding-window size system. In this section, we conduct two targeted experiments to showcase the enhanced localization robustness of PO-KF from the aforementioned characteristics.

1) Robustness during Zero-Velocity States: Considering the frequent prolonged stops in the KAIST urban dataset, we evaluate the localization robustness of PO-KF during zero-velocity states within this dataset. Specifically, in processing the KAIST dataset trajectories, we deliberately avoided using ZUPT in both OpenVINS and PO-KF. For illustration, we show the result trajectories of the *urban39* data in Fig.19. During zero velocity intervals, the limited parallax of the tracked features results in invalid measurements to correct the state vector in the MSCKF-based OpenVINS. This deficiency leads to noticeable trajectory drifts, highlighted by sky-blue rectangles in Fig.19. In contrast, the strategy designed for handling zero-velocity states in PO-KF ensures that the historical tracking data with sufficient parallax is retained. This allows for the formulation of measurement equations for incoming visual measurements, enabling PO-KF to maintain a smooth trajectory even without ZUPT.

For clear comparison and analysis, we calculate and present the absolute pose errors of trajectories solved by OpenVINS and PO-KF, both with and without ZUPT, in Table VIII. Upon deactivating ZUPT, OpenVINS exhibits a significant increase in pose errors, with the *urban28* data even failing to run. In contrast, PO-KF maintains a stable localization accuracy without ZUPT. Moreover, PO-KF without ZUPT even outperforms PO-KF with ZUPT in the position accuracy for several data groups. This phenomenon is attributed to the false zero-velocity state detection and the inaccurate measurement noise for ZUPT. Accurately determining the measurement noise for ZUPT is challenging, as it heavily depends on the IMU's precision and the carrier's stability. Overall, PO-KF not only sustains localization accuracy during zero-velocity intervals but also avoids the negative impacts of false zero-velocity detections and inaccurate ZUPT measurement noise, affirming the localization robustness of PO-KF under this condition.

TABLE VIII
ABSOLUTE POSE ERRORS (DEG/M) WITHOUT ZUPT ON THE KAIST URBAN DATASET

Sequence	with ZUPT		without ZUPT	
	OpenVINS	PO-KF	OpenVINS	PO-KF
urban28	2.88 / 29.95	2.01 / 14.40	failed	**1.67 / 9.60**
urban30	2.40 / 16.67	**2.19** / 13.16	2.98 / 26.04	2.27 / **12.64**
urban32	1.79 / 19.19	**1.61** / 9.59	10.2 / 178.7	1.62 / **8.06**
urban38	1.86 / 11.32	**1.44 / 8.42**	4.89 / 82.16	1.58 / 8.47
urban39	1.70 / 11.86	**1.59 / 11.63**	3.47 / 45.62	1.72 / 13.47
RMS	*2.17 / 19.04*	***1.79*** */ 11.65*	-	***1.79 / 10.68***

2) Robustness with Small Sliding-Window Size: The computational cost in filter-based VIO systems scales cubically with the size of the sliding-window. However, smaller sliding-window sizes generally result in shorter baselines, harming the stability and precision of VIO. In this part, we evaluate the localization robustness of PO-KF with various small sliding-window sizes on our robot dataset. Considering the sliding-window size employed in previous sections is 20 frames, we systematically assess the performance of PO-KF and OpenVINS with window sizes of 15, 10, 5, and 3 frames. The absolute pose errors for different sliding-window sizes are calculated and summarized in Table IX.

Since every image frame is considered a keyframe within the sliding-windows of both PO-KF and OpenVINS, the visual baseline shortens as the sliding-window size decreases, leading to increased pose error for both filters in Table IX. For OpenVINS, the noticeable enlargement of the pose error is caused by the reduction in successful feature triangulation as the sliding-window size decreases. Numerically, when the window size is reduced to 15 frames, the position error in OpenVINS increases by 26% compared to the 20-frame window. As the size further decreases to 10 frames, the maximum baseline distance within the sliding-window is only 1.5m, which is notably shorter than the feature depths in the outdoor scene. This limitation results in half of the data groups failing to run in OpenVINS with the 10-frame sliding-window. More critically, reducing the window size to 5 or 3 frames renders almost all data groups infeasible for OpenVINS due to extremely limited visual constraints.

In contrast, PO-KF shows a significantly milder increase in errors compared to OpenVINS as the sliding-window size decreases. With a 15-frame sliding-window, the position error of PO-KF increases by only 8% compared to the 20-frame window. For a 10-frame sliding-window, the position error increases by 17%. When the sliding-window size is reduced to 5 frames, PO-KF only fails to run on the *Robot-B* data. This benefits from the fact that the feature 3D positions are not required in PO-KF, which ensures its successful operation even with a 5-frame sliding-window. When facing extreme challenges with a 3-frame sliding-window, PO-KF also exhibits

TABLE IX
ABSOLUTE POSE ERRORS (DEG/M) WITH DIFFERENT SLIDING-WINDOW SIZES ON THE ROBOT DATASET

Sequence	OpenVINS					PO-KF				
	20 frames	15 frames	10 frames	5 frames	3 frames	20 frames	15 frames	10 frames	5 frames	3 frames
Robot-A	1.69 / **3.15**	**1.54** / 3.38	failed	failed	failed	**0.96 / 1.71**	1.00 / 1.78	1.05 / 1.77	1.27 / 1.98	1.18 / 2.57
Robot-B	**1.45 / 3.46**	1.89 / 4.26	failed	failed	failed	**0.67** / 2.23	0.70 / **2.22**	0.87 / 2.82	failed	failed
Robot-C	**0.76 / 2.04**	0.93 / 2.19	1.56 / 3.48	failed	failed	**1.05 / 2.32**	1.22 / 2.43	1.18 / 2.45	1.26 / 3.80	1.55 / 5.70
Robot-D	**1.31 / 1.99**	1.48 / 2.12	1.35 / 3.07	failed	1.75 / 8.35	**0.77 / 1.06**	0.93 / 1.43	1.15 / 1.98	1.49 / 2.80	failed
Robot-E	1.00 / **2.39**	**0.94** / 4.25	failed	failed	failed	**0.46 / 1.51**	0.55 / 1.92	0.53 / 1.42	0.79 / 2.49	failed
Robot-F	**0.87** / 2.02	1.00 / **1.87**	1.80 / 3.17	failed	failed	**0.56 / 1.11**	0.63 / 1.30	0.64 / 1.31	0.57 / 1.74	failed
Robot-G	**0.90 / 2.32**	1.11 / 3.38	failed	failed	failed	**0.50 / 1.20**	0.53 / **1.20**	0.55 / 1.49	0.71 / 2.02	1.00 / 2.40
Robot-H	**0.86 / 1.13**	0.90 / 1.27	1.50 / 2.07	4.02/ 4.40	7.08 / 7.02	**0.73** / 1.09	0.74 / 1.08	0.75 / **1.05**	0.79 / 1.20	0.89 / 2.72
RMS	***1.15 / 2.41***	*1.27 / 3.03*	-	-	-	***0.74 / 1.60***	*0.82 / 1.73*	*0.88 / 1.87*	-	-

worsened localization robustness, with half of the trajectories in this dataset drifting. Nonetheless, under a 3-frame sliding-window, the successful rate and localization accuracy of PO-KF are still better than OpenVINS. Generally, PO-KF demonstrates superior localization accuracy and robustness with a small-size sliding-window.

E. Experiments on Runtime

We also conduct an experiment to analyze the runtime performance of PO-KF and the MSCKF-based OpenVINS. For this analysis, the sliding-window sizes of both systems are set to 20 frames. Their runtimes are evaluated on a desktop PC (equipped with an AMD R7950X CPU and 32GB RAM), using our robot dataset. The average processing times for one image frame in different data groups are summarized in Table X. Since OpenVINS and PO-KF share the same front-end and state equations, their running times for feature tracking and state propagation are nearly identical. However, the measurement update in PO-KF takes slightly longer than in OpenVINS, resulting in a marginal increase of 0.33 ms in the overall runtime of PO-KF compared to OpenVINS. Nonetheless, the increase is considerably smaller than the total running time for one image frame.

TABLE X
AVERAGE RUNTIMES (MS) OF OPENVINS AND PO-KF ON THE ROBOT DATASET

Sequence	OpenVINS			PO-KF		
	Track&Prop.	Update	Total	Track&Prop.	Update	Total
Robot-A	5.61	1.64	7.26	5.63	1.99	7.61
Robot-B	5.39	1.63	7.03	5.39	1.90	7.29
Robot-C	5.59	1.69	7.27	5.59	1.97	7.56
Robot-D	5.48	1.73	7.21	5.50	2.08	7.58
Robot-E	6.15	1.77	7.91	6.00	2.05	8.06
Robot-F	6.20	1.62	7.82	6.11	2.01	8.12
Robot-G	7.69	1.86	9.55	7.63	2.21	9.84
Robot-H	8.13	1.52	9.65	8.17	2.09	10.25
RMS	*6.36*	*1.68*	*8.04*	*6.33*	*2.04*	*8.37*

Track&Prop. denotes feature tracking and state propagation

The detailed running times of main modules in the measurement update process are counted in Table X. This table also presents visual measurement sizes for runtime analysis. In Table X, the 'Model' category encompasses the running times for measurement model construction and outlier checking, including the nullspace projection time for OpenVINS. The 'KFU.' category aggregates the time required for measurement compression, Kalman filter update, and error-state feedback. 'Input Meas.' denotes the size of visual measurements input into the measurement update function, while 'Updated Meas.' indicates the size of measurements utilized in the final measurement update equations. Due to some failed triangulations, the size of updated measurements in OpenVINS reduces notably compared to PO-KF, coinciding with the analysis in Sec.V-C. As a result of more visual measurements for measurement equation construction, the running times for 'Model' and 'KFU.' in PO-KF become slightly larger than those in OpenVINS.

TABLE XI
DETAIL RUNNING TIMES AND VISUAL MEASUREMENT SIZES IN MEASUREMENT UPDATE ON THE ROBOT DATASET

Sequence	Method	Running Time [ms]			Input Meas.	Updated Meas.
		Tri.	Model	KFU.		
Robot-A	OpenVINS	0.07	0.27	1.30	110	65
	PO-KF	-	0.42	1.56	111	85
Robot-B	OpenVINS	0.07	0.26	1.30	106	67
	PO-KF	-	0.39	1.51	109	82
Robot-C	OpenVINS	0.07	0.27	1.33	110	67
	PO-KF	-	0.41	1.55	111	83
Robot-D	OpenVINS	0.08	0.28	1.36	112	67
	PO-KF	-	0.44	1.64	115	87
Robot-E	OpenVINS	0.08	0.30	1.38	108	67
	PO-KF	-	0.44	1.61	109	81
Robot-F	OpenVINS	0.08	0.27	1.26	107	63
	PO-KF	-	0.42	1.58	109	82
Robot-G	OpenVINS	0.09	0.33	1.43	109	67
	PO-KF	-	0.51	1.69	111	82
Robot-H	OpenVINS	0.09	0.27	1.16	105	58
	PO-KF	-	0.49	1.59	107	81
RMS	OpenVINS	*0.08*	*0.28*	*1.32*	*108*	*65*
	PO-KF	-	*0.44*	*1.59*	*110*	*83*

Tri. denotes feature triangulation, KFU. denotes Kalman filter update

In summary, the total running time for one image frame of PO-KF only slightly increases compared to MSCKF-based OpenVINS. Thus, PO-KF also demonstrates good real-time performance similar to the proven efficiency of OpenVINS.

VIII. CONCLUSION

In this paper, we propose PO-KF, a pose-only representation-based Kalman filter for visual-inertial

odometry, along with an information matrix-derived base-frame selection algorithm. Benefiting from the pose-only representation, PO-KF explicitly eliminates feature 3D positions from the measurement equation. This enables PO-KF to efficiently address the challenges present in MSCKF-based VIO, including linearization errors on feature 3D positions and delayed visual measurement updates. Moreover, our proposed base-frame selection algorithm efficiently identifies the most suitable two base-frames for each feature point, ensuring a pose-only measurement model with the optimal constraints on camera poses.

Comprehensive analysis and extensive experiments across three diverse datasets demonstrate the superior localization accuracy of PO-KF compared to SOTA VIO systems, including the optimization-based VINS-Mono, IC-VINS, and the MSCKF-based OpenVINS. Specifically, PO-KF achieves significantly enhanced localization performance relative to OpenVINS, reducing the relative rotation error and position error over a 100m trajectory by 52% and 38%, respectively. Additionally, our results validate the effectiveness of the proposed base-frame selection algorithm. Furthermore, dedicated experiments highlight the exceptional robustness of PO-KF under challenging conditions, while achieving real-time performance comparable to OpenVINS.

Commonly used keyframe strategies in VIO are designed to increase the baseline for long-tracking features within a fixed sliding-window size, wasting the valid measurements from short-tracking features. Future work will focus on developing a dynamic keyframe strategy for PO-KF that maintains a fixed sliding-window size while ensuring a sufficient baseline for long-tracking features and efficiently utilizing all measurements from short-tracking features.

ACKNOWLEDGMENTS

The authors would like to thank Prof. YuanXin Wu and Dr. Qi Cai from Shanghai Jiao Tong University for their valuable suggestions, comments and discussions. We would also like to thank Mr. Linfu Wei, Guan Wang and Man Yuan from our research group for their help in data collection.

APPENDIX A
JACOBIAN OF FEATURE DEPTH TO CAMERA POSES

As derived in Eq.(20), the feature depth in c_i-frame that represented with the i-th and j-th camera poses is denoted as:

$$d_{f,i}^{(i,j)} \triangleq \frac{\lambda_f^{\mathrm{c}_j}}{\theta_{i,j}} = \frac{\| -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} \|}{\| \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j} \boldsymbol{x}_f^{\mathrm{u}_i} \|}, \tag{38}$$

where, the $\lambda_f^{\mathrm{c}_j}$ and $\theta_{i,j}$ are calculated as follows:

$$\begin{aligned} \lambda_f^{\mathrm{c}_j} =& \sqrt{a} = \sqrt{\boldsymbol{A}^T\boldsymbol{A}}, \qquad \boldsymbol{A} = -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j} \\ \theta_{i,j} =& \sqrt{b} = \sqrt{\boldsymbol{B}^T\boldsymbol{B}}, \qquad \boldsymbol{B} = \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j} \boldsymbol{x}_f^{\mathrm{u}_i} \end{aligned}. \tag{39}$$

Based on the algebraic relations, part of the Jacobians can be calculated as follows:

$$\begin{aligned} \boldsymbol{J}_{\lambda_f}^{d_{f,i}} =& \frac{1}{\theta_{i,j}}, \qquad \boldsymbol{J}_a^{\lambda_f} = \frac{1}{2\sqrt{a}}, \qquad \boldsymbol{J}_A^a = 2\boldsymbol{A}^T \\ \boldsymbol{J}_{\theta_{i,j}}^{d_{f,i}} =& -\frac{\lambda_f^{\mathrm{c}_j}}{\theta_{i,j}^2}, \qquad \boldsymbol{J}_b^{\theta_{i,j}} = \frac{1}{2\sqrt{b}}, \qquad \boldsymbol{J}_B^b = 2\boldsymbol{B}^T \end{aligned}. \tag{40}$$

Performing error disturbance on the expressions of $\boldsymbol{A}$ and $\boldsymbol{B}$, we obtain their Jacobian matrices with respect to the i-th and j-th camera poses as follows:

$$\begin{aligned} \boldsymbol{J}_{T_{\mathrm{c}_i}^{\mathrm{w}}}^A &= \left[\mathbf{0}_3 \quad -\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_j}\right], \\ \boldsymbol{J}_{T_{\mathrm{c}_j}^{\mathrm{w}}}^A &= \left[-\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \left[\boldsymbol{p}_{\mathrm{c}_i}^{\mathrm{c}_j}\right]_\times \quad \left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{w}}^{\mathrm{c}_j}\right], \\ \boldsymbol{J}_{T_{\mathrm{c}_i}^{\mathrm{w}}}^B &= \left[-\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j} \left[\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \quad \mathbf{0}_3\right], \\ \boldsymbol{J}_{T_{\mathrm{c}_j}^{\mathrm{w}}}^B &= \left[\left[\boldsymbol{x}_f^{\mathrm{u}_j}\right]_\times \left[\mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\boldsymbol{x}_f^{\mathrm{u}_i}\right]_\times \quad \mathbf{0}_3\right]. \end{aligned} \tag{41}$$

Then we obtain the Jacobian matrices of the feature depth to the two base-frames based on the chain rule, as shown below:

$$\begin{aligned} \boldsymbol{J}_{T_{\mathrm{c},i}^{\mathrm{w}}}^{d_{f,i}} =& \boldsymbol{J}_{\lambda_f}^{d_{f,i}} \boldsymbol{J}_a^{\lambda_f} \boldsymbol{J}_A^a \boldsymbol{J}_{T_{\mathrm{c}_i}^{\mathrm{w}}}^A + \boldsymbol{J}_{\theta_{i,j}}^{d_{f,i}} \boldsymbol{J}_b^{\theta_{i,j}} \boldsymbol{J}_B^b \boldsymbol{J}_{T_{\mathrm{c}_i}^{\mathrm{w}}}^B \\ \boldsymbol{J}_{T_{\mathrm{c},j}^{\mathrm{w}}}^{d_{f,i}} =& \boldsymbol{J}_{\lambda_f}^{d_{f,i}} \boldsymbol{J}_a^{\lambda_f} \boldsymbol{J}_A^a \boldsymbol{J}_{T_{\mathrm{c}_j}^{\mathrm{w}}}^A + \boldsymbol{J}_{\theta_{i,j}}^{d_{f,i}} \boldsymbol{J}_b^{\theta_{i,j}} \boldsymbol{J}_B^b \boldsymbol{J}_{T_{\mathrm{c}_j}^{\mathrm{w}}}^B \end{aligned}. \tag{42}$$

APPENDIX B
JACOBIAN OF THE DISTORTION AND PROJECTION FUNCTIONS

For a camera with the pin-hole projection model and the rad-tan distortion model, the distortion and projection process are expressed as:

$$\begin{aligned} x_{f,d}^{\mathrm{u}_l} =& x_f^{\mathrm{u}_l}\gamma + 2p_1 x_f^{\mathrm{u}_l} y_f^{\mathrm{u}_l} + p_2\left(r^2 + 2(x_f^{\mathrm{u}_l})^2\right) \\ y_{f,d}^{\mathrm{u}_l} =& y_f^{\mathrm{u}_l}\gamma + 2p_2 x_f^{\mathrm{u}_l} y_f^{\mathrm{u}_l} + p_1\left(r^2 + 2(p_{f,y}^{\mathrm{u}_l})^2\right), \\ u^{\mathrm{p}_l} =& f_x x_{f,d}^{\mathrm{u}_l} + c_x, \quad v^{\mathrm{p}_l} = f_y y_{f,d}^{\mathrm{u}_l} + c_y \end{aligned} \tag{43}$$

where, $r^2 = \left(x_f^{\mathrm{u}_l}\right)^2 + \left(y_f^{\mathrm{u}_l}\right)^2$, $\gamma = \left(1 + k_1 r^2 + k_2 r^4\right)$. f_x, f_y, c_x, and c_y are the projection parameters of the camera, while k_1, k_2, p_1, and p_2 are the distortion parameters.

According to the algebraic relations in Eq.(43), we derive the Jacobian matrices of the distortion and projection functions as follows:

$$\boldsymbol{J}_{x_f^{\mathrm{u}_l}}^{x_{f,d}^{\mathrm{u}_l}} = \begin{bmatrix} c_1 & c_2 \\ e_1 & e_2 \end{bmatrix}, \quad \boldsymbol{J}_{x_{f,d}^{\mathrm{u}_l}}^{p_f^{\mathrm{p}_l}} = \begin{bmatrix} \frac{1}{f_x} & 0 \\ 0 & \frac{1}{f_y} \end{bmatrix}, \tag{44}$$

where, the expressions of c_1, c_2, e_1, and e_2 are as below:

$$\begin{aligned} c_1 =& \gamma + 2\left(p_{f,x}^{\mathrm{u}_l}\right)^2 k_1 + 4r^2\left(p_{f,x}^{\mathrm{u}_l}\right)^2 k_2 + 2p_{f,y}^{\mathrm{u}_l} p_1 + 6p_{f,x}^{\mathrm{u}_l} p_2 \\ c_2 =& 2p_{f,x}^{\mathrm{u}_l} p_{f,y}^{\mathrm{u}_l} k_1 + 4r^2 p_{f,x}^{\mathrm{u}_l} p_{f,y}^{\mathrm{u}_l} k_2 + 2p_{f,x}^{\mathrm{u}_l} p_1 + 2p_{f,y}^{\mathrm{u}_l} p_2 \\ e_1 =& 2p_{f,x}^{\mathrm{u}_l} p_{f,y}^{\mathrm{u}_l} k_1 + 4r^2 p_{f,x}^{\mathrm{u}_l} p_{f,y}^{\mathrm{u}_l} k_2 + 2p_{f,y}^{\mathrm{u}_l} p_2 + 2p_{f,x}^{\mathrm{u}_l} p_1 \\ e_2 =& \gamma + 2\left(p_{f,y}^{\mathrm{u}_l}\right)^2 k_1 + 4r^2\left(p_{f,y}^{\mathrm{u}_l}\right)^2 k_2 + 2p_{f,x}^{\mathrm{u}_l} p_2 + 6p_{f,y}^{\mathrm{u}_l} p_1 \end{aligned}. \tag{45}$$

REFERENCES

[1] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing Systems for Autonomous Driving: State of the Art and Challenges," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469–6486, Apr. 2021.

[2] S. Javed, A. Hassan, R. Ahmad, W. Ahmed, R. Ahmed, A. Saadat, and M. Guizani, "State-of-the-Art and Future Research Challenges in UAV Swarms," *IEEE Internet of Things Journal*, vol. 11, no. 11, 2024.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[4] G. Huang, "Visual-Inertial Navigation: A Concise Review," in *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada: IEEE, May 2019, pp. 9572–9582.

[5] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, Apr. 2007, pp. 3565–3572.

[6] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 4666–4672.

[7] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[8] Q. Cai, Y. Wu, L. Zhang, and P. Zhang, "Equivalent Constraints for Two-View Geometry: Pose Solution/Pure Rotation Identification and 3D Reconstruction," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 163–180, Feb. 2019.

[9] Q. Cai, L. Zhang, Y. Wu, W. Yu, and D. Hu, "A Pose-only Solution to Visual Reconstruction and Navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 73–86, 2021.

[10] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization," in *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, Jun. 2013.

[11] 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 Robotics and Automation Letters*, pp. 1–8, 2022.

[12] R. Mur-Artal and J. D. Tardós, "Visual-Inertial Monocular SLAM With Map Reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[13] 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 Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[14] B. Song, X. Yuan, Z. Ying, B. Yang, Y. Song, and F. Zhou, "DGM-VINS: Visual–Inertial SLAM for Complex Dynamic Environments With Joint Geometry Feature Extraction and Multiple Object Tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.

[15] S. Song, H. Lim, A. J. Lee, and H. Myung, "*DynaVINS:* A Visual-Inertial SLAM for Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 523–11 530, Oct. 2022.

[16] A. Samadzadeh and A. Nickabadi, "SRVIO: Super Robust Visual Inertial Odometry for Dynamic Environments and Challenging Loop-Closure Conditions," *IEEE Transactions on Robotics*, pp. 1–14, 2023.

[17] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 2657–2664.

[18] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos, "Inertial Aiding of Inverse Depth SLAM using a Monocular Camera," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, Apr. 2007, pp. 2797–2802.

[19] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick, "SC-KF Mobile Robot Localization: A Stochastic Cloning Kalman Filter for Processing Relative-State Measurements," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 717–730, Aug. 2007.

[20] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

[21] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-Inertial-Wheel Odometry with Online Calibration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4559–4566.

[22] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "VINS on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5155–5162.

[23] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017.

[24] M. Li and A. I. Mourikis, "Improving the accuracy of EKF-based visual-inertial odometry," in *2012 IEEE International Conference on Robotics and Automation*. St Paul, MN, USA: IEEE, May 2012, pp. 828–835.

[25] L. Mingyang, "Visual-Inertial Odometry on Resource-Constrained Systems," Pd.D., University of California, Riverside, Riverside, Dec. 2014.

[26] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online Self-Calibration for Visual-Inertial Navigation: Models, Analysis, and Degeneracy," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3479–3498, Oct. 2023.

[27] M. Li and A. I. Mourikis, "Online temporal calibration for camera–IMU systems: Theory and algorithms," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, Jun. 2014.

[28] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A First-Estimates Jacobian EKF for Improving SLAM Consistency," in *Experimental Robotics*, B. Siciliano, O. Khatib, F. Groen, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 54, pp. 373–382.

[29] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency Analysis and Improvement of Vision-aided Inertial Navigation," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[30] Z. Huai and G. Huang, "Robocentric Visual-Inertial Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6319–6326.

[31] K. Wu, A. Ahmed, G. Georgiou, and S. Roumeliotis, "A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices," in *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, Jul. 2015.

[32] J. H. Jung, J. Cha, J. Y. Chung, T. I. Kim, M. H. Seo, S. Y. Park, J. Y. Yeo, and C. G. Park, "Monocular Visual-Inertial-Wheel Odometry Using Low-Grade IMU in Urban Areas," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 925–938, 2022.

[33] L. Wang, X. Niu, T. Zhang, H. Tang, and Q. Chen, "Accuracy and Robustness of ODO/NHC Measurement Models for Wheeled Robot Positioning," *Measurement*, p. 111720, Aug. 2022.

[34] T. Hua, L. Pei, T. Li, J. Yin, G. Liu, and W. Yu, "M2C-GVIO: Motion manifold constraint aided GNSS-visual-inertial odometry for ground vehicles," *Satellite Navigation*, vol. 4, no. 1, p. 13, Dec. 2023.

[35] H. Li and J. Stueckler, "Visual-Inertial Odometry With Online Calibration of Velocity-Control Based Kinematic Motion Models," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6415–6422, Jul. 2022.

[36] X. Qiu, H. Zhang, and W. Fu, "Lightweight hybrid visual-inertial odometry with closed-form zero velocity update," *Chinese Journal of Aeronautics*, vol. 33, no. 12, pp. 3344–3359, Dec. 2020.

[37] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[38] 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.

[39] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-Inertial Odometry With Structural Regularity of Man-Made Environments," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 999–1013, 2019.

[40] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging Planar Regularities for Point Line Visual-Inertial Odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5120–5127.

[41] J. Montiel, J. Civera, and A. Davison, "Unified Inverse Depth Parametrization for Monocular SLAM," in *Robotics: Science and Systems II*. Robotics: Science and Systems Foundation, Aug. 2006.

[42] L. Liu, T. Zhang, Y. Liu, B. Leighton, L. Zhao, S. Huang, and G. Dissanayake, "Parallax Bundle Adjustment on Manifold with Improved Global Initialization," in *Algorithmic Foundations of Robotics XIII*, M. Morales, L. Tapia, G. Sánchez-Ante, and S. Hutchinson, Eds. Cham: Springer International Publishing, 2020, vol. 14, pp. 621–638.

[43] Y. Yang and G. Huang, "Observability Analysis of Aided INS With Heterogeneous Features of Points, Lines, and Planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1399–1418, 2019.

[44] 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," *arXiv preprint arXiv:2305.12644*, May 2023.
[45] 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 Transactions on Robotics*, pp. 1–14, 2024.
[46] H. Tang, T. Zhang, X. Niu, J. Fan, and J. Liu, "Impact of the Earth Rotation Compensation on MEMS-IMU Preintegration of Factor Graph Optimization," *IEEE Sensors Journal*, vol. 22, no. 17, pp. 17 194–17 204, Sep. 2022.
[47] E.-H. Shin, "Estimation techniques for low-cost inertial navigation," Pd.D., University of Calgary, Calgary, 2005.
[48] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko, "Zero-Velocity Detection—An Algorithm Evaluation," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2657–2666, Nov. 2010.
[49] W. Ouyang, Y. Wu, and H. Chen, "INS/Odometer Land Navigation by Accurate Measurement Modeling and Multiple-Model Adaptive Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 245–262, Feb. 2021.
[50] X. Qiu, H. Zhang, W. Fu, C. Zhao, and Y. Jin, "Monocular Visual-Inertial Odometry with an Unbiased Linear System Model and Robust Feature Tracking Front-End," *Sensors*, vol. 19, no. 8, p. 1941, Apr. 2019.
[51] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC: IEEE, Sep. 2017, pp. 4225–4232.
[52] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, May 2019.
[53] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stuckler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 1680–1687.
[54] M. Grupp, "Evo: Python Package for the Evaluation of Odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

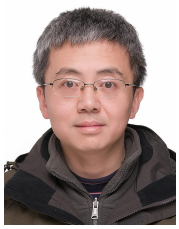
Liqiang Wang received the B.E. and M.E. degrees from Wuhan University, China, in 2020 and 2023, respectively. He is currently pursuing a Ph.D. degree at GNSS Research Center, Wuhan University. His primary research interests include GNSS/INS integrations, visual SLAM, and multi-sensor fusion navigation systems.

Hailiang Tang received the M.E. and Ph.D. degrees from Wuhan University, China, in 2020 and 2023, respectively. He is currently a postdoctoral fellow at GNSS Research Center, Wuhan University. His research interests include autonomous robotics systems, visual and LiDAR SLAM, GNSS/INS integration, and deep learning.

Tisheng Zhang is an Associate Professor at GNSS Research Center, Wuhan University, China. He received his 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-Doctor at Hong Kong Polytechnic University from 2018 to 2019. His research interests focus on the fields of GNSS receiver and multi-sensor integrated navigation.

Yan Wang received the M.E. degree from Computer Applied Technology, China University of Mining and Technology, China, in 2019, and the Ph.D. degree from GNSS Research Center, Wuhan University, China, in 2023. He is currently a postdoctoral fellow at GNSS Research Center, Wuhan University. His research interests focus on indoor navigation, sensor fusion algorithms, and computer vision.

Quan Zhang (Member, IEEE) received the B.S. degree in Geomatics Engineering from Shandong University of Science and Technology, China, in 2009, and the Ph.D. degree from Wuhan University, China, in 2015. From 2017 to 2018, he was a Post-Doctoral Researcher with the Digital Photogrammetry Research Group (DPRG), Lyles School of Civil Engineering of Purdue University. He is currently an Associate Professor at GNSS Research Center in Wuhan University. His research interests include vehicle-mounted GNSS/INS integration technology and the integrity of multi-sensor integrated navigation for intelligent driving.



Xiaoji Niu is a Professor at GNSS Research Center, Wuhan University, China. He was conferred Ph.D. and bachelor's degrees (with honors) by the Department of Precision Instruments at Tsinghua University in 2002 and 1997, respectively. He performed post-doctoral research at the University of Calgary, Canada, and worked as a senior scientist at SiRF Technology Inc. Dr. Niu is currently leading the Integrated & Intelligent Navigation (i2Nav) group. His research interests focus on GNSS/INS integration, low-cost navigation sensor fusion, and relevant new applications. Dr. Niu has published 200+ academic papers and owns 30+ patents.