# PO-VINS: An Efficient and Robust Pose-Only Visual–Inertial State Estimator With LiDAR Enhancement

Hailiang Tang, Tisheng Zhang, Liqiang Wang, Guan Wang, and Xiaoji Niu

***Abstract*—The pose adjustment with a pose-only visual representation has been proven equivalent to the bundle adjustment (BA) while significantly improving the computational efficiency. However, the pose-only solution has not yet been properly considered in a tightly coupled visual–inertial state estimator (VISE) with a normal configuration for real-time navigation. Meanwhile, the light detection and ranging (LiDAR) can provide accurate depth estimation for visual landmarks and thus can enhance the robustness of VISEs. Hence, we propose a tightly coupled LiDAR-enhanced VISE, named PO-VINS, which can be an efficient and robust odometry framework for autonomous applications. The pose-only solution is integrated for higher state-estimation efficiency, while the lightweight LiDAR enhancement is employed to improve robustness and accuracy. Specifically, based on the pose-only visual representation, we derive the analytical depth uncertainty, which is employed for rejecting LiDAR-depth outliers. Besides, we propose a multistate constraint (MSC)-based LiDAR-depth measurement (LDM) model within the pose-only framework, to balance efficiency and robustness. The pose-only visual and LDMs and the IMU-preintegration measurements are tightly integrated under the factor graph optimization (FGO) framework to perform efficient and accurate state estimation. Exhaustive experimental results on private and public datasets indicate that the proposed PO-VINS yields improved or comparable accuracy to state-of-the-art (SOTA) methods. Compared to the baseline method LE-VINS, the state-estimation efficiency of PO-VINS on the private dataset is improved by 33% and 56% on the laptop PC (Intel i7-13700H) and the onboard ARM computer (NVIDIA Xavier), respectively. Besides, PO-VINS yields higher accuracy and robustness than LE-VINS by employing the proposed uncertainty-based outlier culling method and MSC-based measurement model for LiDAR depth.**

***Index Terms*—Factor graph optimization (FGO), multisensor fusion navigation, pose-only state estimation, visual–inertial navigation.**

Received 18 July 2025; revised 16 October 2025; accepted 21 October 2025. Date of publication 13 November 2025; date of current version 21 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 42374034 and Grant 42504027, in part by the Key Research and Development Program of Hubei Province under Grant 2024BAB024, in part by the Major Program (JD) of Hubei Province under Grant 2023BAA02602, and in part by the High Quality Development Project of the Ministry of Industry and Information Technology (MIIT) under Grant 2024-182. The Associate Editor coordinating the review process was Dr. Alvaro Hernandez. *(Corresponding author: Tisheng Zhang.)*

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

Liqiang Wang and Guan Wang are with the GNSS Research Center, Wuhan University, Wuhan 430079, China (e-mail: wlq@whu.edu.cn; wanguan@whu.edu.cn).

Digital Object Identifier 10.1109/TIM.2025.3632451

## NOMENCLATURE

| | |
|---|---|
| $p^{\mathrm{p}}$ | Coordinate $(u, v)$ in the image pixel frame. |
| $p^{\mathrm{u}}$ | Coordinate in the normalized camera frame (u-frame) with a unit depth. |
| $p^{\mathrm{r}}$ | LiDAR point in the LiDAR range frame (r-frame). |
| $\mathbf{q}, \mathbf{R}, \phi$ | Attitude quaternion, rotation matrix, and rotation vector. |
| $\otimes$ | Quaternion product. |
| Log,Exp | Transformation between the quaternion and rotation vector. |
| $p^{\mathrm{w}}_{\mathrm{wb}}, \mathbf{q}^{\mathrm{w}}_{\mathrm{b}}$ | IMU pose body frame (b-frame) with respect to the world frame (w-frame). |
| $v^{\mathrm{w}}_{\mathrm{wb}}$ | IMU velocity in the world frame. |
| $b_g, b_a$ | Gyroscope and accelerometer biases. |
| $p^{\mathrm{b}}_{\mathrm{bc}}, \mathbf{q}^{\mathrm{b}}_{\mathrm{c}}$ | Camera–IMU extrinsic parameters. |
| $p^{\mathrm{r}}_{\mathrm{rc}}, \mathbf{q}^{\mathrm{r}}_{\mathrm{c}}$ | Camera–LiDAR extrinsic parameters. |
| $t_d$ | Time-delay parameter between the camera and the IMU data. |
| $X, x$ | State vector. |
| ATE | Absolute translation error. |
| BA | Bundle adjustment. |
| FGO | Factor graph optimization. |
| GNSS | Global navigation satellite system. |
| IMU | Inertial measurement unit. |
| INS | Inertial navigation system. |
| LDM | LiDAR-depth measurement. |
| LVINS | LiDAR–visual–inertial navigation system. |
| MEMS | Micro-electromechanical system. |
| MSC | Multistate constraint. |
| RTE | Relative translation error. |
| SLAM | Simultaneous localization and mapping. |
| VINS | Visual–inertial navigation system. |
| VISE | Visual–inertial state estimator. |
| VRM | Visual-reprojection measurement. |

## I. INTRODUCTION

THE VINS, due to its lower cost and smaller size, has been broadly used in mobile robots and autonomous vehicles

[1]. A single camera cannot estimate the absolute scale of the ego motion, resulting in the scale uncertainty of the monocular camera [2]. The ego motion scale can be retrieved by utilizing a low-cost MEMS-IMU. Meanwhile, the IMU errors, such as biases and scale-factor errors [3], can also be estimated with the visual measurements derived from the frame-to-frame data association. Besides, compared to LiDAR-based methods, visual-based methods have a lower cost with satisfied accuracy. Hence, the tightly coupled VINS can achieve reliable and accurate relative positioning, especially in GNSS-denied environments [4].

Typically, the visual features [5], [6] are detected and tracked frame by frame to obtain measurements, or the raw pixels [7] and image patches [8] are directly treated as the measurements. With the visual measurements from the front-end and the IMU measurements, state estimation [9] can be performed to determine the pose and velocity. Conventionally, the filtering-based methods, including the extended Kalman filter (EKF) [10], [11] and iterated extended Kalman filter (IEKF) [12], are employed to solve the state-estimation problem in VINS due to lower computational costs. Particularly, the multistate constraint Kalman filter (MSCKF)-based VINS [10], [11], [13], [14] has exhibited quite good performance. FGO [15] has been proven more accurate than filtering in VINS [16]. This is because the nonlinear error of the visual measurement can be reduced notably by multiple iterations. Hence, the optimization-based VINS has attracted a lot of attention, including feature-based [17], [18], [19], [20] and direct methods [7], [21], [22].

Though consuming many computational resources, the optimization-based VINS can still achieve superior real-time performance, such as VINS-Mono [17], especially with the rapid development of computer technology. Commonly, the BA [2], [9] is adopted to estimate the pose and visual landmark parameters simultaneously. However, the high dimensional parameter space of visual landmarks significantly increases the computational complexity in solving the state-estimation problem in VINS. In contrast, the dimension of the pose parameters is much lower. Researchers have studied new methods to improve the efficiency of the BA, such as incremental BA [23], [24], fixed-lag smoother or sliding-window optimizer [17], [25], and preserving sparsity [26]. Nevertheless, too many computational resources have been used to estimate the parameter space of visual landmarks. In some edge devices, it is essential to decrease the computational costs, especially when minimal processor cores are available for positioning tasks.

Recently, the two-view imaging geometry has been proven equivalent to a pair of pose-only constraints decoupling camera poses from visual landmark parameters [27]. This work has been extended to the multiple-view imaging geometry [28]. The pose-only state estimation is conducted by adopting a pose adjustment rather than the BA. Besides, the visual landmark parameters can be analytically reconstructed from the obtained camera poses [28]. Specifically, the pose adjustment is conducted by explicitly expressing the visual landmark parameter with the pose and feature measurements. The pose-only solution has demonstrated that computational efficiency is significantly improved by 2–4 orders of magnitude for 3-D scene reconstruction [28]. PIPO-SLAM [29], a visual–inertial SLAM system based on ORB-SLAM3 [20], employs a local pose adjustment rather than the BA in the local mapping thread. The employed local pose adjustment yields significantly decreased computational and memory costs when more than 100 camera poses and 1000 landmarks are employed for local mapping optimization [29]. Note that, the local pose adjustment in PIPO-SLAM [29] is a visual-only optimization, the same as in [28], rather than a tightly coupled visual–inertial optimization. Besides, using 100 cameras and 1000 landmarks is not a normal configuration for real-time navigation. Consequently, the pose-only solution has not yet been properly considered in a tightly coupled VISE with a typical configuration for real-time navigation. Specifically, the impact of the pose-only solution on the accuracy and efficiency of the real-time VINS is still unknown.

Except for the efficiency issue, accurate depth estimation for visual landmarks is a critical factor that influences the accuracy and robustness of a VINS. In contrast, light detection and ranging (LiDAR) can directly measure long-distance accurate depths [30], and thus, the 3-D LiDAR can be employed to enhance the VINS by providing accurate depth [31], [32], [33]. Therefore, the LiDAR-depth-enhanced VINSs can achieve a similar accuracy to LiDAR-centric methods in most scenarios, while without integrating an LiDAR odometry (LO) and reducing computational costs. In short, the lightweight LiDAR-enhanced VINS can be an efficient and accurate method in many applications. In LE-VINS [33], LiDAR has been proven to improve the robustness and accuracy of the VINS by constructing an LDM in the state estimator. Hence, the inverse-depth parameter [34] of the visual landmark can be accurately estimated with the LDM, and the pose accuracy can be improved [33]. However, the state parameters of the visual landmarks will be removed from the state vector in the pose-only solution. The pose-only LDM model should be further studied for accurate and efficient state estimation.

In this study, we aim to construct a pose-only VISE with LiDAR enhancement, named PO-VINS, to achieve an efficient and robust state estimation. The pose-only solution is incorporated to improve the efficiency by removing the visual landmark states from the state estimator. Meanwhile, the lightweight LiDAR enhancement is involved to improve the accuracy by providing accurate depth measurements for visual landmarks, while the LDM model is also constructed within the pose-only framework. The main contributions of this study can be listed as follows.

1) A tightly coupled VISE with LiDAR enhancement is presented under the framework of FGO. The proposed state estimator is a complete pose-only solution for VRM and LDM, which balances efficiency and robustness.
2) The analytical uncertainty for visual landmark depth is derived from the pose-only visual presentation. Hence, an uncertainty-based outlier rejection is proposed to detect and reject LiDAR-depth outliers, which has been proven to improve the robustness of the state estimation.

3) The famous MSC method is adopted to build a pose-only LDM model by combining all visual and LiDAR-depth observations of a landmark into a single measurement. Hence, the measurement number can be decreased, and the estimation efficiency can be improved. Besides, it can also reduce the impact of the LiDAR-depth outliers and improve estimation robustness.
4) Comprehensive experiments are conducted on public and private datasets involving a low-speed robot, a handheld device, and a vehicle. Besides, we also verify the real-time performance on an embedded onboard ARM computer.

The remainder of this article is organized as follows. We first briefly discuss the related works about the VINS and the LiDAR-enhanced VINS. The system overview is described in Section III. Section IV presents the pose-only visual representation and derives the analytical depth uncertainty for landmarks. Then, the proposed LiDAR-enhanced VISE is given in Section V, with a full pose-only form for visual and LDMs. The experiments and results are shown and discussed in Section VI for quantitative evaluation. Finally, we conclude the proposed PO-VINS.

## II. Related Works

In this study, we aim to construct a tightly coupled VINS with LiDAR enhancement for efficient and robust state estimation. Hence, the related works can be categorized into VISE and LiDAR-enhanced VISE. The LiDAR enhancement described here denotes the method using the LiDAR depth to improve the accuracy and robustness of VISE. Tightly coupled methods have been proven more accurate than loosely coupled methods, and thus, we will discuss the former.

### A. Visual–Inertial State Estimator

In the early stage, the visual–inertial estimator is modeled using the Kalman filter [9] due to the limitations in computing resources. MonoSLAM is a real-time monocular SLAM system that runs at 30 Hz with a standard personal computer within the EKF framework [35]. FEJ-EKF is proposed to improve the estimator's consistency by employing the first-estimate Jacobians [36]. The IEKF is also adopted for visual–inertial state estimation [12]. An MSCKF-based estimator is presented for vision-aided inertial navigation without including the visual landmarks in the state vector [10]. The delayed linearization in MSCKF avoids the computational burden and information loss, and thus, MSCKF yields improved precision and efficiency [10]. MSCKF 2.0 builds upon MSCKF by adopting the first-estimate Jacobians to achieve consistent state estimation and perform online estimation of the camera–IMU extrinsic parameters [11]. Due to superior real-time performance, MSCKF has become a famous VISE framework. Hence, many recent works have focused on MSCKF-based estimators, such as R-VIO [37] and OpenVINS [13]. However, the one-time linearization for nonlinear visual measurements in filtering-based methods possibly introduces large linearization errors into the estimator and degrades performance [16].

In contrast, the nonlinear visual measurements can be repeatedly linearized by using the FGO [15]. Hence, the FGO should be more accurate than the filter for visual–inertial estimation [16], [38], though with higher computation complexity. The FGO or the nonlinear optimization is usually implemented by employing a sliding-window estimator [39] to reduce the dimensions of the state vector and improve estimation efficiency. Besides, the marginalization is conducted using the Schur complement [40] by converting all marginalized states into a prior to avoid loss of information [7], [25], [39]. OKVIS [25] is a keyframe-based visual–inertial odometry (VIO) using nonlinear optimization, exhibiting improved accuracy than MSCKF-based methods. VINS-Mono [17] further employs a robust visual–inertial initialization and online relocalization using a global pose graph optimization (PGO). Kimera [18] includes a similar VIO module developed based on the famous nonlinear optimization library GTSAM [41]. ORB-SLAM-VI [42] and ORB-SLAM3 [20], which are both based on ORB-SLAM [43], [44], also employ marginalization to construct the prior. BASALT [19] extracts relevant information from VIO using nonlinear factor recovery [45] for visual–inertial mapping. The direct spare method [7] is incorporated into VI-DOS [46], which includes dynamic marginalization. DM-VIO [22] is a delayed marginalization VIO that maintains a second factor graph for marginalization, yielding improved accuracy.

The above optimization-based methods use marginalization to improve real-time performance without information loss. Incremental BA, such as ICE-BA [23] and iSAM2 [24], [41], can also enhance the efficiency of VISE by only updating limited states that are related to the observations. However, the incremental BA is mainly designed for global mapping rather than odometry. Except for these methods, using different feature parametrization, such as inverse-depth [34], ParallaxBA [47], and PMBA [48], may result in various iterations and running time. Nevertheless, the high dimensional parameter space of visual landmarks is the main obstacle for efficient visual–inertial state estimation. The pose-only solution, i.e., pose adjustment, has exhibited higher efficiency than BA for large-scale visual-only state estimation [28], [29]. Nevertheless, the pose-only solution has not yet been considered for a real time, tightly coupled VISE, in which the dimension of landmark parameters is very limited. Specifically, the effects of the pose-only solution on the efficiency and accuracy of tightly coupled VISE have not been demonstrated.

### B. LiDAR-Enhanced VISE

The LiDAR-enhanced VISE can be further classified into direct and feature-based methods [33]. Direct methods employ LiDAR depths for visual pixels, and pose estimation is achieved by conducting photometric BA. In feature-based methods, visual features are associated with LiDAR depths, and BA is conducted to perform state estimation.

As the visual pixels are used in direct methods, sparse LiDARs, such as a 16-beam spinning LiDAR, can be adopted to provide depths. A direct laser–visual odometry is achieved by conducting a photometric alignment with occlusion handling [49]. DVL-SLAM is a similar direct visual SLAM

system using window-based optimization, but it does not consider occlusion. Accurate LiDAR depths have also been employed for pixels in tightly coupled LiDAR-VIO, such as R3LIVE [50] and FAST-LIVO [51], [52]. However, the occlusion should be considered to avoid wrong LiDAR-depth usages, or it will degrade accuracy severely. Besides, the impact of the camera–LiDAR extrinsic errors is more notable in the direct methods, as the LiDAR depth at the corresponding pixel will be used.

Hence, many LiDAR-enhanced VISEs employ feature-based methods, in which visual features are first detected and then associated with LiDAR depths. In DEMO [31], visual features and LiDAR point clouds are associated in a unit sphere of the camera frame. This depth-association method is adopted by [53], but the LiDAR depth is treated as a constant without considering the depth error. LE-VINS [33] proposes an LDM model that constructs constraints to visual landmark states while considering the depth errors. In contrast, the depth association is achieved in the image plane in [32] and [54], leading to high computational costs when projecting the point clouds into the image plane. A voxel-map-based depth-association method is used in [55], but the LiDAR depth is only treated as an initial value. A nonrepetitive solid-state LiDAR is employed in CamVox [56] to build depth images, which are incorporated into ORB-SLAM2 [44] with RGB images to achieve an RGB-D SLAM. However, the nonoverlapping area between the LiDAR and camera field of view in CamVox will be partially wasted.

In conclusion, feature-based methods should be more suitable for LiDAR-enhanced VISE in terms of practicality and reliability. However, wrong or inaccurate depth associations may occur, no matter what depth-association methods are used. The reason is mainly because the occlusion may frequently happen on object edges, such as trees and vehicles. For example, a visual feature is detected on the edge of a tree, and it may be associated with an LiDAR point on a wall behind the tree due to camera–LiDAR extrinsic errors. Hence, it is critical to reject LiDAR-depth outliers to improve robustness. Besides, the LDM model should be further improved to perform the high-efficiency characteristic of the pose-only framework. Consequently, we can achieve efficient and robust LiDAR-enhanced VISE by constructing pose-only visual and LDMs.

## III. System Overview

The proposed PO-VINS is built upon LE-VINS [33] by further incorporating the pose-only solution. The pose-only solution is integrated to improve the state-estimation efficiency in PO-VINS. The LiDAR is employed to provide accurate depths for visual landmarks to improve the system's accuracy and robustness while incorporating an LO and saving computational resource.

The system pipeline of the proposed PO-VINS is depicted in Fig. 1. We employ an INS-centric [4] processing framework to fully utilize the short-term accuracy of the INS. The INS is initialized by gravity leveling [3] to estimate roll and pitch angles roughly while the initial position, velocity, and heading angle are set to zero. We can also derive the initial gyroscope

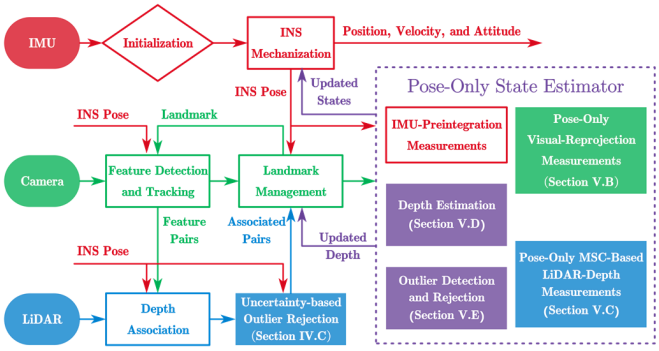



Fig. 1. System overview of the proposed PO-VINS. Here, the pose-only solution is integrated to improve the state-estimation efficiency. The LiDAR is employed to provide accurate depths for visual landmarks to improve the system's accuracy and robustness while consuming too many computational resources.

biases during stationary conditions. Note that, in-motion visual alignment methods [17] can also be adopted, but they are out of the scope of this article. The high-frequency INS mechanization can be conducted with the initialized states, and the obtained INS pose will be used for the following camera and LiDAR data processing.

With continuous visual image frames derived from the camera, Shi and Tomasi [5] feature points are detected in parallel from separated image grids to improve efficiency. The prior INS pose is then employed to assist the Lucas–Kanade optical flow [57] to enhance the tracking continuity. The tracking efficiency can also be improved with the INS aiding by reducing the usage of the image pyramid [4]. The visual keyframe is selected by judging the keyframe interval and relative motions [4]. Only feature points in the keyframes will be added to the landmark manager for state estimation to improve the efficiency.

Meanwhile, the accumulated LiDAR point clouds are projected to the visual keyframe time using the high-frequency INS pose with distortion removed. Hence, the feature pairs and LiDAR points can be associated in a unit sphere of the camera frame, and LiDAR depths for corresponding landmarks can be estimated by employing a plane-fitting method [33]. Several outlier culling methods have been adopted to remove LiDAR-depth outliers, such as using only foreground point clouds and plane checking [33]. However, LiDAR-depth outliers may exist, especially in unstructured environments. Thus, we derive the analytical depth uncertainty for visual landmarks from the pose-only representation and employ the uncertainty to remove LiDAR-depth outliers.

Finally, we obtain a series of visual features, some associated with LiDAR depths. The visual features, LiDAR depths, and IMU measurements are tightly fused using a sliding-window optimizer to perform maximum-a-posterior estimation. The gyroscope and accelerometer observations from the IMU are used to construct an IMU-preintegration measurement model [58]. Those visual features without LiDAR depth are utilized to build pose-only VRMs. Hence, the landmark states are not incorporated into the state vector; thus, the dimension of the state vector is reduced. The proposed VRM is in a tightly coupled form, as the IMU pose states

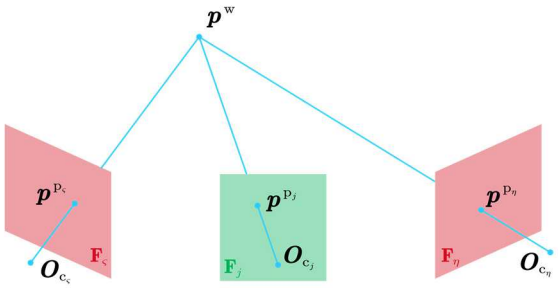



Fig. 2. Illustration of the multiview geometry. The red frame s$\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ represent the anchored frames of the landmark $\boldsymbol{p}^{\mathrm{w}}$. Note that, only keyframes are employed for state estimation.

are adopted in the state vector. For those visual features with LiDAR depths, we present MSC-based LDMs in pose-only form to compress the measurement number, thus improving efficiency. The system is also more robust to LiDAR-depth outliers by combining all feature observations of a landmark and LiDAR depth into one measurement. The pose-only state estimation is achieved by integrating visual points, LiDAR-depth, and IMU measurements to conduct nonlinear optimization. After that, the depths for visual landmarks are updated, and visual feature outliers are detected and rejected with the updated depths. Meanwhile, the updated INS states, including navigation states and IMU biases, are returned to the INS mechanization module for real-time high-frequency navigation.

## IV. Pose-Only Representation for Visual Landmarks

This section presents the methodology of the pose-only visual representation. We first derive the pose-only representation for visual landmarks from the multiple-view geometry. The analytical depth uncertainty of the landmark can be obtained by conducting error propagation. Based on the derived uncertainty, the LiDAR-depth outliers can be further detected and rejected.

### A. Pose-Only Visual Representation

considers a 3-D visual landmark in the world frame (w-frame) $\boldsymbol{p}^{\mathrm{w}}$ observed in several images, as depicted in Fig. 2. The origin of the camera frame (c-frame) is represented by $\boldsymbol{O}_{\mathrm{c}}$. $\boldsymbol{p}^{\mathrm{p}} = (u, v)$ is the observed feature in the image plane, i.e., pixel frame (p-frame). $\boldsymbol{p}^{\mathrm{p}}$ can be converted into the normalized camera frame (u-frame) using the back projection function $\pi_c^{-1}$ and camera intrinsic parameters [2], denoted as $\boldsymbol{p}^{\mathrm{u}} = (x^{\mathrm{u}}, y^{\mathrm{u}}, 1)$. The projection equation of the 3-D visual landmark $\boldsymbol{p}^{\mathrm{w}}$ in the u-frame can be written as

$$\boldsymbol{p}^{\mathrm{u}} = \frac{1}{z^{\mathrm{c}}}\boldsymbol{p}^{\mathrm{c}} = \frac{1}{z^{\mathrm{c}}}\left(\mathbf{R}_{\mathrm{c}}^{\mathrm{w}}\right)^{-1}\left(\boldsymbol{p}^{\mathrm{w}} - \boldsymbol{p}_{\mathrm{wc}}^{\mathrm{w}}\right) \tag{1}$$

where $\boldsymbol{p}^{\mathrm{c}} = (x^{\mathrm{c}}, y^{\mathrm{c}}, z^{\mathrm{c}})$ is the coordinate of the visual landmark in the c-frame and $\{\boldsymbol{p}_{\mathrm{wc}}^{\mathrm{w}}, \mathbf{R}_{\mathrm{c}}^{\mathrm{w}}\}$ is the camera pose with respect to the w-frame.

For two images $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ in Fig. 2, we can derive the following equation from (1) as:

$$z^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\eta} = z^{\mathrm{c}_\varsigma}\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma} + \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta} \tag{2}$$

where $\{\boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}, \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\}$ is the relative transformation from the c-frame $\mathrm{c}_\varsigma$ to the c-frame $\mathrm{c}_\eta$ and they can be written as

$$\begin{cases} \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta} = \left(\mathbf{R}_{\mathrm{c}_\eta}^{\mathrm{w}}\right)^{-1}\left(\boldsymbol{p}_{\mathrm{wc}_\varsigma}^{\mathrm{w}} - \boldsymbol{p}_{\mathrm{wc}_\eta}^{\mathrm{w}}\right) \\ \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta} = \left(\mathbf{R}_{\mathrm{c}_\eta}^{\mathrm{w}}\right)^{-1}\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{w}}. \end{cases} \tag{3}$$

Left multiply the skew-symmetric matrix $[\boldsymbol{p}^{\mathrm{u}_\eta}]_\times$ [59] on both sides of (2), and we can obtain

$$z^{\mathrm{c}_\varsigma}\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma} = -\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}. \tag{4}$$

Taking the magnitude in (4), the landmark depth $z^{\mathrm{c}_\varsigma}$ in the c-frame $\mathrm{c}_\varsigma$ can be written as

$$z^{\mathrm{c}_\varsigma} = \frac{\left\|\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right\|}{\theta_{\varsigma,\eta}} \triangleq d_\varsigma^{(\varsigma,\eta)} \tag{5}$$

where $\theta_{\varsigma,\eta} = \|[\boldsymbol{p}^{\mathrm{u}_\eta}]_\times \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}\|$. Similarly, we can obtain $z^{\mathrm{c}_\eta}$ by left multiply $[\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}]_\times$ on both sides of (2) as

$$z^{\mathrm{c}_\eta} = \frac{\left\|\left[\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}\right]_\times \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right\|}{\theta_{\varsigma,\eta}} \triangleq d_\eta^{(\varsigma,\eta)}. \tag{6}$$

Hence, the pose-only constraint for two-view imaging geometry in (2) can be written as

$$d_\eta^{(\varsigma,\eta)}\boldsymbol{p}^{\mathrm{u}_\eta} = d_\varsigma^{(\varsigma,\eta)}\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma} + \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}. \tag{7}$$

The pose-only constraint in (7) has been proven to be equivalent to the two-view imaging geometry [27]. Supposing that $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ are two anchored frames of the visual landmark $\boldsymbol{p}^{\mathrm{w}}$, we can derive a set of constraints $C(\varsigma, \eta)$ as

$$C(\varsigma,\eta) = \left\{d_j^{(\varsigma,j)}\boldsymbol{p}^{\mathrm{u}_j} = d_\varsigma^{(\varsigma,\eta)}\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_j}\boldsymbol{p}^{\mathrm{u}_\varsigma} + \boldsymbol{p}_{\mathrm{c}_j\mathrm{c}_\varsigma}^{\mathrm{c}_j}, j \neq \varsigma\right\} \tag{8}$$

where $\mathbf{F}_j$ denotes another observed frame, as shown in Fig. 2. Equation (8) is the pose-only visual representation for multiply-view geometry and has been proved equivalent to the projection (1) [28]. The key idea in pose-only visual representation (8) is that we can explicitly use the pose of the anchored frames and feature observations to express the landmark state, i.e., the depth. Hence, the visual landmark states can be removed from the state vector, and thus, the state-estimation efficiency can be improved.

Consequently, the pose-only visual measurements can be derived from (8). It should be noted that the anchored frames $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ should construct the largest parallax of the landmark $\boldsymbol{p}^{\mathrm{w}}$. Fortunately, the parameter $\theta_{\varsigma,\eta}$ can be employed to represent the magnitude of the parallax [28]. In practice, the anchored frame $\mathbf{F}_\varsigma$ is set to the first observed frame or the frame associated with LiDAR depth to facilitate the sliding-window marginalization. The anchored frame $\mathbf{F}_\eta$ can be searched within the other frames to meet the largest parallax, thus maximizing visual constraints and improving accuracy.

### B. Analytical Depth Uncertainty for Visual Landmark

With the pose-only depth (5), the landmark depth $d_\varsigma^{(\varsigma,\eta)}$ in the anchored frame $\mathbf{F}_\varsigma$ can be analytically obtained using camera pose and feature observations. We can also derive the analytical depth uncertainty using error perturbation [3]. In

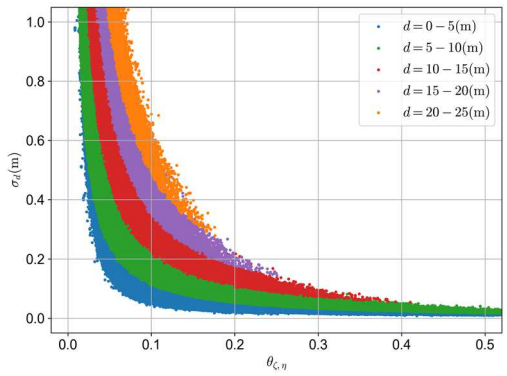



Fig. 3. Relationship between the standard deviation of depth $\sigma_d$ and the parallax parameter $\theta_{\varsigma,\eta}$. The results are obtained from real-world datasets.

tightly coupled VISE, the camera pose is calculated by IMU pose states and camera–IMU extrinsic parameters. Due to the high short-term accuracy of the INS pose, the relative pose errors of the camera can be ignored within the sliding window, whose length is usually about several seconds. Hence, the depth uncertainty is mainly caused by feature noise. Typically, the feature noise $\sigma_p$ is manually set to a fixed value, such as in [13] and [17], and it is set as $\sigma_p = 1.5$ pixels in PO-VINS. We can obtain the covariance matrix of feature points $\Sigma_p^{\mathrm{u}}$ in the u-frame as

$$\Sigma_p^{\mathrm{u}} = \begin{bmatrix} \left(\sigma_p/f_x\right)^2 & 0 \\ 0 & \left(\sigma_p/f_y\right)^2 \end{bmatrix} \tag{9}$$

where $f_x$ and $f_y$ are focal lengths in intrinsic parameters.

According to the pose-only depth formulation (5), we can write the Jacobians with respect to the feature error $\boldsymbol{p}^{\mathrm{u}_\varsigma}$ as

$$\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\varsigma}} = \left(-\frac{\beta}{\theta^3}\left(\left(\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right)^{\mathrm{T}}\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}\right)^{\mathrm{T}}\right)_{[0:1]} \tag{10}$$

where the real number $\beta$ is expressed as

$$\beta = \left\|\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right\|. \tag{11}$$

Note that, the third dimension in (10) is not used, as the raw feature point is in two dimensions. Similarly, the Jacobians with respect to the feature error $\boldsymbol{p}^{\mathrm{u}_\eta}$ can be written as

$$\begin{aligned}\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\eta}} = &\left(-\frac{1}{\beta\theta}\left(\left(\left[\boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right]_\times\right)^{\mathrm{T}}\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\right)^{\mathrm{T}}\right. \\ &\left.+\frac{\beta}{\theta^3}\left(\left(\left[\mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}\right]_\times\right)^{\mathrm{T}}\left[\boldsymbol{p}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}_{\mathrm{c}_\varsigma}^{\mathrm{c}_\eta}\boldsymbol{p}^{\mathrm{u}_\varsigma}\right)^{\mathrm{T}}\right)_{[0:1]}.\end{aligned} \tag{12}$$

Finally, the uncertainty $\Sigma_d = \sigma_d^2$ of the landmark depth $d_\varsigma^{(\varsigma,\eta)}$ can be analytically calculated as

$$\Sigma_d = \left(\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\varsigma}}\right)^{\mathrm{T}}\Sigma_p^{\mathrm{u}}\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\varsigma}} + \left(\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\eta}}\right)^{\mathrm{T}}\Sigma_p^{\mathrm{u}}\frac{\partial d_\varsigma^{(\varsigma,\eta)}}{\partial \boldsymbol{p}^{\mathrm{u}_\eta}}. \tag{13}$$

Fig. 3 shows the depth uncertainty $\sigma_d$, i.e.,the standard deviation, with respect to different parallax parameters $\theta_{\varsigma,\eta}$ and depth. It exhibits that the depth uncertainty is minor when the parallax parameter is larger. Besides, if the depth is larger, the depth uncertainty is larger. The results satisfy the theory of the two-view geometry [2]. Hence, it demonstrates that the derived analytical depth uncertainty in (13) is correct. We also noted that the depth uncertainty $\sigma_d$ may be larger than several decimeters, which is far larger than the LiDAR measurement error. This is why the estimation accuracy can be improved by incorporating the LiDAR enhancement.

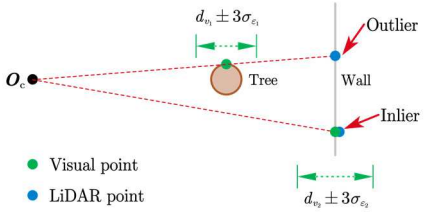



Fig. 4. Illustration of the LiDAR-depth outliers and inliers. The visual points denote the observed position of landmarks, where the LiDAR points represent the calculated position using associated LiDAR depths.

### C. Uncertainty-Based Outlier Rejection for LiDAR Depths

The depth uncertainty can be analytically calculated within the pose-only framework using (13). The depth uncertainty can be further employed to detect and reject LiDAR-depth outliers. Considering a visual landmark, its depth $d_v$ in the anchored frame $\mathbf{F}_\varsigma$ can be calculated using the camera pose of the two anchored frames $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ in (5). Meanwhile, the depth uncertainty $\Sigma_{d_v} = \sigma_{d_v}^2$ can be calculated using (13). Suppose that the landmark is associated with an LiDAR-depth $d_r$, and the LiDAR depth has an standard deviation $\sigma_{d_r}$. Here, the standard deviation $\sigma_{d_r}$ is set to a fixed value of 0.1 m, which is determined by the depth-association methods in LE-VINS [33]. Note that, the depth $d_v$ and $d_r$ are all in the c-frame.

The proposed uncertainty-based outlier rejection for LiDAR depths is based on three-sigma verification. The difference between the calculated depth $d_v$ and LiDAR-depth $d_r$ is written as

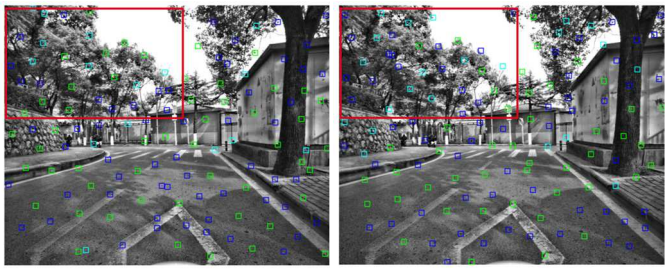
$$\varepsilon_d = d_v - d_r. \tag{14}$$

According to the property of random variables, the standard deviation of the difference $\varepsilon_d$ can be calculated as

$$\sigma_{\varepsilon_d} = \sqrt{\sigma_{d_r}^2 + \sigma_{d_v}^2}. \tag{15}$$

Hence, if the difference $\varepsilon_d$ is not within $\pm 3\sigma_{\varepsilon_d}$, the associated LiDAR depth should be treated as an outlier. Note that, the LiDAR-depth outliers will be removed, while the visual feature pairs associated with the LiDAR-depth outliers will be reserved to construct visual measurements.

Fig. 4 exhibits an illustration of outliers and inliers of LiDAR depths. As shown in Fig. 4, the outlier happens on a tree edge, and the visual point is on the tree, while the LiDAR point is on the wall. The visual and LiDAR points are all on the wall for inliers. According to the results, the LiDAR-depth outliers mainly occur in unstructured environments, especially with many foreground and background objects. We also show a qualitative result in Fig. 5. With the proposed outlier rejection method, fewer visual landmarks are associated



(a) (b)

Fig. 5. Impact of the outlier culling method for LiDAR depths. The blue rectangles denote the uninitialized landmarks. The cyan and green rectangles denote the initialized landmarks, and the green rectangles are associated with LiDAR depths. The big red rectangles represent the unstructured areas, i.e.,the tree leaves, where outliers may frequently occur. With the proposed outlier culling algorithm, fewer landmarks are associated with LiDAR depth within the red rectangles. (a) Without outlier culling. (b) With outlier culling.

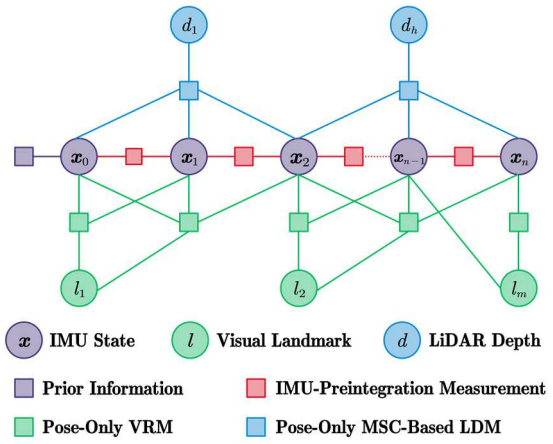



Fig. 6. FGO framework of the proposed PO-VINS. Here, the VRM and LDM represent the visual-reprojection measurement and LiDAR-depth measurement, respectively. The pose-only VRM may include two or three states. The two-state VRM is employed for all landmarks, while the three-state VRM is used for those landmarks with more than three feature observations. The pose-only MSC-based LDM compresses all observations of a single landmark and the LiDAR depth in to one measurement. The IMU velocity and biases are not depicted in the figure for better visualization.

with LiDAR depths (the green rectangle) in the unstructured leaves (within the big red rectangle). Meanwhile, it almost exhibits the same results on the ground and the wall, with or without using the proposed outlier culling method, as shown in Fig. 5. All in all, the LiDAR-depth outliers can be detected and removed by using the presented uncertainty-based outlier rejection, and thus, the system robustness can be improved.

## V. Pose-Only VISE With LiDAR Enhancement

This section presents the proposed pose-only VISE with LiDAR-depth enhancement. The state-estimation problem is first presented to discuss the FGO framework of PO-VINS, as shown in Fig. 6. Then, the VRM and LDM are all modeled in pose-only form. We also study the depth-estimation method, as the landmark depth will be employed for subsequent outlier culling and feature tracking. Finally, we briefly introduce the outlier culling method for visual features.

### A. Tightly Coupled State Estimator

The proposed tightly coupled state estimator is based on the sliding-window optimizer, and the state vector $\boldsymbol{X}$ in PO-VINS can be defined as follows:

$$\begin{aligned} \boldsymbol{X} &= \left[\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_{\mathrm{c}}^{\mathrm{b}}\right] \\ \boldsymbol{x}_k &= \left[\boldsymbol{p}_{\mathrm{wb}_k}^{\mathrm{w}}, \mathbf{q}_{\mathrm{b}_k}^{\mathrm{w}}, \boldsymbol{v}_{\mathrm{wb}_k}^{\mathrm{w}}, \boldsymbol{b}_{g_k}, \boldsymbol{b}_{a_k}, \boldsymbol{t}_{d_k}\right], \quad k \in [0, n] \\ \boldsymbol{x}_{\mathrm{c}}^{\mathrm{b}} &= \left[\boldsymbol{p}_{\mathrm{bc}}^{\mathrm{b}}, \mathbf{q}_{\mathrm{c}}^{\mathrm{b}}\right] \end{aligned} \tag{16}$$

where $\boldsymbol{x}_k$ is the IMU state at each time node, including the position $\boldsymbol{p}_{\mathrm{wb}}^{\mathrm{w}}$, the attitude quaternion $\mathbf{q}_{\mathrm{b}}^{\mathrm{w}}$, and the velocity $\boldsymbol{v}_{\mathrm{wb}}^{\mathrm{w}}$ in the w-frame; $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$ are the gyroscope and the accelerometer biases, respectively; b denotes the IMU body frame (b-frame); $n$ is the number of the IMU-preintegration in the sliding window; and $\boldsymbol{x}_{\mathrm{c}}^{\mathrm{b}}$ is the camera–IMU extrinsic parameters. The attitude quaternion $\mathbf{q}$ and the rotation matrix $\mathbf{R}$ are equivalent [59]. Note that, the time-delay parameter $t_d$ between the camera and IMU is modeled as a random walk [3] for better robustness, as the camera and IMU may not be well synchronized in some datasets. For convenience, the time-delay parameters for the camera–IMU are omitted in the following parts. For details about the implementation of the time-delay model, we can refer to the open-sourced LE-VINS [33] on GitHub (https://github.com/i2Nav-WHU/LE-VINS).

Note that, the visual landmark state parameter, such as the inverse-depth or the 3-D position, is not contained in (16). Thus, the proposed state estimator is in pose-only form regarding the VRM and LDMs. Besides, the proposed pose-only estimator is a tightly coupled form. The state estimation in PO-VINS is achieved by constructing a nonlinear optimization problem, which can be solved by minimizing the sum of the Mahalanobis norm [9] of all measurements and the prior as (17), shown at the bottom of the next page, where $\boldsymbol{e}^V$ is a residual for the pose-only VRM; $l_i$ is a visual landmark; $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ are its two anchored keyframes, mentioned in Section IV-A; $\mathbf{F}_j$, $j \neq \varsigma$ is an another keyframe of the landmark $l_i$; $\boldsymbol{e}^D$ is a residual for the pose-only MSC-based LDM, which compresses all feature observations of a visual landmark and the associated LiDAR depth into one measurement; $\boldsymbol{e}^{\mathrm{Pre}}$ is a residual for IMU-preintegration measurements; and $\boldsymbol{e}^{\mathrm{Prior}}$ is a residual for prior constraints, which are derived from the marginalization [7], [25].

Ceres Solver [60], an open-sourced library for modeling and solving large optimization problems, is adopted in PO-VINS. Specifically, the Levenberg–Marquardt (LM) [15] algorithm is employed to solve the nonlinear optimization problem in (17). Meanwhile, we adopt the Huber robust cost function [60] for VRM and LDM to reduce the effects of the visual feature and LiDAR-depth outliers. Fig. 6 depicts the FGO framework of the proposed PO-VINS. The pose-only VRM may include two or three states. A two-state VRM is constructed when $\mathbf{F}_j = \mathbf{F}_\eta$, and a two-view constraint is employed in FGO.

The IMU-preintegration measurement constructs relative pose constraints between two consecutive IMU states while incorporating the velocity and IMU biases constraints. We

follow our previous work to model the IMU-preintegration measurement [58], and the residuals can be written as:

$$\boldsymbol{e}^{\mathrm{Pre}}\left(\tilde{z}^{\mathrm{Pre}}_{k-1,k}, X\right) = \begin{bmatrix} \left(\mathbf{R}^{\mathrm{w}}_{\mathrm{b}_{k-1}}\right)^{\mathrm{T}}\left(\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_k} - \boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_{k-1}} - \boldsymbol{v}^{\mathrm{w}}_{\mathrm{wb}_{k-1}}\Delta t_{k-1,k} - \frac{1}{2}\boldsymbol{g}^{\mathrm{w}}\Delta t^2_{k-1,k}\right) - \Delta\tilde{\boldsymbol{p}}^{\mathrm{Pre}}_{k-1,k} \\ \left(\mathbf{R}^{\mathrm{w}}_{\mathrm{b}_{k-1}}\right)^{\mathrm{T}}\left(\boldsymbol{v}^{\mathrm{w}}_{\mathrm{wb}_k} - \boldsymbol{v}^{\mathrm{w}}_{\mathrm{wb}_{k-1}} - \boldsymbol{g}^{\mathrm{w}}\Delta t_{k-1,k}\right) - \Delta\tilde{\boldsymbol{v}}^{\mathrm{Pre}}_{k-1,k} \\ \mathrm{Log}\left(\left(\mathbf{q}^{\mathrm{w}}_{\mathrm{b}_k}\right)^{-1} \otimes \mathbf{q}^{\mathrm{w}}_{\mathrm{b}_{k-1}} \otimes \tilde{\mathbf{q}}^{\mathrm{Pre}}_{k-1,k}\right) \\ \boldsymbol{b}_{g_k} - \boldsymbol{b}_{g_{k-1}} \\ \boldsymbol{b}_{a_k} - \boldsymbol{b}_{a_{k-1}} \end{bmatrix} \tag{18}$$

where $\Delta\tilde{\boldsymbol{p}}^{\mathrm{Pre}}_{k-1,k}, \Delta\tilde{\boldsymbol{v}}^{\mathrm{Pre}}_{k-1,k}$, and $\tilde{\mathbf{q}}^{\mathrm{Pre}}_{k-1,k}$ are the position, velocity, and attitude preintegration observations [58], respectively; $\mathbf{R}^{\mathrm{w}}_{\mathrm{b}}$ denotes the rotation matrix of the quaternion $\mathbf{q}^{\mathrm{w}}_{\mathrm{b}}$; $g^{\mathrm{w}}$ denotes the gravity vector in the w-frame; $\mathrm{Log}(\cdot)$ represents the transformation from quaternion to rotation vector [59]; and $\Delta t_{k-1,k}$ is the time length between the two IMU states, i.e., the interval of visual keyframes. The covariance matrix $\Sigma^{\mathrm{Pre}}$ is obtained by noise propagation [58].

### B. Pose-Only VRM

This part derives the residuals for pose-only VRM. From the pose-only multiply-view constraints (8), the landmark depth in the anchored keyframe $\mathbf{F}_\varsigma$ can be expressed as the function of the camera pose of the anchored keyframe $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$ as

$$d^{(\varsigma,\eta)}_{\varsigma} \triangleq \frac{\left\|\left[\tilde{\boldsymbol{p}}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}^{\mathrm{c}_\eta}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}\right\|}{\theta_{\varsigma,\eta}}$$
$$\theta_{\varsigma,\eta} = \left\|\left[\tilde{\boldsymbol{p}}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}^{\mathrm{c}_\eta}_{\mathrm{c}_\varsigma}\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma}\right\| \tag{19}$$

where the relative pose $\{\boldsymbol{p}^{\mathrm{c}_\eta}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}, \mathbf{R}^{\mathrm{c}_\eta}_{\mathrm{c}_\varsigma}\}$ can be calculated using (3). Besides, the camera pose $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wc}}, \mathbf{R}^{\mathrm{w}}_{\mathrm{c}}\}$ can be written as the function of the IMU pose and the extrinsic parameters as

$$\begin{cases} \boldsymbol{p}^{\mathrm{w}}_{\mathrm{wc}} = \boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}} + \mathbf{R}^{\mathrm{w}}_{\mathrm{b}}\boldsymbol{p}^{\mathrm{b}}_{\mathrm{bc}} \\ \mathbf{R}^{\mathrm{w}}_{\mathrm{c}} = \mathbf{R}^{\mathrm{w}}_{\mathrm{b}}\mathbf{R}^{\mathrm{b}}_{\mathrm{c}} \end{cases} \tag{20}$$

where $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}}\}$ is the IMU pose state in (16) and $\{\boldsymbol{p}^{\mathrm{b}}_{\mathrm{bc}}, \mathbf{R}^{\mathrm{b}}_{\mathrm{c}}\}$ denotes the camera–IMU extrinsic parameters in (16).

For an observed keyframe $\mathbf{F}_j, j \neq \varsigma$ of the landmark $l_i$, the pose-only VRM residuals can be derived from (8) as

$$e^{V}\left(\tilde{z}^{V_{\varsigma,\eta}}_{j,l_i}, \boldsymbol{X}\right) = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix}^{\mathrm{T}}\left(\hat{\boldsymbol{p}}^{\mathrm{u}_j} - \tilde{\boldsymbol{p}}^{\mathrm{u}_j}\right)$$
$$\hat{\boldsymbol{p}}^{\mathrm{u}_j} = \frac{\hat{\boldsymbol{p}}^{\mathrm{c}_j}}{\|\hat{\boldsymbol{p}}^{\mathrm{c}_j}\|} \tag{21}$$

where $\hat{\boldsymbol{p}}^{\mathrm{u}_j}$ is the calculated coordinate in the u-frame of the keyframe $\mathbf{F}_j$. $\boldsymbol{b}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{b}_2 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^{\mathrm{T}}$ are two orthogonal bases [4], [33]. It should be noted that the residuals are equivalent to the two-view reprojection residuals if we have $\mathbf{F}_j = \mathbf{F}_\eta$. The covariance $\Sigma^{V_{\varsigma,\eta}}_{l_i}$ is also propagated from the pixel plane onto the tangent plane, similar to (9). From (8), the calculated c-frame coordinate $\hat{\boldsymbol{p}}^{\mathrm{c}_j}$ can be rewritten as

$$\hat{\boldsymbol{p}}^{\mathrm{c}_j} = d^{(\varsigma,\eta)}_{\varsigma}\mathbf{R}^{\mathrm{c}_j}_{\mathrm{c}_\varsigma}\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma} + \boldsymbol{p}^{\mathrm{c}_j}_{\mathrm{c}_j\mathrm{c}_\varsigma}. \tag{22}$$

By considering (19), $\boldsymbol{p}^{\mathrm{u}_j}$ in (21) can be rewritten in more concise form

$$\hat{\boldsymbol{p}}^{\mathrm{u}_j} = \frac{\theta_{\varsigma,\eta}\hat{\boldsymbol{p}}^{\mathrm{c}_j}}{\theta_{\varsigma,\eta}\|\hat{\boldsymbol{p}}^{\mathrm{c}_j}\|} = \frac{\hat{\boldsymbol{p}}^{j}}{\|\hat{\boldsymbol{p}}^{j}\|}$$
$$\hat{\boldsymbol{p}}^{j} = \left\|\left[\tilde{\boldsymbol{p}}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}^{\mathrm{c}_\eta}_{\mathrm{c}_\varsigma}\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma}\right\|\left(\mathbf{R}^{\mathrm{c}_j}_{\mathrm{c}_\varsigma}\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma} + \boldsymbol{p}^{\mathrm{c}_j}_{\mathrm{c}_j\mathrm{c}_\varsigma}\right) \tag{23}$$

where $\hat{\boldsymbol{p}}^{j}$ is a coordinate in a scaled camera frame (only used for better representation); the relative pose $\{\boldsymbol{p}^{\mathrm{c}_j}_{\mathrm{c}_j\mathrm{c}_\varsigma}, \mathbf{R}^{\mathrm{c}_j}_{\mathrm{c}_\varsigma}\}$ can also be obtained from (3) and (20). Consequently, the calculated term $\hat{\boldsymbol{p}}^{\mathrm{u}_j}$ is the function of the IMU poses $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_\varsigma}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}_\varsigma}\}, \{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_j}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}_j}\}$, and $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_\eta}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}_\eta}\}$, and the camera–IMU extrinsic parameters $\{\boldsymbol{p}^{\mathrm{b}}_{\mathrm{bc}}, \mathbf{R}^{\mathrm{b}}_{\mathrm{c}}\}$. Consequently, the VRM residuals in (21) are in pose-only form, without involving the visual landmark states.

### C. Pose-Only LDM

If a visual landmark is associated with an LiDAR depth, we should construct a pose-only LDM, as no depth state is available in the state vector. This part presents two pose-only LDM models, including the direct LDM and MSC-based LDM, and the latter is employed in PO-VINS. Note that, the LiDAR-depth $d_i$ associated with the landmark $l_i$ is always in the c-frame of the anchored keyframe $\mathbf{F}_\varsigma$, which is determined by the depth-association method [33].

*1) Direct LDM:* As the landmark depth can be calculated using the IMU pose states and camera–IMU extrinsic parameters, a naive LDM model is to directly constrain the landmark depth expressed in (19). Hence, the residual for the direct LDM can be written as

$$e^{D}\left(\tilde{z}^{D_i}, X\right) = \frac{\left\|\left[\tilde{\boldsymbol{p}}^{\mathrm{u}_\eta}\right]_\times \boldsymbol{p}^{\mathrm{c}_\eta}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}\right\|}{\left\|\left[\tilde{\boldsymbol{p}}^{\mathrm{u}_\eta}\right]_\times \mathbf{R}^{\mathrm{c}_\eta}_{\mathrm{c}_\varsigma}\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma}\right\|} - \tilde{d}_i \tag{24}$$

where $\tilde{d}_i$ is the LiDAR-depth observation for landmark $l_i$, whose two anchored keyframes are $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$; and the relative pose $\{\boldsymbol{p}^{\mathrm{c}_\eta}_{\mathrm{c}_\eta\mathrm{c}_\varsigma}, \mathbf{R}^{\mathrm{c}_\eta}_{\mathrm{c}_\varsigma}\}$ is the same as that in (19). The LiDAR-depth residual in (24) is the function of the IMU poses $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_\varsigma}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}_\varsigma}\}$ and $\{\boldsymbol{p}^{\mathrm{w}}_{\mathrm{wb}_\eta}, \mathbf{R}^{\mathrm{w}}_{\mathrm{b}_\eta}\}$, and the camera–IMU extrinsic parameters $\{\boldsymbol{p}^{\mathrm{b}}_{\mathrm{bc}}, \mathbf{R}^{\mathrm{b}}_{\mathrm{c}}\}$. Hence, the direct LDM is pose-only, which can directly constrain the pose states.

However, the VRM and LDM should be employed together for a single landmark to avoid a loss of observations. Thus, the measurement number may be increased inevitably, resulting in increased computational costs in solving the state-estimation problem. Besides, the LiDAR-depth constraints in (24) are only imposed on the poses of the two anchored keyframes.

$$\arg\min_{\boldsymbol{X}} \frac{1}{2}\left\{\begin{array}{l} \sum_{j\neq\eta, l_i,\ i\in[1,m]} \left\|\boldsymbol{e}^{V}\left(\tilde{z}^{V_{\varsigma,\eta}}_{j,l_i}, \boldsymbol{X}\right)\right\|^2_{\Sigma^{V_{\varsigma,\eta}}_{l_i}} + \sum_{i\in[1,h]}\left\|\boldsymbol{e}^{D}\left(\tilde{z}^{D_i}, \boldsymbol{X}\right)\right\|^2_{\Sigma^{D_i}} \\ + \sum_{k\in[1,n]}\left\|\boldsymbol{e}^{\mathrm{Pre}}\left(\tilde{z}^{\mathrm{Pre}}_{k-1,k}, \boldsymbol{X}\right)\right\|^2_{\Sigma^{\mathrm{Pre}}_{k-1,k}} + \left\|\boldsymbol{e}^{\mathrm{Prior}}\left(\tilde{z}^{\mathrm{Prior}}, \boldsymbol{X}\right)\right\|^2 \end{array}\right\} \tag{17}$$

Therefore, the impacts of the LiDAR-depth outliers are larger, because not all pose states related to the landmark are employed in the direct LDM. All in all, the direct LDM may result in lower efficiency and reduced reliability.

*2) MSC-Based LDM:* In the famous MSCKF, the landmark states can be eliminated using the left nullspace projection, and a linearized measurement can be employed for Kalman update [10], [11]. In short, the MSC update can be treated as a kind of pose-only form to some extent. Inspired by the MSC update, we propose an MSC-based LDM, which compresses all feature observations of a landmark and its LiDAR depth into a single measurement while retaining the pose-only form. Hence, the measurement number can be reduced, as the pose-only VRM (presented in Section V-B) should not be used again to avoid information reusage. Besides, the impact of the LiDAR-depth outliers can be reduced by employing all feature observations.

Considering a landmark $l_i$, its inverse-depth parameter in the anchored keyframe $\mathbf{F}_\varsigma$ is expressed as $\tau_i$. For the feature observation in the keyframe $\mathbf{F}_j$, the residuals of the VRM in LE-VINS [33] can be written as

$$e^V\left(\tilde{z}_{j,l_i}^{V_\varsigma}, \boldsymbol{X}\right) = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix}^{\mathrm{T}} \left( \frac{\hat{\boldsymbol{p}}^{\mathrm{c}_j}}{\|\hat{\boldsymbol{p}}^{\mathrm{c}_j}\|} - \tilde{\boldsymbol{p}}^{\mathrm{u}_j} \right) \tag{25}$$

where $\hat{\boldsymbol{p}}^{\mathrm{c}_j}$ is the calculated coordinate using $\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma}$. $\tilde{\boldsymbol{p}}^{\mathrm{u}_j}$ and $\tilde{\boldsymbol{p}}^{\mathrm{u}_\varsigma}$ denote the observed features in the u-frame. Similarly, the residual of the LDM in LE-VINS is calculated as

$$e^D\left(\tilde{z}^{D_i}, \boldsymbol{X}\right) = \frac{1}{\tau_i} - \tilde{d}_i. \tag{26}$$

Finally, we obtain several VRMs from multiple observed keyframes and an LDM for the landmark $l_i$.

Linearizing the estimation errors for the states in (16) and the inverse-depth state $\tau_i$, the residuals in (25) and (26) can be approximated as

$$e_i \simeq \mathbf{J}_{\boldsymbol{Y}_i}\delta\boldsymbol{Y}_i + \mathbf{J}_{\tau_i}\delta\tau_i + n_i \tag{27}$$

where $\boldsymbol{Y}_i$ is a part of states related to the landmark $l_i$ in (16), $n_i$ is the measurement noise matrix, and $\mathbf{J}_{\boldsymbol{Y}_i}$ and $\mathbf{J}_{\tau_i}$ are the corresponding Jacobians. We can now obtain a matrix $\mathbf{A}_i$, whose columns form a basis of the left nullspace of $\mathbf{J}_{\tau_i}$ [10], [11]. The residuals in (27) can be rewritten by left multiplying $\mathbf{A}_i^{\mathrm{T}}$ on both sides as

$$\mathbf{A}_i^{\mathrm{T}} e_i \simeq \mathbf{A}_i^{\mathrm{T}}\mathbf{J}_{\boldsymbol{Y}_i}\delta\boldsymbol{Y}_i + \mathbf{A}_i^{\mathrm{T}} n_i. \tag{28}$$

Hence, we obtain the residuals of the MSC-based LDM as

$$e_i^0 = \mathbf{A}_i^{\mathrm{T}} e_i \simeq \mathbf{J}_{\boldsymbol{Y}_i}^0\delta\boldsymbol{Y}_i + n_i^0 \tag{29}$$

where $\mathbf{J}_{\boldsymbol{Y}_i}^0 = \mathbf{A}_i^{\mathrm{T}}\mathbf{J}_{\boldsymbol{Y}_i}$ and $n_i^0 = \mathbf{A}_i^{\mathrm{T}} n_i$. The residuals $e_i^0$ are now independent of the landmark inverse-depth errors and thus can be used for state estimation. As the dimension of the matrix $\mathbf{J}_{\tau_i}$ for a single landmark is very small, we can derive $\mathbf{A}_i^{\mathrm{T}}$ by employing a QR decomposition. The null-space operation and Schur complement have been proven to retain the same amount of information under certain conditions [61]. Hence, the Schur complement can be an alternative method to derive the MSC-based LDM in (29), but the detailed discussions are out of the scope of this article.

The MSC-based LDM residuals in (29) are expressed in pose-only form, without employing landmark states while incorporating both visual feature and the LiDAR-depth observations. Hence, we employ the pose-only MSC-based LDM for those landmarks with LiDAR depths; otherwise, the pose-only VDM is employed. Note that, the proposed MSC-based LDM yields optimal state estimation, as linearization errors can be reduced by multiple iterations during optimization. In contrast, the conventional MSCKF suffers from the one-time linearization, resulting in suboptimal estimation [16].

### D. Depth Estimation for Visual Landmarks

The pose-only state estimator does not include the landmark depth states, yielding improved efficiency. The depths can be analytically estimated to facilitate the outlier culling and the INS-aided feature tracking. For those landmarks without LiDAR depth, their depths are explicitly expressed by the pose of their anchored keyframes $\mathbf{F}_\varsigma$ and $\mathbf{F}_\eta$. Hence, the landmark depth in the anchored keyframe $\mathbf{F}_\varsigma$ can be updated by directly adopting the pose-only depth representation in (19) using the updated IMU pose and camera–IMU extrinsic parameters.

For those landmarks with LiDAR depths, we employ a depth-only optimization by combining the visual and LiDAR observations. Specifically, we use the VRM and LDM expressed in (25) and (26) to construct a new nonlinear optimizer, respectively. The IMU pose and camera–IMU extrinsic states are fixed without estimation, and only the inverse-depth states are estimated. The estimated inverse-depth states are employed to update the landmark depths and their positions in the w-frame. The LiDAR depth can be reflected in the landmark depth by using the depth-only optimization. Thus, the LiDAR-depth outliers can be further rejected in the outlier culling module, which will be described in the next part. As a result, the state-estimation efficiency should be improved a little.

### E. Outlier Detection and Rejection

Once the nonlinear optimization and depth estimation are finished, we conduct outlier detection and rejection for visual features. As the Huber robust cost function is employed in the estimator, the impacts of the outliers may not be notable during optimization. Nevertheless, they can be easily detected using error statistics. We can obtain the landmark positions in the w-frame with the estimated visual landmark depths. The reprojection errors can be calculated for each observed keyframe, and we can have a series of reprojection errors for the landmark. If the average reprojection error exceeds the setting feature noise $\sigma_p$, this landmark, its feature observations, and the possible LiDAR depth will be treated as outliers. Otherwise, only those feature observations with a reprojection error larger than $3\sigma_p$ will be removed, while the visual landmark will be reserved. The reserved visual landmarks will be employed for INS-aided feature tracking [4]. By adopting the outlier culling method, those visual features affected by moving objects, repetitive textures, and illumination changes will be removed, and thus, the system's robustness can be improved.

TABLE I
DATASET DESCRIPTIONS

| Datasets | *Robot* | *R3LIVE* | *FusionPortableV2* |
|---|---|---|---|
| Carrier | Wheeled robot | Handheld | Vehicle |
| Camera | 1280*1024, 20 Hz | 1280*720, 15 Hz | 1024*768, 20 Hz |
| IMU | ADI ADIS16465, 200Hz | BOSCH BMI088, 200 Hz | Sensonor STIM300, 200Hz |
| LiDAR | Livox Mid-70, 10 Hz | Livox AVIA, 10 Hz | Ouster OS1-128, 10 Hz |
| Length | 7 sequences with 9550 s and 13.4 km | 6 sequences with 4462 s and 5.0 km | 5 sequences with 2527 s and 20.8 km |

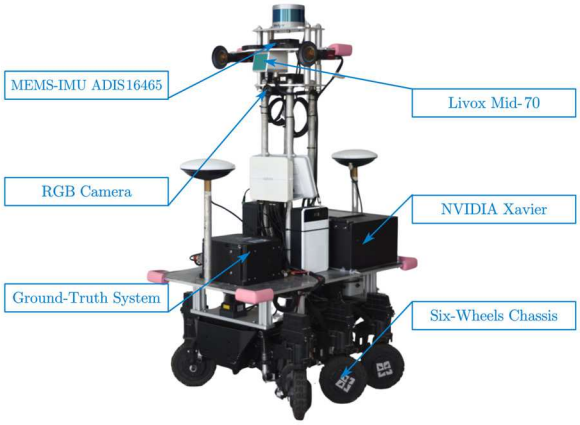



Fig. 7. Equipment setup in the Robot dataset, and the carrier is a low-speed wheeled robot. All sensors are well-synchronized through hardware triggers, and the dataset is collected by the onboard ARM computer (Xavier).

## VI. EXPERIMENTS AND RESULTS

We conduct comprehensive experiments on both public and private datasets with different carriers. We first introduce the datasets and evaluation methods. Then, the accuracy and efficiency results are presented to examine the proposed PO-VINS. We also conduct a series of ablation experiments to evaluate and verify the proposed methods fully. Finally, the real-time performance of PO-VINS on an onboard ARM computer is presented.

### A. Datasets and Evaluation Setup

*1) Datasets:* The public and private datasets are adopted to evaluate the proposed methods fully. Besides, these datasets are collected by carriers with different dynamic conditions, including a low-speed robot, a handheld device, and a high-speed vehicle. Only an MEMS-IMU, a camera, and an LiDAR are used on each dataset for quantitative evaluation. The details about the datasets are shown in Table I.

A low-speed wheeled robot is used in the private Robot dataset, and the maximum speed is about 1.5 m/s. The employed sensors in the Robot dataset include the MEMS-IMU ADIS16465, an RGB camera, and the solid-state LiDAR Mid-70 from Livox, as depicted in Fig. 7. The solid-state LiDAR Mid-70 has a nonrepetitive and irregular scanning pattern, which is conducive to associating the visual features with LiDAR depths. The sensors are all well-synchronized through hardware triggers driven by a GNSS receiver. The high-accuracy ground truth pose (0.02 m for position) is obtained by a postprocessing GNSS/INS integrated navigation software [3]. Seven sequences, which are employed in LE-VINS [33], are used for a fair comparison. As the Robot dataset is collected on campus, the main challenging scenes are moving objects, illumination changes, and repetitive textures. Another two large-scale sequences are employed to evaluate the real-time performance of PO-VINS on the onboard ARM computer, i.e., the NVIDIA Xavier (8-core CPU with 16-GB RAM) in Fig. 7.

The adopted public datasets are the R3LIVE [50] and FusionPortableV2 [62]. The R3LIVE dataset includes indoor and outdoor environments, making it highly challenging. Besides, large angular motions may frequently happen due to the handheld carrier, and thus, motion blurs occur, especially in dim indoor environments. As no ground truth is included in the R3LIVE dataset, the end-to-end errors can be utilized for quantitative evaluation. Hence, only six sequences with end-to-end trajectories are adopted in the R3LIVE dataset. For the FusionPortableV2 dataset, the five vehicle sequences with good ground truth are used. As the vehicle travels very fast, the FusionPortableV2 dataset is more challenging than the other two datasets, especially for the highway sequences. The GNSS/INS integrated navigation results are employed as the ground truth in the FusionPortableV2 dataset. More details about the two datasets can be found in Table I.

*2) Evaluation Setup:* The proposed PO-VINS is implemented by building upon LE-VINS (https://github.com/i2Nav-WHU/LE-VINS) [33]. Hence, LE-VINS is employed as the baseline system for PO-VINS. The pure VINS systems without the LiDAR enhancement are expressed as LE-VINS-WO and PO-VINS-WO. As PO-VINS is an optimization-based system, the state-of-the-art (SOTA) VINS VINS-Mono [17] is employed due to its superior robustness. The SOTA tightly coupled LVINSs LVI-SAM [53], R2LIVE [63], R3LIVE [50], [64], and FAST-LIVO2 [52] are also involved for comprehensive evaluations. We extract 150 visual features for feature-based systems on the Robot dataset due to the lower dynamics, and 200 features on the R3LIVE and FusionPortableV2 datasets due to larger motions. According to our experiments, the sliding-window size is set to 10 for LE-VINS-WO, LE-VINS, PO-VINS-WO, and PO-VINS to bound the accuracy and efficiency. Note that, no explicit loop closure is used for these systems. These systems are all run within the robot operation system (ROS) framework on a laptop (Intel i7-13700H with 32-GB RAM). Note that, LVI-SAM, R2LIVE, R3LIVE, and FAST-LIVO2 failed to run in real time on FusionPortableV2 dataset because of the dense point clouds from Ouster OS1-128. In contrast, PO-VINS and LE-VINS can run in real time on all datasets, as the LiDAR-depth enhancement is lightweight and efficient without using an LO.

As the ground truth pose is included in the Robot and FusionPortableV2 datasets, the ATE is utilized for quantitative evaluations. Specifically, the EVO [65] is adopted to calculate the ATEs. Besides, we also employ the RTE to evaluate the robustness of PO-VINS. The reason is that the RTE, over a

TABLE II
COMPARISON OF THE ATEs ON THE ROBOT DATASET

| ATE (m) | VINS-Mono | FAST-LIO2 | LVI-SAM | R2LIVE | FAST-LIVO2 | R3LIVE | LE-VINS-WO | LE-VINS | PO-VINS-WO | PO-VINS |
|---|---|---|---|---|---|---|---|---|---|---|
| *Exp-1* (2554 m) | 4.67 | 4.67 | 9.16 | 5.05 | 2.32 | 4.49 | 1.86 | **1.46** | 1.74 | 1.50 |
| *Exp-2* (2533 m) | 2.53 | 4.96 | 1.96 | 2.62 | 2.74 | 2.65 | 1.28 | **1.16** | 1.52 | 1.20 |
| *Exp-3* (1151 m) | 2.51 | 0.78 | 2.05 | 1.14 | 2.65 | 0.80 | 0.84 | 0.81 | 0.86 | **0.77** |
| *Exp-4* (1657 m) | 1.65 | 0.83 | 0.82 | 0.67 | **0.22** | 0.75 | 0.93 | 0.84 | 1.05 | 0.90 |
| *Exp-5* (2321 m) | 3.64 | 5.91 | 7.37 | 2.48 | 4.53 | 4.93 | 1.53 | 2.45 | 1.71 | **0.80** |
| *Exp-6* (1539 m) | 5.39 | 2.73 | 5.19 | 2.02 | 2.36 | 7.90 | 2.05 | **1.61** | 1.78 | 1.86 |
| *Exp-7* (1715 m) | 3.73 | 3.67 | 2.24 | 3.06 | 0.97 | 3.45 | 1.34 | 0.76 | **0.75** | 0.81 |
| Average | 3.45 | 3.22 | 4.11 | 2.43 | 2.26 | 3.57 | 1.40 | 1.30 | 1.34 | **1.12** |

LE-VINS-WO and PO-VINS-WO denote the configurations without LiDAR enhancement. The bold result represents the best among different methods, while the underlined result is the second best.

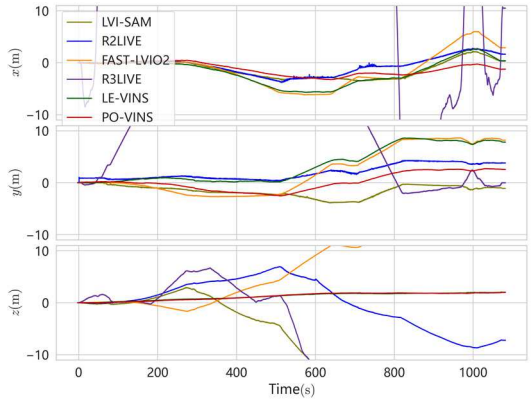



Fig. 8. Position errors on the Robot-Exp-5 sequence with the initial point aligned. Here, only the LVINSs are employed for better visualization.

distance such as 25 m, can reflect the short-time consistency. For the R3LIVE dataset, the 3-D end-to-end errors are used for evaluations.

### B. Evaluation of the Accuracy

*1) Private Robot Dataset:* The evaluation results on the private Robot dataset are shown in Table II. The seven sequences are originally employed in LE-VINS, and they are all collected in complex campus environments. According to the results, LE-VINS and PO-VINS yield notably improved accuracy than SOTA methods regarding the average ATE, using only lightweight LiDAR enhancement rather than complex LO. This demonstrates the practicality and reliability of LiDAR-enhanced VINS. Nevertheless, LO-involved systems, i.e., FAST-LIO2, LVI-SAM, R2LIVE, R3LIVE, and FAST-LIVO2, exhibit superior results on Exp-4, which is in a small-scale environment. The LiDAR subsystem can match with their built point-cloud map, and thus, the drifts can be eliminated. As a visual-based odometry, PO-VINS has no capability to eliminate drifts, even in the same scenarios. We further show the position errors for LVINSs on Exp-5, as depicted in Fig. 8. LE-VINS and PO-VINS exhibit smaller drifts along the $z$-axis than LVI-SAM, R2LIVE, R3LIVE, and FAST-LIVO2. PO-VINS also exhibits superior performance along the $x$-axis and $y$-axis, while LE-VINS shows the largest drift along the $y$-axis.

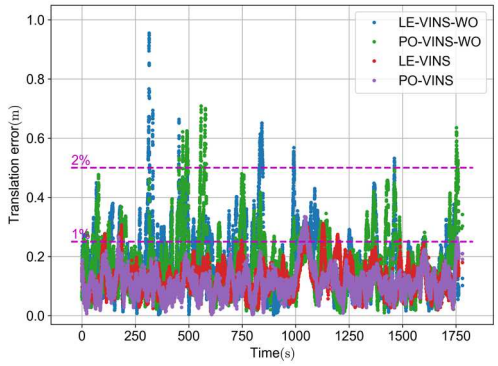
Besides, PO-VINS-WO almost achieves the same average accuracy as LE-VINS-WO in Table II. It demonstrates that the pose-only visual representation is suitable for tightly coupled VINS without losing accuracy. Compared to LE-VINS-WO, LE-VINS also exhibits improved accuracy by incorporating the LiDAR enhancement on all sequences except for Exp-5. The reason for the degradation on Exp-5 may be the LiDAR-depth outliers. In contrast, PO-VINS yields the best on Exp-5, and the proposed uncertainty-based outlier rejection is one of the reasons. Due to the proposed MSC-based LDM, PO-VINS is more robust to LiDAR-depth outliers by employing all visual features and LiDAR depth to construct the MSC-based LDM. More ablation experimental results are presented in Section VI-D. The average absolute translation accuracy of PO-VINS is improved by 16.4% compared to PO-VINS-WO. As the original VINS system PO-VINS-WO already achieves satisfactory accuracy, the improvement for PO-VINS is not very significant. Note that, the minor degradation on Exp-6 and Exp-7 for PO-VINS should be a normal phenomenon, as the system is affected by many factors, especially in such complex experimental environments.

As the short-term relative errors can reflect the robustness to some extent, we also evaluate RTEs over 25 m to evaluate the proposed pose-only methods. Note that, the rotation or attitude errors can be reflected in the RTE; thus, only the RTE is employed. The RTEs over 25 m on Exp-2 and Exp-7 are shown in Fig. 9. According to the results, LE-VINS-WO and PO-VINS-WO almost achieve a similar RTE, though the maximum RTE for PO-VINS-WO on each sequence is much smaller. Hence, the results illustrate that the proposed pose-only VINS can achieve similar robustness regarding short-time accuracy. With the LiDAR enhancement, PO-VINS also exhibits the same RTE as LE-VINS. It demonstrates that the proposed pose-only MSC-based LDM can perform the same local consistency as the conventional LiDAR-enhanced VINS.

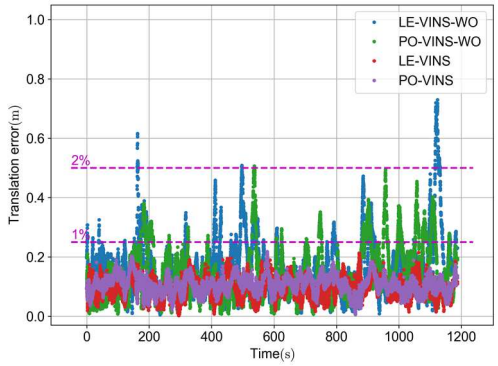
*2) Public R3LIVE-Handheld Dataset:* The low-speed Robot dataset is less challenging in terms of dynamic conditions. Hence, the R3LIVE-Handheld dataset is adopted to evaluate the proposed methods, as large angular motions may frequently happen. The end-to-end errors for the employed systems are shown in Table III, and Fig. 10 depicts the trajectories on the hkust-campus01 sequence for LVINSs. According

TABLE III
COMPARISON OF THE END-TO-END ERRORS ON THE R3LIVE DATASET

| ATE (m) | VINS-Mono | FAST-LIO2 | LVI-SAM | R2LIVE | FAST-LIVO2 | R3LIVE | LE-VINS-WO | LE-VINS | PO-VINS-WO | PO-VINS |
|---|---|---|---|---|---|---|---|---|---|---|
| *main-building* (1037 m) | failed | 1.38 | 8.33 | 0.54 | 1.27 | **0.11** | 2.70 | 0.46 | 4.03 | 0.33 |
| *hkust-campus00* (1317 m) | failed | 5.29 | 15.69 | **0.05** | 5.96 | 5.20 | 13.76 | 12.59 | 11.43 | 10.72 |
| *hkust-campus01* (1524 m) | failed | **2.19** | 17.00 | 3.28 | 5.68 | 19.57 | 5.71 | 3.03 | 5.72 | 2.43 |
| *hkust-campus02* (504 m) | 11.67 | 0.08 | 15.05 | 1.63 | **0.01** | 0.08 | 5.61 | 3.06 | 6.14 | 2.54 |
| *hku-park0* (247 m) | 1.60 | 0.06 | 2.59 | 0.08 | **0.04** | 0.08 | 0.76 | 1.36 | 1.39 | 1.64 |
| *hku-park1* (402 m) | 26.61 | 0.56 | 0.55 | 0.55 | **0.54** | 0.6 | 1.11 | 0.84 | 1.27 | 0.83 |
| Average | Invalid | 1.59 | 9.87 | **1.02** | 2.25 | 4.27 | 4.94 | 3.55 | 5.00 | 3.08 |




(a)




(b)

Fig. 9. Comparison of the RTEs over 25 m. The results demonstrate the necessity of the proposed LiDAR-depth enhancement. (a) Robot-Exp-2 sequence. (b) Robot-Exp-7 sequence.

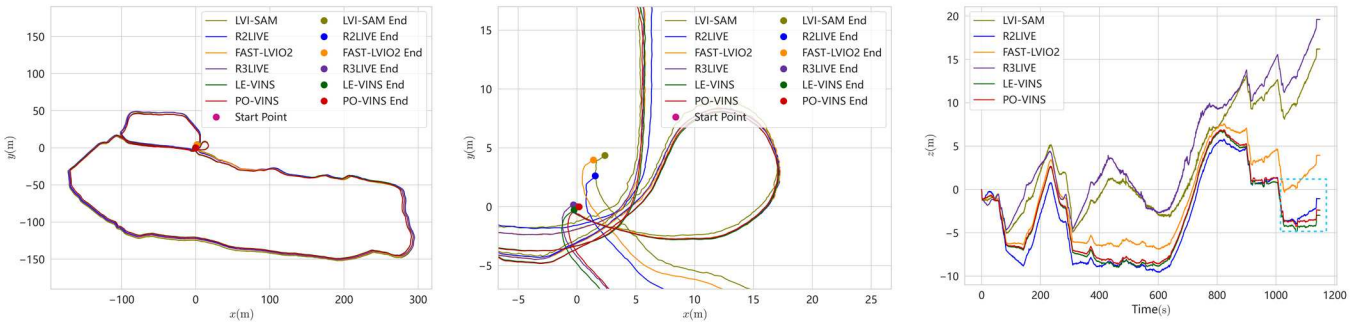
to the results, VINS-Mono fails on three sequences, mainly because of the motion blur caused by large angular motion in indoor environments. Besides, the visual–inertial time-delay parameter has changed notably over time. Thus, VINS-Mono cannot estimate the changed time-delay parameter, which is modeled as a random constant. In contrast, LE-VINS-WO and PO-VINS-WO succeed in running on all the sequences benefiting from the INS-centric design. Meanwhile, we model the visual–inertial time-delay parameter as a random walk, and thus, it can be estimated accurately over time.

The LiDAR system can perform better on the R3LIVE-Handheld dataset due to the structured environments. As a result, R2LIVE yields the best average results, while FAST-LIO2 and FAST-LIVO2 also exhibit superior results. R3LIVE shows worse results on the hkust-campus01 sequence, and thus, the average end-to-end error is much larger. LVI-SAM exhibits worse results because it is mainly designed for spinning LiDARs rather than the Livox LiDAR employed in the R3LIVE dataset.Nevertheless, the LiDAR-based systems can match their previously built point-cloud map, resulting in lower end-to-end errors, according to the R2LIVE results in Fig. 10(c). That is, why FAST-LIO2, R2LIVE, R3LIVE, and FAST-LIVO2 can achieve centimeter-level end-to-end errors on some sequences. Nevertheless, they may exhibit worse results, if the ATE can be used for evaluation.

The pure VINSs PO-VINS-WO and LE-VINS-WO show similar average errors. With the LiDAR-depth enhancement, LE-VINS and PO-VINS yield notably improved accuracy than their VINS systems on all the sequences except for hku-park0. The reason is that hku-park0 is in a dim park with many unstructured trees and bushes. Thus, only a few valid visual features will be reserved after using the outlier rejection method (Section V-E) due to the impacts of the LiDAR-depth outliers. Besides, PO-VINS and LE-VINS show worse results on the hkust-campus00 sequence, mainly because the larger angular motions may frequently interrupt the feature tracking. Nevertheless, PO-VINS shows a lower average error than LE-VINS. Overall, PO-VINS demonstrates superior robustness than the baseline system LE-VINS on the R3LIVE dataset. PO-VINS also exhibits comparable accuracy to tightly coupled LVINSs on sequences main-building, hkust-campus01, and hku-park1, which also demonstrate the practicality of the lightweight LiDAR-enhanced VINS.

*3) Public FusionPortableV2-Vehicle Dataset:* As the robot and pedestrian are very slow on the previous datasets, about several meters per second, the high-speed FusionPortableV2-Vehicle dataset is employed. However, we fail to run LVI-SAM and R2LIVE because of the sensor incompatibility. The ATEs on the FusionPortableV2 dataset are shown in Table IV. VINS-Mono exhibits the worst ATE, especially on high-speed highway sequences. FAST-LIO2, R3LIVE, and FAST-LIVO2 also exhibit worse results, and they fail on high-way sequences. In contrast, the proposed PO-VINS succeeds to run on all sequences and exhibits the best results. Therefore, the proposed LiDAR-enhanced VINS demonstrates superior robustness and reliability, even without using LO.

PO-VINS-WO shows a lower average ATE than LE-VINS-WO, which is caused by the notable difference in the highway00 sequence.Nevertheless, LE-VINS-WO and PO-VINS-WO almost yield similar accuracy on other

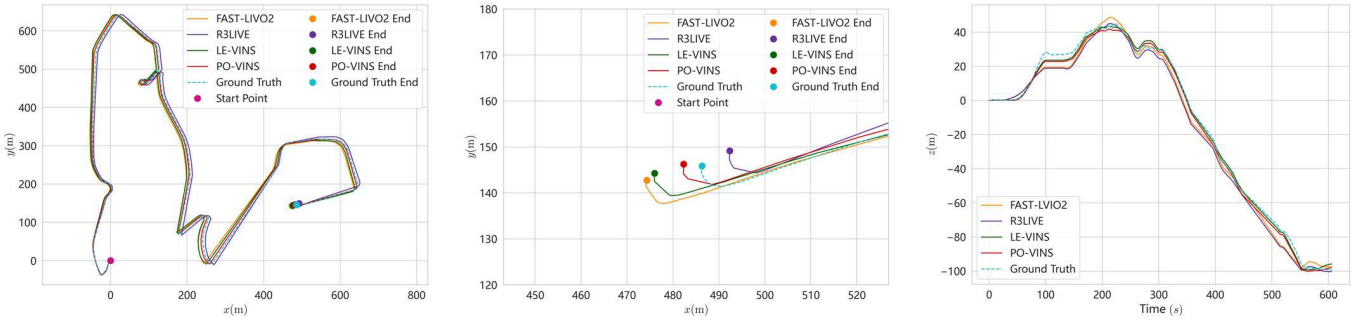



(a) (b) (c)

Fig. 10. Trajectories on the R3LIVE-hkust-campus01 sequence for LVINSs. The cyan rectangle in (c) denotes the abnormal height result for R2LIVE. (a) Whole trajectories. (b) Trajectories at the end point. (c) Height ($z$-axis) changes.

TABLE IV
COMPARISON OF THE ATES ON THE FUSIONPORTABLEV2 DATASET

| ATE (m) | VINS-Mono | FAST-LIO2 | FAST-LIVO2 | R3LIVE | LE-VINS-WO | LE-VINS | PO-VINS-WO | PO-VINS |
|---|---|---|---|---|---|---|---|---|
| *campus00* (2.60 km) | 24.37 | 5.24 | 6.02 | 5.42 | 6.32 | 4.49 | 3.75 | **2.48** |
| *campus01* (2.06 km) | 13.87 | **5.51** | 7.7 | 5.64 | 6.35 | 5.99 | 6.22 | 6.90 |
| *downhill00* (3.74 km) | 71.6 | 36.5 | 40.87 | 157.89 | 4.05 | 3.43 | 4.58 | **2.81** |
| *highway00* (9.33 km) | failed | failed | failed | failed | 31.98 | 25.91 | 13.88 | **9.27** |
| *highway01* (3.05 km) | 141.98 | 20.81 | 162.17 | failed | 12.9 | 7.99 | 10.45 | **7.93** |
| Average | Invalid | Invalid | Invalid | Invalid | 12.32 | 9.56 | 7.78 | **5.88** |




(a) (b) (c)

Fig. 11. Trajectories on the FusionPortableV2-campus00 sequence for LVINSs. (a) Whole trajectories. (b) Trajectories at the end point. (c) Height ($z$-axis) changes.

sequences. PO-VINS shows improved accuracy over PO-VINS-WO by incorporating the LiDAR enhancement, especially on the highway sequences. PO-VINS also exhibits superior accuracy to LE-VINS, benefiting from the proposed pose-only solutions for LiDAR depth.

Fig. 11 depicts the trajectories on the campus00 sequence. PO-VINS shows a better-aligned trajectory to ground truth than other systems, and the endpoint is the closest to the ground truth. In conclusion, the results demonstrate that the proposed pose-only methods are robust on the high-speed vehicle dataset and achieve improved accuracy compared to the baseline system LE-VINS.

### C. Evaluation of the Efficiency

The accuracy results in Section VI-B illustrate that the pose-only VINS, i.e., PO-VINS-WO, exhibits a similar accuracy to the conventional baseline system LE-VINS-WO. Besides, the pose-only LiDAR-enhanced VINS, i.e., PO-VINS, yields improved accuracy and robustness to LE-VINS by incorporating the proposed uncertainty-based outlier rejection and MSC-based LDM. Nevertheless, the main advantage of the pose-only solution lies in the higher state-estimation efficiency. The statistical results of the state-estimation time, which includes the FGO and the marginalization, are shown in Table V.

Compared to the VINS baseline LE-VINS-WO, the average state-estimation time of PO-VINS-WO is reduced by about 33.8%. The results are reasonable, as the dimension of the state vector can only be reduced by about 50% by using the pose-only VRM, according to our analysis. The time costs for LE-VINS increased a little as extra LDMs are employed. PO-VINS exhibits shorter time costs, and the average

TABLE V
COMPARISON OF THE STATE-ESTIMATION TIME ON THE ROBOT DATASET

| Time (ms) | LE-VINS-WO | LE-VINS | PO-VINS-WO | PO-VINS |
|---|---|---|---|---|
| *Exp-1* | 24.6 | 24.5 | 16.2 | 16.3 |
| *Exp-2* | 22.8 | 24.2 | 15.6 | 16.2 |
| *Exp-3* | 22.0 | 24.3 | 13.8 | 16.1 |
| *Exp-4* | 22.0 | 24.7 | 14.3 | 15.9 |
| *Exp-5* | 24.5 | 24.6 | 15.7 | 16.2 |
| *Exp-6* | 24.6 | 24.2 | 15.5 | 16.7 |
| *Exp-7* | 21.0 | 23.2 | 15.7 | 16.2 |
| Average | 23.1 | 24.2 | 15.3 | 16.2 |

The time costs for the state estimation include the factor graph optimization and the marginalization.

state-estimation time is reduced by 33.1% compared to the baseline LE-VINS. This benefits from the proposed MSC-based LDM and the depth-only optimization. The MSC-based LDM can combine all visual features and the LiDAR-depth observations of a visual landmark into a single measurement, and thus, the measurement number can be reduced notably. Meanwhile, the LiDAR-depth outliers can be exposed by using depth-only optimization, resulting in fewer measurements while maintaining superior robustness. Ablation experiments will be conducted in the next section to evaluate the impacts of these employed methods.

The efficiency results demonstrate that the pose-only solution can also achieve notable improved state-estimation efficiency in the tightly coupled VINS, even with LiDAR enhancement. Besides, the efficiency improvement should be more significant on the onboard ARM computer with limited computational resources. Although the improvement is not larger than 100% in such a real-time configuration due to the small dimension of the state-estimation problem, the results are satisfactory to improve the real-time performance.

### D. Ablation Experiments

Ablation experiments are conducted to evaluate the impacts of the uncertainty-based outlier rejection method, MSC-based LDM, and depth-only optimization. Table VI exhibits the ATEs and state-estimation time using different configurations.

*1) Impact of the Uncertainty-Based Outlier Rejection:* The proposed uncertainty-based outlier rejection method can detect and reject the LiDAR-depth outliers and thus can improve the localization accuracy. As shown in Table VI, the ATEs are increased on four sequences without using the proposed outlier rejection method, especially on the Exp-1 and Exp-6 sequences. Meanwhile, the ATEs are almost the same on the other three sequences. The average ATE of the proposed PO-VINS is reduced by about 21.7% compared to the configuration without using the outlier rejection. Besides, the state-estimation time costs are almost the same with or without the outlier rejection. The state-estimation efficiency is improved a little for PO-VINS, as the LDMs are decreased after the outlier rejection. The ablation results demonstrate that the uncertainty-based outlier rejection for LiDAR depth can effectively detect and reject outliers and improve localization accuracy. Note that, the proposed outlier rejection can run fast, and the average time cost is about 0.2 ms.

TABLE VI
ATES AND THE STATE-ESTIMATION TIME ON THE ROBOT DATASET WITH DIFFERENT CONFIGURATIONS

| | w/o outlier rejection[1] | | Using direct LDM[2] | | w/o depth-only optimization[3] | | Proposed PO-VINS | |
|---|---|---|---|---|---|---|---|---|
| | ATE (m) | Time (ms) | ATE (m) | Time (ms) | ATE (m) | Time (ms) | ATE (m) | Time (ms) |
| *Exp-1* | 2.27 | 16.7 | 2.2 | 22.1 | 1.74 | 21.3 | **1.50** | 16.3 |
| *Exp-2* | **1.18** | 16.2 | 1.69 | 21.9 | 1.38 | 19.8 | 1.20 | 16.2 |
| *Exp-3* | 0.77 | 16.6 | **0.66** | 19.9 | 0.81 | 19.3 | 0.77 | 16.1 |
| *Exp-4* | **0.85** | 16.3 | 1.25 | 20.0 | 0.86 | 18.8 | 0.90 | 15.9 |
| *Exp-5* | 1.01 | 17.6 | 1.49 | 21.0 | **0.57** | 20.7 | 0.80 | 16.2 |
| *Exp-6* | 2.89 | 16.7 | 2.37 | 20.4 | **1.70** | 20.6 | 1.86 | 16.7 |
| *Exp-7* | 1.02 | 15.6 | 1.01 | 20.9 | 0.85 | 18.4 | **0.81** | 16.2 |
| Average | 1.43 | 16.5 | 1.52 | 20.9 | 1.13 | 19.8 | **1.12** | 16.2 |

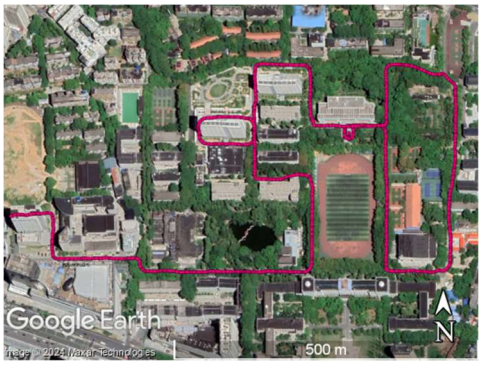
[1]Without using the uncertainty-based outlier rejection method proposed in Section IV.C.
[2]Using the direct LDM in Section V.C.1.
[3]Without using the depth-only optimization presented in Section V.D.

*2) Impact of the MSC-Based LDM:* The proposed pose-only MSC-based LDM combines all the visual features and LiDAR-depth observations of a landmark into a single measurement. Hence, it can improve both robustness and efficiency, as illustrated in Table VI. Compared to the direct LDM configuration, the proposed PO-VINS is more robust to LiDAR-depth outliers, as all feature observations are employed. In contrast, only the feature observations in the two anchored keyframes are used in the direct LDM. Meanwhile, the LDM number can be decreased notably by combining all related observations into one measurement, and thus, the state-estimation efficiency can also be reduced. The results show that the proposed pose-only MSC-based LDM is accurate and efficient.

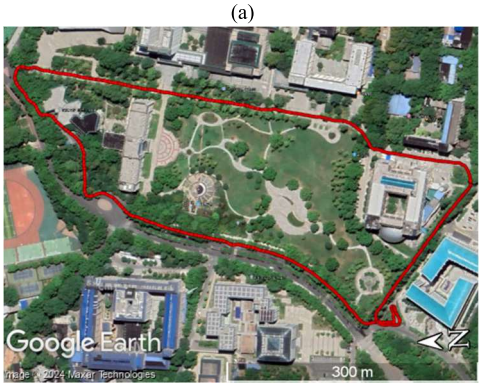
*3) Impact of the Depth-Only Optimization:* The depth-only optimization is conducted to update the depths of those landmarks associated with the LiDAR depth. According to the results in Table VI, the proposed PO-VINS yields improved efficiency to the configuration without the depth-only optimization while achieving similar ATEs. The improvement of efficiency is because the LiDAR-depth outliers can be exposed after the depth-only optimization and can be detected and rejected. As a result, the LDM number can be decreased a little, and state-estimation efficiency can be improved without sacrificing the accuracy. The average time cost of the depth-only optimization is about 1.8 ms. Thus, it is meaningful to employ the proposed depth-only optimization.

### E. Real-Time Performance

We conduct experiments on an onboard ARM computer (NVIDIA Xavier with 8-core CPU and 16-GB RAM) to evaluate the real-time performance of the proposed PO-VINS. The same configurations from the previous experiments, which are running on the laptop, are adopted on the onboard computer for LE-VINS and PO-VINS. Both PO-VINS and LE-VINS can run in real time on the onboard computer, even with the same configurations employed on the laptop.




(a)




(b)

Fig. 12. Trajectories on the real-time experiments. (a) Test trajectory on the Seq-1 (2012 s and 2565 m). (b) Test trajectory on the Seq-2 (1162 s and 1462 m).

TABLE VII
PROCESSING TIME AND ATES ON THE REAL-TIME EXPERIMENTS

| | Front-end[1] (ms) | | Back-end[2] (ms) | | ATE (m) | |
|---|---|---|---|---|---|---|
| | LE-VINS | PO-VINS | LE-VINS | PO-VINS | LE-VINS | PO-VINS |
| *Seq-1* | 28 | **26** | 109 | **48** | 2.82 | **2.05** |
| *Seq-2* | 25 | **24** | 104 | **45** | 2.20 | **1.64** |
| Average | 26.5 | **25.0** | 106.5 | **46.5** | 2.51 | **1.85** |

[1]The front-end processing time includes image preprocessing, feature detection and tracking, and the depth association.
[2]The back-end processing time, *i.e.,* the state-estimation time, includes the factor graph optimization and the marginalization.
Note that the depth association and the factor graph optimization are only conducted when a new keyframe is selected, and the average keyframe interval is about 0.3 s. Thus, both LE-VINS and PO-VINS can run real-time on the on-board ARM computer.

The trajectories of the employed large-scale sequences Seq-1 (2012 s and 2565 m) and Seq-2 (1162 s and 1462 m) are depicted in Fig. 12. The two sequences are also collected by the wheeled robot depicted in Fig. 7. As shown in Table VII, the front-end time for PO-VINS is almost the same, while the average back-end time is reduced by 56.3%, compared to the baseline system LE-VINS. This is because the FGO is only conducted when a new keyframe is selected, and the average keyframe interval is about 0.3 s. It demonstrates that the proposed pose-only solution is more efficient on the onboard ARM computer with limited computational resources. Meanwhile, PO-VINS yields notably improved accuracy to LE-VINS on the two sequences regarding the ATE.

TABLE VIII
TOTAL RUNNING TIME AND EQUIVALENT FPS ON THE REAL-TIME EXPERIMENTS

| | Sequence length (s) | Total running time (s) | | Equivalent FPS | |
|---|---|---|---|---|---|
| | | LE-VINS | PO-VINS | LE-VINS | PO-VINS |
| *Seq-1* | 2012 | 1315 | **944** | 30.6 | **42.6** |
| *Seq-2* | 1162 | 830 | **610** | 28.0 | **38.1** |

The equivalent FPS is calculated by dividing the sequence length by the running time and multiplying by the camera frame rate, and the camera frame rate is 20 FPS.

We evaluate the total running time of LE-VINS and PO-VINS and calculate the equivalent frame per second (FPS), as shown in Table VIII. Here, the equivalent FPS is calculated by dividing the sequence length by the running time and multiplying by the camera frame rate (20 frames/s). The total running time of PO-VINS on the two sequences is decreased by 28.2% and 26.5%. Besides, the FPSs of PO-VINS are increased by 39.2% and 36.1%.

According to our statistics, the average CPU usage on Xavier for PO-VINS and LE-VINS is about 14% (using 250-MB RAM) and 18% (using 260-MB RAM), respectively. In short, two CPU cores (CPU load of 25%) are definitely enough for PO-VINS and LE-VINS to run in real time on Xavier, even using the same configurations on the laptop. Besides, the system latency for PO-VINS is mainly caused by the front- and back-end processes, and it is about 72 ms, which is slightly larger than the camera interval of 50 ms. Nevertheless, the back-end optimization is conducted every 300 ms. Besides, the low-latency IMU can be used to provide high-frequency INS prior poses without waiting for visual processes. The real-time experimental results demonstrate that the proposed pose-only solution and PO-VINS effectively improve the efficiency on the onboard ARM computer. Hence, PO-VINS can be an efficient and robust odometry solution.

## VII. CONCLUSION AND DISCUSSION

This study proposes a tightly coupled LiDAR-enhanced VINS using a full pose-only solution to achieve an efficient and robust state estimation. In the proposed pose-only VINS, the landmark depths are explicitly expressed by the pose of the anchored keyframes. Hence, the landmark states can be explicitly removed from the state vector. Experimental results demonstrate that the pose-only VINS yields a similar accuracy to the conventional VINS, while the state efficiency is improved by 33.8% even in a real-time configuration. The depth uncertainty for visual landmarks is analytically derived within the pose-only framework. Hence, we can employ the depth uncertainty to detect and reject LiDAR-depth outliers, which has been proven to be effective. Besides, we propose an MSC-based LDM to incorporate LiDAR enhancement into the pose-only solution seamlessly. The proposed PO-VINS achieves improved efficiency and robustness compared to the baseline system, LE-VINS, on a series of datasets with different carriers. Specifically, the state-estimation efficiency on the private Robot dataset is improved by more than 30% and 50% on the laptop (Intel i7-13700H) and the onboard ARM computer (NVIDIA Xavier), respectively.

We believe that the pose-only VISE can be a substitute for the conventional BA-based methods, as it is more efficient while maintaining the same accuracy. By employing a lightweight LiDAR-depth enhancement, the proposed PO-VINS can achieve comparable accuracy to tightly coupled LVINSs with LiDAR odometry involved in most cases, demonstrating superior practicality. However, we noted that PO-VINS shows unsatisfactory results on the R3LIVE-Handheld dataset due to the visually degraded scenarios. Future works include adaptively incorporating the LO to construct a tightly coupled LVINS while maintaining a superior efficiency and robustness. Besides, we intend to integrate GNSS to construct a multisensor fusion system so as to achieve a drift-free positioning scheme.

## REFERENCES


[1] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 1–3, pp. 1–308, 2020.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[3] P. D. Groves, *Multiple View Geometry in Computer Vision*. Norwood, MA, USA: Artech House, 2008.

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

[5] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 1994, pp. 593–600.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2564–2571.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[8] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[9] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

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

[11] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, May 2013.

[12] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017.

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

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

[15] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, nos. 1–2, pp. 1–139, 2017.

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

[17] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[18] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1689–1696.

[19] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 422–429, Apr. 2020.

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

[21] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[22] L. V. Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1408–1415, Apr. 2022.

[23] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1974–1982.

[24] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[25] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2014.

[26] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-KLAM: Constrained keyframe-based localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3638–3643.

[27] Q. Cai, Y. Wu, L. Zhang, and P. Zhang, "Equivalent constraints for two-view geometry: Pose solution/pure rotation identification and 3D reconstruction," *Int. J. Comput. Vis.*, vol. 127, no. 2, pp. 163–180, Feb. 2019.

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

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

[30] T. Zhang, L. Wei, H. Tang, M. Yuan, L. Wang, and X. Niu, "SE-LIO: Semantic-enhanced solid-state-LiDAR–inertial odometry for tree-rich environments," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–13, 2025.

[31] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *Auto. Robots*, vol. 41, no. 1, pp. 31–43, Jan. 2017.

[32] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "LiDAR-monocular visual odometry using point and line features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1091–1097.

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

[34] J. Civera, A. J. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[35] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[36] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 473–479.

[37] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6319–6326.

[38] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[39] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, Sep. 2010.

[40] F. Zhang, *The Schur Complement and Its Applications*, vol. 4. Cham, Switzerland: Springer, 2006.

[41] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GT-RIM-CP&R-2012-002, 2012.

[42] R. Mur-Artal and J. D. Tardos, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[43] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[44] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[45] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 50–72, Jan. 2016.

[46] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2510–2517.

[47] L. Zhao, S. Huang, Y. Sun, L. Yan, and G. Dissanayake, "ParallaxBA: Bundle adjustment using parallax angle feature parametrization," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 493–516, Apr. 2015.

[48] L. Liu, T. Zhang, B. Leighton, L. Zhao, S. Huang, and G. Dissanayake, "Robust global structure from motion pipeline with parallax on manifold bundle adjustment and initialization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2164–2171, Apr. 2019.

[49] K. Huang, J. Xiao, and C. Stachniss, "Accurate direct visual-laser odometry with explicit occlusion handling and plane detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 1295–1301.

[50] J. Lin and F. Zhang, "$R^3$LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 10672–10678.

[51] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "FAST-LIVO: Fast and tightly-coupled sparse-direct LiDAR-inertial-visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 4003–4009.

[52] C. Zheng et al., "FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry," *IEEE Trans. Robot.*, vol. 41, pp. 326–346, 2024.

[53] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5692–5698.

[54] J. Graeter, A. Wilczynski, and M. Lauer, "LIMO: LiDAR-monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7872–7879.

[55] P. Wang, Z. Fang, S. Zhao, Y. Chen, M. Zhou, and S. An, "Vanishing point aided LiDAR-visual-inertial estimator," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13120–13126.

[56] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong, "CamVox: A low-cost and accurate LiDAR-assisted visual SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5049–5055.

[57] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[58] 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 J.*, vol. 22, no. 17, pp. 17194–17204, Sep. 2022.

[59] J. Sola, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*.

[60] S. Agarwal and K. Mierle. (Mar. 2022). *Ceres Solver*. The Ceres Solver Team. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[61] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6749–6755.

[62] H. Wei et al., "FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments," 2024, *arXiv:2404.08563*.

[63] J. Lin, C. Zheng, W. Xu, and F. Zhang, "$R^2$LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7469–7476, Oct. 2021.

[64] J. Lin and F. Zhang, "$R^3$LIVE++: A robust, real-time, radiance reconstruction package with a tightly-coupled LiDAR-inertial-visual state estimator," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 11168–11185, Dec. 2024.

[65] Michael Grupp. (Mar. 2023). *EVO*. [Online]. Available: https://github.com/MichaelGrupp/evo